.REM %

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZQMB-G-D

PRODUCT NAME: 0-124K MEMORY EXERCISER

DATE CREATED: FEBRUARY, 1974

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: JOHN ADAMS

TABLE OF CONTENTS

PDP-11 D-124K MEMORY EXERCISER
TABLE OF CONTENTS

TABLE OF CONTENTS (CONT'D)

ABSTRACT

PROGRAM DZQMB TESTS CONTIGUOUS MEMORY ADDRESS FROM 000000 TO 757776. IT VERIFIES THAT EACH ADDRESS IS UNIQUE (AN ADDRESS TEST) AND THAT EACH MEMORY LOCATION CAN BE READ/WRITTEN RELIABLY (WORST CASE NOISE TESTS). IF MEMORY MANAGEMENT IS AVAILABLE, ALL TESTING IS PERFORMED WITH MEMORY MANAGEMENT ENABLED, (UNLESS DISABLED).

THIS PROGRAM MAY BE USED TO ADJUST MARGIN MEMORY.

ALSO INCLUDED IS A TOGGLE IN ADDRESS TEST.

ALSO INCLUDED IS THE BRANCH GOBBLE MOS TEST. NOTE THAT ONLY SECTIONS 9.1 THROUGH 9.4 APPLY TO BRANCH GOBBLE.

CHAPTER 1
REQUIREMENTS

1.1 EQUIPMENT

THE PDP-11 FAMILY PROCESSOR WITH 8K MEMORY.

OPTIONAL...

KT11-C OR KT11-D MEMORY MANAGEMENT OPTION OR MF11 PARITY OPTION.

1.2 STORAGE

PROGRAM STORAGE - THE PROGRAM USES MEMORY 0-17777.

1.3 PRELIMINARY PROGRAMS

IPDP-11 FAMILY INSTRUCTION EXERCISER .DCGKC OR `DZQKC` KT11-C KT11-D LOGIC TESTS.

CHAPTER 2

LOADING AND STARTING PROCEDURE

LOAD PROGRAM INTO MEMORY USING ABS LOADER.

LOAD ADDRESS 200
SET SW12 IN DESIRED POSITION (SEE CHAPTER 3).
PRESS START.

ASTERISK "*" WILL BE PRINTED AFTER EACH PASS.
"DZQMB DONE!" WILL BE PRINTED AFTER 8 PASSES.

PASS COUNT MAY BE MONITORED IN THE DISPLAY REGISTER (11/45 OR LOCATION 756.

NOTE

THIS PROGRAM SAVES THE LOADERS BOOT AND ABS. TO RESTORE THE LOADERS, RESTART AT 162. BEFORE RESTARTING INSURE THAT THE PROGRAM IS NOT RELOCATED. IF THE PROGRAM IS RELOCATED, THE PC WILL INDICATE WHICH BANK CONTAINS THE PROGRAM. NEXT START THE PROGRAM AT *+ 12354, WHERE * = BITS 13-15 OF THE PC. THE PROGRAM WILL RELOCATE BACK TO 0-4K AND HALT AT 176. PRESS CONTINUE TO RESUME TESTING.

2.1 ACT11 OPERATION

IF THE PROGRAM IS RUN IN QUICK VERIFY MODE UNDER ACT11 THE PROGRAM IS DONE AFTER THE FIRST PASS. ALSO THE PROGRAM DOES NOT RELOCATE TO TEST THE LOWER 4K OF MEMORY.

CHAPTER 3

SWITCH SETTINGS

SW15=1 OR UP HALT ON ERROR

NOTE

IF SW15=1 WHEN AN ERROR OCCURS THE PROGRAM WILL HALT, AND THE CORRECT DATA WILL NOT BE LOADED INTO THE FAILING ADDRESS. IF SW15 IS RAISED AFTER THE ERROR TYPEOUT BEGINS, THE PROGRAM WILL HALT WHEN THE TYPEOUT COMPLETES AND THE CORRECT DATA WILL BE LOADED INTO THE FAILING ADDRESS.

SW14=1 OR UP LOOP SUBTEST

SW13=1 OR UP INHIBIT ERROR TYPEOUT

SW12=1 OR UP INHIBIT USE OF MEMORY MANAGEMENT

NOTE

INHIBITING THE USE OF MEMORY MANAGEMENT CAN BE DONE ONLY WHEN THE PROGRAM IS STARTED. IF THE USE OF MEMORY MANAGEMENT IS INHIBITED THE LAST ADDRESS AS TYPED BY THE PROGRAM WILL ONLY REFLECT THE AMOUNT OF MEMORY UP TO 28K (LAST ADDRESS = 160000).

SW11=1 OR UP INHIBIT SUBTEST ITERATION

```
382
383                                 SW10=1 OR UP        RING BELL ON ERROR
```

PDP-11 0-124K MEMORY EXERCISER PAGE 9
SWITCH SETTINGS

SW9=1 OR UP DISPLAY ERROR COUNT IN DISPLAY REGISTER

SW9=0 OR DOWN DISPLAY PASS COUNT IN DISPLAY REGISTER

SW8=1 OR UP HALT PROGRAM UNRELOCATED AND RESTORE LOADERS

SW5=1 OR UP INHIBIT PARITY ERROR DETECTION (INITIAL STARTUP ONLY)

NOTE

WITH PARITY ERROR DETECTION ENABLED, A MEMORY FAILURE WILL CAUSE A PARITY ERROR. THE ERROR PRINTOUT ON A PARITY ERROR DOES NOT TYPE THE GOOD DATA. THUS, A BIT DROP OR PICKUP WILL NOT BE TYPED AS SUCH. IT IS BEST TO RUN THE PROGRAM FOR 1 PASS (UNTIL AN * IS TYPED) WITH PARITY DISABLED, THEN RESTART THE PROGRAM WITH PARITY ENABLED.

CHAPTER 4

SUBROUTINE ABSTRACTS

4.1 SCOPE

THE PROGRAM STORES IN R1 THE PC OF THE LAST TEST SUCCESSFULLY EXECUTED AND MAY BE USED AS AN AID IN DEBUGGING IF THE PROGRAM 'BOMBS' BECAUSE OF A HARDWARE FAILURE.

CHAPTER 5

ERRORS

THESE TESTS PRINT OUT THE PC WHERE THE ERROR WAS DETECTED, THE FAILING ADDRESS, THE GOOD DATA, AND THE BAD DATA I.E.

PC=XXXXXX ADDRESS AAAAAA GOOD DATA GGGGGG BAD DATA BBBBBB

THE ADDRESS OF THE FAILING LOCATION IS THE TRUE 18 BIT PHYSICAL ADDRESS.

NOTE

WHEN TESTING MEMORY LOCATIONS 0-17776 THE PC TYPED WILL BE A MULTIPLE OF 20000 GREATER THAN REFLECTED IN THE PROGRAM LISTING.

THE ADDRESS OF THE BAD DATA IS IN (R2) -2

THE GOOD DATA IN R0

THE BAD DATA IN R3

THE ADDRESS OF GOOD DATA IS IN R4 (RANDOM DATA TEST ONLY). WHEN AN ERROR IS DETECTED WHEN EXERCISING THE MEMORY USING THE WORST CASE NOISE PATTERNS, THE USER SHOULD RESTART THE PROGRAM SELECTING PROGRAM (SEE CHAPTER 8 FOR DETAILS) SELECTING THE APPROPRIATE PARAMETERS. THE USER CAN USE THE PC AND ADDRESS OF THE FAILURE TO SELECT THE PROPER CORE BANK(S) AFFECTED AND ALSO THE SPECIFIC PATTERN. THIS ALLOWS MAXIMUM SCOPE CAPABILITIES.

5.1 PARITY ERROR

```
                IF THE MEMORY PARITY OPTIONS ARE INSTALLED THE PROGRAM RUNS  WITH  THE
```

ERRORS

ACTION ENABLE BIT SET (BIT 0). IF A PARITY ERROR IS DETECTED THE PROGRAM WILL TYPE:

PARITY ERROR

AND SCAN MEMORY FOR THE ADDRESS(ES) CAUSING THE PARITY ERROR(S). WHEN THE PARITY ERROR IS DETECTED AN ERROR WILL BE TYPED AS SHOWN BELOW:

PC=XXXXXX ADDRESS AAAAAA BAD DATA BBBB3B

PRESS CONTINUE OR RESTART TO RESUME TESTING. IF A PARITY ERROR IS NOT DETECTED ON SCAN THE PROGRAM WILL TYPE:

PARITY ERROR NOT FOUND ON SCAN

PC=XXXXXX ADDRESS=AAAAAA

WHERE:

AAAAAA=PC AT TIME PARITY ERROR WAS DETECTED.

NOTE

PARITY IS DISABLED WHEN THE PROGRAM IS RELOCATED.

CHAPTER 6
RESTRICTIONS

6.1 STARTING RESTRICTION

PROGRAM MUST NOT BE RELOCATED WHEN RESTARTING.

6.2 OPERATIONAL RESTRICTION

PROGRAM CHECKS CONTIGUOUS MEMORY. IF A PARITY ERROR TRAP OCCURS WHEN THE PROGRAM IS RELOCATED PROGRAM ACTION IS UNDEFINED. IF PARITY MEMORY IS AVAILABLE OR SELECTED THE 3 XOR 5 TEST PATTERN WILL FIND PARITY MEMORY ONLY. DO NOT POWER FAIL THE PROGRAM WHEN THE PROGRAM IS RUNNING IN MOS MEMORY OR RELOCATED.

CHAPTER 7

MISCELLANEOUS

IF THE PROGRAM HALTS IN THE TRAP/INTERRUPT VECTOR AREA (0-1000), EXAMINE REGISTER 6 (THE STACK PTR). R6 CONTAINS THE ADDRESS WHERE THE PC OF THE INSTRUCTION THAT CAUSED THE TRAP ABORT IS STORED. SEE ALSO R1 (R1 SPECIFIES THE LAST TEST COMPLETED).

NOTE

THE PDP11/45 WILL DISPLAY THE TRAP VECTOR ADDRESS+4 IN THE ADDRESS LIGHTS. THUS, A TRAP TO 4 (BUS ERROR) WILL DISPLAY 10 IN THE ADDRESS LIGHTS.

7.1 STACK POINTER

THE STACK POINTER IS INITIALLY SET TO 600, AND IS RESET TO THIS VALUE AT THE START OF EACH SUBTEST.

7.2 PASS COUNT

SEVEN PASSES ARE REQUIRED FOR COMPLETION OF THIS PROGRAM, AT WHICH TIME AN "*" WILL BE PRINTED. THE PASS COUNT MAY BE OBSERVED BY TURNING THE SWITCH TO THE DISPLAY POSITION. (THE PASS COUNT IS ALSO STORED IN LOCATION 1000.) THE PASS COUNT SHOULD BE MONITORED IN THE EVENT THAT THE PROGRAM ENTERS AN UNDEFINED LOOP. NOTE THAT BIT 15 OF THE DISPLAY REGISTER IS NOT PART OF THE PASS COUNT. BIT 15, IF ON, INDICATES THAT THE PROGRAM IS IN ITS RELOCATED CYCLE.

615 7.3 ERROR COUNT

EACH TIME AN ERROR OCCURS, THE ERROR COUNT IS INCREMENTED. THE ERROR COUNT CAN BE OBSERVED BY TURNING THE SWITCH TO THE DISPLAY POSITION AND SETTING SWITCH 9. THE ERROR COUNT IS ALSO STORED IN LOCATION 1002.) THE PROGRAM WILL COUNT 1777(OCTAL) ERRORS; THE ERROR COUNT IS NOT INCREMENTED PAST THIS VALUE. NOTE THAT BIT 15 OF THE DISPLAY REGISTER IS NOT PART OF THE ERROR COUNT. BIT 15, IF ON, INDICATES THAT THE PROGRAM IS IN ITS RELOCATED CYCLE.

7.4 DISPLAY REGISTER

EITHER THE PASS COUNT OR THE ERROR COUNT IS DISPLAYED IN THE DISPLAY REGISTER. THE COUNT TO BE DISPLAYED IS CONTROLLED BY THE SETTING OF SWITCH 9. BIT 15 OF THE DISPLAY REGISTER, HOWEVER, IS USED AS A RELOCATION INDICATOR AND IS NOT PART OF EITHER THE PASS COUNT OR THE ERROR COUNT. WHEN BIT 15 IS ON, THE PROGRAM IS PERFORMING A RELOCATED CYCLE. WHEN THE PROGRAM IS RELOCATED, THE SPECIAL RESTART PROCEDURES OF CHAPTER 2 MUST BE FOLLOWED.

7.5 PROGRAM RELOCATION

WHEN THE PROGRAM IS RELOCATED, VERIFICATION IS MADE THAT THE PROGRAM HAS BEEN RELOCATED CORRECTLY. IF THE PROGRAM CANNOT BE RELOCATED UPWARD, THE RELOCATED TEST PHASE IS BYPASSED. IF AN ERROR OCCURS WHILE RELOCATING THE PROGRAM BACK TO THE LOWER 4K, AN ERROR MESSAGE IS TYPED AND THE PROGRAM HALTS. CONTINUING THE PROGRAM RETRIES THE DOWNWARD RELOCATION. DOWNWARD RELOCATION WILL BE ATTEMPTED UNTIL IT IS SUCCESSFUL OR THE PROGRAM IS RELOADED.

7.6 POWER FAIL

THE PROGRAM MAY BE POWER FAILED WHEN RUNNING. WHEN THE POWER RETURNS THE PROGRAM WILL CONTINUE IN SEQUENCE.

CAUTION

PROGRAM ACTION IS UNDEFINED IF THE PROGRAM IS RELOCATED OR IN MOS MEMORY.

DO NOT TURN POWER OFF/ON UNTIL THE MESSAGE 'POWER FAILED' HAS BEEN TYPED. THIS IS BECAUSE THE STACK MAY OVERFLOW.

7.7 EXECUTION TIME

```
672
673                                      EXECUTION TIME IS DEPENDENT ON TYPE OF PROCESSER, TYPE OF MEMORY, AND
```

PDP-11 D-12?A MEMORY EXERCISER PAGE 1B
MISCELLANEOUS

AMOUNT OF MEMORY. SOME REPRESENTATIVE TIMES (PER PASS) ARE:

11 05 WITH 28K MEMORY - 1 MIN.
11 45 WITH 96K MEMORY - 3 MIN.

CHAPTER 9

PROGRAM DESCRIPTION

THE PROGRAM VERIFIES EACH ADDRESS BY WRITING THE VALUE OF EACH ADDRESS INTO ITSELF STARTING AT LOCATION 20000 AND ENDING AT THE LAST LOCATION IN MEMORY. THE VALUE OF THE LAST LOCATION +2 IS TYPED ON THE TTY. NEXT THE VALUES WRITTEN ARE VERIFIED. TO COMPLETE THE ADDRESS TEST THE COMPLEMENT VALUE OF EACH MEMORY ADDRESS IS WRITTEN STARTING AT THE LAST MEMORY ADDRESS AND ENDING AT ADDRESS 20000. THE WRITTEN COMPLEMENT VALUES ARE THEN VERIFIED. THE NEXT PHASE OF TESTING INCLUDES READING, WRITING AND CHECKING MEMORY USING SEVERAL WORST CASE NOISE TEST PATTERNS (1 XOR 8, 3 XOR 9, AND 8 XOR 13). A SUBTEST IS DEDICATED TO CHECKING EACH PATTERN. THE TEST PROCEEDS BY EXERCISING EACH BANK OF MEMORY USING THE TEST PATTERNS NOTED ABOVE. NOTE THAT WITH THE MEMORY MANAGEMENT OPTION INSTALLED THAT ALL ADDRESSES ARE WRITTEN, READ AND CHECKED WITH THE MEMORY MANAGEMENT ENABLED. AFTER ALL MEMORY FROM 20000 TO THE LAST ADDRESS HAS BEEN TESTED, THE PROGRAM RELOCATES TO THE NEXT 4K MEMORY BANK AND TESTS LOCATIONS 0-17776 USING (1 XOR 8). THE PROGRAM THEN RELOCATES TO 40000 (100000 IF AVAILABLE) AND CHECKS MEMORY USING 3 XOR 9, AND 8 XOR 13 TEST PATTERN. THE PROGRAM THEN CHECKS MEMORY USING RANDOM DATA (RANTST). THIS ROUTINE MOVES THE PROGRAM CODE THROUGHOUT MEMORY STARTING AT LOCATION 20000, AND RELOCATES THE DATA BY A 32(DECIMAL) WORD OFFSET ON EACH SUBSEQUENT RELOCATION. I.E., FIRST RELOCATION IS TO 20000, NEXT IS TO 20100, THEN 20200, ETC. AFTER RELOCATION THE CODE MOVED IS CHECKED AGAINST THE ORIGINAL CODE (0-17776). WHEN THE RANDOM DATA TEST IS COMPLETE THE PROGRAM THEN SUCCESSIVELY ROTATES A '0' BIT (ROT0) AND A '1' BIT (ROT1) THROUGH ALL OF MEMORY. WHEN ALL TESTING IS COMPLETE THE PROGRAM RELOCATES TO ITS ORIGINAL POSITION, INCREMENTS THE PASS COUNT (LOCATION 1000) AND RESTARTS BEGINNING WITH THE WORST CASE NOISE TESTS. AN ASTERISK (*) WILL BE TYPED ON COMPLETION OF EACH PASS, AND WHEN 8 PASSES HAVE BEEN COMPLETED THE PROGRAM WILL TYPE 'DZQMB DONE' AND RESTART THE PROGRAM BEGINNING WITH THE MEMORY ADDRESS TESTS.

T38 8.1 PROGRAM 2 (USER SELECTIONS)

THIS PROGRAM IS PROVIDED TO ALLOW THE USER TO SPECIFY CERTAIN TEST PARAMETERS AS SHOWN BELOW:

1. ENABLE/DISABLE PARITY ERROR INTERRUPTS
2. STARTING BANK NUMBER FOR TEST
3. NUMBER OF 4K BANKS TO TEST
4. PATTERN TO BE USED

NOTE

ALL INPUTS ARE IN OCTAL.

8.1.1 PROGRAM 2 USER PARAMETERS

1. ENABLE PARITY? 1/0 = YES/NO. TYPE 1 TO ENABLE INTERRUPT ON PARITY ERROR. TYPE 0 TO DISABLE INTERRUPT.

2. STARTING BANK #(8)? TYPE THE 4K BANK WHERE YOU WISH TO BEGIN TESTING.

TYPE	TO START AT	TYPE	TO START AT
0	000000		
1	020000	20	400000
2	040000	21	420000
3	060000	22	440000
4	100000	23	460000
5	120000	24	500000
6	140000	25	520000
7	160000	26	540000
10	200000	27	560000
11	220000	30	600000
12	240000	31	620000
13	260000	32	640000
14	300000	33	660000
15	320000	34	700000
16	340000	35	720000
17	360000	36	740000

NOTE

TYPE ONLY NUMBERS SHOWN!!!

3. INUMBER OF 4K BANKS TO TEST (8)? TYPE IN OCTAL THE NUMBER OF 4K BANKS TO TEST.

4. PATTERN #?

K02


```
795
796                                    TYPE      TO SELECT
```

0	1 XOR 8 TEST PATTERN
1	3 XOR 9 TEST PATTERN
2	8 XOR 13 TEST PATTERN
3	USER CONSTANT
4	ROTATING 0
5	ROTATING 1
6	3 XOR 9 PARITY PATTERN
7	0,1,2,4,5 ABOVE

NOTE

PROGRAM WILL NOT ALLOW AN ODD NUMBER OF 4K BANKS TO BE TESTED IF PATTERN 2 OR 7 IS SELECTED. IF PATTERN #3 IS SELECTED THE PROGRAM WILL REQUEST A CONSTANT. TYPE A 6 DIGIT OCTAL NUMBER. TO ENTER A NEW CONSTANT TYPE AN 'A' AND WAIT FOR THE PROGRAM TO RESPOND. THE STARTING ADDRESS IS 214.

8.1.2 PROGRAM 2 USE

PROGRAM 2 CAN BE EFFECTIVELY USED TO MAKE PROPER ADJUSTMENTS TO A SPECIFIC MEMORY BANK AND ALSO TO 'MARGIN' MEMORY. THIS IS SO BECAUSE THE PROGRAM IS NOT RUNNING IN THE MEMORY BANK(S) BEING ADJUSTED/MARGINED. THUS ALL MEMORY FROM 0-124K MAY BE ADJUSTED/MARGINED. PARITY SHOULD BE DESELECTED WHEN MAKING ANY ADJUSTMENTS PARTICULARLY WHEN TESTING THE FIRST 4K BANK(S).

8.2 PROGRAM 3

THIS PROGRAM IS THE SAME AS PROGRAM 2 WITH THE FOLLOWING EXCEPTIONS:

1. INSTEAD OF NUMBER OF 4K BANKS TO TEST, TYPE NUMBER OF 256(DECIMAL),400(OCTAL) WORD BLOCKS TO TEST.

2. DO NOT SELECT PATTERN 2 OR 7.

THE STARTING ADDRESS IS 220.

8.3 PROGRAM 4

PROGRAM 4 CAN BE USED TO WRITE/READ USER DEFINED DATA INTO ANY SINGLE ADDRESS. THE PROGRAM WRITES THE DATA AND CHECKS IT.

```
THE PROGRAM WILL REQUEST AN 18 BIT ADDRESS AND IF SWITCH 0 = 0, A 16
BIT CONSTANT (DATA). IF SWITCH 0 = 1 THE PROGRAM WILL TYPE THE
```

PROGRAM DESCRIPTION

CONTENTS OF SEQUENTIAL ADDRESSES UNTIL EITHER SWITCH 9 = 0 OR A NEW ADDRESS IS ENTERED.

TO ENTER A NEW ADDRESS AND CONSTANT TYPE AN 'A' AND WAIT FOR THE PROGRAM TO RESPOND.

THE STARTING ADDRESS IS 224.

3.4 PROGRAM 5

PROGRAM 5 IS A TOGGLE IN MEMORY ADDRESS TEST. THIS TEST IS USEFUL WHEN AN ADDRESS SELECTION FAILURE IS SUSPECTED INVOLVING THE FIRST 4K OF MEMORY. THIS PROGRAM WRITES THE VALUE OF EACH ADDRESS INTO ITSELF STARTING WITH THE LOWER LIMIT AND CONTINUING TO THE UPPER LIMIT. AFTER ALL ADDRESSES HAVE BEEN WRITTEN EACH ADDRESS IS CHECKED FOR THE CORRECT CONTENTS STARTING WITH THE UPPER LIMIT AND CONTINUING TO THE LOWER LIMIT.

```
        LOCATION  CONTENTS       MNEMONIC     COMMENT

          10      012700         MOV #50,R0   ;GET FIRST ADDRESS
        * 12      000050                      ;TO TEST
          14      010001         MOV R0,R1    ;SAVE IN R1
          16      020037    1$:  CMP R0,@#SWR ;CHECK UPPER LIMIT
          20      177570                      ;(IN SWITCH REGISTER)
          22      001403         BEQ 2$       ;BRANCH IF AT UPPER LIMIT
          24      010010         MOV R0,(R0)  ;LOAD VALUE INTO ADDRESS
          26      005720         TST (R0)+    ;STEP TO NEXT ADDRESS
          30      000772         BR 1$        ;LOOP UNTIL DONE

          32      010004    2$:  MOV R0,R4    ;SAVE UPPER LIMIT
          34      020001    3$:  CMP R0,R1    ;CHECK IF AT LOWER LIMIT
        * 36      001767         BEQ 1$       ;BRANCH IF DONE
          40      024000         CMP -(R0),R0 ;CHECK DATA WRITTEN
          42      001774         BEQ 3$       ;BRANCH IF OK
          44      000000         HALT         ;ERROR
          46      000772         BR 3$        ;LOOP BACK
```

AFTER TOGGLING THE PROGRAM LA=10 **SET UPPER LIMIT**, START.

NOTE

THE UPPER LIMIT ADDRESS OBTAINED FROM THE SWITCH REGISTER MAY BE CHANGED DURING PROGRAM OPERATION. HOWEVER, OCCASIONALLY THE PROGRAM MAY HALT BECAUSE OF 'SWITCH BOUNCE'. (THE BEST PROCEDURE WHEN CHANGING LIMITS IS TO STOP THE PROGRAM MAKE THE CHANGE AND

8.5 PROGRAM 6

PROGRAM 6 IS ALSO A TOGGLE IN PROGRAM TO BE USED WITH PROGRAM 5 FOR MORE COMPLETE ADDRESS TESTING. THIS PROGRAM WRITES THE COMPLEMENT VALUE OF EACH ADDRESS INTO ITSELF STARTING WITH THE UPPER LIMIT AND CONTINUING TO THE LOWER LIMIT. AFTER ALL ADDRESSES HAVE BEEN WRITTEN EACH ADDRESS IS CHECKED FOR THE CORRECT CONTENTS STARTING WITH THE LOWER LIMIT ADDRESS AND CONTINUING TO THE UPPER LIMIT. TOGGLE IN THE FOLLOWING PATCHES TO PROGRAM 5 ABOVE.

LOCATION	CONTENTS		MNEMONIC	COMMENT
12	100			;CHANGE LOWER LIMIT
36	001404		BEQ 4$	;BRANCH TO PROGRAM 6
50	010402	4$:	MOV R4,R2	;GET UPPER LIMIT
52	005142	5$:	COM -(R2)	;COMPLEMENT ADDRESS
54	020201		CMP R2,R1	;CHECK IF AT LOWER LIMIT
56	001375		BNE 5$	;LOOP UNTIL DONE
60	020204	6$:	CMP R2,R4	;CHECK IF AT UPPER LIMIT
62	001755		BEQ 1$	;GO TO PROGRAM 5 IF DONE
64	010203		MOV R2,R3	;GET VALUE OF ADDRESS
66	005103		COM R3	;COMPLEMENT VALUE
70	020322		CMP R3,(R2)+	;CHECK ADDRESS
72	001772		BEQ 6$	;BRANCH IF OK
74	000000		HALT	;ERROR
76	[illegible]		BR 6$	;GO CHECK NEXT ADDRESS

CHAPTER 9

BRANCH GOBBLE MOS TEST

9.1 ABSTRACT

THE BRANCHGOBBLE PROGRAM IS USED TO TEST MOS MEMORY. CONTIGUOUS LOCATIONS ARE TESTED BETWEEN TWO LIMITS IN A MINIMUM 8K, MAXIMUM 124K MEMORY MACHINE. IF PARITY IS AVAILABLE IT IS ENABLED.

9.2 OPERATING PROCEDURE

1. LOADING: LOAD THE DZCMBG PROGRAM INTO MEMORY USING THE ABSOLUTE LOADERS.

2. STARTING: LOAD ADDRESS 200 AND PRESS THE START BUTTON.

3. THE PROGRAM WILL FIRST IDENTIFY ITSELF ON TTY:

   BRANCH GOBBLE

4. THEN THE ABSOLUTE LOADER WILL BE SAVED.

5. A CHECK WILL BE MADE FOR PARITY REGISTERS. IF NONE ARE FOUND THE MESSAGE:

   NO PARITY

   WILL BE TYPED TO THE USER. IF PARITY IS FOUND IT IS TURNED ON, AND THE MESSAGE,

   PARITY ENABLED

   WILL BE TYPED TO THE USER. THIS WILL BE FOLLOWED BY A LIST OF THE UNIBUS ADDRESSES OF THE PARITY REGISTERS FOUND AND ENABLED.

999
1000

6. THE USER WILL THEN BE ASKED IF HE WANTS THE PENDING TEST TO

```
        BE RUN USING MEMORY MANAGEMENT.

             USE KT11?  (Y OR N)

        IF THE USER TYPES Y  THEN  MEMORY  MANAGEMENT  WILL  BE  USED
        DURING  THE  PENDING  TEST.  IF  HE  TYPES  N,  THEN  MEMORY
        MANAGEMENT WILL NOT BE USED.  TYPING ANYTHING ELSE OTHER THAN
        Y OR N WILL CAUSE THE QUESTION TO BE REPEATED.

     7. THE USER WILL THEN BE ASKED TO GIVE THE LIMITS  OF  THE  TEST
        SPAN:

             HIGH LIMIT?

        AND:

             LOW LIMIT?

        RESTRICTIONS ON THE USER'S RESPONSE ARE:

        A.  THE  NUMBERS  MUST  BE  VALID  18-BIT,  6-DIGIT  OCTAL
            ADDRESSES, REAL NOT VIRTUAL.

        B.  THE NUMBERS SHOULD BE MULTIPLES OF 100 (OCTAL).

        C.  THE HIGH LIMIT MUST BE GREATER THAN THE LOW LIMIT.

        D.  IF MEMORY MANAGEMENT IS NOT USED, HIGH LIMIT MUST BE LESS
            THAN OR EQUAL TO 160000.

        E.  HIGH LIMIT CAN BE 1 + THE HIGHEST REAL CORE ADDRESS.  FOR
            EXAMPLE, IN AN 8K MACHINE, HIGH LIMIT CAN EQUAL 40000.

        VIOLATIONS TO THESE RESTRICTIONS WILL BE DEALT WITH  IN  THIS
        WAY:

        A.  A QUESTION MARK AND THE PROMPT WILL BE ISSUED:

                  ?

            THE USER IS THEN EXPECTED  TO  INPUT  THAT  LIMIT  AGAIN;
            THIS TIME CORRECTLY.

        B.  WHAT EVER THE LAST TWO OCTAL DIGITS OF THE  NUMBER  WHICH
            THE USER TYPED THEY WILL BE ASSEMBLED AS ZEROES.

        C.  THE USER WILL BE ASKED FOR OTHER LIMITS BY REPEATING THIS
            STEP (7).  BEFORE STEP (7) IS REPEATED
```

```
1357
1358                                    NOT VALID!
```

WILL BE TYPED.

D. SAME AS 3.

8. THE TEST STARTS. THE TEST WILL LOOP INDEFINITELY BETWEEN THE TWO LIMITS UNLESS THE USER HALTS THE PROGRAM OR AN ERROR IS ENCOUNTERED. AN ASTERISK IS TYPED AT THE BEGINNING OF EACH PASS MODE.

9. TO STOP THE TEST RUNNING AND START ANOTHER HIT THE HALT SWITCH AND RETURN STEP 2. ANY TEST BUT THE FIRST WILL NOT INCLUDE STEP 4.

10. TO STOP THE TEST AND RESTORE THE LOADER, HIT THE HALT SWITCH, LOAD ADDRESS 162 AND START. WHEN THE LOADER IS RESTORED THE PROGRAM WILL HALT AT LOCATION 200.

11. TO STOP THE TEST AND START THE 0-124K MEMORY TEST HIT THE HALT SWITCH, LOAD ADDRESS 200, AND START.

12. DATA LIGHTS. THE DATA LIGHTS WILL DISPLAY THE CURRENT LOCATION BEING TESTED DURING A BRANCH GOBBLE TEST. WHEN A MOS MEMORY FAILURE OCCURS THESE LIGHTS WILL CONTAIN VIRTUAL (16-BIT) ADDRESS "NEAR" THE FAILURE.

9.3 ERRORS

1. ERRORS IN OPERATING THE PROGRAM ARE DESCRIBED IN OPERATING PROCEDURE 9.2.

2. IF A PARITY ERROR IS DETECTED THE USER IS TOLD THE PC+2 AT THET IME OF THE ERROR:

   PARITY ERROR
   PC=XXXXXX

   THEN THE SCAN IS MADE THROUGH ALL OF MEMORY TO TRY TO FORCE THE ERROR TO ARISE AGAIN. IF IT IS NOT FOUND THE MESSAGE

   SCAN COMPLETE

   IS TYPED AND THE TEST IS RESTARTED.

   IF THE ERROR IS DETECTED ON THE SCAN THEN THE USER IS GIVEN THE ADDRESS OF THE LOCATION CAUSING THE PARITY ERROR AND THE CONTENTS OF THAT LOCATION:

   XXXXXX HAD BAD DATA XXXXXX

   IF MEMORY MANAGEMENT WAS OFF DURING THE SCAN FOR THE ERROR

```
1115                                            THE ADDRESS GIVEN FOR THE ERROR IS REAL PC:
```

KT11 OFF

IS TYPED.

IF MEMORY MANAGEMENT WAS ON DURING THE SCAN:

KT11 ON PAR=XXXXX

IS TYPED, WHERE THE PAR (PAGE ADDRESS REGISTER) GIVEN IS THAT PAR WHICH SHOULD BE USED IN RELOCATING THE VIRTUAL ADDRESS GIVEN FOR THE ERROR ONTO A REAL CORE ADDRESS. THE METHOD FOR THIS RELOCATION IS GIVEN IN THE NOTE BELOW.

AFTER ANY PARITY IS ENCOUNTERED AND THE USER NOTIFIED, THE TEST WILL BE RESTARTED.

3. WHENEVER THE BRANCH GOBBLE TEST BRINGS OUT AN ERROR IN MOS MEMORY IT MAY SURFACE AS A PARITY AND BE HANDLED AS DESCRIBED ABOVE. OTHERWISE THE ADDRESS (VIRTUAL IF MEMORY MANAGEMENT IS ON) IN THE DATA LIGHTS WILL DESIGNATE THE VICINITY OF THE ERROR. IF MEMORY MANAGEMENT IS ON, RELOCATE THE ADDRESS IN THE DATA LIGHTS IN THE MANNER DESCRIBED IN THE NOTE BELOW.

NOTE

TO COMPUTE THE REAL ADDRESS OF AN ADDRESS RELOCATED BY MEMORY MANAGEMENT, ADD THE LOW ORDER 13-BITS OF THE VIRTUAL ADDRESS TO THE CORRESPONDING PAR SHIFTED, 6 BITS TO THE LEFT:

VIRTUAL ADDRESS = 0 00X XXX XXX XXX XXX

PAR = YYY YYY YYY YYY 000 000

REAL ADDRESS = ZZZ ZZZ ZZZ ZZZ ZZZ ZZZ

TO DETERMINE WHICH PAR TO USE REMEMBER THAT ON KERNEL SPACE IS USED IN ANY TEST HERE. TAKE THE HIGH ORDER 3-BITS OF THE VIRTUAL ADDRESS AND USE THEM TO DESIGNATE THE KIPAR TO USE. FOR INSTANCE, IF THE VIRTUAL ADDRESS IS 031676 USE KIPAR1 BECAUSE THE UPPER 3 BITS OF THE VIRTUAL ADDRESS ARE 001=1.

4. IF AN ERROR CONDITION ARISES, EITHER AS A PARITY ERROR OR ONE NOT A APPARENT SUCH AS THE PROGRAM HALTS OR IT IS CLEAR FROM THE ADDRESS AND DISPLAY LIGHTS THAT THE PROGRAM IS NOT RUNNING ITS NORMAL COURSE, THE USER CAN ENTER CONSOLE MODE AN EXAMINE THE CONTENTS OF THE MEMORY LOCATIONS STARTING AT THE

```
                                        ADDRESS IN THE DISPLAY REGISTER. THE CONTENTS OF THESE
                                        LOCATIONS SHOULD BE COMPARED TO THE CONTENTS IN THE
```

LISTINGS) OF LOCATIONS 15226 THROUGH 15304 (WITH THE
EXCEPTIONS: 15232 IS UNDETERMINABLE AND 15304 SHOULD CONTAIN
EITHER 15370 OR 35370). IN THIS WAY THE USER SHOULD BE ABLE
TO DETERMINE WHICH BITS WERE LOST IN WHAT WORDS.

9.4 PROGRAM DESCRIPTION

THIS VERSION OF THE BRANCH GOBBLE TEST IS TAKEN ALMOST DIRECTLY FROM
THE DZQKA-A INSTRUCTION EXERCISER WHICH CONTAINED THE ORIGINAL BRANCH
GOBBLE. WHAT HAS BEEN DONE HERE IS TO GIVE THAT TEST AN INTERFACE TO
THE USER AND MEMORY MANAGEMENT FACILITIES. THESE ADDITIONS HAVE BEEN
DONE IN A WAY WHICH ALLOWS THE TEST TO RUN AS IT DID IN ITS ORIGINAL
FORM. DATA IS COLLECTED FROM THE USER AND IF MEMORY MANAGEMENT IS
NEEDED IT IS SET UP AND THAT TEST IS ALLOWED TO RUN BETWEEN THE
DESIGNATED LIMITS.

```
        .
        .NLIST  MD,MC
        .LIST   ME
        .ABS
        .MCALL  $TYPE
        .TITLE  TEST DZQMB-G D-124K MEMORY EXERCISER
        .SBTTL  STARTING INST & DEFINITIONS
;COPYRIGHT 1973 DIGITAL EQUIPMENT CORP., MAYNARD,MASS.

;THIS TEST CHECKS THAT ALL MEMORY ADDRESSES ARE UNIQUE USING ADDRESS TESTS
;AND CHECKS DATA RELIABILITY OF MEMORY USING WORST CASE NOISE TEST PATTERNS
;A RANDOM # PATTERN (PROGRAM CODE RELOCATED), A ROTATING 0 AND ROTATING
;1 PATTERN.
;ALSO INCLUDED ARE USER TESTS WHICH CAN BE USED TO TEST SPECIFIED SEG-
;MENTS OF MEMORY USING THE PATTERNS MENTIONED ABOVE. ADDITIONALLY A
;28 WORD TOGGLE IN PROGRAM IS DOCUMENTED (SEC 9.5 OF THE DOCUMENT) WHICH
;CAN BE USED IF AN ADDRESSING MALFUNCTION IS SUSPECTED INVOLVING THE FIRST
;4K OF MEMORY.
;A MOS MEMORY TEST HAS BEEN ADDED, THE BRANCH GOBBLE ROUTINE.
;THE PROGRAM MAY BE POWER FAILED WHEN RUNNING. THE PROGRAM WILL PRINT
;A MESSEGE (POWER FAILED) AND CONTINUE IN SEQUENCE WHEN THE POWER COMES
;BACK UP. **CAUTION** DO NOT POWER FAIL THE PROGRAM IF THE PROGRAM IS IN
;MOS MEMORY OR IF THE PROGRAM IS RELOCATED.

;LOADING AND STARING INSTRUCTIONS
;LOAD ADDRESS 200 AND START
;NOTE:  PROGRAM WILL RUN WORST CASE TEST PATTERNS IN LOWEST 4K
;THUS THE PROGRAM CANNOT BE RESTARTED AT 200 IF RELOCATED. TO PREVENT
;RELOCATION FROM OCCURING DEPOSIT 200 INTO LOCATION 42 (NOT NECESSARY
;IF LOADED VIA ACT11). THIS ACTION WILL PREVENT RELOCATION AND ALSO
;INHIBIT TESTING MEMORY IN LOWEST 4K.
;THIS PROGRAM ALSO RELOCATES THE ABS AND BOOT LOADERS TO ALLOW TESTING
;OF MEMORY. TO RESTORE THE LOADERS RESTART AT 162.
;       STACK POINTER IS SET AT 500
```

```
                                  ;       AN ASTERISK '*' WILL BE PRINTED ON COMPLETION OF EACH PASS, AND
                                  ;       THE PROGRAM NAME WILL BE PRINTED WHEN TEST IS COMPLETE.

                                  ;GENERAL REGISTER ASSIGNMENTS
        000000                            R0=%0
        000001                            R1=%1
        000002                            R2=%2
        000003                            R3=%3
        000004                            R4=%4
        000005                            R5=%5
        000006                            SP=%6
        000007                            PC=%7
        000000                            R10=%0
        000001                            R11=%1
        000002                            R12=%2
        000003                            R13=%3
        000004                            R14=%4
        000005                            R15=%5

                                  ;STATUS REGISTER (PSW) BIT ASSIGNMENTS
        000001                            C=1                     ;C BIT
        000002                            V=2                     ;V BIT
        000004                            Z=4                     ;Z BIT
        000010                            N=10                    ;N BIT
        000020                            T=20                    ;'T' BIT
        000340                            PRTY7=340               ;PRIORITY LEVEL 7
        000200                            PRTY4=200               ;PRIORITY LEVEL 4
        000000                            KM=000000               ;KERNEL MODE
        040000                            SM=040000               ;SUPERVISORY MODE
        140000                            UM=140000               ;USER MODE
        000000                            PKM=000000              ;PREVIOUS KERNEL MODE
        010000                            PSM=010000              ;PREVIOUS SUPERVISORY MODE
        030000                            PUM=030000              ;PREVIOUS USER MODE
        004000                            REG=004000              ;SELECT R10-R15

                                  ;VECTOR ADDRESSES
        000004                            ERRVEC=4                ;ADDRESS OF ERROR VECTOR
        000010                            RESVEC=10               ;ADDRESS OF RESERVED INST. TRAP VECTOR
        000014                            TBITVEC=14              ;ADDRESS OF 'T' BIT TRAP VECTOR
        000014                            TRTVEC=14               ;ADDRESS OF 'TRACE' TRAP VECTOR
        000014                    BPTVEC=14               ;ADDRESS OF 'BREAKPOINT' TRAP VECTOR
        000020                            IOTVEC=20               ;ADDRESS OF IOT TRAP VECTOR
        000024                            PFVEC=24                ;ADDRESS OF POWER FAIL TRAP VECTOR
        000030                            EMTVEC=30               ;ADDRESS OF EMT VECTOR
        000034                            TRAPVEC=34              ;ADDRESS OF TRAP VECTOR
        000060                            TKVEC=60                ;ADDRESS OF TTY KEYBOARD INTERRUPT VECTOR
        000064                            TPVEC=64                ;ADDRESS OF TTY PRINTER INTERRUPT VECTOR
        000240                            PIRVEC=240              ;ADDRESS OF PIRQ VECTOR
        000244                            FPEVEC=244              ;ADDRESS OF FLOATING POINT INT. VECTOR
        000250                            MMVEC=250               ;ADDRESS OF MEM MGMT ERROR TRAP VECTOR

                                  ;REGISTER ADDRESSES
        177776                            PSW=177776              ;ADDRESS OF STATUS REGISTER
        177774                            SLR=177774              ;ADDRESS OF STACK LIMIT REGISTER
        177772                            PIRQ=177772             ;ADDRESS OF PROGRAM INTERRUPT REQUEST
        177770                            UBREAK=177770           ;ADDRESS OF MICRO BREAK REGISTER
```

```
1296    177560                          TKS=177560              ;ADDRESS OF KEYBOARD CSR
1297    177562                          TKB=177562              ;ADDRESS OF KEYBOARD BUFFER
1298    177564                          TPS=177564              ;ADDRESS OF TELEPRINTER CSR
1299    177566                          TPB=177566              ;ADDRESS OF TELEPRINTER BUFFER
1300    177570                          SWR=177570              ;ADDRESS OF CONSOL SWITCH REGISTER
1301    177570                          DISPLAY=177570          ;ADDRESS OF CONSOL DISPLAY REGISTER
1302
1303                            ;INITIAL STACK POINTER SETTING
1304    000500                          STKPTR=500
1305
1306                            ;MISCELLANEOUS BIT ASSIGNMENTS
1307    000100                          BIT15=  100
1308    040000                          BIT14=  040000
1309    020000                          BIT13=  020000
1310    010000                          BIT12=  010000
1311    001000                          BIT9=   001000
1312    000400                          BIT8=   000400
1313    000100                          BIT6=   000100
1314
1315                            ;MEMORY MANAGEMENT REGISTER ADDRESS ASSIGNMENTS
1316    177572                          SR0=177572              ;ADDRESS OF MEM MGMT REGISTER SR0
1317    177574                          SR1=177574              ;     "    "    "          "    SR1
1318    177576                          SR2=177576              ;     "    "    "          "    SR2
1319    172516                          SR3=172516              ;ADDRESS OF MEM MGMT REGISTER SR3
1320
1321    172300                          KIPDR0=172300           ;ADDRESS OF KERNEL 'I' PAGE
1322    172302                          KIPDR1=172302           ;DESCRIPTOR REGISTERS
1323    172304                          KIPDR2=172304
1324    172306                          KIPDR3=172306
1325    172310                          KIPDR4=172310
1326    172312                          KIPDR5=172312
1327    172314                          KIPDR6=172314
1328    172316                          KIPDR7=172316
1329
1330    172340                          KIPAR0=172340           ;ADDRESSES OD KERNEL 'I' SPACE
1331    172342                          KIPAR1=172342           ;PAGE ADRESS REGISTERS
1332    172344                          KIPAR2=172344
1333    172346                          KIPAR3=172346
1334    172350                          KIPAR4=172350
1335    172352                          KIPAR5=172352
1336    172354                          KIPAR6=172354
1337    172356                          KIPAR7=172356
1338
1339
1340                            ;INSTRUCTION EQUATES
1341    104400                          HLT=TRAP
1342    104000                          SCOPE=EMT               ;SCOPE IS AN EMT TRAP
1343
1344                            ;MISC. EQUATES
1345    000006                          RW=6                    ;R/W BIT IN PDR REGISTERS
1346    000C00                          UP=0                    ;UP BIT IN PDR REGISTERS
1347
1348
1349
1350
1351
```

```
                                        .=[illegible]
                                        .WORD   [illegible]             ;SPECIAL TRAP INTERRUPT [illegible]
                                        .WORD   [illegible]             ;[illegible]
                                                                        ;LOADED PROPERLY FROM [illegible]
                                        .WORD   ERRTRP
                                        .WORD   [illegible]
                                        .=TRAPVEC
                                        .WORD   ERROR
                                        .WORD   [illegible]
                                        .=46
000046                                  .WORD   LOGICAL
                                        .=52
000052                                  .WORD   BIT14
                                        .=120
                                SELF.:  .WORD   0
                                SAVPC2: .WORD   0
                                ;THE SUBROUTINE WHERE IS CALLED BEFORE ANY TEST IS RUN TO SEE
                                ;IF BRANCH GOBBLE RELOCATED THE ENTIRE FIRST FOUR K OF CORE INTO
                                ;THE SECOND FOR K AND DIDN'T RELOCATE EVERYTHING BACK. IF THIS IS
                                ;THE CASE THE WHERE WILL PUT THE PROGRAM BACK INTO THE FIRST FOUR K
                                ;AND RETURN TO THE BEGINNING OF THE TEST THE USER DESIGNATED
                                ;BY LOADING HIS STARTING ADDRESS. NOTE THST THIS ROUTINE WILL NOT
                                ;RELOCATE THE PROGRAM IF IT HAS BEEN MOVED BY ANY OTHER SUBPROGRAM
                                ;EXCEPT THE BRANCH GOBBLE PROGRAM. THE RELOCATION OF THE PROGRAM
                                ;TO THE FIRST FOUR K IS INACTED BY THE USER IN THE SAME WAY IT WAS
                                ;IN THE DZQMBF AND DZQMBE VERSIONS OF THE 0-124 TEST. IF IT HAS BEEN
                                ;RELOCATED BY THE PART OF THIS TEST WHICH WAS TAKEN FROM DZQMB. THAT
                                ;IS ALL THIS PROGRAM EXCEPT BRANCH GOBBLE.
[illegible] 005737  000120      WHERE:  TST     @#120
[illegible] 100401                      BMI     1$
[illegible] 000207                      RTS     PC
[illegible] 011667  017762      1$:     MOV     (SP),SAVPC2+20000
000140 004567  026102                   JSR     R5,RELOC+20000
000144 020000                           .WORD   20000
000146 000000                           .WORD   0
000150 016716  177746                   MOV     SAVPC2,(SP)
000154 005037  000120                   CLR     @#120
000160 000207                           RTS     PC
                                ;
       000162                           .=162
000162 012706  000500           PONE:   MOV     #500,SP ;STARTING ADDRESS TO RELOCATE LOADERS.
[illegible] 004767  177732              JSR     PC,WHERE
000172 004767  001736                   JSR     PC,$RLDR
000176 000000                           HALT
[illegible] 012706  000500      PTWO:   MOV     #500,SP ;STARTING ADDRESS OF 0-124K MEMORY EXERCISER.
[illegible] 004767  [illegible]         JSR     PC,WHERE
```

```
000210  000137  002310             JMP     @#START          ;GO TO START OF TEST
000214  012706  000500   PTHREE:   MOV     #500,SP  ;STARTING ADDRESS OF PROGRAM #2.
000220  004767  177700             JSR     PC,WHERE
000224  000137  004630             JMP     @#PRG2           ;GO START PROGRAM #2
000230  012706  000500   PFOUR:    MOV     #500,SP  ;STARTING ADRESS OF PROGRAM #3.
000234  004767  177664             JSR     PC,WHERE
000240  000137  005706             JMP     @#PRG3           ;GO START PROGRAM #3
        000250                     .=250
000250  000000                     .WORD   0        ;MEMORY MANAGEMENT TRAP VECTOR.
000252  000000                     .WORD   0
000254  012706  000500   PFIVE:    MOV     #500,SP  ;START ADDRESS OF PROGRAM #4.
000260  004767  177640             JSR     PC,WHERE
000264  000137  005740             JMP     @#PRG4
000270  012706  000500   PSIX:     MOV     #500,SP  ;STARTING ADDRESS OF BRANCH GOBBLE MOS TEST.
000274  004767  177624             JSR     PC,WHERE
000300  000167  012134             JMP     BRANCH
                         ;

                         ;ROUTINE TO SAVE REGISTERS ON THE STACK
                         ;CALLED BY SAVE MACRO OR JSR    PC,$SAVR
000304  012767  000016   $SAVR:    MOV     (SP)+,1$         ;SAVE RETURN PC
000310  010546                     MOV     %5,-(SP)
000312  010446                     MOV     %4,-(SP)
000314  010346                     MOV     %3,-(SP)
000316  010246                     MOV     %2,-(SP)
000320  010146                     MOV     %1,-(SP)
000322  010046                     MOV     %0,-(SP)
000324  012707                     MOV     (PC)+,PC         ;RETURN
000326  000000           1$:       0                        ;CONTAINS RETURN ADDRESS

                         ;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
                         ;CALLED BY RESTORE MACRO OR JSR  PC,$RESTR
000330  012667  000016   $RESTR:   MOV     (SP)+,1$         ;SAVE RETURN PC
000334  012600                     MOV     (SP)+,%0
000336  012601                     MOV     (SP)+,%1
000340  012602                     MOV     (SP)+,%2
000342  012603                     MOV     (SP)+,%3
000344  012604                     MOV     (SP)+,%4
000346  012605                     MOV     (SP)+,%5
000350  012707                     MOV     (PC)+,PC         ;RETURN
000352  000000           1$:       0                        ;CONTAINS RETURN ADDRESS

                                   .SBTTL  POWER FAIL ROUTINE
        000502                     .=502
                         ;POWER FAIL ROUTINE
                         ;THE POWER DOWN ROUTINE SAVES THE KEYBOARD STATUS,THE GENERAL REGISTERS
                         ;(R0-R5),AND MEM MGMT REGISTERS (KIPDR0-KIPDR7,KIPAR0-KIPAR7,SR2,SR0)
                         ;ON THE STACK AND SAVES THE STACK POINTER IN PFSTK BELOW.
000502  013746  177560   PDWN:     MOV     @#TKS,-(SP)      ;SAVE KEYBOARD STATUS
000506  004767  177572             JSR     PC,$SAVR         ;GO SAVE REGISTERS ON THE STACK
000512  005737  000752             TST     @#MMAVA          ;CHECK IF MEM MGMT IS AVAILABLE
000516  001417                     BEQ     3$               ;BRANCH IF NOT AVAILABLE
000520  013746  177572             MOV     @#SR0,-(SP)      ;SAVE SR0
000524  013746  177576             MOV     @#SR2,-(SP)      ;SAVE SR2
000530  012700  172300             MOV     #KIPDR0,R0       ;GET ADDRESS OF KIPDR0
000534  012702  000010             MOV     #8.,R2
```

D04


```
000540  010203                        MOV     R2,R3
000542  012046                1$:     MOV     (R0)+,-(SP)     ;SAVE KIPDR0-KIPDR7
000544  077302                        SOB     R2,1$
000546  012700  172340                MOV     #KIPAR0,R0      ;GET ADDRESS OF KIPAR0
000552  012046                2$:     MOV     (R0)+,-(SP)     ;SAVE KIPAR0-KIPAR7
000554  077302                        SOB     R3,2$
000556  010627                3$:     MOV     SP,(PC)+        ;SAVE STACK PTR IN FOLLOWING LOCATION
000560  000000                PFSTK:  .WORD   0               ;CONTAINS STACK PTR AFTER POWER FAIL
000562  012737  000572  000024        MOV     #PUP,@#PFVEC    ;SET POWER FAIL VECTOR TO PUP ROUTINE
000570  000000                        HALT

                              ;POWER UP ROUTINE.
000572  000240                PUP:    NOP
000574  013706  000560                MOV     @#PFSTK,SP      ;SET STACK PTR
000600  005767  000146                TST     MMAVA           ;CHECK IF MEM MGMT IS AVAILABLE
000604  001417                        BEQ     4$
000606  012700  172360                MOV     #KIPAR7+2,R0    ;GET ADDRESS OF KIPAR7+2
000612  012702  000010                MOV     #8.,R2
000616  010203                        MOV     R2,R3
000620  012640                1$:     MOV     (SP)+,-(R0)     ;RESTORE KIPAR7-KIPAR0
000622  077302                        SOB     R3,1$
000624  012700  172320                MOV     #KIPDR7+2,R0    ;GET ADDRESS OF KIPDR7+2
000630  012640                2$:     MOV     (SP)+,-(R0)     ;RESTORE KIPDR7-KIPDR0
000632  077202                        SOB     R2,2$
000634  012637  177576                MOV     (SP)+,@#SR2     ;RESTORE SR2
000640  012637  177572                MOV     (SP)+,@#SR0     ;RESTORE SR0
000644  005767  006144        4$:     TST     PARAVA          ;CHECK IF PARITY REGISTERS ARE ENABLED
000650  001402                        BEQ     5$              ;BRANCH IF NOT
000652  004767  006056                JSR     PC,.MAMF        ;GO ENABLE PARITY REGISTERS
000656                        5$:
000656  004767  177446                JSR     PC,$RESTR       ;RESTORE REGISTERS FROM STACK
000662  012637  177560                MOV     (SP)+,@#TKS
000666  012737  000502  000024        MOV     #PDWN,@#PFVEC   ;SET POWER FAIL TRAP TO PDWN ROUTINE
000674  005027                        CLR     (PC)+
000676  000000                10$:    .WORD   0
000700  005267  177772        11$:    INC     10$             ;DELAY WAITING FOR TTY MOTOR
000704  100375                        BPL     11$
000706  004567  000046                JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
000712  000720                        PWRFAIL
000714  000240                6$:     NOP
000716  000002                        RTI                     ;RETURN

000720  005015  047520  042527  PWRFAIL:.ASCIZ  <15><12>'POWER FAILED'<15><12>
000726  020122  040506  046111
000734  042105  005015     000

                                      .SBTTL  TAGS & PRINT ROUTINE
        000742                        .EVEN
000742  000000                ICNT:   .WORD   0               ;CONTAINS PASS COUNT
000744  000000                ICOUNT: .WORD   0               ;CONTAINS ITERATION PATTERN
000746  000000                ERCNT:  0                       ;CONTAINS ERROR COUNT
000750  000000                LDDISP: 0                       ;CONTAINS DISPLAY REGISTER IMAGE
000752  000000                MMAVA:  0                       ;MEM MGMT AVAILABLE INDICATOR
                                                              ;0=NOT AVAIL,-1=AVAIL
000754  000000                RELOCF: .WORD   0               ;CONTAINS RELOCATION FACTOR
000756  000000                COUNT:  .WORD   0               ;TEMPORARY WORKING LOCATION
```

```
                                            ;ROUTINE TO PASS MESSAGE ADDRESS TO TYPE ROUTINE BELOW
                                            ;CALL:  JSR     R5,$PRINT
                                            ;       MESSAGE ADDRESS
        000762  000240                      $PRINT: NOP
        000762  012567  000016                      MOV     (R5)+,1$                ;GET MESSAGE ADDRESS
        000766  066767  177762  000010              ADD     RELOCF,1$               ;ADD RELOCATION FACTOR
        000774  013746  177776                      MOV     @#PSW,-(SP)             ;PUSH PSW ON THE STACK
        001000  004767  000014                      JSR     PC,.TYPE                ;CALL TYPE ROUTINE
        001004  000000                      1$:     .WORD   0                       ;CONTAINS MESSAGE ADDRESS
        001006  000205                              RTS     R5                      ;RETURN

                                            ;ROUTINE TO TYPE ASCII MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                            ;THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                            ;CALL:  TYPE
                                            ;               MESADR                  ;MESADR IS FIRST ADDRESS OF ASCIZ STRING
                                            ;

                                            ;TAGS USED BY THE TYPE ROUTINE BELOW
        001010     000                      $NULL:  .BYTE   0                       ;CONTAINS NULL CHARACTER
        001011     002                      $FILLS: .BYTE   2                       ;CONTAINS # OF FILLER CHARACTERS
        001012     000                      $TPFLG: .BYTE   0                       ;CONTAINS TELEPRINTER AVAILABLE FLAG
                                                                                    ;0/377 = AVAIL/NOT AVAIL
        001013     000                      $TKFLG: .BYTE   0                       ;CONTAINS KEYBOARD AVAILABLE FLAG
        001014  177564                      $TPS:   .WORD   177564                  ;ADDRESS OF TLEPRINTER STATUS REGISTER
        001016  177566                      $TPB:   .WORD   177566                  ;ADDRESS OF TELEPRINTER DATA BUFFER
        001020  010046                      .TYPE:  MOV     R0,-(SP)                ;SAVE R0
        001022  017600  000002                      MOV     @2(SP),R0               ;GET MESSAGE ADDRESS
        001026  062766  000002  000002              ADD     #2,2(SP)                ;ADJUST RETURN PC

        001034  112046                      1$:     MOVB    (R0)+,-(SP)             ;PUSH CHARACTER TO BE TYPED ONTO STACK
        001036  001003                              BNE     2$                      ;BRANCH IF NOT THE TERMINATOR
        001040  005726                              TST     (SP)+                   ;POP TERMINATOR CHAR OFF THE STACK
        001042  012600                              MOV     (SP)+,R0                ;RESTORE R0
        001044  000002                              RTI                             ;RETURN TO CALLER

        001046  004767  000026              2$:     JSR     PC,5$                   ;TYPE CHARACTER
        001052  122726  000012              3$:     CMPB    #12,(SP)+               ;CHECK IF CHARACTER WAS A LINE FEED
        001056  001366                              BNE     1$                      ;BRANCH IF NOT LINE FEED
        001060  016746  177724                      MOV     $NULL,-(SP)             ;GET # OF FILLERS REQUIRED AND FILLER
                                                                                    ;CHARACTER.

        001064  105366  000001              4$:     DECB    1(SP)                   ;DECREMENT FILLERS REQ. COUNT
        001070  002770                              BLT     3$                      ;BRANCH IF NO MORE FILLERS ARE REQUIRED
        001072  004767  000002                      JSR     PC,5$                   ;TYPE FILLER CHARACTER
        001076  000772                              BR      4$

        001100  105777  177710              5$:     TSTB    @$TPS                   ;WAIT FOR OUTPUT DEVICE
        001104  100375                              BPL     .-4
        001106  116677  000002  177702              MOVB    2(SP),@$TPB             ;OUTPUT CHARACTER
        001114  000207                              RTS     PC

                                            ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                            ;ERROR TRAP SERVICE ROUTINE
        001116  005737  177570              ERRTRP: TST     @#SWR                   ;CHECK IF HALT ON ERROR
        001122  100001                              BPL     .+4                     ;BRANCH IF NO HALT ON ERROR
        001124  000000                              HALT                            ;HALT
```

F04

```
TEST DZQMB-G 0-124K MEMORY EXERCISER   MACY11 27(732)   10-SEP-76  11:59  PAGE 45
DZQMBG.P11     TAGS $ PRINT ROUTINE

001126  005737                          TST     @#4             ;CHECK IF PREV TRAP TO 4 REPORTED
001130  000000                  1$:     .WORD   0               ;CONTAINS ERROR REPORTED FLAG
001132  001013                          BNE     2$              ;BRANCH IF NOT REPORTED
001134  010667  177772                  MOV     SP,1$           ;SET "NOT REPORTED"
001140  011602                          MOV     (SP),R2         ;GET PC OFF STACK
001142  004767  000356                  JSR     PC,$FORMO       ;GO TO FORMAT ROUTINE
001146  004567  177606                  JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
001152  001452                          TRAP4
001154  004567  177600                  JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
001160  002275                          DIGITS
001162  000000                  2$:     HALT                    ;ERROR! SECOND TRAP TO 4 OCCURRED
                                                                ;BEFORE FIRST WAS PRINTED
001164  005067  177740                  CLR     1$
001170  000137  000200                  JMP     @#200           ;RESTART AT 200

                                        .SBTTL  ERROR SERVICE ROUTINE
                                ;ERROR SERVICE CALLED BY JSR  PC,ERROR INSTRUCTION
                                ;OR HLT (A TRAP INST)
001174  000240                  ERROR:  NOP
001176  022767  017777  177542          CMP     #17777,ERCNT    ;CHECK FOR MAX ERROR CNT
001204  001403                          BEQ     4$
001206  062767  000001  177532          ADD     #1,ERCNT        ;INCREMENT ERROR COUNT
001214  032737  001000  177570  4$:     BIT     #BIT9,@#SWR     ;SWITCH 9 UP?
001222  001411                          BEQ     5$
001224  042767  017777  177516          BIC     #17777,LODISP   ;SAVE RELOCATION BITS
001232  056767  177510  177510          BIS     ERCNT,LODISP    ;LOAD ERROR COUNT
001240  016737  177504  177570          MOV     LODISP,@#DISPLAY        ;LOAD DISPLAY REGISTER
001246  005737  177570          5$:     TST     @#SWR           ;HALT ON ERROR
001252  100002                          BPL     .+6
001254  000000                          HALT
001256  000471                          BR      3$
001260  032737  020000  177570          BIT     #20000,@#SWR    ;PRINT OUT DESIRED?
001266  001052                          BNE     1$              ;BRANCH IF NO PRINTOUT
001270  004767  177010                  JSR     PC,$SAVR        ;GO SAVE REGISTERS ON THE STACK
001274  016602  000014                  MOV     14(SP),R2       ;GET PC OF ERROR CALL
001300  004767  000220                  JSR     PC,$FORMO       ;GO TO FORMAT ROUTINE
001304  004567  177450                  JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
001310  001467                          ERRPC
001312  004567  177442                  JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
001316  002275                          DIGITS
001320  016602  000004                  MOV     4(SP),R2        ;GET FAILING ADDRESS (IN R2
001324  004767  000174                  JSR     PC,$FORMO       ;GO TO FORMAT ROUTINE
001330  004567  177424                  JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
001334  002253                          ADRESS
001336  105767  005157                  TSTB    PENFLG          ;BRANCH IF PARITY ERROR DETECTED
001342  001017                          BNE     11$             ;BUT NOT FOUND
001344  105767  005150                  TSTB    PEFLG           ;BRANCH IF PARITY ERROR DETECTED
001350  001006                          BNE     10$             ;BUT FOUND
001352  004567  177402                  JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
001356  001473                          XMTDAT
001360  010046                          MOV     R0,-(SP)        ;PUSH VALUE TO TYPED ONTO STACK
001362  004767  000360                  JSR     PC,O2A          ;GO PRINT VALUE
001366                          10$:
001366  004567  177366                  JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
001372  001506                          RECDAT
001374  010346                          MOV     R3,-(SP)        ;PUSH VALUE TO BE TYPED ONTO STACK
```

```
001376  004767  000344                          JSR     PC,02A
001402                                  11$:
001402  004567  177352                          JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
001406  015020                                  $CREF
001410  004767  176714                          JSR     PC,$RESTR       ;RESTORE REGISTERS FROM STACK
001414  032737  002000  177570          1$:     BIT     #2000,@#SWR     ;RING BELL ON ERROR
001422  001403                                  BEQ     2$
001424  004567  177330                          JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
001430  001521                                  BELL
001432  005737  177570                  2$:     TST     @#SWR           ;HALT AFTER PRINT OUT
001436  100001                                  BPL     .+4
001440  000000                                  HALT
001442  010042                          3$:     MOV     R0,-(R2)        ;RESTORE CORRECT DATA TO ADDRESS
001444  062702  000002                          ADD     #2,R2
001450  000002                                  RTI

001452  051124  050101  042520  TRAP4:  .ASCII  'TRAPPED TO 4 '
001460  020104  047524  032040
001466     040
001467     120  036503     000  ERRPC:  .ASCIZ  'PC='
001473     107  047517  020104  XMTDAT: .ASCIZ  'GOOD DATA='
001500  040504  040524  000075
001506  041040  042101  042040  RECDAT: .ASCIZ  ' BAD DATA='
001514  052101  036501     000
001521     007     000          BELL:   .ASCIZ  <7>
        001524                          .EVEN
                                ;ROUTINE TO PLACE ASCII VALUE OF AN ADDRESS IN TO ADDRESS MESSAGE
001524  066767  177224  000014  $FORMO: ADD     RELOCF,11$+2
001532  066767  177216  000134          ADD     RELOCF,41$+2
001540  004767  176540                  JSR     PC,$SAVR        ;GO SAVE REGISTERS ON THE STACK
001544  012704  002275          11$:    MOV     #DIGITS,R4      ;ADDRESS WHERE ASCII VALUES ARE STORED
001550  005003                          CLR     R3              ;WORKING & INDEX REGISTER
001552  162702  000002                  SUB     #2,R2           ;ADJUST ADDRESS
001556  010205                          MOV     R2,R5           ;SAVE
001560  010501                          MOV     R5,R1
001562  005767  177164                  TST     MMAVA           ;CHECK IF MEM MGMT IS AVAILABLE
001566  001426                          BEQ     1$              ;BRANCH IF NOT AVAILABLE
001570  032737  000001  177572          BIT     #1,@#SR0        ;IS MEM MGMT ENABLED
001576  001422                          BEQ     1$
001600  042701  017777                  BIC     #17777,R1       ;SAVE PAR SELECTOR BITS
001604  000301                          SWAB    R1              ;SWAP BYTES
001606  006001                          ROR     R1
001610  006001                          ROR     R1              ;FORM INDEX VALUE
001612  006001                          ROR     R1
001614  006001                          ROR     R1
001616  017102  001726                  MOV     @PARTAB(R1),R2  ;GET CONTENTS OF PAR
001622  012700  000006                  MOV     #6,R0           ;SHIFT COUNT
001626  006302                          ASL     R2              ;SHIFT KIPAR1 6 PLACES LEFT
001630  006103                          ROL     R3              ;2 MSB'S GO INTO R3
001632  077003                          SOB     R0,.-4
001634  042705  160000                  BIC     #160000,R5      ;CLEAR PAR SELECTOR BITS
001640  060502                          ADD     R5,R2           ;FORM 18 BIT ADDRESS
001642  005503                          ADC     R3              ;IN R2 & R3
001644  006302                  1$:     ASL     R2              ;FIRST DIGIT TO R3
001646  006103                          ROL     R3
001650  012700  000006                  MOV     #6,R0           ;DIGIT COUNT
```

```
001654  000404                          BR      3$              ;PRINT FIRST DIGIT
001656  006302                  2$:     ASL     R2
001660  006103                          ROL     R3
001662  005305                          DEC     R5
001664  001374                          BNE     2$
001666  012705  000003          3$:     MOV     #3,R5           ;DIGIT SHIFT COUNT
001672  116324  002236          41$:    MOVB    DIGTAB(R3),(R4)+ ;LOAD DIGIT INTO MESSAGE
001676  005003                          CLR     R3              ;CLEAR INDEX
001700  005300                          DEC     R0              ;DEC DIGIT COUNT
001702  001365                          BNE     2$
001704  004767  176420                  JSR     PC,$RESTR       ;RESTORE REGISTERS FROM STACK
001710  046767  177040  177630          BIC     RELOCF,11$+2
001716  046767  177032  177750          BIC     RELOCF,41$+2
001724  000207                          RTS     PC              ;RETURN

001726  172340                  PARTAB: KIPAR0
001730  172342                          KIPAR1
001732  172344                          KIPAR2
001734  172346                          KIPAR3
001736  172350                          KIPAR4
001740  172352                          KIPAR5
001742  172354                          KIPAR6
001744  172356                          KIPAR7

                                ;ROUTINE TO TYPE OCTAL VALUE PUSHED ONTO STACK
                                ;CALL:  MOV     VALUE,-(SP)     ;PUSH VALUE ONTO STACK
                                        JSR     PC,02A          ;CALL ROUTINE
                                ;
001746                          02A:
001746  004767  176332                  JSR     PC,$SAVR        ;GO SAVE REGISTERS ON THE STACK
001752  016600  000016                  MOV     16(SP),R0       ;GET VALUE
001756  012703  000006                  MOV     #6,R3           ;COUNTER
001762  005002                          CLR     R2              ;WORKING REGISTER
001764  006100                          ROL     R0
001766  006102                          ROL     R2
001770  062702  000260          1$:     ADD     #260,R2         ;FORM ASCII VALUE
001774  010267  000040                  MOV     R2,2$           ;MOVE CHAR TO TYPE LOCATION
002000  004567  175754                  JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
002004  002040                          2$
002006  005002                          CLR     R2
002010  006100                          ROL     R0
002012  006102                          ROL     R2
002014  006100                          ROL     R0
002016  006102                          ROL     R2
002020  006100                          ROL     R0
002022  006102                          ROL     R2
002024  005303                          DEC     R3
002026  001360                          BNE     1$
002030  004767  176274                  JSR     PC,$RESTR       ;RESTORE REGISTERS FROM STACK
002034  012616                          MOV     (SP)+,(SP)
002036  000207                          RTS     PC
002040  000000                  2$:     .WORD   0               ;CONTAINS CHARACTER TO BE TYPED

002042  000000                  LODFLO: .WORD   0
                                ;ROUTINE TO SAVE ABS LOADER
002044  005767  177772          $LDR:   TST     LODFLO
002050  001401                          BEQ     3$
```

```
        002052  000207                                  RTS     PC
        002054  012700  017776                  3$:     MOV     #17776,R0
        002060  012737  002072  000004                  MOV     #2$,@#ERRVEC    ;SET TIME OUT TRAP VECTOR
        002066  005720                                  TST     (R0)+
        002070  000776                                  BR      .-2
        002072  022626                          2$:     CMP     (SP)+,(SP)+
        002074  162700  002202                          SUB     #2202,R0        ;POINT R0 BACK TO LOADER
        002100  010067  000102                          MOV     R0,$LDR1        ;SAVE FOR RESTORE ROUTINE
        002104  012702  001100                          MOV     #1100,R2        ;WORD COUNT
        002110  012703  015454                          MOV     #LODAR,R3       ;WHERE LOADER IS TO BE STORED
        002114  012023                          1$:     MOV     (R0)+,(R3)+     ;STORE LOADER
        002116  005302                                  DEC     R2
        002120  001375                                  BNE     1$
        002122  014367  000042                          MOV     -(R3),LSTLOC    ;SAVE LAST WORD OF LOADERS
        002126  005367  177710                          DEC     LODFLG
        002132  000207                                  RTS     PC              ;RETURN

                                                ;ROUTINE TO RESTORE LOADER
        002134  005767  177702                  $RLDR:  TST     LODFLG
        002140  001001                                  BNE     2$
        002142  000207                                  RTS     PC
        002144  016705  000036                  2$:     MOV     $LDR1,R5        ;GET FIRST ADDRESS OF WHERE LOADER IS
                                                                                ;TO BE RESTORED
        002150  012704  015454                          MOV     #LODAR,R4       ;ADDRESS WHERE LOADER IS STORED
        002154  012702  001100                          MOV     #1100,R2        ;WORD COUNT
        002160  012425                          1$:     MOV     (R4)+,(R5)+     ;RESTORE
        002162  005302                                  DEC     R2
        002164  001375                                  BNE     1$
        002166  012745                                  MOV     (PC)+,-(R5)     ;RESTORE LAST LOCATION (SAVED BY SAVE
        002170  000000                          LSTLOC: .WORD   0               ;LOADERS ROUTINE ABOVE)
        002172  004567  176562                          JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
        002176  002210                                  $LDRM
        002200  005067  177636                          CLR     LODFLG
        002204  000207                                  RTS     PC              ;RETURN TO CALLER

        002206  000000                          $LDR1:  .WORD   0               ;FIRST ADDRESS WHERE LOADERS ARETO BE
                                                                                ;RESTORED TO
        002210  047514  042101  051105          $LDRM:  .ASCIZ  'LOADER IS RESTORED'<15><12>
        002216  044440  020123  042522
        002224  052123  051117  042105
        002232  005015  000
                002236                                  .EVEN
                                                ;DIGIT TABLE
        002236  030460                          DIGTAB: "01
        002240  031462                                  "23
        002242  032464                                  "45
        002244  033466                                  "67

                                                ;MESSAGES
        002246  040514  052123  040             LST:    .ASCII  'LAST '
        002253     115  046505  051117          ADRESS: .ASCII  'MEMORY ADDRESS IS '
        002260  020131  042101  051104
        002266  051505  020123  051511
        002274     040
        002275     060  030060  030060          DIGITS: .ASCIZ  '000000 '
        002302  020060     000
```

```
1790            002306                  .EVEN
```

```
        002306  000000                          PLACE:  .WORD   0
                                                        .SBTTL  MEMORY ADDRESS TESTS

                                                ;THIS TEST ADDRESS MEMORY UP TO 128K AND PROVES 'UNIQNESS' OF ALL
                                                ;MEMORY ADDRESS IN A 32K SEGMENT. THE TEST WRITES INTO EACH MEMORY
                                                ;ADDRESS THE VALUE OF THAT ADDRESS AND THEN CHECKS FOR THE CORRECT
                                                ;DATA IN EACH ADDRESS.
                                                ;THE TWELVE MOST SIGNIFICANT BITS OF THE LAST AVAILABLE MEMORY ADDRESS
                                                ;IS STORED IN R5.
                                                ;STARTING INTRUCTIONS
                                                ;       LOAD ADDRESS=200
                                                ;       PRESS START
                                                ;       STACK POINTER IS AT 500
                                                ;************RESTART AT 162 TO RESTORE LOADER*****************
                                                ;MEMORY ADDRESS TEST
        002310  012737  002352  000212          START:  MOV     #START1,@#212   ;CHANGE START ADDRESS
        002316  012706  000500                          MOV     #STKPTR,SP      ;SET UP STACK PTR
        002322  004767  177516                          JSR     PC,$LDR         ;GO SAVE MONITOR & LOADERS
        002326  004567  176426                          JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
        002332  011752                                  RESLDR
        002334  005037  000746                          CLR     @#ERCNT         ;CLEAR ERROR COUNT
        002340  005037  000750                          CLR     @#LDDISP        ;CLEAR DISPLAY REGISTER STORAGE LOCN
        002344  013737  000750  177570                  MOV     @#LDDISP,@#DISPLAY      ;CLEAR DISPLAY REGISTER
        002352  012706  000500                  START1: MOV     #STKPTR,SP      ;SET STACK PTR
        002356  005037  006520                          CLR     @#PEFLG         ;CLEAR PARITY ERROR INDICATORS
        002362  012727  002352                          MOV     #START1,(PC)+   ;LOAD PARITY ERROR RESTART ADDRESS
        002366  000000                          PERSTRT:.WORD   0               ;CONTAINS RESTART ADDRESS AFTER PAR ERR
        002370  005037  000742                          CLR     @#ICNT          ;CLEAR PASS COUNT
        002374  005037  000754                          CLR     @#RELOCF        ;CLEAR RELOACATION FACTOR
        002400  012737  000502  000024                  MOV     #PDWN,@#PFVEC   ;SET POWER FAIL TRAP VECTOR
        002406  005037  000026                          CLR     @#PFVEC+2

                                                ;CHECK IF MEMORY MANAGEMENT IS AVAILABLE
        002412  005067  176334                          CLR     MMAVA           ;CLEAR MEM MGMT AVAILABLE INDICATOR
        002416  032737  010000  177570                  BIT     #BIT12,@#SWR    ;CHECK IF TO RUN WITH MEM MGMT
        002424  001007                                  BNE     1$              ;DO NOT USE MEM MGMT IF SW12 WAS SET
        002426  012737  002444  000004                  MOV     #1$,@#ERRVEC    ;SET TIME OUT TRAP
        002434  005037  177572                          CLR     @#SR0           ;REFERENCE MEM MGMT
        002440  005167  176306                          COM     MMAVA           ;SET INDICATOR TO -1 IF AVAILABLE
        002444  004767  004274                  1$:     JSR     PC,.MAMF        ;GO ENABLE PARITY ACTION


                                                ;ROUTINE TO WRITE VALUE OF MEMORY ADDRESS INTO MEMORY ADDRESS
                                                ;FOR EXAMPLE ROUTINE WRITES 20000 INTO LOCATION 20000
        002450  012737  002510  000004          WRTUP:  MOV     #DONEC,@#ERRVEC ;SET TIME OUT TRAP VECTOR
        002456  010701                                  MOV     PC,R1           ;LOAD TRACE REGISTER
        002460  004767  004360                          JSR     PC,LDMMC
        002464  012737  007142  000250                  MOV     #MMABTO,@#MMVEC ;SET MEM MGMT ABORT VECTOR
        002472  012702  020000                          MOV     #20000,R2       ;FIRST ADDRESS
        002476  010203                                  MOV     R2,R3           ;LOAD CONSTANT
        002500  010322                                  MOV     R3,(R2)+        ;WRITE VALUE OF ADDRESS INTO ADDRESS
        002502  062703  000002                          ADD     #2,R3           ;NEXT VALUE
        002506  000774                                  BR      .-6             ;WRITE UNTIL DONE

        002510  012706  000500                  DONEC:  MOV     #STKPTR,SP      ;SET STACK PTR
```

```
002514  004767  177034                 JSR     PC,$FORMAT      ;GO TO FORMAT ROUTINE
002520  004567  176234                 JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
002524  002246                         LST
002526  004567  176226                 JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
002532  015020                         $CRLF

                               ;ROUTINE TO CHECK THAT VALUE OF MEMORY ADDRESS WAS WRITTEN CORRECTLY
002534  010701                         MOV     PC,R1           ;LOAD TRACE REGISTER
002536  012702  020000                 MOV     #20000,R2       ;SET R2
002542  012737  002600  000004         MOV     #DONE1,@#ERRVEC ;SET TIME OUT TRAP
002550  010200                         MOV     R2,R0
002552  162700  000002                 SUB     #2,R0           ;SUBTRACT 2
002556  004767  004262                 JSR     PC,LDMMO
002562  062700  000002         1$:     ADD     #2,R0
002566  012203                         MOV     (R2)+,R3        ;GET WRITTEN VALUE
002570  020003                         CMP     R0,R3           ;CHECK
002572  001773                         BEQ     1$
002574  104400                         HLT                     ;ERROR! TO DETERMINE WHICH ADDRESS WAS
                               ;WRITTEN IMPROPERLY EXAMINE R2.  NEXT EXAMINE MEM MGMT REGISTER KIPAR1
                               ;(IF MEM MGMT IS AVAILABLE). ADD R2 AND KIPAR1 TOGETHER AS SHOWN BELOW

                               :       R2-2                    0 00X XXX XXX XXX XXX
                               :       KIPAR1(772342)  0 000 YYY YYY YYY YYY
                               :       ADDRESS               ZZZ ZZZ ZZZ ZZZ ZZZ ZZZ

002576  000771                         BR      1$
002600  012706  000500         DONE1:  MOV     #STKPTR,SP      ;SET STACK PTR
002604  010701                         MOV     PC,R1           ;LOAD TRACE REGISTER

                               ;ROUTINE TO WRITE 1'S COMPLEMENT VALUE OF ADDRESS INTO ADDRESS
                               ;FOR EXAMPLE ROUTINE WRITES 157777 INTO ADDRESS 20000

002606  005767  176140                 TST     MMAVA           ;MEMORY MAGNAGEMENT AVAILABLE?
002612  001420                         BEQ     3$
002614  013703  172342                 MOV     @#KIPAR1,R3     ;FIND LAST ADDRESS IF MEM MANAGE USED
002620  006303                         ASL     R3
002622  006303                         ASL     R3
002624  006303                         ASL     R3
002626  006303                         ASL     R3
002630  006303                         ASL     R3
002632  006303                         ASL     R3
002634  010246                         MOV     R2,-(SP)        ;DEVELOP COMPLEMENT OF LAST ADDRESS
002636  042716  020000                 BIC     #20000,(SP)     ;SAVE BITS IF MEMORY IS NOT A MULTIPLE OF 4K
002642  062603                         ADD     (SP)+,R3
002644  012737  007174  000250         MOV     #MMABT1,@#MMVEC ;SET ABORT VECTOR
002652  000403                         BR      2$
002654  162702  000002         3$:     SUB     #2,R2           ;R2=LAST ADDRESS
002660  010203                         MOV     R2,R3
002662  005103                 2$:     COM     R3              ;COMPLEMENT VALUE IN R3
002664  062703  000002         1$:     ADD     #2,R3
002670  010342                         MOV     R3,-(R2)        ;WRITE COMPLIMENT VALUE INTO ADDRESS
002672  102403                         BVS     DONE3
002674  020227  017776                 CMP     R2,#17776
002700  001371                         BNE     1$

                               ;SET UP TO CHECK COMPLEMENT DATA WRITTEN DOWN
```

```
002702  000240                      DONE3:  NOP
002704  010701                              MOV     PC,R1           ;LOAD TRACE REGISTER
002706  005767  176040                      TST     MMAVA           ;CHECK IF MM IS AVAIL
002712  001406                              BEQ     1$
002714  012737  000200  172342              MOV     #200,@#KIPAR1   ;INIT KIPAR1
002722  012737  007142  000250              MOV     #MMABTO,@#MMVEC ;SET ABORT VECTOR
002730  012737  002770  000004      1$:     MOV     #DONE4,@#ERRVEC
002736  012702  020000                      MOV     #20000,R2       ;FIRST ADDRESS
002742  010200                              MOV     R2,R0
002744  005100                              COM     R0              ;FIRST DATA (COM OF ADDRESS)
002746  062700  000002                      ADD     #2,R0
002752  162700  000002              2$:     SUB     #2,R0
002756  012203                              MOV     (R2)+,R3        ;GET VALUE
002760  020003                              CMP     R0,R3           ;CHECK
002762  001773                              BEQ     2$
002764  104400                              HLT
002766  000771                              BR      2$
002770  000240                      DONE4:  NOP

                                    ;ROUTINE TO WRITE BANK # INTO ALL ADDRESSES IN A 4K BANK
002772  012737  003040  000004              MOV     #DONE4A,@#ERRVEC;SET TIME OUT TRAP VECTOR
003000  010701                              MOV     PC,R1
003002  004767  004036                      JSR     PC,LDMMO
003006  012737  007142  000250              MOV     #MMABTO,@#MMVEC
003014  012702  020000                      MOV     #20000,R2
003020  005000                              CLR     R0
003022  005200                      1$:     INC     R0              ;R0 WILL BE DATA WRITTEN
003024  012704  010000                      MOV     #4096.,R4       ;SET 4K COUNTER
003030  010022                      2$:     MOV     R0,(R2)+        ;WRITE BANK # INTO ALL ADDRESSES
003032  005304                              DEC     R4
003034  001375                              BNE     2$
003036  000771                              BR      1$

003040  022626                      DONE4A: CMP     (SP)+,(SP)+     ;ADJUST STACK PTR

                                    ;CHECK THAT DATA WRITTEN ABOVE CAN BE READ
003042  012737  003110  000004              MOV     #DONE4B,@#ERRVEC
003050  010701                              MOV     PC,R1
003052  004767  003766                      JSR     PC,LDMMO
003056  012702  020000                      MOV     #20000,R2
003062  005000                              CLR     R0
003064  005200                      1$:     INC     R0
003066  012704  010000                      MOV     #4096.,R4
003072  012203                      2$:     MOV     (R2)+,R3
003074  020003                              CMP     R0,R3
003076  001401                              BEQ     .+4
003100  104400                              HLT
003102  005304                              DEC     R4
003104  001372                              BNE     2$
003106  000766                              BR      1$
003110  022626                      DONE4B: CMP     (SP)+,(SP)+

                                    ;ROUTINE TO WRITE CONSTANT DATA INTO 4K
                                    ;BANK STARTING WITH LAST MEMORY LOCATION
003112  010701                              MOV     PC,R1
003114  012737  007174  000250              MOV     #MMABT1,@#MMVEC
```

```
 1959  003122  162702  000002              SUB     #2,R2
 1960  003126  005000                      CLR     R0
 1961  003130  005300              1$:     DEC     R0
 1962  003132  012704  010000              MOV     #4096.,R4
 1963  003136  010042              2$:     MOV     R0,-(R2)
 1964  003140  102406                      BVS     DONE4C
 1965  003142  020227  017776              CMP     R2,#17776       ;CHECK IF DONE
 1966  003146  001403                      BEQ     DONE4C
 1967  003150  005304                      DEC     R4
 1968  003152  001371                      BNE     2$
 1969  003154  000765                      BR      1$
 1970
 1971  003156  012737  003252  000004  DONE4C: MOV     #DONE4D,@#ERRVEC
 1972  003164  010701                      MOV     PC,R1
 1973  003166  004767  003652              JSR     PC,LDMMO
 1974  003172  012737  007142  000250      MOV     #MMABTO,@#MMVEC ;SET ABORT VECTOR
 1975  003200  012702  020000              MOV     #20000,R2
 1976  003204  022704  010000          1$: CMP     #4096.,R4       ;CHECK IF WRITE ABOVE STARTED ON
 1977                                                              ;4K BOUNDARY
 1978  003210  001406                      BEQ     2$
 1979  003212  012203                      MOV     (R2)+,R3
 1980  003214  020003                      CMP     R0,R3
 1981  003216  001401                      BEQ     .+4
 1982  003220  104400                      HLT
 1983  003222  005204                      INC     R4
 1984  003224  001367                      BNE     1$
 1985  003226  005200              2$:     INC     R0
 1986  003230  012704  010000              MOV     #4096.,R4
 1987  003234  012203              3$:     MOV     (R2)+,R3
 1988  003236  020003                      CMP     R0,R3
 1989  003240  001401                      BEQ     .+4
 1990  003242  104400                      HLT
 1991  003244  005304                      DEC     R4
 1992  003246  001372                      BNE     3$
 1993  003250  000766                      BR      2$
 1994
 1995  003252  022626              DONE4D: CMP     (SP)+,(SP)+
 1996  003254  005737  000042              TST     @#42            ;BRANCH IF PROGRAM WAS NOT
 1997  003260  001406                      BEQ     BEGIN1          ;LOADED VIA ACT11 IN QV OR AA MODES
 1998  003262  005767  001330              TST     LOGICAL+2       ;BRANCH IF NOT IN QV MODE
 1999  003266  100003                      BPL     BEGIN1
 2000  003270  012737  000006  000742      MOV     #6,@#ICNT       ;SET ICNT TO DO 1 PASS ONLY IN QV
 2001
```

```
                                [illegible] TEST [illegible] TESTS
                        ;[illegible] TEST [illegible] NOISE TEST PATTERNS [illegible]
                        ;[illegible] AND CHECKS THAT THE [illegible] BE WRITTEN AND READ.
                        ;[illegible] TRAP [illegible]
                        [illegible]:    [illegible]     #STKPTR,SP       ;SET STACK PTR
                                [illegible]     [illegible]      ;[illegible] ENABLE PARITY [illegible]
                                [illegible]     [illegible]      ;[illegible] CHECK SWITCHES
 [illegible]     [illegible]     [illegible]  [illegible]  [illegible]  #DONES,@#ERRVEC  ;SET UP TIME OUT TRAP

                        ;WRITE [illegible] 9 TEST PATTERN STARTING AT ADDRESS 20000
 [illegible]     [illegible]                            [illegible]     PC,R1            ;UPDATE TRACE REGISTER
 [illegible]     [illegible]     000001                 [illegible]     #[illegible],-SP ;PUSH STARTING BANK # ON THE STACK
 [illegible]     [illegible]                            [illegible]     -SP              ;PUSH # OF 128. WORD [illegible] TO WRITE
 [illegible]     [illegible]     004054                 JSR     PC,..1X9         ;GO TO ROUTINE TO WRITE 1 [illegible] PATTERN

                        ;CHECK 1 XOR 9 TEST PATTERN WRITTEN ABOVE
 [illegible]     [illegible]     [illegible]  00000[illegible]  [illegible]: [illegible]     #ERRTRP,@#ERRVEC
 [illegible]     [illegible]     [illegible]            [illegible]: [illegible] #STKPTR,SP   ;SET STACK PTR
 [illegible]     [illegible]                            [illegible]     PC,R1            ;UPDATE TRACE REGISTER
 [illegible]     [illegible]     000001                 [illegible]     #1,-SP           ;PUSH STARTING BANK # ON THE STACK
 [illegible]     [illegible]                            NEG     R3               ;R3 CONTAINS # OF 128. WORD BLOCKS
 003356          [illegible]                            MOV     R3,-SP           ;WRITTEN BY .1X9 ROUTINE ABOVE
 003360          [illegible]     004136                 JSR     PC,..1X9         ;GO CHECK 1X9 PATTERN

 00336[illegible]   005027                              CLR     (PC)+            ;SET INDICATOR TO WRITE NORMAL 3X9 PAT
 00336[illegible]   000000          PARPAT: .WORD   0

                        ;WRITE 3 XOR 9 TEST PATTERN STARTING AT ADDRESS 20000
                        ;NOTE PATTERN IS NORMAL 3 XOR 9 IF NO PARITY MEMORY IS AVAILABLE.
                        ;AND IS A MODIFIED PATTERN IF PARITY MEMORY IS AVAILABLE.
                        ;THE CONTENTS OF PARPAT IF 0 NOT 0 INDICATE IF NORMAL MODIFIED PATTERN
                        ;IS BEING USED IN TESTS BELOW.
 003372  012706  000500         DONES:  MOV     #STKPTR,SP       ;SET STACK PTR
 [illegible]  010701                           MOV     PC,R1            ;UPDATE TRACE REGISTER
 [illegible]  012737  003416  00000[illegible]    MOV     #DONE7,@#ERRVEC  ;SET TIME OUT TRAP VECTOR
 [illegible]  012746  000001                   MOV     #1,-(SP)         ;PUSH STARTING BANK # ON STACK
 [illegible]  005046                           CLR     -(SP)            ;PUSH # OF 256. WORD BLOCKS TO WRITE
 003412  004767  004340                   JSR     PC,..3X9         ;CALL ROUTINE TO WRITE 3XOR9 PATTERN

                        ;CHECK 3 XOR 9 TEST PATTERN WRITTEN ABOVE
 003416  012737  001116  00000[illegible]  DONE7:  MOV     #ERRTRP,@#ERRVEC
 003424  016600  000006                   MOV     6(SP),R0         ;GET # OF 256. WORD BLOCKS WRITTEN
 003430  005400                           NEG     R0               ;FORM TWO'S COMPLEMENT
 003432  010027                           MOV     R0,(PC)+         ;SAVE # OF 256 WORD BLOCKS
 003434  000000          WDS.256: .WORD   0                ;CONTAINS # OF 256 WORD BLOCKS IN MEM.
 003436  012706  000500                   MOV     #STKPTR,SP       ;SET STACK PTR
 003442  010701                           MOV     PC,R1            ;SET SCOPE PTR
 003444  012746  000001                   MOV     #1,-(SP)         ;PUSH BANK # ON THE STACK
 003450  010046                           MOV     R0,-(SP)         ;PUSH # OF 256. WORD BLOCKS TO WRITE
 003452  004767  004520                   JSR     PC,..3X9         ;GO CHECK DATA WRITTEN

                        ;SETUP TO RUN MODIFIED 3 XOR 9 PATTERN IF PARITY MEMORY IS AVAILABLE
 003456  005737  007014                   TST     @#PARAVA         ;BRANCH IF PARITY MEMORY IS NOT AVAIL
 [illegible]  [illegible]                  BEQ     DONE8
 [illegible]  005737  003366                   TST     @#PARPAT         ;BRANCH IF PARITY PAT JUST WRITTEN
```

```
        003470  001003                          BNE     DONE8
        003472  010637  003366                  MOV     SP,@#PARPAT     ;SET INDICATOR TO WRITE 3X9 PAR PAT
        003476  000734                          BR      DONE6           ;REPEAT TEST USING MODIFIED 3X9 PATTERN

                                        ;WRITE 9 XOR 13 TEST PATTERN STARTING AT ADDRESS 40000
        003500  012706  000500          DONE8:  MOV     #STKPTR,SP      ;SET STACK PTR
        003504  012737  003526  000004          MOV     #DONE9,@#ERRVEC ;SET TIME OUT TRAP VECTOR
        003512  010701                          MOV     PC,R1           ;UPDATE TRACE REGISTER
        003514  012746  000002                  MOV     #2,-(SP)        ;PUSH STARTING BANK # ON THE STACK
        003520  005046                          CLR     -(SP)           ;PUSH # OF BANKS TO WRITE ON THE STACK
        003522  004767  005120                  JSR     PC,.9X13        ;GO TO ROUTINE TO WRITE DATA

                                        ;CHECK 9 XOR 13 TEST PATTERN WRITTEN ABOVE
        003526  012706  000500          DONE9:  MOV     #STKPTR,SP      ;SET STACK PTR
        003532  010701                          MOV     PC,R1           ;UPDATE TRACE REGISTER
        003534  012737  001116  000004          MOV     #ERRTRP,@#ERRVEC
        003542  012746  000002                  MOV     #2,-(SP)
        003546  005404                          NEG     R4
        003550  042704  000001                  BIC     #1,R4           ;SET 4K BANK COUNT TO 8K INCREMENT
        003554  001403                          BEQ     DONE10          ;DO NOT CHECK IF ONLY 12K
        003556  010446                          MOV     R4,-(SP)
        003560  004767  005142                  JSR     PC,..9X13       ;GO CHECK 9 XOR 13 PATTERN WRITTEN ABOVE


                                        ;RELOCATE PROGRAM TO CHECK ADDRESSES FROM 000000-017776 USING 1 XOR 8 PATTERN
        003564  000005                  DONE10: RESET                   ;DISABLE MEM MGMT AND PARITY ACTION
        003566  005737  000042          1$:     TST     @#42            ;CHECK IF PROGRAM LOADED VIA ACT11
        003572  001402                          BEQ     2$
        003574  000137  004256                  JMP     @#RANTST        ;DO NOT RELOCATE IF ACT11
        003600  012706  000500          2$:     MOV     #STKPTR,SP      ;SET STACK PTR
        003604  004567  002436                  JSR     5,RELOC         ;RELOCATE PROGRAM CODE
        003610  000000                          000000                  ;FROM 000000 TO
        003612  020000                          20000                   ;20000
        003614  005704                          TST     R4              ;WAS RELOCATION SUCCESSFUL?
        003616  001402                          BEQ     3$              ;BRANCH IF SUCCESSFUL I.E. WAS THERE
                                                                        ;SUFFICIENT MEMORY TO RELOCATE TO.
        003620  000137  004256                  JMP     @#RANTST        ;END OF TEST
        003624  062707  020000          3$:     ADD     #20000,PC       ;RELOCATE PC
        003630  062706  020000                  ADD     #20000,SP       ;SET NEW STACK PTR
        003634  052767  100000  175106          BIS     #100000,LDDISP  ;SET RELOCATION INDICATOR
        003642  016737  175102  177570          MOV     LDDISP,@#DISPLAY;LOAD DISPLAY REGISTER

                                        ;**********************IMPORTANT NOTE*********************************
                                        ;PROGRAM IS NOW EXECUTING CODE FROM PC AS SHOWN BELOW +20000.
                                        ;CAUTION: DO NOT ATTEMPT TO RESTART PROGRAM AT 200
                                        ;*********************************************************************

        003650  010701                  DONE11: MOV     PC,R1           ;RESTART ADDRESS TO LOOP TEST

                                        ;WRITE 1 XOR 8 TEST PATTERN IN LOCATIONS 000000-017776
        003652  005046                          CLR     -(SP)           ;PUSH STARTING BANK # ON THE STACK
        003654  012746  000040                  MOV     #32.,-(SP)      ;PUSH # OF 128. WORD BLOCKS TO WRITE
        003660  004767  003524                  JSR     PC,.1X8         ;GO TO ROUTINE TO WRITE 1 XOR 8 DATA

        003664  010701                          MOV     PC,R1           ;RESTART & LOOP TEST ADDRESS
```

D05


```
                                           ;CHECK 1 XOR 5 TEST PATTERN AS WRITTEN ABOVE
003666  005046                                    CLR     -(SP)              ;PUSH STARTING BANK # ON THE STACK
[illegible] [illegible] [illegible]               MOV     #32.,-(SP          ;PUSH # OF 128. WORD BLOCKS TO WRITE
003674  004767  003622                            JSR     PC,.1X8            ;GO TO ROUTINE TO CHECK DATA

                                           ;RELOCATE PROGRAM TO CHECK ADDRESSES 000000 - 037776 USING
                                           ;3 XOR 9 AND 9 XOR 13 PATTERNS
003700  010701                             DONE12: MOV     PC,R1              ;UPDATE TRACE REGISTER
003702  004567  002340                            JSR     R5,RELOC           ;MOVE PROGRAM BACK  TO LOWEST 4K
003706  020000                                    20000                      ;FROM 20000 TO
003710  000000                                    000000                     ;000000
003712  000137  003716                            JMP     @#.+4              ;RETURN UNRELOCATED
003716  012767  040000  000136                    MOV     #40000,2$
003724  012767  000040  [illegible]               MOV     #32.,BLKCNT
003732  042737  100000  000750                    BIC     #100000,@#LDDISP          ;CLEAR RELOCATION INDICATOR
003740  016737  175004  177570                    MOV     LDDISP,@#DISPLAY          ;DISPLAY NOT RELOCATED
003746  012737  004040  000004                    MOV     #3$,@#ERRVEC       ;SET TIME OUT TRAP
003754  005737  057776                            TST     @#057776           ;CHECK IF 12K OF MEMORY IS AVAILABLE
003760  012737  004002  000004                    MOV     #1$,@#ERRVEC
003766  005737  117776                            TST     @#117776           ;CHECK IF 20K OF MEMORY
003772  006367  000034                            ASL     2$
003776  006367  000100                            ASL     BLKCNT
004002  012706  000500                     1$:    MOV     #STKPTR,SP         ;SET STACK POINTER
004006  010701                                    MOV     PC,R1              ;UPDATE TRACE REGISTER
004010  042737  100000  000750                    BIC     #100000,@#LDDISP          ;CLEAR RELOCATION INDICATOR
004016  013737  000750  177570                    MOV     @#LDDISP,@#DISPLAY        ;LOAD DISPLAY REGISTER
004024  004567  002216                            JSR     R5,RELOC           ;RELOCATE PROGRAM
004030  000000                                    000000                     ;FROM 000000
004032  040000                             2$:    40000                      ;TO 40000
004034  005704                                    TST     R4                 ;RELOCATION SUCCESSFUL?
004036  001402                                    BEQ     4$                 ;YES
004040  000137  004256                     3$:    JMP     @#RANTST           ;GO TO RANDOM DATA TEST
004044  066707  177762                     4$:    ADD     2$,PC              ;RELOCATE PC
004050  066706  177756                            ADD     2$,SP                     ;SET NEW STACK PTR
004054  052767  100000  174666                    BIS     #100000,LDDISP     ;SET RELOCATION INDICATOR
004062  016737  174652  177570                    MOV     LDDISP,@#DISPLAY;RELOAD DISPLAY REGISTER

                                           ;*********************IMPORTANT NOTE*********************************
                                           ;PROGRAM IS NOW EXECUTING CODE FROM PC AS SHOWN BELOW +40000 OR +100000
                                           ;CAUTION: DO NOT ATTEMPT TO RESTART PROGRAM AT 200
                                           ;*******************************************************************

004070  005037  003366                            CLR     @#PARPAT           ;SET INDICATOR TO WRITE NORMAL 3X9 PAT
004074  010701                             DONE13: MOV     PC,R1              ;UPDATE TRACE REGISTER

                                           ;WRITE 3XOR9 TEST PATTERN IN LOCATIONS 000000-037776 OR 000000-077776
004076  005046                                    CLR     -(SP)              ;PUSH BANK # 0 ON THE STACK
004100  012746                                    MOV     (PC)+,-(SP)        ;PUSH 256. BLOCK WORD COUNT ON STACK
004102  000000                             BLKCNT: .WORD   0                  ;CONTAINS 256. BLOCK WORD COUNT LOADED
                                                                              ;BY ABOVE ROUTINE. 40 100 IF 8 16K
004104  004767  003646                            JSR     PC,.3X9

                                           ;CHECK PATTERN WRITTEN IN LOCATIONS 000000-037776 OR 000000-077776
004110  010701                                    MOV     PC,R1
004112  005046                                    CLR     -(SP)              ;PUSH STARTING BANK # ON THE STACK
004114  016746  177762                            MOV     BLKCNT,-(SP        ;PUSH 256. WORD BLOCK COUNT ON THE STACK
```

E05


```
004120  004767  004052                JSR     PC,..3X9          ;CHECK

                               ;ROUTINE TO CHECK IF MODIFIED 9 XOR9 PATTERN SHOULD BE RUN
004124  005737  000014                TST     @#PARAVA          ;BRANCH IF PARITY MEMORY NOT AVAIL
004130  001406                        BEQ     DONE14
004132  005737  003366                TST     @#PARPAT          ;BRANCH IF MODIFIED PATTERN JUST RUN
004136  001003                        BNE     DONE14
004140  010667  177222                MOV     SP,PARPAT         ;SET INDICATOR TO WRITE MODIFIED PAT
004144  000753                        BR      DONE13            ;LOOP TEST USING MODIFIED PATTERN
                               ;WRITE 9 XOR 13 TEST PATTERN IN LOCATIONS 000000-037777
004146  010701                 DONE14: MOV    PC,R1             ;UPDATE RESTART & LOOP ADDRESS
004150  005046                        CLR     -(SP)             ;PUSH STARTING BANK # ON THE STACK
004152  012746  000002                MOV     #2,-(SP)          ;PUSH # OF BANKS TO WRITE ON THE STACK
004156  004767  004464                JSR     PC,..8X13         ;GO TO ROUTINE TO WRITE DATA

                               ;CHECK 9 XOR 13 PATTERN WRITTEN ABOVE
004162  010701                        MOV     PC,R1             ;UPDATE RESTART & LOOP ADDRESS
004164  005046                        CLR     -(SP)             ;PUSH BANK # ON THE STACK
004166  012746  000002                MOV     #2,-(SP)          ;AND # OF BANKS TO CHECK (2)
004172  004767  004526                JSR     PC,..8X13         ;GO TO CHECK ROUTINE

                               ;RELOCATE PROGRAM BACK TO LOWER MEMORY
004176  012767  040000  000022 DONE15: MOV    #40000,1$
004204  022767  000040  177570        CMP     #40,BLKCNT
004212  001402                        BEQ     .+6
004214  006367  000006                ASL     1$
004220  010701                        MOV     PC,R1             ;UPDATE TRACE REGISTER
004222  004567  002020                JSR     R5,RELOC          ;MOVE PROGRAM BACK INTO LOWER
004226  040000                 1$:    40000                     ;FROM 40000 OR 100000
004230  000000                        000000                    ;TO 000000
004232  000137  004236                JMP     @#.+4             ;RETURN UNRELOCATED
004236  012706  000500                MOV     #STKPTR,SP        ;RESET STACK POINTER
004242  042737  100000  000750        BIC     #100000,@#LDDISP         ;CLEAR RELOCATION INDICATOR
004250  013737  000750  177570        MOV     @#LDDISP,@#DISPLAY       ;LOAD DISPLAY REGISTER

                                      .SBTTL  RANDOM DATA,ROTATING 1 0 TESTS
                               ;RANDOM DATA TEST. THIS TEST MOVES THE PROGRAM CODE THROUGHOUT MEMORY
004256  010701                 RANTST: MOV    PC,R1             ;SET TRACE POINTER
004260  012737  004416  000004        MOV     #7$,@#ERRVEC      ;SET TIME OUT TRAP
004266  005767  174460                TST     MMAVA             ;CHECK IF MEM MGMT IS AVAILABLE
004272  001412                        BEQ     1$                ;BRANCH IF NOT AVAILABLE
004274  004767  002544                JSR     PC,LDMMO          ;GO SET UP MEM MGMT
004300  105237  172301                INCB    @#KIPDR0+1        ;ALLOW 4K ADDRESSING IN FIRST 4K
004304  012737  077406  172304        MOV     #200*256.-400+UP+RW,@#KIPDR2   ;SET KIPDR2=RW UP 200 BLOCKS
004312  012737  000400  172344        MOV     #400,@#KIPAR2
004320  012702  020000         1$:    MOV     #20000,R2         ;SET 'TO' ADDRESS POINTER
004324  005004                        CLR     R4                ;SET 'FROM' ADDRESS POINTER
004326  012705  004000         2$:    MOV     #2048.,R5         ;SET 4K WORD COUNT
004332  012422                 3$:    MOV     (R4)+,(R2)+       ;MOVE CODE
004334  012422                        MOV     (R4)+,(R2)+
004336  005305                        DEC     R5                ;DECREMENT 4K WORD COUNTER
004340  001374                        BNE     3$

004342  012705  005473                MOV     #4096.-PLACE+1,R5        ;SET 4K WORD COUNTER
004346  014400                 4$:    MOV     -(R4),R0          ;GET 'GOOD' DATA
004350  014203                        MOV     -(R2),R3          ;GET 'BAD' DATA
```

F05


```
004352  020003                              CMP     R0,R3           ;COMPARE 'GOOD' & 'BAD' DATA
004354  001403                              BEQ     5$
004356  005722                              TST     (R2)+           ;STEP ADDRESS FOR ERROR ROUTINE
004360  104400                              HLT                     ;REPORT ERROR
004362  005742                              TST     -(R2)           ;RESTORE ADDRESS POINTER
004364  005305                      5$:     DEC     R5              ;DECREMENT 4K WORD COUNTER
004366  001367                              BNE     4$              ;LOOP UNTIL 4K WORDS CHECKED

004370  005767  174356                      TST     MMAVA           ;CHECK IF MEM MGMT IS AVAILABLE
004374  001405                              BEQ     6$              ;BRANCH IF NOT AVAILABLE
004376  005237  172342                      INC     @#KIPAR1
004402  005237  172344                      INC     @#KIPAR2
004406  000744                              BR      1$
004410  062702  000100              6$:     ADD     #64.,R2         ;STEP ADDRESS
004414  000744                              BR      2$
004416  012706  000500              7$:     MOV     #STKPTR,SP      ;RESET STACK PTR
004422  012737  001116  000004              MOV     #ERRTRP,@#ERRVEC;RESTORE ERROR TRAP VECTOR

                                    ;ROTATING 0 TEST. THIS TEST ROTATES A SINGLE '0' THROUGH MEMOR
004430  012767  177777  000744      ROT0:   MOV     #-1,.CONST      ;SET CONSTANT =177777
004436  012746  000001                      MOV     #1,-(SP)        ;SET BANK #1
004442  016746  175756                      MOV     WDS.256,-(SP)   ;GET # OF 256. WORD BLOCKS IN MEMOR
004446  004767  004676                      JSR     PC,.USER        ;GO WRITE 1'S THROUGHOUT MEMORY
004452  010701                              MOV     PC,R1           ;SET SCOPE PTR
004454  012746  000001                      MOV     #1,-(SP)        ;SET STARTING BANK #
004460  016746  175750                      MOV     WDS.256,-(SP)   ;SET # OF 256. WORD BLOCKS TO CHECK
004464  004767  004430                      JSR     PC,.ROT0        ;GO TO ROTATE 0 ROUTINE

                                    ;ROTATING 1 TEST THIS TEST ROTATES A SINGLE '1' BIT THROUGH ALL OF
                                    ;MEMORY
004470  005067  000706              ROT1:   CLR     .CONST          ;CLEAR CONSTANT
004474  012746  000001                      MOV     #1,-(SP)        ;PUSH STARTING BANK ONTO STACK
004500  016746  175730                      MOV     WDS.256,-(SP)   ;AND # OF 256. WORD BLOCKS IN MEMOR
004504  004767  004640                      JSR     PC,.USER        ;GO WRITE 0'S THROUGHOUT MEMORY
004510  010701                              MOV     PC,R1           ;SET SCOPE PTR
004512  012746  000001                      MOV     #1,-(SP)        ;SET STARTING BANK #
004516  016746  175712                      MOV     WDS.256,-(SP)   ;SET # OF 256. WORD BLOCKS TO CHECK
004522  004767  004456                      JSR     PC,.ROT1        ;GO ROTATE A '1' BIT THROUGHOUT MEMOR

                                    ;END OF CYCLE
004526  000005                      END:    RESET
004530  010701                              MOV     PC,R1           ;UPDATE TRACE REGISTER
004532  012706  000500                      MOV     #STKPTR,SP      ;SET STACK PTR
004536  005237  000742                      INC     @#ICNT          ;INCREMENT PASS COUNT
004542  022737  000007  000742              CMP     #7,@#ICNT       ;8 PASSES?
004550  001405                              BEQ     DONE            ;BRANCH IF 8 PASSES COMPLETED
004552  004567  174202                      JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
004556  012340                              ASTERISK
004560  000137  003276                      JMP     @#BEGIN1
004564                              DONE:
004564  004567  174170                      JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
004570  012342                              ENDMSG
004572  105737  177564                      TSTB    @#TPS           ;WAIT FOR BELL TO RING
004576  100375                              BPL     .-4
004600  013700  000042                      MOV     @#42,R0         ;GET DECTAPE MONITOR RETURN ADDRESS
004604  001407                              BEQ     FINISH
```

```
004606  004767  175222                    JSR     PC,$RLDR         ;RESTORE MONITOR & LOADERS
004612  000005                            RESET
004614  004710                   LOGICAL: JSR     PC,(R0)          ;GO TO DECTAPE MONITOR
004616  000240                            NOP
004620  000240                            NOP
004622  000240                            NOP
004624  000167  175522           FINISH:  JMP     START1

                                          .SBTTL  USER TESTS
                                          .SBTTL  PROGRAM # 2
                                 ;PROGRAM # 2
                                 ;THIS PROGRAM ALLOWS THE USER TO SELECT A STARTING 4K BANK #,# OF 4K
                                 ;BANKS TO TEST, AND A WORST CASE PATTERN TO WRITE AND CHECK.
004630  005067  001062           PRG2:    CLR     PRG3FLG          ;CLEAR PROGRAM # 3 FLAG
                                 ;NOTE:   PROGRAM 3 ENTERS PROGRAM # 2 HERE, WITH PRG3FLG = 1.
004634  012706  000500           PRG2A:   MOV     #STKPTR,SP       ;SET STACK PTR
004640  005037  006520                    CLR     @#PEFLG          ;CLEAR PARITY ERROR INDICATORS
004644  005027                            CLR     (PC)+            ;CLEAR RUNNING ALL PATTERNS FLAG
004646  000000                   ALLFLG:  .WORD   0                ;CONTAINS RUNNING ALL PATERNS FLAG
                                                                   ;0/-1 = RUNNING SELECTED/ALL PATTERNS
004650  012737  000502  000024            MOV     #PDWN,@#PFVEC    ;SET POWER FAIL TRAP VECTOR
004656  012737  000006  000004            MOV     #ERRVEC+2,@#ERRVEC
004664  005067  174062                    CLR     MMAVA
004670  032737  010000  177570            BIT     #BIT12,@#SWR     ;CHECK IF TEST IS TO BE RUN
004676  001006                            BNE     1$               ;WITH MEM MGMT ENABLED
004700  000261                            SEC
004702  005737  177572                    TST     @#SR0            ;CHECK IF MEM MGMT IS AVAILABLE
004706  103402                            BCS     1$               ;BRANCH IF NOT AVAILABLE
004710  005167  174036                    COM     MMAVA
004714  012737  001116  000004   1$:      MOV     #ERRTRP,@#ERRVEC
004722  005037  000742                    CLR     @#ICNT
004726  005037  000750                    CLR     @#LODISP
004732  005037  003366                    CLR     @#PARPAT         ;SET INDICATOR TO WRITE NORM 3X9 PAT
004736  004567  174016                    JSR     R5,$PRINT        ;GO TO PRINT ROUTINE
004742  012016                            PARITY
004744  004767  004616                    JSR     PC,RECD          ;GO GET ANSWER TO THE PARITY QUESTION
004750  000000                   .PARIT:  .WORD   0                ;TYPE 1 IF PARITY DESIRED 0 IF NOT
004752  005767  177772                    TST     .PARIT
004756  001402                            BEQ     1$               ;BRANCH IF PARITY NOT DESIRED
004760  004767  001760                    JSR     PC,.MAMF         ;GO ENABLE PARITY ACTION
004764                           1$:
004764  004567  173770                    JSR     R5,$PRINT        ;GO TO PRINT ROUTINE
004770  012053                            STBANK                   ;ASK USER FOR STARTING BANK
004772  004767  004570                    JSR     PC,RECD
004776  000000                   .STBANK: .WORD   0                ;CONTAINS STARTING BANK #
005000  005767  177772                    TST     .STBANK          ;CHECK IF STARTING AT BANK #0
005004  001002                            BNE     3$
005006  004767  001336                    JSR     PC,RELOCP        ;GO RELOCATE THE PROGRAM TO TOP OF MEM
005012                           3$:
005012  004567  173742                    JSR     R5,$PRINT        ;GO TO PRINT ROUTINE
005016  012102                            BANKS                    ;ASK USER FOR # OF 4K BANKS TO TEST
005020  004767  004542                    JSR     PC,RECD
005024  000000                   .BANKS:  .WORD   0                ;CONTAINS # OF BANKS TO CHECK
005026  004567  173726                    JSR     R5,$PRINT        ;GO TO PRINT ROUTINE
005032  012137                            PAT                      ;ASK USER WHICH PATTERN
005034  004767  004526                    JSR     PC,RECD
```

```
                                                .PAT:   .WORD   0               ;CONTAINS PATTERN #
                                        175316  PRG2R:  MOV     #PRG2R,PERSTRT  ;SET PAR ERROR RESTART ADDRESS
                                                        JSR     PC,CKSWR        ;GO CHECK SWITCHES
                                                        MOV     .PAT,R0
                                                        ASL     R0              ;SHIFT PATTERN # TO FORM INDEX
                                                        ADD     RELOCF,R0       ;ADD RELOCATION FACTOR
                                                        MOV     WRTTAB(0),R0    ;GET UNRELOCATED PC OF ROUTINE
                                                        ADD     RELOCF,R0       ;ADD RELOCATION FACTOR
                                                        MOV     R0,PC           ;GO TO APPROPRIATE ROUTINE

                                                ;TABLE OF ROUTINES TO WRITE SELECTED PATTERNS
                        005120  005120          WRTTAB: .WORD   $1X8            ;1 XOR 8 ROUTINE
                        005122  005210                  .WORD   $3X9            ;3 XOR 9 ROUTINE
                        005124  005270                  .WORD   $8X13           ;8 XOR 13 ROUTINE
                        005126  005370                  .WORD   $USER           ;USER ROUTINE
                        005130  005476                  .WORD   $ROT0           ;ROTATING '0' ROUTINE
                        005132  005572                  .WORD   $ROT1           ;ROTATING '1' ROUTINE
                        005134  005666                  .WORD   $3X9P           ;PARITY 3 XOR 9 PATTERN
                        005136  005676                  .WORD   $ALL            ;ALL EXCEPT USER

                                                ;ROUTINES
                        005120  016746  177652  $1X8:   MOV     .STBANK,-(SP)   ;GET STARTING BANK #
                        005124  016746  177674          MOV     .BANKS,-(SP)    ;GET # OF 4K BANKS
                        005130  006316                  ASL     (SP)            ;MULTIPLY BY 32.
                        005132  005767  000560          TST     PRG3FLG         ;IF PROGRAM # 3 STOP WITH 256.
                        005136  001004                  BNE     1$              ;WORD BLOCK COUNT
                        005140  006316                  ASL     (SP)            ;TO FORM 128.
                        005142  006316                  ASL     (SP)            ;WORD BLOCK
                        005144  006316                  ASL     (SP)            ;COUNT
                        005146  006316                  ASL     (SP)
                        005150  011627          1$:     MOV     (SP),(PC)+      ;SAVE
                        005152  000000          2$:     .WORD   0               ;CONTAINS 128. WORD BLOCK COUNT
                        005154  004767  002230          JSR     PC,,1X8         ;GO WRITE 1 XOR 8 TEST PATTERN

                        005160  016746  177612          MOV     .STBANK,-(SP)   ;GET STARTING BANK #
                        005164  016746  177762          MOV     2$,-(SP)        ;GET 128. WORD BLOCK COUNT
                        005170  004767  002336          JSR     PC,,1X8         ;GO CHECK PATTERN
                        005174  005267  173542          INC     ICNT
                        005200  005767  177442          TST     ALLFLG          ;CHECK IF RUNNING ALL PATERNS
                        005204  001001                  BNE     $3X9            ;GO START 3X9 PATTERN IF RUNNING ALL
                        005206  000744                  BR      $1X8            ;OTHERWISE LOOP

                        005210  016746  177562  $3X9:   MOV     .STBANK,-(SP    ;GET STARTING BANK #
                        005214  016746  177604          MOV     .BANKS,-(SP)    ;GET # OF BANKS
                        005220  005767  000472          TST     PRG3FLG         ;IF PROGRAM # 2 STOP WITH 255.
                        005224  001004                  BNE     1$              ;WORD BLOCK COUNT
                        005226  006316                  ASL     (SP)            ;MULTIPLY BY 16.
                        005230  006316                  ASL     (SP)            ;TO FORM
                        005232  006316                  ASL     (SP)            ;256. WORD
                        005234  006316                  ASL     (SP)            ;BLOCK COUNT
                        005236  011627          1$:     MOV     (SP),(PC)+      ;SAVE
                        005240  000000          2$:     .WORD   0               ;CONTAINS 256. WORD BLOCK COUNT
                        005242  004767  002510          JSR     PC,,3X9         ;GO WRITE PATTERN

                                                ;CHECK PATTERN WRITTEN ABOVE
                        005246  016746  177562  3$:     MOV     .STBANK,-(SP    ;GET STARTING BANK #
```

```
                                        MOV     2$,-(SP)        ;GET # OF 256. WORD BLOCKS
                                        JSR     PC,..3X9        ;GO CHECK PATTERN
                                        INC     ICNT
                                        TST     ALLFLG          ;CHECK IF RUNNING ALL PATTERNS
                                        BNE     $8X13           ;GO START 8X13 PATTERN IF RUNNING ALL
                                        BR      3$

                                $8X13:  TST     @#PRG3FLG       ;CANNOT DO 8X13 PATTERN USING
                                        BNE     1$              ;PROGRAM # 3
                                        MOV     .STBANK,-(SP)   ;GET STARTING BANK #
                                        MOV     .BANKS,-(SP)    ;AND # OF 4K BANKS
                                        BIT     #1,(SP)         ;MUST BE AN EVEN # OF 4K BANKS
                                        BEQ     2$
                                1$:
                                        JSP     R5,$PRINT       ;GO TO PRINT ROUTINE
                                        QUEST                   ;PRINT ?
                                        JMP     PRG2            ;RESTART

                                2$:     JSR     PC,.8X13        ;GO WRITE PATTERN
                                        MOV     .STBANK,-(SP)
                                        MOV     .BANKS,-(SP)
                                        JSR     PC,..8X13       ;GO CHECK PATTERN
                                        INC     ICNT
                                        TST     ALLFLG          ;CHECK IF RUNNING ALL PATERNS
                                        BNE     $ROTO           ;GO DO ROTO PATTERN IF ALL SELECTED
                                        BR      $8X13


                                ;ROUTINE TO WRITE & CHECK USER CONSTANT
                                $USER:
                                        JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
                                        CONST                   ;ASK FOR USER CONSTANT
                                        JSR     PC,RECD
                                .CONST: .WORD   0               ;CONTAINS USER CONSTANT
                                1$:     MOV     .STBANK,-(SP)   ;GET STARTING BANK #
                                        MOV     .BANKS,-(SP)    ;GET 4K COUNT
                                        TST     PRG3FLG
                                        BNE     2$
                                        ASL     (SP)            ;MULTIPLY 4K BANK COUNT BY 16.
                                        ASL     (SP)            ;TO FORM 256. WORD BLOCK COUNT
                                        ASL     (SP)
                                        ASL     (SP)
                                2$:     MOV     (SP),(PC)+      ;SAVE
                                3$:     .WORD   0
                                        JSR     PC,.USER        ;GO WRITE USER CONSTANT
                                        MOV     .STBANK,-(SP)   ;GET STARTING BANK #
                                        MOV     3$,-(SP)        ;AND # OF 256. WORD BLOCKS
                                        JSR     PC,..USER       ;GO TO USER CHECK ROUTINE
                                        INC     ICNT
                                        TSTB    @#TKS           ;CHECK IF USER HAS TYPED A CHARACTER
                                        BPL     1$
                                        TST     @#TKB           ;CLEAR FLAG
                                        BR      $USER


                                ;ROTATING '0' ROUTINE
                                $ROTO:  MOV     .STBANK,-(SP)   ;GET STARTING BANK #
```

```
                                                        MOV     .BANKS,-(SP)    ;GET # OF BANKS
                                                        TST     PRG3FLG
                                                        BNE     2$
                                                        ASL     (SP)            ;MULTIPLY 4K BANK COUNT BY 16.
                                                        ASL     (SP)            ;TO FORM 256. WORD BLOCK COUNT
                                                        ASL     (SP)
                                                        ASL     (SP)
                                                2$:     MOV     (SP),(PC)+      ;SAVE
                                                3$:     .WORD   0               ;CONTAINS 256. WORD BLOCK COUNT
                                                        MOV     #-1,.CONST      ;SET CONSTANT
                                                        JSR     PC,.USER        ;GO WRITE ALL 1'S THROUGH MEMORY
                                                        MOV     .STBANK,-(SP)   ;GET STARTING BANK #
                                                        MOV     3$,-(SP)        ;GET # OF 256. WORD BLOCKS TO CHECK
                                                        JSR     PC,.ROT0        ;GO CHECK ROTATING 0 PATTERN
                                                        INC     ICNT            ;INCREMENT DISPLAY COUNT
                                                        TST     ALLFLG          ;CHECK IF RUNNING ALL PATERNS
                                                        BNE     $ROT1           ;GO TO $ROT1 IF RUNNING ALL PATTERNS
                                                        BR      $ROT0           ;LOOP

                                                $ROT1:  MOV     .STBANK,-(SP)   ;GET STARTING BANK #
                                                        MOV     .BANKS,-(SP)    ;GET # OF 4 K BANKS
                                                        TST     PRG3FLG         ;CHECK IF RUNNING PROGRAM 3
                                                        BNE     2$              ;BRANCH IF RUNNING PROGRAM 3
                                                        ASL     (SP)            ;SHIFT 4K BANK COUNT BY 16.
                                                        ASL     (SP)            ;TO FORM 256. WORD BLOCK COUNT
                                                        ASL     (SP)
                                                        ASL     (SP)
                                                2$:     MOV     (SP),(PC)+      ;SAVE
                                                3$:     .WORD   0               ;CONTAINS 256. WORD BLOCK COUNT
                                                        CLR     .CONST          ;SET CONSTANT
                                                        JSR     PC,.USER        ;GO WRITE 0'S THROUGHOUT
                                                        MOV     .STBANK,-(SP)   ;GET STARTING BANK #
                                                        MOV     3$,-(SP)        ;AND 256. WORD BLOCK COUNT
                                                        JSR     PC,.ROT1        ;GO CHECK ROTATING 1 PATTERN
                                                        INC     ICNT            ;INCREMENT PASS COUNT
                                                        TST     ALLFLG          ;CHECK IF RUNNING ALL PATTERNS
                                                        BEQ     $ROT1           ;LOOP IF NOT RUNNING ALL PATTERNS
                                                        JMP     $1X8            ;GO DO $1X8 PATTERN

                                                ;ROUTINE TO CHECK MEMORY USING 3 XOR 9 PARITY PATTERN
                                                $3X9P:  MOV     SP,PARPAT       ;SET INDICATOR TO WRITE PARITY PATTERN
                                                        JMP     $3X9

                                                ;ALL PATTERNS
                                                $ALL:   MOV     SP,ALLFLG       ;SET INDICATOR
                                                        JMP     $1X8            ;BEGIN WITH 1X8 TEST PATTERN
                                                        .SBTTL  PROGRAM # 3
                                                ;THIS PROGRAM IS THE SAME AS PROGRAM # 2 ABOVE EXCEPT THAT 256. WORD
                                                ;DATA BLOCKS MAY BE WRITTEN
                                                PRG3:   MOV     #STKPTR,SP      ;SET STACK PTR
                                                        MOV     #1,(PC)+        ;SET PROGRAM 3 FLAG
                                                PRG3FLG:.WORD   0               ;CONTAINS PRG3 INDICATOR
                                                        JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
                                                        PRG3M
                                                        JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
                                                        BANKS
```

```
005734  000167  176674                    JMP     PRG2A            ;GO TO PROGRAM 2

                                          .SBTTL  PROGRAM # 4
                                  ;THIS PROGRAM MAY BE USED TO READ/WRITE A USER CONTANT INTO ANY
                                  ;MEMORY LOCATION
005740  012706  000500            PRG4:   MOV     #STKPTR,SP       ;SET STACK PTR
005744  005037  006520                    CLR     @#PEFLG          ;CLEAR PARITY ERROR INDICATORS
005750  012737  001116  000004            MOV     #ERRTRP,@#ERRVEC
005756  000005                            RESET
005760  012707  005764                    MOV     #.+4,PC          ;RELOCATE BACK TO FIRST 4K
005764  004567  172770                    JSR     R5,$PRINT        ;GO TO PRINT ROUTINE
005770  012262                            PRG4M
005772  004767  003570                    JSR     PC,RECD
005776  000000                    $ADRSO: .WORD   0                ;CONTAINS ADDRESS BITS (15-0)
006000  016767  003634  000236            MOV     .1617,.EXTAD     ;GET EXTENDED ADDRESS BITS (17-16)
006006  006037  177570                    ROR     @#SWR            ;CHECK SWITCH 0
006012  103406                            BCS     PRG4A            ;GO TO PRG4A IF SET
006014  004567  172740                    JSR     R5,$PRINT        ;GO TO PRINT ROUTINE
006020  012161                            CONST
006022  004767  003540                    JSR     PC,RECD
006026  000000                    $CONST: .WORD   0                ;CONTAINS USER CONSTANT
006030  012767  006030  174330    PRG4A:  MOV     #PRG4A,PERSTRT   ;SET RESTART ADDRESS ON PAR ERROR
006036  016703  177734                    MOV     $ADRSO,R3        ;GET ADDRESS BITS (15-0)
006042  016704  000176                    MOV     .EXTAD,R4        ;GET ADDRESS BITS (17-16)
006046  001020                            BNE     10$              ;BRANCH IF MEM MGMT REQUIRED
006050  022703  020000                    CMP     #20000,R3        ;CHECK IF ADDRESS IS LESS THAN 20000
006054  101412                            BLOS    1$               ;BRANCH IF IT IS NOT
006056  006037  177570                    ROR     @#SWR            ;CHECK IF READING
006062  103431                            BCS     3$               ;BRANCH IF READING
006064  004767  172214                    JSR     PC,$SAVR         ;GO SAVE REGISTERS ON THE STACK
006070  004767  000254                    JSR     PC,RELOCP        ;GO RELOCATE PROGRAM
006074  004767  172230                    JSR     PC,$RESTR        ;RESTORE REGISTERS FROM STACK
006100  000422                            BR      3$               ;GO TO 3$
006102  022703  160000            1$:     CMP     #160000,R3       ;CHECK IF MEM MGMT WILL BE REQUIRED
006106  101017                            BHI     3$               ;GO TO 3$ IF NOT REQUIRED
006110  010302                    10$:    MOV     R3,R2            ;GET ADDRESS BITS (15-0)
006112  012700  000006                    MOV     #6,R0            ;SET SHIFT COUNT
006116  006204                    2$:     ASR     R4               ;SHIFT 18 BIT ADDRESS
006120  006003                            ROR     R3               ;6 PLACES RIGHT
006122  077003                            SOB     R0,2$
006124  004767  000714                    JSR     PC,LDMMO         ;GO SETUP MEM MGMT
006130  010337  172342                    MOV     R3,@#KIPAR1      ;SET KIPAR1
006134  042702  177700                    BIC     #177700,R2       ;CLEAR ADDRESS BITS (15-6)
006140  052702  020000                    BIS     #20000,R2        ;SET ADDRESS REGISTER
006144  000401                            BR      4$               ;GO TO 4$
006146  010302                    3$:     MOV     R3,R2            ;SET ADDRESS REGISTER
006150  016700  177652            4$:     MOV     $CONST,R0        ;GET USER CONSTANT
006154  012737  005740  000060            MOV     #PRG4,@#TKVEC    ;SET KEYBOARD INTERRUPT VECTOR
006162  052737  000100  177560            BIS     #100,@#TKS       ;SET IE BIT IN KEYBOARD CSR
006170  006037  177570                    ROR     @#SWR            ;CHECK SWITCH 0
006174  103014                            BCC     5$               ;BRANCH IF NOT SET
006176  011246                            MOV     (R2),-(SP)       ;PUSH DATA TO BE TYPED ONTO STACK
006200  004767  173542                    JSR     PC,02A           ;GO TYPE DATA
006204  004567  172550                    JSR     R5,$PRINT        ;GO TO PRINT ROUTINE
006210  015020                            $CRLF
006212  062767  000002  177556            ADD     #2,$ADRSO        ;STEP ADDRESS
```

```
006220  005567  000020                    ADC     .EXTAD
006224  000701                            BR      PRG4A
006226  010012                   5$:      MOV     R0,(R2)          ;WRITE USER CONSTANT INTO ADDRESS
006230  012203                            MOV     (R2)+,R3         ;GET DATA WRITTEN
006232  020003                            CMP     R0,R3            ;CHECK DATA
006234  001401                            BEQ     6$
006236  104400                            HLT                      ;REPORT ERROR
006240  005742                   6$:      TST     -(R2)            ;RESTORE ADDRESS
006242  000771                            BR      5$               ;LOOP BACK
006244  000000                   .EXTAD:  .WORD   0                ;CONTAINS EXTENDED ADDRESS BITS
                                          .SBTTL  PROGRAM SUBROUTINES
                                          .SBTTL  RELOCATION ROUTINES
                                 ;ROUTINE TO RELOCATE PROGRAM CODE
006246  012500                   RELOC:   MOV     (R5)+,R0         ;GET FROM ADDRESS
006250  011502                            MOV     (R5),R2          ;GET TO ADDRESS
006252  010203                            MOV     R2,R3
006254  062703  017776                    ADD     #17776,R3        ;MOVES 4K
006260  012737  006330  000004            MOV     #4$,@#ERRVEC     ;SET TIME OUT TRAP
006266  005004                            CLR     R4               ;CLEAR RELOCATION SUCCESSFUL INDICATOR
006270  005723                            TST     (R3)+            ;CHECK IF MEMORY IS AVAILABLE
006272  012022                   1$:      MOV     (R0)+,(R2)+      ;RELOCATE
006274  020203                            CMP     R2,R3            ;RELOCATION COMPLETE?
006276  001375                            BNE     1$
006300  011503                            MOV     (R5),R3
006302  020203                   2$:      CMP     R2,R3
006304  001413                            BEQ     5$               ;BRANCH IF DONE
006306  024042                            CMP     -(R0),-(R2)      ;CHECK THAT DATA WAS RELOCATED PROPERLY
006310  001774                            BEQ     2$
006312  005703                            TST     R3               ;CHECK IF RELOCATING BACK TO 000000
006314  001403                            BEQ     3$
006316  104400                            HLT                      ;ERROR! CANNOT RELOCATE PROGRAM CODE
                                                                   ;TO UPPER MEMORY BANK PROPERLY
006320  000000                            HALT
006322  000767                            BR      2$               ;CONTINUE RELOCATING AT YOUR PERIL
006324  000000                   3$:      HALT                     ;ERROR! CANNOT RELOCATE CODE BACK TO
                                                                   ;TO 000000 PROPERLY
006326  000777                            BR      .
006330  022626                   4$:      CMP     (SP)+,(SP)+      ;RESTORE STACK PTR
006332  005104                            COM     R4
006334  000240                   5$:      NOP
006336  012702  000754                    MOV     #RELOCF,R2       ;GET ADDRESS OF RELOCATION FACTOR
006342  061502                            ADD     (R5),R2          ;ADD FACTOR
006344  012512                            MOV     (R5)+,(R2)       ;RELOCATED RELOCF NOW CONTAINS RELOCATION
                                                                   ;FACTOR
006346  000205                            RTS     5                ;RETURN, R4=-1 IF NO RELOCATION


                                 ;ROUTINE TO RELOCATE PROGRAM CODE FROM ORIGINAL POSITION (0-4K) TO
                                 ;TOP OF MEMORY.
006350  012700  020000           RELOCP:  MOV     #20000,R0        ;SET UP TO SCAN FOR TOP OF MEMORY
006354  012737  000006  000004            MOV     #ERRVEC+2,@#ERRVEC
006362  062700  020000           1$:      ADD     #20000,R0        ;INCREMENT SCAN ADDRESS
006366  000261                            SEC                      ;SET TIME OUT INDICATOR
006370  005710                            TST     (R0)             ;CHECK FOR EXISTANT MEMORY
006372  103373                            BCC     1$               ;'C' WILL BE CLEAR IF MEMORY EXISTS
006374  012737  001116  000004            MOV     #ERRTRP,@#ERRVEC
```

```
006402  162700  020000                  SUB     #20000,R0           ;ADJUST TO LAST EXISTANT 4K
006406  010067  000006                  MOV     R0,2$               ;PASS RELOCATION ADDRESS TO RELOC ROUTINE
006412  004567  177530                  JSR     R5,RELOC            ;RELOCATE PROGRAM
006416  000000                          000000                      ;FROM ADDRESS 000000
006420  000000                  2$:     .WORD   0                   ;TO LAST 4K BANK
006422  004567  172332                  JSR     R5,$PRINT           ;GO TO PRINT ROUTINE
006426  012301                          RELOCM
006430  016746  177764                  MOV     2$,-(SP)            ;PASS TO 02A ROUTINE
006434  062716  012356                  ADD     #REL24K,(SP)        ;SET UP RESTART ADDRESS
006440  004767  173302                  JSR     PC,02A              ;TYPE RESTART ADDRESS
006444  011667  000006                  MOV     (SP),3$             ;SAVE RETURN ADDRESS IN 3$ BELOW
006450  066706  177744                  ADD     2$,SP               ;RESET STACK PTR
006454  012716                          MOV     (PC)+,(SP)          ;GET RETURN ADDRESS
006456  000000                  3$:     .WORD   0                   ;CONTAINS RETURN PC
006460  066716  177734                  ADD     2$,(SP)             ;ADJUST RETURN PC
006464  000207                          RTS     PC

                                        .SBTTL  PARITY ERROR SERVICE ROUTINE
                                ;WHEN A PARITY ERROR IS DETECTED THIS ROUTINE SCANS MEMORY FOR THE AD-
                                ;DRESS CAUSING THE PARITY ERROR. WHEN THE ADDRESS IS LOCATED THE ROUTINE
                                ;HALTS WITH THE ADDRESS+2 IN R0. TO CONTINUE AFTER THE ERROR PRESS CONTINUE.
006466  010067  000173          .PARSRV:MOV     R0,SAVR0            ;SAVE R0 IN SAVR0
006472  012700  006664                  MOV     #SAVR0+2,R0
006476  010120                          MOV     R1,(R0)+
006500  010220                          MOV     R2,(R0)+
006502  010320                          MOV     R3,(R0)+
006504  010420                          MOV     R4,(R0)+
006506  010520                          MOV     R5,(R0)+
006510  004567  172244                  JSR     R5,$PRINT           ;GO TO PRINT ROUTINE
006514  006676                          PARERR
006516  005027                          CLR     (PC)+               ;CLEAR PARITY ERROR INDICATORS
006520     000                  PEFLG:  .BYTE   0                   ;NOT 0/0 =PAR ERR/NO PAR ERR
006521     000                  PENFLG: .BYTE   0                   ;NOT 0/0=PAR ERR DETECTED/NOT DETECTED ON SCAN
006522  012737  006570  000114          MOV     #2$,@#PARVEC                ;SET PARITY ERROR TRAP
006530  012737  006626  000004          MOV     #4$,@#ERRVEC                ;SET TIME OUT TRAP VECTOR
006536  005002                          CLR     R2
006540  005767  172206                  TST     MMAVA               ;CHECK IF MEM MGMT IS AVAILABLE
006544  001407                          BEQ     1$                  ;BRANCH IF NOT AVAILABLE
006546  004767  000272                  JSR     PC,LDMM0            ;SET UP MEM MGMT
006552  105237  172301                  INCB    @#KIPDR0+1          ;ALLOW FULL 4K PAGE ADDRESSING
006556  012737  007142  000250          MOV     #MMABTO,@#MMVEC ;SET MEM MGMT ABORT TRAP VECTOR
006564  012200                  1$:     MOV     (R2)+,R0            ;SCAN ALL ADDRESSES
006566  000776                          BR      1$
006570  110667  177724          2$:     MOVB    SP,PEFLG            ;SET PARITY ERROR FOUND INDICATOR
006574  010003                          MOV     R0,R3
006576  104400                          HLT                         ;PARITY ERROR! ADDRESS+2 IS IN R2 DATA
                                                                    ;IS IN R0
006600  000002                          RTI                         ;CONTINUE SCAN
006602  000240                  3$:     NOP                         ;INSERT HALT INST TO EXAMINE PARITY REGS
006604  005067  177710                  CLR     PEFLG               ;CLEAR PARITY ERROR INDICATORS
006610  012706  000500                  MOV     #STKPTR,SP          ;RESET STACK PTR
006614  000005                          RESET
006616  004767  000122                  JSR     PC,.MAMF            ;GO ENABLE PARITY ERROR DETECTION
006622  000177  173540                  JMP     @PERSTRT            ;RESTART SELECTED PROGRAM

                                ;SERVICE ROUTINE IF PARITY ERROR NOT DETECTED ON SCAN
```

```
2674  006626  105767  177666          4$:     TSTB    PEFLG           ;BRANCH IF PARITY ERROR WAS
2675  006632  001363                          BNE     3$              ;DETECTED ON SCAN
2676  006634  016602  000004                  MOV     4(SP),R2        ;GET PC AT TIME OF ERROR
2677  006640  162702  000002                  SUB     #2,R2           ;BACK IT UP
2678  006644  110667  177651                  MOVB    SP,PEFLG        ;SET IND = NO PAR ERROR DETECTED ON SCAN
2679  006650  004567  172104                  JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
2680  006654  006717                          NOFIND
2681  006656  104400                          HLT                     ;ERROR! PARITY ERROR NOT DETECTED ON SCAN
2682  006660  000750                          BR      3$
2683                                  ;THE BELOW 6 WORDS CONTAINS THE SAVED CONTENTS OF R0-R5 WHEN THE
2684                                  ;PARITY ERROR OCCURRED
2685  006662  000000                  SAVR0:  .WORD   0
2686  006664  000000                  SAVR1:  .WORD   0
2687  006666  000000                  SAVR2:  .WORD   0
2688  006670  000000                  SAVR3:  .WORD   0
2689  006672  000000                  SAVR4:  .WORD   0
2690  006674  000000                  SAVR5:  .WORD   0
2691
2692  006676  005015  040520  044522  PARERR: .ASCIZ  <15><12>'PARITY ERROR'<15><12>
2693  006704  054524  042440  051122
2694  006712  051117  005015     000
2695  006717     116  052117  043040  NOFIND: .ASCIZ  'NOT FOUND ON SCAN'<15><12>
2696  006724  052517  042116  047440
2697  006732  020116  041523  047101
2698  006740  005015     000
2699          006744                          .EVEN
2700
2701                                  ;ROUTINE TO ENABLE PARITY ERROR ACTION ON MA/MF PARITY MEMORIES
2702          172100                          PARCSR=172100           ;ADDRESS OF FIRST PARITY REGISTER
2703          000114                          PARVEC=114              ;PARITY ERROR INTERRUPT VECTOR ADDRESS
2704
2705  006744  032737  000040  177570  .MAMF:  BIT     #40,@#SWR       ;CHECK IF PARITY ERROR DETECTION IS TO
2706  006752  001033                          BNE     DISPAR          ;BE ENABLED. BRANCH IF NOT TO BE ENABLED
2707  006754  013746  000004                  MOV     @#ERRVEC,-(SP)  ;SAVE ERROR TRAP VECTOR
2708  006760  012737  000006  000004          MOV     #ERRVEC+2,@#ERRVEC      ;SET TIME OUT TRAP TO RETURN (VIA RTI)
2709  006766  012737  006466  000114          MOV     #.PARSRV,@#PARVEC       ;SET PARITY ERROR TRAP VECTOR
2710  006774  012737  000340  000116          MOV     #340,@#PARVEC+2 ;PRIORITY LEVEL 7 ON TRAP
2711  007002  012700  172100                  MOV     #PARCSR,R0      ;GET FIRST ADDRESS OF PARITY REGISTER
2712  007006  012702  000001                  MOV     #1,R2
2713  007012  005027                          CLR     (PC)+           ;CLEAR AVAILABILITY INDICATOR
2714  007014  000000                  PARAVA: .WORD   0               ;CONTAINS AVAILABILITY INDICATOR
2715
2716                                  ;ENABLE ALL AVAILABLE PARITY REGISTERS
2717  007016  000262                  1$:     SEV                     ;SET TIME OUT INDICATOR
2718  007020  012720  000001                  MOV     #1,(R0)+        ;SET ACTION ENABLE IF AVAILABLE
2719  007024  102402                          BVS     2$              ;BRANCH IF NO PARITY AVAILABLE
2720  007026  050267  177762                  BIS     R2,PARAVA       ;SET AVAILABLITY INDICATOR
2721  007032  006302                  2$:     ASL     R2              ;SHIFT INDICATOR
2722  007034  103370                          BCC     1$
2723  007036  012637  000004                  MOV     (SP)+,@#ERRVEC  ;RESTORE ERROR TRAP VECTOR
2724  007042  000207                  DISPAR: RTS     PC              ;RETURN
2725
2726                                          .SBTTL  MEM MGMT ROUTINES
2727                                  ;ROUTINE TO INITIALIZE MEMORY MANAGEMENT REGISTERS
2728  007044  000240                  LDMMO:  NOP
2729  007046  005767  171700                  TST     MMAVA
```

B06


```
[illegible]                                 BEQ     1$
[illegible]                                 MOV     #[illegible]*256.-400+UP+RW,@#KIPDR0   ;SET [illegible]
[illegible]                                 MOV     #200*256.-400+UP+RW,@#KIPDR1   ;SET [illegible]
[illegible]                                 CLR     @#KIPDR2
[illegible]                                 MOV     #0,@#KIPAR2
[illegible]                                 MOV     #200*256.-400+UP+RW,@#KIPDR7   ;SET KIPDR7=RW UP 200 BLOCKS
[illegible]                                 CLR     @#KIPAR0
[illegible]                                 MOV     #200,@#KIPAR1
[illegible]                                 MOV     #[illegible],@#KIPAR7
[illegible]                                 MOV     #1,@#SR0                ;ENABLE MEM MGMT
[illegible]                                 NOP
[illegible]                         1$:     RTS     PC

                                    ;MEMORY MANAGEMENT ABORT ROUTINE FOR WRITE UP
[illegible]  [illegible]  020000            MMABT0: MOV     #20000,R2       ;RESET R2
[illegible]  [illegible]  [illegible]  172342       ADD     #200,@#KIPAR1   ;ADVANCE TO NEXT [illegible]
[illegible]                                 MOV     @#SR2,(SP)      ;RETURN TO INSTRUCTION THAT
[illegible]                                 CLR     @#SR0           ;DISABLE MEM MGMT
[illegible]  [illegible]  000001  177572    MOV     #1,@#SR0        ;ENABLE MEM MGMT
[illegible]  000002                         RTI                     ;CAUSED THE ABORT

                                    ;MEM MGMT ABORT SERVICE FOR WRITE DOWN
[illegible]  012702  040000         MMABT1: MOV     #40000,R2       ;RESET R2
[illegible]200  [illegible]  000200  172342 SUB     #200,@#KIPAR1
[illegible]  001436                         BEQ     2$
[illegible]  013716  177576                 MOV     @#SR2,(SP)
[illegible]  012737  000001  177572         MOV     #1,@#SR0        ;ENABLE MEM MGMT
[illegible]  000002                         RTI
[illegible]                         2$:
[illegible]  005037  177572                 CLR     @#SR0           ;DISABLE MEM MGMT
[illegible]  052766  000002  000002         BIS     #V,2(SP)
007236  000002                              RTI

                                    ;ROUTINE TO SET UP MEMORY MANAGEMENT FOR PATTERN TESTS
007240  005702                      STMM2:  TST     R2              ;CHECK IF TESTING BANK # 0
007242  001442                              BEQ     2$              ;EXIT IF BANK # 0
007244  005767  171502                      TST     MMAVA
007250  001005                              BNE     1$              ;BRANCH IF MEM MGMT AVAILABLE
007252  006002                              ROR     R2              ;ADJUST ADDRESS
007254  006002                              ROR     R2
007256  006002                              ROR     R2
007260  006002                              ROR     R2
007262  000207                              RTS     PC              ;RETURN

007264  004767  177554              1$:     JSR     PC,LDMMC        ;GO MAKE INITIAL SET UP
007270  000302                              SWAB    R2
007272  006002                              ROR     R2
007274  010237  172344                      MOV     R2,@#KIPAR2
007300  062702  000200                      ADD     #200,R2
007304  010237  172346                      MOV     R2,@#KIPAR3
007310  012737  077406  172304              MOV     #200*256.-400+UP+RW,@#KIPDR2   ;SET KIPDR2=RW UP 200 BLOCKS
007316  012737  077406  172306              MOV     #200*256.-400+UP+RW,@#KIPDR3   ;SET KIPDR3=RW UP 200 BLOCKS
007324  005037  172310                      CLR     @#KIPDR4
007330  012702  040000                      MOV     #40000,R2
007334  012737  007352  000250              MOV     #MMABT2,@#MMVEC
[illegible]  012737  000001  177572         MOV     #1,@#SR0        ;ENABLE MEM MGMT
```

```
                                3$:     RTS     PC

                                ;ROUTINE TO SERVICE 8 XOR 13 ABORTS
                                MMABT2: NOP
                                        MOV     #40000,R2
                                        ADD     #400,@#KIPAR2
                                        ADD     #400,@#KIPAR3
                                        MOV     @#SR2,(SP)      ;SET RETURN TO INSTRUCTION THAT ABORTED
                                        MOV     #1,@#SR0        ;ENABLE MEM MGMT
                                        RTI

                                        .SBTTL  1 XOR 8 ROUTINES
                                ;ROUTINE TO WRITE 1 XOR 8 WORST CASE NOISE PATTERN
                                ;CALL:  MOV     BANK #,-(SP)    ;PUSH STARTING BANK # ON THE STACK
                                ;       MOV     BLKCNT,-(SP)    ;PUSH 128. WORD BLOCK COUNT ON THE STACK
                                ;       JSR     PC,.1X8
                                .1X8:   MOV     2(SP),R3        ;GET # OF 128. WORD BLOCKS TO WRITE
                                        MOV     4(SP),R2        ;GET STARTING BANK #
                                        JSR     PC,STMM2        ;GO SET UP MEM MGMT
                                1$:     MOV     #-1,R0          ;SET UP DATA REGISTERS
                                        MOV     R0,R5
                                        COM     R5

                                2$:     COM     R0
                                        COM     R5
                                        MOV     #8.,R4          ;SET 128. WORD COUNTER
                                3$:     MOV     R0,(R2)+        ;WRITE 128. WORDS
                                        MOV     R5,(R2)+
                                        MOV     R0,(R2)+
                                        MOV     R5,(R2)+

                                        MOV     R0,(R2)+
                                        MOV     R5,(R2)+
                                        MOV     R0,(R2)+
                                        MOV     R5,(R2)+

                                        MOV     R0,(R2)+
                                        MOV     R5,(R2)+
                                        MOV     R0,(R2)+
                                        MOV     R5,(R2)+

                                        MOV     R0,(R2)+
                                        MOV     R5,(R2)+
                                        MOV     R0,(R2)+
                                        MOV     R5,(R2)+

                                        DEC     R4              ;DECREMENT 128. WORD COUNTER
                                        BNE     3$
                                        DEC     R3              ;DECREMENT BLOCK COUNT
                                        BNE     2$
                                        MOV     (SP)+,(SP)      ;ADJUST STACK
                                        MOV     (SP)+,(SP)
                                        RTS     PC              ;RETURN TO CALLER

                                ;ROUTINE TO CHECK 1 XOR 8 PATTERN WRITTEN ABOVE
```

D06


```
                                        ;CALL:  MOV     BANK #,-(SP)    ;PUSH STARTING BANK # ON THE STACK
                                        ;       MOV     BLKCNT,-(SP)    ;PUSH 128. WORD BLOCK COUNT ON STACK
                                        ;       JSR     PC,..1X8

        007522  000240                  ..1X8:  NOP
        007524  004767  002120                  JSR     PC,CKSWR        ;GO CHECK SWITCH REGISTER
        007530  016667  000002  171220  10$:    MOV     2(SP),COUNT     ;GET BLOCK COUNT
        007536  016602  000004                  MOV     4(SP),R2        ;GET STARTING BANK #
        007542  004767  177472                  JSR     PC,STMM2        ;GO SET UP MEM MGMT
        007546  005000                  1$:     CLR     R0              ;CLEAR TEST WORD
        007550  005767  171170                  TST     ICOUNT          ;IF BIT 15 OF ICOUNT =1 THEN PATTERN
        007554  100401                          BMI     .+4             ;IS COMPLEMENTED
        007556  005100                          COM     R0              ;COMPLEMENT TEST WORD
        007560  012705  000040          2$:     MOV     #32.,R5         ;SET 128. WORD COUNTER

        007564  005100                  3$:     COM     R0
        007566  012203                          MOV     (R2)+,R3        ;GET TEST DATA
        007570  020003                          CMP     R0,R3           ;COMPARE WITH CHECK WORD
        007572  001403                          BEQ     .+10
        007574  005046                          CLR     -(SP)           ;PUSH FAKE STATUS ON THE STACK
        007576  004767  171372                  JSR     PC,ERROR        ;ERROR! MEM DATA (R3) NOT = TEST DATA
                                                                        ;(R0), ADDRESS=(R2)-2

        007602  005100                          COM     R0
        007604  012203                          MOV     (R2)+,R3        ;GET TEST DATA
        007606  020003                          CMP     R0,R3           ;COMPARE WITH CHECK WORD
        007610  001403                          BEQ     .+10
        007612  005046                          CLR     -(SP)           ;PUSH FAKE STATUS ON THE STACK
        007614  004767  171354                  JSR     PC,ERROR        ;ERROR! MEM DATA (R3) NOT = TEST DATA
                                                                        ;(R0), ADDRESS=(R2)-2

        007620  005100                          COM     R0
        007622  012203                          MOV     (R2)+,R3        ;GET TEST DATA
        007624  020003                          CMP     R0,R3           ;COMPARE WITH CHECK WORD
        007626  001403                          BEQ     .+10
        007630  005046                          CLR     -(SP)           ;PUSH FAKE STATUS ON THE STACK
        007632  004767  171336                  JSR     PC,ERROR        ;ERROR! MEM DATA (R3) NOT = TEST DATA
                                                                        ;(R0), ADDRESS=(R2)-2

        007636  005100                          COM     R0
        007640  012203                          MOV     (R2)+,R3        ;GET TEST DATA
        007642  020003                          CMP     R0,R3           ;COMPARE WITH CHECK WORD
        007644  001403                          BEQ     .+10
        007646  005046                          CLR     -(SP)           ;PUSH FAKE STATUS ON THE STACK
        007650  004767  171320                  JSR     PC,ERROR        ;ERROR! MEM DATA (R3) NOT = TEST DATA
                                                                        ;(R0), ADDRESS=(R2)-2

        007654  005305                          DEC     R5              ;DECREMENT 128. WORD COUNTER
        007656  001342                          BNE     3$
        007660  005100                          COM     R0              ;COMPLEMENT CHECK WORD
        007662  005367  171070                  DEC     COUNT           ;DECREMENT BLOCK COUNT
        007666  001334                          BNE     2$
        007670  016602  000004                  MOV     4(SP),R2        ;GET BANK #
        007674  004767  177340                  JSR     PC,STMM2
        007700  016667  000002  171050  4$:     MOV     2(SP),COUNT     ;GET # OF 128. WORD BLOCKS TO COMPLEMENT
        007706  006367  171032                  ASL     ICOUNT
```

E06


```
007712  102306                   BVC     10$
007714  012705  000040   50$:    MOV     #32.,R5
007720  [illegible]      5$:     COM     (R3)+          ;COMPLEMENT PATTERN
007722  [illegible]              XOR     (R2)+
007724  005122                   COM     (R2)+
007726  005122                   COM     (R2)+
007730  005305                   DEC     R5
007732  [illegible]              BNE     5$
007734  005367  171016           DEC     COUNT
007740  001365                   BNE     50$
007742  005767  170775           TST     ICOUNT
007746  [illegible]              BNE     10$
007750  012616                   MOV     (SP)+,(SP)
007752  012616                   MOV     (SP)+,(SP)
007754  000207                   RTS     PC

                                 .SBTTL  3 XOR 9 ROUTINES
                         ;ROUTINE TO WRITE 3XOR9 WORST CASE NOISE TEST PATTERN
                         ;CALL:  MOV     BANK #,-(SP)    ;PUSH STARTING BANK # ON STACK
                         ;       MOV     BLKCNT,-(SP)    ;PUSH 256. WORD BLOCK COUNT ON STACK
                         ;       JSR     PC,.3X9         ;CALL ROUTINE
                         ;
007756  016602  000004   .3X9:   MOV     4(SP),R2        ;GET STARTING BANK #
007762  004767  177252           JSR     PC,STMM2
007766  005000                   CLR     R0
007770  010003                   MOV     R0,R3
007772  005103                   COM     R3              ;R0 (0) AND R3 (-1) IS THE DATA WRITTEN
007774  005767  173366           TST     PARPAT          ;BRANCH IF PARITY MEMORY PATTERN IS
010000  001402                   BEQ     1$              ;NOT TO BE WRITTEN

010002  012700  000401           MOV     #401,R0         ;WRITE PARITY 3X9 PATTERN
010006  012704  000020   1$:     MOV     #16.,R4         ;EACH LOOP WRITES 256. WORDS

010012  010022           2$:     MOV     R0,(R2)+
010014  010022                   MOV     R0,(R2)+
010016  010022                   MOV     R0,(R2)+
010020  010022                   MOV     R0,(R2)+

010022  010322                   MOV     R3,(R2)+
010024  010322                   MOV     R3,(R2)+
010026  010322                   MOV     R3,(R2)+
010030  010322                   MOV     R3,(R2)+

010032  010022                   MOV     R0,(R2)+
010034  010022                   MOV     R0,(R2)+
010036  010022                   MOV     R0,(R2)+
010040  010022                   MOV     R0,(R2)+

010042  010322                   MOV     R3,(R2)+
010044  010322                   MOV     R3,(R2)+
010046  010322                   MOV     R3,(R2)+
010050  010322                   MOV     R3,(R2)+

010052  005304                   DEC     R4
010054  001356                   BNE     2$
010056  005100                   COM     R0
```

```
010060  005103                            COM     R3
010062  005767  173300                    TST     PARPAT          ;BRANCH IF PARITY MEMORY PATTERN IS
010066  001402                            BEQ     3$              ;NOT TO BE WRITTEN

010070  004767  000014                    JSR     PC,.XOR39       ;GO GET CONSTANTS
010074  005366  000002            3$:     DEC     2(SP)           ;DECREMENT 256. WORD BLOCK COUNT
010100  001342                            BNE     1$
010102  012616                            MOV     (SP)+,(SP)      ;ADJUST STACK
010104  012616                            MOV     (SP)+,(SP)
010106  000207                            RTS     PC

                                  ;ROUTINE TO SET CONSTANTS FOR WRITING/CHECKING 3 XOR PATTERN WIT-
                                  ;PARITY.
010110  032702  000010            .XOR39: BIT     #10,R2          ;CHECK BIT 3
010114  001404                            BEQ     .3IS0           ;BRANCH IF BIT 3 = 0
010116  032702  001000            .3IS1:  BIT     #1000,R2        ;CHECK BIT 9
010122  001404                            BEQ     .3NOT9          ;BRANCH IF BIT 9 =0
010124  000407                            BR      .3IS9
010126  032702  001000            .3IS0:  BIT     #1000,R2        ;CHECK BIT 9
010132  001404                            BEQ     .3IS9           ;BRANCH IF 0
010134  005767  170624            .3NOT9: TST     ICOUNT          ;CHECK IF NORMAL OR COMPLEMENT DATA
010140  100004                            BPL     LDCOMP          ;GO LOAD COMPLEMENT CONSTANTS
010142  100410                            BMI     LDNORM          ;GO LOAD NORMAL CONSTANTS
010144  005767  170574            .3IS9:  TST     ICOUNT          ;CHECK IF NORMAL OR COMPLEMENT DATA
010150  100005                            BPL     LDNORM          ;GO LOAD NORMAL CONSTANTS
010152  012700  177777            LDCOMP: MOV     #-1,R0          ;SET COMPLEMENT CONSTANTS
010156  012703  000401                    MOV     #401,R3
010162  000207                            RTS     PC              ;RETURN
010164  012700  000401            LDNORM: MOV     #401,R0         ;LOAD NORMAL CONSTANTS
010170  012703  177777                    MOV     #-1,R3
010174  000207                            RTS     PC

                                  ;ROUTINE TO CHECK 3 XOR 9 WORST CASE NOISE PATTERN
                                  ;CALL:   MOV     BANK#,-(SP)     ;PUSH STARTING BANK # ONTO STACK
                                  ;        MOV     BLKCNT,-(SP)    ;AND 256. WORD BLOACK COUNT
                                  ;        JSR     PC,..3X9        ;CALL ROUTINE

010176  000240                    ..3X9:  NOP
010200  004767  001444                    JSR     PC,CKSWR        ;GO CHECK SWITCH REGISTER

                                  ;CHECK WORST CASE PATTERN
010204  016604  000002            1$:     MOV     2(SP),R4        ;GET 256. BLOCK WORD COUNT
010210  016602  000004                    MOV     4(SP),R2        ;GET FIRST BANK #
010214  004767  177020                    JSR     PC,STMM2        ;GO SET UP MEM MGMT
010220  005000                            CLR     R0              ;SET CHECK WORD
010222  005767  170516                    TST     ICOUNT          ;IF ICOUNT IS NEG AM CHECKING COMP-
010226  100001                            BPL     .+4             ;LEMENTED PATTERN
010230  005100                            COM     R0              ;SO COMPLEMENT CHECK WORD
010232  012705  000100            2$:     MOV     #64.,R5         ;SET 256. WORD COUNTER

010236  005767  173124            3$:     TST     PARPAT          ;BRANCH IF PARITY MEMORY PATTERN IS
010242  001402                            BEQ     30$             ;NOT TO BE CHECKED

010244  004767  177640                    JSR     PC,.XOR39       ;GO GET CONSTANT
010250                            30$:
010250  012203                            MOV     (R2)+,R3        ;GET TEST DATA
```

```
010252  020003                          CMP     R0,R3           ;COMPARE WITH CHECK WORD
010254  001403                          BEQ     .+10
010256  005046                          CLR     -(SP)           ;PUSH FAKE STATUS ON THE STACK
010260  004767  170710                  JSR     PC,ERROR        ;ERROR! MEM DATA (R3) NOT = TEST DATA
                                                                ;(R0), ADDRESS=(R2)-2

010264  012203                          MOV     (R2)+,R3        ;GET TEST DATA
010266  020003                          CMP     R0,R3           ;COMPARE WITH CHECK WORD
010270  001403                          BEQ     .+10
010272  005046                          CLR     -(SP)           ;PUSH FAKE STATUS ON THE STACK
010274  004767  170674                  JSR     PC,ERROR        ;ERROR! MEM DATA (R3) NOT = TEST DATA
                                                                ;(R0), ADDRESS=(R2)-2

010300  012203                          MOV     (R2)+,R3        ;GET TEST DATA
010302  020003                          CMP     R0,R3           ;COMPARE WITH CHECK WORD
010304  001403                          BEQ     .+10
010306  005046                          CLR     -(SP)           ;PUSH FAKE STATUS ON THE STACK
010310  004767  170660                  JSR     PC,ERROR        ;ERROR! MEM DATA (R3) NOT = TEST DATA
                                                                ;(R0), ADDRESS=(R2)-2

010314  012203                          MOV     (R2)+,R3        ;GET TEST DATA
010316  020003                          CMP     R0,R3           ;COMPARE WITH CHECK WORD
010320  001403                          BEQ     .+10
010322  005046                          CLR     -(SP)           ;PUSH FAKE STATUS ON THE STACK
010324  004767  170644                  JSR     PC,ERROR        ;ERROR! MEM DATA (R3) NOT = TEST DATA
                                                                ;(R0), ADDRESS=(R2)-2


010330  005100                          COM     R0              ;COMPLEMENT CHECK WORD
010332  005305                          DEC     R5              ;DECREMENT 256. WORD COUNTER
010334  001340                          BNE     3$
010336  005100                          COM     R0              ;COMPLEMENT CHECK WORD
010340  005304                          DEC     R4              ;DECREMENT BLOCK COUNTER
010342  001333                          BNE     2$

010344  032737  040000  177570          BIT     #40000,@#SWR    ;LOOP ON TEST?
010352  001314                          BNE     1$              ;BRANCH IF LOOP ON TEST DESIRED
010354  016667  000002  170374  40$:    MOV     2(SP),COUNT     ;GET # OF 256. WORD BLOCKS TO CHECK
010362  016602  000004                  MOV     4(SP),R2        ;GET STARTING BANK #
010366  004767  176546                  JSR     PC,STMM2        ;GO SET UP MEM MGMT IF REQUIRD

                                ;CHECK WORST CASE BIT COMPLEMENT PATTERN
010372  005000                          CLR     R0
010374  005767  170344                  TST     ICOUNT          ;CHECK IF COMPLEMENT PATERN
010400  100001                          BPL     .+4
010402  005100                          COM     R0              ;COMPLEMENT CHECK WORD
010404  012704  000100          4$:     MOV     #64.,R4         ;SET 256. WORD COUNTER
010410  012705  000004          5$:     MOV     #4,R5           ;SET 4 WORD COUNTER
010414  005767  172746          6$:     TST     PARPAT          ;BRANCH IF PARITY MEMORY PATTERN IS
010420  001402                          BEQ     60$             ;NOT TO BE CHECKED
010422  004767  177462                  JSR     PC,.XOR39
010426  012203                  60$:    MOV     (R2)+,R3        ;GET DATA
010430  020003                          CMP     R0,R3           ;CHECK DATA
010432  001403                          BEQ     .+10
010434  005046                          CLR     -(SP)
010436  004767  170632                  JSR     PC,ERROR
```

H06


```
010442 005100                        COM   R0              ;COMPLEMENT CHECK WORD
010444 005142                        COM   -(R2)           ;COMPLEMENT TEST DATA
010446 012203                        MOV   (R2)+,R3        ;GET DATA
010450 020003                        CMP   R0,R3           ;CHECK
010452 001403                        BEQ   .+10
010454 005046                        CLR   -(SP)           ;PUSH FAKE STATUS ON THE STACK
010456 004767 170512                 JSR   PC,ERROR
010462 005100                        COM   R0              ;COMPLEMENT CHECK WORD
010464 005162 177776                 COM   -2(R2)          ;RESTORE DATA
010470 005305                        DEC   R5              ;DECREMENT 4 WORD COUNTER
010472 001350                        BNE   6$
010474 005100                        COM   R0              ;COMPLEMENT CHECK WORD
010476 005304                        DEC   R4              ;DECREMENT 256. WORD COUNTER
010500 001343                        BNE   5$
010502 005100                        COM   R0              ;COMPLEMENT CHECK WORD
010504 005367 170246                 DEC   COUNT           ;DECREMENT BLOCK COUNTER
010510 001335                        BNE   4$

010512 016602 000004                 MOV   4(SP),R2        ;GET BANK #
010516 004767 176516                 JSR   PC,STMM2
010522 016603 000002                 MOV   2(SP),R3        ;GET BLOCK COUNT
010526 032737 040000 177570          BIT   #40000,@#SWR    ;LOOP ON TEST
010534 001307                        BNE   40$             ;BRANCH IF LOOP ON TEST
010536 006367 170202                 ASL   ICOUNT
010542 102220                        BVC   1$
010544 012705 000040          7$:    MOV   #32.,R5         ;COMPLEMENT PATTERN
010550 011200                 10$:   MOV   (R2),R0         ;GET FIRST DATA WORD
010552 016204 000010                 MOV   10(R2),R4       ;GET FIFTH DATA WORD
010556 110422                        MOVB  R4,(R2)+        ;SWAP WORDS 1-4
010560 110422                        MOVB  R4,(R2)+        ;WITH 5-8
```

```
        010562  110422                          MOVB    R4,(R2)+
        010564  110422                          MOVB    R4,(R2)+
        010566  110422                          MOVB    R4,(R2)+
        010570  110422                          MOVB    R4,(R2)+
        010572  110422                          MOVB    R4,(R2)+
        010574  110422                          MOVB    R4,(R2)+
        010576  110022                          MOVB    R0,(R2)+                ;AND VICE VERSA
        010600  110022                          MOVB    R0,(R2)+
        010602  110022                          MOVB    R0,(R2)+
        010604  110022                          MOVB    R0,(R2)+
        010606  110022                          MOVB    R0,(R2)+
        010610  110022                          MOVB    R0,(R2)+
        010612  110022                          MOVB    R0,(R2)+
        010614  110022                          MOVB    R0,(R2)+
        010616  005305                          DEC     R5
        010620  001353                          BNE     10$
        010622  005303                          DEC     R3
        010624  001347                          BNE     7$

        010626  005767  170112                  TST     ICOUNT
        010632  001402                          BEQ     11$
        010634  000167  177344                  JMP     1$
        010640  012616                  11$:    MOV     (SP)+,(SP)
        010642  012616                          MOV     (SP)+,(SP)
        010644  000207                          RTS     PC

                                        ;ROUTINE TO WRITE 8 XOR 13 WORST CASE NOISE TEST PATTERN
                                                .SBTTL  8 XOR 13 ROUTINES
                                        ;CALL:  MOV     BANK #,-(SP)
                                        ;       MOV     #4KBANKS,-(SP)
                                        ;       JSR     PC,.8X13

        010646  016604  000002          .8X13:  MOV     2(SP),R4                ;GET BANK COUNT
        010652  016602  000004                  MOV     4(SP),R2                ;GET FIRST BANK #
        010656  004767  176356                  JSR     PC,STMM2                ;GO SET MEM MGMT
        010662  005000                  1$:     CLR     R0
        010664  012705  000040          2$:     MOV     #32.,R5                 ;EACH LOOP WRITES 4K WORDS
        010670  005100                          COM     R0
        010672  012703  000200          3$:     MOV     #128.,R3                ;EACH SMALL LOOP WRITES 128 WORDS
        010676  005100                          COM     R0
        010700  010022                  4$:     MOV     R0,(R2)+                ;WRITE INTO MEMORY ADDRESSES
        010702  005303                          DEC     R3                      ;DECREMENT WORD COUNT
        010704  001375                          BNE     4$
        010706  005305                          DEC     R5                      ;DECREMENT 128. WORD COUNT
        010710  001370                          BNE     3$
        010712  005304                          DEC     R4                      ;DECREMENT 4K BANK COUNT
        010714  001363                          BNE     2$                      ;LOOP UNTIL DONE
        010716  012616                          MOV     (SP)+,(SP)              ;ADJUST STACK
        010720  012616                          MOV     (SP)+,(SP)
        010722  000207                          RTS     PC

                                        ;ROUTINE TO CHECK 8 XOR 13 WORST CASE NOISE TEST PATTERN
                                        ;CALL:
                                        ;       MOV     BANK#,-(SP)             ;PUSH FIRST BANK # ON THE STACK
                                        ;       MOV     #BANKS,-(SP)            ;PUSH # OF 4K BANKS TO CHECK ON THE STACK
                                        ;       JSR     PC,..8X13               ;CALL ROUTINE
```

```
010724  000240                    ..9X13: NOP
010726  004767  000716                    JSR     PC,CKSWR        ;GO CHECK SWITCH REGISTER
010732  012700  177777                    MOV     #-1,R0          ;SET TEST DATA WORD
010736  016602  000004            10$:    MOV     4(SP),R2        ;GET BANK #
010742  004767  175272                    JSR     PC,STMM2        ;GO SET MEM MGMT IF REQUIRED
010746  016667  000002  170002            MOV     2(SP),COUNT     ;GET # OF 4K BANKS TO CHECK

010754  012704  000040            1$:     MOV     #32.,R4         ;SET 4K WORD COUNTER
010760  005100                    2$:     COM     R0              ;COMPLEMENT TEST WORD
010762  012705  000100                    MOV     #64.,R5         ;SET 128 WORD COUNTER

010766                            3$:
010766  012203                            MOV     (R2)+,R3        ;GET TEST DATA
010770  020003                            CMP     R0,R3           ;COMPARE WITH CHECK WORD
010772  001403                            BEQ     .+10
010774  005046                            CLR     -(SP)           ;PUSH FAKE STATUS ON THE STACK
010776  004767  170172                    JSR     PC,ERROR        ;ERROR! MEM DATA (R3) NOT = TEST DATA
                                                                  ;(R0), ADDRESS=(R2)-2

011002  012203                            MOV     (R2)+,R3        ;GET TEST DATA
011004  020003                            CMP     R0,R3           ;COMPARE WITH CHECK WORD
011006  001403                            BEQ     .+10
011010  005046                            CLR     -(SP)           ;PUSH FAKE STATUS ON THE STACK
011012  004767  170156                    JSR     PC,ERROR        ;ERROR! MEM DATA (R3) NOT = TEST DATA
                                                                  ;(R0), ADDRESS=(R2)-2

011016  005305                            DEC     R5              ;DECREMENT 128 WORD COUNTER
011020  001362                            BNE     3$
011022  005304                            DEC     R4              ;DECREMENT 4096. WORD COUNTER
011024  001355                            BNE     2$
011026  005100                            COM     R0
011030  005367  167722                    DEC     COUNT           ;ALL 4K BANKS CHECKED?
011034  001347                            BNE     1$

011036  016602  000004                    MOV     4(SP),R2        ;GET FIRST BANK ADDRESS
011042  004767  176172                    JSR     PC,STMM2        ;GO SET UP MEM MGMT IF REQUIRED
011046  016604  000002                    MOV     2(SP),R4        ;GET # OF 4K BANKS
011052  006367  167556                    ASL     ICOUNT          ;SHIFT ITERATION PATTERN
011056  001401                            BEQ     .+4
011060  102326                            BVC     10$
011062  012705  004000            40$:    MOV     #2048.,R5       ;SET 4096. WORD COUNTER
011066  005122                    4$:     COM     (R2)+           ;COMPLEMENT TEST PATTERN
011070  005122                            COM     (R2)+
011072  005305                            DEC     R5
011074  001374                            BNE     4$
011076  005304                            DEC     R4
011100  001370                            BNE     40$
011102  005100                            COM     R0              ;COMPLEMENT TEST WORD
011104  005767  167634                    TST     ICOUNT
011110  001312                            BNE     10$
011112  012616                            MOV     (SP)+,(SP)
011114  012616                            MOV     (SP)+,(SP)
011116  000207                            RTS     PC              ;RETURN

                                          .SBTTL  ROTATING 1'S & 0'S ROUTINES
```

```
                                        ;ROUTINE TO CHECK ROTATING '0' BIT THROUGH FILD OF 1'S
                                        ;CALL:  MOV     BANK#,-(SP)     ;SET STARTING BANK #
                                        ;       MOV     BLKCNT,-(SP)    ;SET 256. WORD BLOCK COUNT
                                        ;       JSR     PC,.ROT0        ;CALL ROUITNE
                                        ;
011120  004767  000524          .ROT0:  JSR     PC,CKSWR        ;GO CHECK SWITCHES
011124  016604  000002                  MOV     2(SP),R4        ;GET 256. WORD BLOCK COUNT
011130  016602  000004                  MOV     4(SP),R2        ;GET FIRST BANK #
011134  004767  176100                  JSR     PC,STMM2        ;GO SET UP MEM MGMT (IF AVAIL
011140  012700  177777                  MOV     #-1,R0          ;SET CHECK WORD
                                        ;
011144  012705  000400          1$:     MOV     #256.,R5        ;SET 256. WORD COUNT
011150  000241                  2$:     CLC                     ;CLEAR CARRY BIT IN PSW
011152  004767  000124                  JSR     PC,ROTATE
011156  016203  177776                  MOV     -2(R2),R3       ;GET RESULT
011162  103402                          BCS     3$              ;BRANCH IF 'C' BIT WAS SET
011164  020003                          CMP     R0,R3           ;CHECK RESULT
011166  001403                          BEQ     4$
011170  005046                  3$:     CLR     -(SP)           ;ERROR! COULD NOT ROTATE '0' BIT
011172  004767  167776                  JSR     PC,ERROR        ;THROUGH ADDRESS IN R2
011176  005305                  4$:     DEC     R5              ;DECREMENT 256. WORD COUNT
011200  001363                          BNE     2$              ;LOOP UNTIL DONE
011202  005304                          DEC     R4              ;DECREMENT 256. WORD BLOCK COUNT
011204  001357                          BNE     1$              ;LOOP UNTIL DONE
011206  012616                          MOV     (SP)+,(SP)      ;POP CONSTANTS OFF THE STACK
011210  012616                          MOV     (SP)+,(SP)
011212  000207                          RTS     PC              ;RETURN TO CALLER
                                        ;
                                        ;ROUTINE TO CHECK ROTATING '1' BIT THROUGH A FIELD OF 0'S
                                        ;CALL:  MOV     BANK#,-(SP)     ;SET STARTING BANK #
                                        ;       MOV     BLKCNT,-(SP)    ;SET # OF 256. WORD BLOCKS TO CHEC
                                        ;       JSR     PC,.ROT1        ;CALL ROUTINE
                                        ;
011214  004767  000430          .ROT1:  JSR     PC,CKSWR        ;GO CHECK SWITCHES
011220  016604  000002                  MOV     2(SP),R4        ;GET # OF 256. WORD BLOCKS TO CHEC
011224  016602  000004                  MOV     4(SP),R2        ;GET STARTING BANK #
011230  004767  176004                  JSR     PC,STMM2        ;GO SET UP MEM MGMT (IF AVAIL
011234  005000                          CLR     R0              ;SET CHECK WORD
                                        ;
011236  012705  000400          1$:     MOV     #256.,R5        ;SET 256. WORD COUNTER
011242  000261                  2$:     SEC                     ;SET 'C' BIT IN PSW
011244  004767  000032                  JSR     PC,ROTATE       ;GO ROTATE '1' BIT
011250  016203  177776                  MOV     -2(R2),R3       ;GET RESULT
011254  103002                          BCC     3$              ;BRANCH IF 'C' IS CLEAR
011256  020003                          CMP     R0,R3           ;CHECK RESULT
011260  001401                          BEQ     .+4
011262  104400                  3$:     HLT                     ;ERROR! COULD NOT ROTATE '1' BIT
                                                                ;THROUGH ADDRESS IN R2
011264  005305                          DEC     R5              ;DECREMENT 256. WORD COUNT
011266  001365                          BNE     2$
011270  005304                          DEC     R4              ;DECREMENT 256. WORD BLOCK COUNT
011272  001361                          BNE     1$
011274  012616                          MOV     (SP)+,(SP)      ;ADJUST RETURN ADDRESS
011276  012616                          MOV     (SP)+,(SP)
011300  000207                          RTS     PC              ;RETURN TO CALLER
```

```
                                ;ROUTINE TO ROTATE 'C' BIT THROUGH A MEMORY LOCATION.
011302  106112                  ROTATE: ROLB    (R2)            ;(R2)=177776 OR 000001
011304  106112                          ROLB    (R2)            ;(R2)=177775 OR 000002
011306  106112                          ROLB    (R2)            ;(R2)=177773 OR 000004
011310  106112                          ROLB    (R2)            ;(R2)=177767 OR 000010
011312  106112                          ROLB    (R2)            ;(R2)=177757 OR 000020
011314  106112                          ROLB    (R2)            ;(R2)=177737 OR 000040
011316  106112                          ROLB    (R2)            ;(R2)=177677 OR 000100
011320  106112                          ROLB    (R2)            ;(R2)=177777 OR 000000
011322  106122                          ROLB    (R2)+           ;(R2)=177577 OR 000200
011324  106112                          ROLB    (R2)            ;(R2)=177377 OR 000400
011326  106112                          ROLB    (R2)            ;(R2)=176777 OR 001000
011330  106112                          ROLB    (R2)            ;(R2)=175777 OR 002000
011332  106112                          ROLB    (R2)            ;(R2)=173777 OR 004000
011334  106112                          ROLB    (R2)            ;(R2)=167777 OR 010000
011336  106112                          ROLB    (R2)            ;(R2)=157777 OR 020000
011340  106112                          ROLB    (R2)            ;(R2)=137777 OR 040000
011342  106112                          ROLB    (R2)            ;(R2)=077777 OR 100000
011344  106122                          ROLB    (R2)+           ;(R2)=177777 OR 000000
011346  000207                          RTS     PC              ;RETURN

                                ;ROUTINE TO WRITE USER SLECTED PATTERN INTO MEMORY
                                ;CALL:  MOV     BANK#,-(SP)     ;PUSH STARTING BANK # ONTO STACK
                                ;       MOV     BLKCNT,-(SP)    ;AND 128. WORD BLOCK COUNT
                                ;       JSR     PC,..USER       ;CALL ROUTINE

011350  016604  000002          .USER:  MOV     2(SP),R4        ;GET BLOCK COUNT
011354  016602  000004                  MOV     4(SP),R2        ;GET STARTING BANK #
011360  004767  175654                  JSR     PC,STMM2        ;GO SET UP MEM MGMT
011364  016700  174012                  MOV     .CONST,R0       ;GET USER CONSTANT
011370  012703  000100          1$:     MOV     #64.,R3         ;SET 256. WORD COUNTER
011374  010022                  2$:     MOV     R0,(R2)+        ;WRITE 256. WORDS
011376  010022                          MOV     R0,(R2)+
011400  010022                          MOV     R0,(R2)+
011402  010022                          MOV     R0,(R2)+
011404  005303                          DEC     R3              ;DECREMENT 256. WORD COUNTER
011406  001372                          BNE     2$              ;LOOP UNTIL 256. WORDS HAVE BEEN WRITTEN
011410  005304                          DEC     R4              ;DECREMENT BLOCK COUNT
011412  001366                          BNE     1$
011414  012616                          MOV     (SP)+,(SP)      ;ADJUST STACK
011416  012616                          MOV     (SP)+,(SP)
011420  000207                          RTS     PC

                                        .SBTTL  USER PATTERN ROUTINE
                                ;ROUTINE TO CHECK USER SELECTED PATTERN
                                ;CALL:  MOV     BANK#,-(SP)     ;PUSH STARTING BANK # ONTO STACK
                                ;       MOV     BLKCNT,-(SP)    ;AND 256. WORD BLOCK COUNT
                                ;       JSR     PC,...USER      ;CALL ROUTINE

011422  004767  000222          ..USER: JSR     PC,CKSWR        ;GO CHECK SWITCH REGISTER
011426  016700  173750                  MOV     .CONST,R0       ;GET USER CONSTANT
011432  016604  000002          1$:     MOV     2(SP),R4        ;GET # OF 256. WORD BLOCKS
011436  016602  000004                  MOV     4(SP),R2        ;GET STARTING BANK #
011442  004767  175572                  JSR     PC,STMM2        ;GO SET UP MEM MGMT IF REQUIRED

011446  012705  000100          2$:     MOV     #64.,R5         ;SET WORD COUNT
```

```
3320  011452                          3$:
3321  011452  012203                          MOV     (R2)+,R3        ;GET TEST DATA
3322  011454  020003                          CMP     R0,R3           ;COMPARE WITH CHECK WORD
3323  011456  001403                          BEQ     .+10
3324  011460  005046                          CLR     -(SP)           ;PUSH FAKE STATUS ON THE STACK
3325  011462  004767  167536                  JSR     PC,ERROR        ;ERROR! MEM DATA (R3) NOT = TEST DATA
3326                                                                  ;(R0), ADDRESS=(R2)-2
3327
3328  011466  012203                          MOV     (R2)+,R3        ;GET TEST DATA
3329  011470  020003                          CMP     R0,R3           ;COMPARE WITH CHECK WORD
3330  011472  001403                          BEQ     .+10
3331  011474  005046                          CLR     -(SP)           ;PUSH FAKE STATUS ON THE STACK
3332  011476  004767  167472                  JSR     PC,ERROR        ;ERROR! MEM DATA (R3) NOT = TEST DATA
3333                                                                  ;(R0), ADDRESS=(R2)-2
3334
3335  011502  012203                          MOV     (R2)+,R3        ;GET TEST DATA
3336  011504  020003                          CMP     R0,R3           ;COMPARE WITH CHECK WORD
3337  011506  001403                          BEQ     .+10
3338  011510  005046                          CLR     -(SP)           ;PUSH FAKE STATUS ON THE STACK
3339  011512  004767  167456                  JSR     PC,ERROR        ;ERROR! MEM DATA (R3) NOT = TEST DATA
3340                                                                  ;(R0), ADDRESS=(R2)-2
3341
3342  011516  012203                          MOV     (R2)+,R3        ;GET TEST DATA
3343  011520  020003                          CMP     R0,R3           ;COMPARE WITH CHECK WORD
3344  011522  001403                          BEQ     .+10
3345  011524  005046                          CLR     -(SP)           ;PUSH FAKE STATUS ON THE STACK
3346  011526  004767  167442                  JSR     PC,ERROR        ;ERROR! MEM DATA (R3) NOT = TEST DATA
3347                                                                  ;(R0), ADDRESS=(R2)-2
3348
3349  011532  005305                          DEC     R5              ;DECREMENT WORD COUNT
3350  011534  001346                          BNE     3$
3351  011536  005304                          DEC     R4              ;DECREMENT BLOCK COUNT
3352  011540  001342                          BNE     2$
3353
3354  011542  032737  040000  177570          BIT     #40000,@#SWR    ;CHECK LOOP SWITCH
3355  011550  001330                          BNE     1$              ;LOOP CHECKING THIS PATTERN
3356  011552  006367  167186                  ASL     ICOUNT          ;SHIFT PATTERN INDICATOR
3357  011556  001325                          BNE     1$
3358  011560  012616                          MOV     (SP)+,(SP)      ;ADJUST STACK
3359  011562  012616                          MOV     (SP)+,(SP)
3360  011564  000207                          RTS     PC              ;RETURN TO CALLER
3361
3362                                          .SBTTL  GET TTY INPUT ROUTINE
3363                                  ;ROUTINE TO GET ASCII INPUT FROM TTY,AND CONVERT TO OCTAL.
3364                                  ;ROUTINE LEAVES THE FIRST 16 BITS IN ADDRESS FOLLOWINTG THE CALL
3365                                  ;AND THE LAST 2 BITS IN .1617 BELOW.
3366                                  ;CALL:  JSR  PC,RECD
3367  011566                          RECD:
3368  011566  004767  166512                  JSR     PC,$SAVR        ;GO SAVE REGISTERS ON THE STACK
3369  011572  004767  002172                  JSR     PC,INUM
3370  011576  010267  000042                  MOV     R2,TEMP2
3371  011602  010367  000040                  MOV     R3,TEMP3
3372  011606  004767  166516                  JSR     PC,$RESTR       ;RESTORE REGISTERS FROM STACK
3373  011612  011667  000024                  MOV     (SP),TEMP1
3374  011616  016777  000024  000016          MOV     TEMP3,@TEMP1
3375  011624  016767  000014  000006          MOV     TEMP2,.1617
```

```
 3376  011632  062716  000002                 ADD     #2,(SP)
 3377  011636  000207                         RTS     PC
 3378  011640  000000                 .1617:  .WORD   0
 3379  011642  000000                 TEMP1:  .WORD   0
 3380  011644  000000                 TEMP2:  .WORD   0
 3381  011646  000000                 TEMP3:  .WORD   0
 3382
 3383                                 ;ROUTINE TO CHECK THE SWITCH REGISTER
 3384                                 ;CHECK SWITCH 9: IF SET, LOAD ERROR COUNT INTO THE DISPLAY REGISTER;
 3385                                 ;IF NOT SET, LOAD PASS COUNT INTO THE DISPLAY REGISTER
 3386  011650  042767  017777  167072 CKSWR:  BIC     #17777,LDDISP   ;SAVE RELOCATION BITS
 3387  011656  032737  000400  177570         BIT     #BIT8,@#SWR     ;CHECK SWITCH 8
 3388  011664  001402                         BEQ     10$             ;BRANCH IF SET
 3389  011666  004767  000464                 JSR     PC,REL24K       ;GO RELOCATE PROGRAM BACK TO 4K AND STOP
 3390  011672  032737  001000  177570 10$:    BIT     #BIT9,@#SWR     ;SWITCH 9 SET ?
 3391  011700  001404                         BEQ     1$
 3392  011702  056767  167040  167040         BIS     ERCNT,LDDISP    ;LOAD ERROR COUNT
 3393  011710  000403                         BR      2$
 3394  011712  056767  167024  167030 1$:     BIS     ICNT,LDDISP     ;LOAD PASS COUNT
 3395  011720  016737  167024  177570 2$:     MOV     LDDISP,@#DISPLAY;LOAD THE DISPLAY REGISTER
 3396  011726  012767  040177  167010         MOV     #040177,ICOUNT  ;LOAD ITERATION COUNT WORD
 3397  011734  032737  004000  177570         BIT     #4000,@#SWR     ;CHECK SW11
 3398  011742  001402                         BEQ     .+6
 3399  011744  105067  166774                 CLRB    ICOUNT          ;ICOUNT =040000 IF SW11 =1
 3400  011750  000207                         RTS     PC
 3401
 3402                                 ;MESSAGES
 3403  011752  005015  047524  051040 RESLDR: .ASCIZ  <15><12>'TO RESTORE LOADERS START AT 162'<15><12>
 3404  011760  051505  047524  042522
 3405  011766  046040  040517  042504
 3406  011774  051522  051440  040524
 3407  012002  052122  040440  020124
 3408  012010  033061  006462  000012
 3409  012016  005015  047105  041101 PARITY: .ASCIZ  <15><12> 'ENABLE PARITY? 1/0=YES/NO '
 3410  012024  042514  050040  051101
 3411  012032  052111  037531  030440
 3412  012040  030057  054475  051505
 3413  012046  047057  020117     000
 3414  012053     015  051412  040524 STBANK: .ASCIZ  <15><12>'STARTING BANK #(8)? '
 3415  012060  052122  047111  020107
 3416  012066  040502  045516  021440
 3417  012074  034050  037451  000040
 3418  012102  005015  020043  043117 BANKS:  .ASCIZ  <15><12>'# OF 4K BANKS TO TEST(8)? '
 3419  012110  032040  020113  040502
 3420  012116  045516  020123  047524
 3421  012124  052040  051505  024124
 3422  012132  024470  020077     000
 3423  012137     015  050012  052101 PAT:    .ASCIZ  <15><12>'PATTERN #? '
 3424  012144  042524  047122  021440
 3425  012152  020077     000
 3426  012155     015  037412     000 QUEST:  .ASCIZ  <15><12>'?'
 3427  012161     015  052012  050131 CONST:  .ASCIZ  <15><12>'TYPE CONSTANT'
 3428  012166  020105  047503  051516
 3429  012174  040524  052116     000
 3430  012201     015  044412  050116 PRG3M:  .ASCIZ  <15><12>'INPUT # OF 256. WORD BLOCKS TO TEST INSTEAD OF'
 3431  012206  052125  021440  047440
```

B07

```
FROM:   .ASCII   15  12 "TAPE ADDRESS"

RELOCM: .ASCII   15  12 "TO RESTORE PROGRAM START AT "

ASTERISK: .ASCII '*'
ENDMSG: .ASCII  'DZQMB DONE!'

        .EVEN

;ROUTINE TO RELOCATE PROGRAM BACK TO 0
REL24K: MOV     PC,R0                   ;FORM BASE ADDRESS WHERE CODE
        BIC     #177777,R0              ;IS RELOCATED
        MOV     R0,1$                   ;PUT FROM ADDRESS INTO SUBROUTINE CALL
        JSR     R5,RELOC                ;RELOCATE CODE TO
1$:     0                               ;LOWEST 4K
        MOV     #STKPTR,SP              ;SET STACK PTR
        BIC     #100000,@#LODISP        ;CLEAR RELOCATION INDICATOR
        MOV     @#LODISP,@#DISPLAY      ;LOAD DISPLAY REGISTER
        CLR     @#RELOCF                ;CLEAR RELOCATION FACTOR
        RESET                           ;DISABLE MEM MGMT
        CLR     @#176                   ;PUT A HALT AT 176
        JMP     @#162                   ;RESTORE LOADERS & HALT AT 176
;
        .SBTTL  BRANCH GOBBLE MOS MEMORY TEST
;
        *****PROGRAM DESCRIPTION*****
;
;THIS IS A PSEUDO-MODIFIED VERSION OF THE BRANCH GOBBLE
;MOS MEMORY EXERCISER. PSEUDO-MODIFIED BECAUSE THE
;ORIGINAL CODE, TAKEN FROM THE DZQKA-A INSTRUCTION
;EXERCISER WHICH IS BRANCH GOBBLE IS INCLUDED
;HERE IN ITS ORIGINAL FORM, BUT MEMORY MANAGEMENT
;CABABILITIES HAVE BEEN ADDED TO GIVE IT OPERATING
;ABLITIES IN A 0-128K MEMORY ENVIRONMENT.
;
;
        *****OPERATING PROCEDURE*****
;
;WHEN LOADED THIS PROGRAM'S STARTING ADDRESS IS XXXXXX.
;WHEN RUNNING THE FOLLOWING STEPS ARE TAKEN:
;1.0    A PROGRAM ID IS TYPED ON THE TTY:
                BRANCH GOBBLE MOS TEST
;1.5    THE PROGRAM DETERMINES IF THERE IS MOS PARITY, IF YES IT IS ENABLED
```

C07


```
;                       AND THE USER IS TOLD.
;2.                     THE USER IS ASKED:
;                               USE MEMORY MANAGEMENT?(Y OR N)
;                               >
;                       IF THE USER TYPES Y, THE MEMORY MANAGEMENT
;                       WILL BE USED TO RUN THE TEST. IF HE TYPES
;                       N THEN MEMORY MANAGEMENT WILL NOT BE USED
;                       IN EITHER CASE THE ACTUAL TEST WILL BE
;                       PERFORMED BY THE UNMODIFIED ORIGINAL VERSION
;                       OF BRANCH GOBBLE.
;                       NOTE THAT WHEN THE TEST OF MEMORY
;                       LOCATED IN UNIBUS ADDRESSES HIGHER THAN
;                       177777 IS DESIRED, THE ENABLING OF MEMORY
;                       MANAGEMENT IN THIS PROGRAM IS MANDATORY.
;                       NOTE ALSO THAT WHEN THE TEST IS TO BE
;                       IN THE 0 TO 177777 RANGE OF UNIBUS ADDRESSES
;                       IT IS RECOMMENDED THAT MEMORY MANAGEMENT
;                       BE DISABLED.
;3.                     THE PROGRAM WILL THEN ASK THE USER
;                       FOR THE HIGH ADDRESS LIMIT FOR THE TEST:
;                               WHAT IS THE HIGH LIMIT:
;                               >
;                       AND THEN FOR THE LOW LIMIT:
;                               WHAT IS THE LOW LIMIT:
;                               >
;                       BOTH OF THE ADDRESSES SHOULD BE SPECIFIED IN
;                       OCTAL USING THE FORM XXXX00, THAT IS THE
;                       ADDRESSES MUST BE THE BEGINNING OF 100 BYTE
;                       BLOCKS OF MEMORY. IF THEY ARE NOT THEN THEY
;                       WILL BE TRUNCATED!
;                       THE ADDRESSES WILL BE INTERPRETED AS FULL 18-BIT
;                       UNIBUS ADDRESSES.
;                       VALID ADDRESSES MEET THE FOLLOWING CONDITIONS:
;                               1. THE SPECIFIED SPAN OF THE TEST
;                               SHOULD NOT ENCOMPASS THE ACTUAL
;                               MEMORY LOCATIONS OCCUPIED BY THIS
;                               PROGRAM. THIS PROGRAM IS RELOCATABLE
;                               AT LOADING TIME, SO THAT NO LIMITATIONS
;                               ARE THUS IMPOSED.
;                               2. THE HIGH LIMIT SHOULD BE GREATER
;                               THAN THE LOW LIMIT.
;                               3. THE NUMBER MUST BE SPECIFIED BY
;                               OCTAL DIGITS.
;                               4. IF THE USER HAS DISABLED MEMORY
;                               MANAGEMENT THE ADDRESSES SHOULD BE
;                               IN LOW MEMORY.I.E. LESS THAN 177777.
;                       IF ANY OF THESE CONDITIONS IS NOT MET
;                       THE USER WILL BE ASKED TO INPUT ANOTHER
;                       ADDRESS.
;                       NOTE THAT IF ANY OF THE ADDRESSES IN THE TEST
;                       SPAN IS A NON-EXISTANT MEMORY LOCATION
;                       A TIME OUT ERROR WILL OCCUR FROM WHICH
;                       NO RECOVERY CAN BE MADE EXCEPT TO RESTART
;                       AT THE STARTING ADDRESS.
;4.                     WHEN THE ABOVE INFORMATION HAS BEEN
;                       SUCCESSFULLY GATHERED FROM THE USER, THE
```

D07


```
                                        ;               TEST WILL BEGIN. IF BRANCH GOBBLE ENCOUNTERS
                                        ;               A MEMORY FAULT DURING THE TEST THE
                                        ;               PROCESSOR WILL BE HALTED AT THE "LOCATION"
                                        ;               OF THE MEMORY FAULT.
                                        ;5.             IF THE TEST IS COMPLETE WITHOUT AN
                                        ;               ENCOUNTER WITH A MEMORY FAULT ANY WHERE
                                        ;               IN THE TESTED SPAN, THEN THE TEST
                                        ;               WILL BE REPEATED BY RETURNING TO STEP 4.
                                        ;6.             IF THE USER WISHES TO STOP THE TEST IN PROGRESS AND START
                                        ;               ANOTHER WITH A DIFFERENT RANGE THEN HE SHOULD HIT THE HALT
                                        ;               SWITCH AND START THE TEST AGAIN AT 200.
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
 012436                                 TOP:
                                        ;CONSTANTS:
         077406                         KPDR=077406
                                        ;THIS ROUTINE TAKES CARE OF THE IDENTIFICATION.
                                        ;ASK THE USER IF MEMORY MANAGEMENT SHOULD BE
                                        ;ENABLED, AND IF NOT DOES THE SET UP FOR THE
                                        ;ACTUAL TEST. IF MEMORY MANAGEMENT IS
                                        ;DESIRED THE ROUTINE YMMBGO IS GIVEN CONTROL.
                                        ;
 012436  000000                         LODFLG: .WORD   0
                                        ;
 012440  005067  165454                 BRANCH: CLR     RELFL           ;INITIALIZE THE RELOCATION INDICATOR.
 012444  005067  166302                         CLR     MMAVA           ;INITIALIZE THE MEM. MANAGEMENT FLAG.
 012450  004567  166304                         JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
 012454  015206                                 .WORD   IDMESS
 012456  004767  167352                         JSR     PC,$LDR
 012462  004767  001566                 15$:    JSR     PC,MOSPAR
 012466                                 1$:
 012466  004567  166256                         JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
 012472  015141                                 .WORD   MMMESS          ;MEMORY MANAGEMENT SHOULD
                                                                        ;BE ENABLED
 012474  105737  177560                 2$:     TSTB    @#TKS           ;WAIT FOR A CHARACTER.
 012500  100375                                 BPL     2$
 012502  113746  177562                         MOVB    @#TKB,-(SP)     ;GET THE CHARACTER.
 012506  042716  177600                         BIC     #177600,(SP)
 012512  004767  001520                         JSR     PC,TYPIT        ;ECHO THE CHARACTER.
 012516  005726                                 TST     (SP)+           ;RESET THE STACK.
 012520  022766  000131  177776                 CMP     #'Y,-2(SP)      ;IF IT IS Y, THEN GO
 012526  001513                                 BEQ     YMMBGO          ;TO YMMBGO TO ENABLE
                                                                        ;MEMORY MANAGEMENT
 012530  022766  000116  177776                 CMP     #'N,-2(SP)      ;IS IT N?
 012536  001406                                 BEQ     3$
 012540  012746  000077                         MOV     #'?,-(SP)       ;IF IT WAS NIETHER
```

E07


```
012544  004767  001466                  JSR     PC,TYPIT        ;OR N THEN ASK
012550  005736                          TST     (SP)+           ;THE USER AGAIN.
012552  000745                          BR      1$
012554  004767  001266          3$:     JSR     PC,LIMITS       ;GO GET THE LIMITING
                                                                ;ADDRESSES FOR THE TEST.
                                                                ;THEY WILL BE LEFT AS
                                                                ;BLOCK NUMBERS IN LOLIM
                                                                ;AND HILIM.
012560  022767  001600  001172  SPOT:   CMP     #1600,HILIM
012566  002010                          BGE     SPOT2
012570  012700  177770                  MOV     #SPOT-.,R0
012574  060700                          ADD     PC,R0
012576  062700  177772                  ADD     #-6,R0
012602  010046                          MOV     R0,-(SP)
012604  000167  001140                  JMP     LIMERR
012610  005046                  SPOT2:  CLR     -(SP)
012612  012746  002502                  MOV     #FIRST1-.,-(SP)
012616  060716                          ADD     PC,(SP)
012620  062716  177772                  ADD     #-6,(SP)
012624  004767  000774                  JSR     PC,LOSEG4
012630  004767  000436                  JSR     PC,LOSEG1
012634  012746  000052          REPET1: MOV     #'*,-(SP)
012640  004767  001372                  JSR     PC,TYPIT
012644  005726                          TST     (SP)+
012646  022767  000200  001104          CMP     #200,HILIM
012654  002026                          BGE     REPET3
012656  016700  001076                  MOV     HILIM,R0
012662  004767  000054          4$:     JSR     PC,ROT
012666  010067  002556                  MOV     R0,HI           ;SET THE PARAMETERS
                                                                ;IN THE ACTUAL TEST
                                                                ;ROUTINE.
012672  016700  001064                  MOV     LOLIM,R0        ;DO THE SAME FOR THE
                                                                ;LOW ADDRESS LIMIT.
012676  004767  000040                  JSR     PC,ROT
012702  010067  002544                  MOV     R0,LO
012706  005037  000036                  CLR     @#36            ;SET UP THE VECTORS
012712  012704  000020                  MOV     #REPET3-.,R4    ;FOR AN INTERUPT
012716  060704                          ADD     PC,R4           ;FROM A TRAP INSTRUCTION
012720  062704  177772                  ADD     #-6,R4          ;WHICH WILL BE USED TO
012724  010437  000034                  MOV     R4,@#34         ;RETURN FROM THE TEST
                                                                ;ROUTINE WHEN IT IS DONE.
012730  000002                          RTI                     ;START THE TEST.
012732  004767  000404          REPET3: JSR     PC,LOSEG2
012736  000167  177672                  JMP     REPET1
                                ;
                                ;THIS ROUTINE IS CALLED TO SHIFT R0 TO THE LEFT SIX BITS.
012742  012701  177772          ROT:    MOV     #-6,R1
012746  006300                  1$:     ASL     R0
012750  005201                          INC     R1
012752  002775                          BLT     1$
012754  000207                          RTS     PC
                                ;
                                ;YMMBGO SETS UP FOR USING MEMORY MANAGEMENT TO
                                ;DO THE MOS TEST.
                                ;ALL THE MEMORY MANAGEMENT REGISTERS WHICH ARE
                                ;TO REMAIN STATIC FOR THE TESTS DURATION ARE SET.
```

F07


```
                                        ;THE LIMITS ARE THEN GOTTEN AND CHECKED FOR VALIDITY.
                                        ;THE TRAP INTERRUPT VECTORS ARE SET AND YMMBG1
                                        ;IS CALLED. YMMBG1 IS A ROUTINE THE SETS THOSE
                                        ;MEMORY MANAGEMENT REGISTERS WHICH NEED TO
                                        ;BE CHANGED DYNAMICALLY DURING THE TEST.
                                        ;
012756  012767  177777  165766  YMMBG0: MOV     #-1,MMAVA
012764  005037  177572                  CLR     @#SR0           ;SET ALL THE STATIC
012770  012700  172340                  MOV     #KIPAR0,R0      ;REGISTERS.
012774  012701  172300                  MOV     #KIPDR0,R1
013000  005003                          CLR     R3
013002  012704  177770                  MOV     #-10,R4
013006  010320                  1$:     MOV     R3,(R0)+
013010  062703  000200                  ADD     #200,R3
013014  012721  077406                  MOV     #KPDR,(R1)+
013020  005204                          INC     R4
013022  001371                          BNE     1$
013024  004767  000616                  JSR     PC,LIMITS       ;GET THE LIMITS FOR THE PENDING
                                                                ;TEST. THEY ARE LEFT IN BLOCK NUMBER
                                                                ;FORM IN HILIM AND LOLIM.
013030  004767  000570                  JSR     PC,LOSEG4       ;SEE IF PERMANENT RELOCATION IS
                                                                ;APPROPRIATE FOR THIS TEST SPAN.
013034  012737  007600  172356          MOV     #7600,@#KIPAR7  ;MAP THE UNIBUS DEVICE PAGE INTO
                                                                ;INTO HIGH VIRTUAL MEMORY.
013042  005046                  REPET2: CLR     -(SP)
013044  012746  002250                  MOV     #FIRST1-.,-(SP)
013050  060716                          ADD     PC,(SP)
013052  062716  177772                  ADD     #-6,(SP)        ;SET UP THE STACK
                                                                ;TO SIMULATE THE OCCURRENCE OF AN
                                                                ;AN INTERRUPT SO THAT THE TEST CAN
                                                                ;BE STARTED USING AN RTI.
013056  016767  000754  000674          MOV     HISAV,HILIM
013064  016767  000674  000670          MOV     LOSAV,LOLIM
013072  004767  000174                  JSR     PC,LOSEG1
013076  012746  000052                  MOV     #'*,-(SP)
013102  004767  001130                  JSR     PC,TYPIT
013106  005726                          TST     (SP)+
013110  012737  013122  000034          MOV     #YMMBG1,@#34
013116  005037  000036                  CLR     @#36            ;SET UP THE TRAP INTERRUPT VECTOR
                                                                ;WHICH WILL BE USED TO RETURN FROM THE
                                                                ;TESTING ROUTINE.
                                        ;
                                        ;YMMBG1 IS USED TO DYNAMICALLY ALLOCATE MEMORY UNDER MEMORY MANAGEMENT
                                        ;WHILE A TEST IS IN PROGRESS. WORKING UPWARDS FROM THE LOLIM
                                        ;MEMORY MANAGEMENT IR SET TO ENABLE BRGOB TO WORK THROUGH AS
                                        ;MUCH OF THE TEST SPAN AS POSSIBLE IN A SINGLE MANAGEMENT SET UP
                                        ;BEFORE HAVING TO RESET THE MANAGEMENT REGISTERS.
                                        ;BRGOB ALWAYS IS IN LOW VIRTUAL MEMORY AND UPPER VIRTUAL
                                        ;ADDRESSES ARE ALWAYS MAPPED INTO UNIBUS DEVICE ADDRESSES.
                                        ;
013122  005037  177572          YMMBG1: CLR     @#SR0
013126  026767  000626  000626          CMP     HILIM,LOLIM     ;IS THE TEST DONE?
013134  101451                          BLOS    DONIT           ;YES, THEN BRANCH.
013136  012700  172342                  MOV     #KIPAR1,R0      ;ELSE GET READY TO SET
013142  016705  000614                  MOV     LOLIM,R5        ;THE KERNAL PAGE ADDRESS REGISTERS.
013146  012701  177772                  MOV     #-6,R1
```

```
013152  012767  020000  002272          MOV     #20000,LO
013160  010520                  1$:     MOV     R5,(R0)+        ;RESET THE KIPAR'S
013162  062705  000200          2$:     ADD     #200,R5
013166  026705  000566          25$:    CMP     HILIM,R5        ;REACHED HILIM?
013172  101407                          BLOS    3$              ;YES, GOTO 3$.
013174  005201                          INC     R1              ;NO, INCREMENT R1 AND SEE IF ALL THE
                                                                ;KIPAR'S HAVE BEEN SET.
013176  002770                          BLT     1$              ;ALL THE KIPAR'S HAVE NOT BEEN SET SO
                                                                ;LOOP TO GET THE NEXT ONE.
013200  010567  002244                  MOV     R5,HI           ;DO THIS IS ALL THE TEST SPAN HAS NOT
                                                                ;BEEN ALLOCATED TO SOME VIRTUAL ADDRESSES
                                                                ;IS THE KERNAL INSTRUCTION SPACE.
013204  162705  000002                  SUB     #2,R5
013210  000403                          BR      4$
013212  016767  000542  002230  3$:     MOV     HILIM,HI        ;DO THIS IF ALL THE TEST SPAN HAS BEEN
                                                                ;ALLOCATED TO THE VIRTUAL KERNAL SPACE
                                                                ;JUST ALLOCATED.
013220  166767  000536  002222  4$:     SUB     LOLIM,HI        ;COMPUTE THE VIRTUAL LIMIT
                                                                ;OF THE TEST SPAN.
013226  016700  002216                  MOV     HI,R0
013232  004767  177504                  JSR     PC,ROT
013236  062700  020000                  ADD     #20000,R0
013242  010067  002202                  MOV     R0,HI
013246  010567  000510                  MOV     R5,LOLIM
013252  005237  177572                  INC     @#SR0           ;TURN ON MEMORY MANAGEMENT.
013256  000002                          RTI                     ;RETURN TO BRGOB TO
                                                                ;PERFORM THE TEST IN
                                                                ;THE SPAN INDICATED
                                                                ;BY THE RESULT OF THE
                                                                ;ABOVE
013260  004767  000056          CONIT:  JSR     PC,LOSEG2
013264  022626                          CMP     (SP)+,(SP)+
013266  000167  177550                  JMP     REPET2          ;TEST COMPLETED, SO
                                                                ;RESTART
                                ;
                                ;LOSEG1 IS CALLED TO DECIDE WHETHER OR NOT THE LIMITS ARE SUCH THAT
                                ;THE FIRST 4K OF MEMORY WILL HAVE TO BE CHECKED IN THE TEST. THAT
                                ;IS DO THE LIMITS INCLUDE ADDRESSES WHICH LIEM IN THE  FIRST 4K BLOCK
                                ;OF MEMORY THUS REQUIRING THAT THE CONTENTS OF THIS FIRST BLOCK OF
                                ;MEMORY BE MOVED INTO THE SECOND 4K BLOCK SO THAT THE TEST CAN BE
                                ;RUN THROUGH THE FIRST 4K BLOCK. IF RELOCATION IS NECESSARY LOSEG1
                                ;SETS LOSFL TO -1 AND RESETS THE LIMITS. LOLIM AND HILIM.
                                ;APPPOPRIATELY.
013272  005067  000324          LOSEG1: CLR     LOSFL           ;INITIALIZE.
013276  026727  000462  000200          CMP     LOSAV,#200      ;SEE IF THE LOW LIMIT OR THE HIGH LIMIT
013304  103011                          BHIS    2$              ;LIE IN THE FIRST 4K BLOCK OF MEMORY.
013306  005367  000310                  DEC     LOSFL
013312  022767  000200  000446          CMP     #200,HISAV
013320  103004                          BHIS    3$
013322  012767  000200  000432  1$:     MOV     #200,LOLIM
013330  000207                  2$:     RTS     PC
013332  012767  000200  000420  3$:     MOV     #200,HILIM
013340  000770                          BR      1$
                                ;
                                ;LOSEG2 IS CALLED TO DO THE ACTUAL RELOCATION OF THE FIRST 4K MEMORY
```

```
                                      ;BANK INTO THE SECOND 4K MEMORY BANK, THEN RUN THE TEST
                                      ;THROUGH THE DESIGNATED PARTS OF THE FIRST BANK AND THEN RESTORE
                                      ;THE CONTENTS OF THE FIRST BANK BY MOVING THE SECOND BANK'S
                                      ;CONTENTS BACK INTO THE FIRST BANK.
013342  005767  000254               LOSEG2: TST     LOSFL           ;SEE IF RELOCATION IS NECESSARY.
013346  100401                               BMI     1$              ;IF NOT RETURN.
013350  000207                               RTS     PC
013352  012667  000242               1$:     MOV     (SP)+,SAVPC
013356  005737  000120                       TST     @#120           ;SEE IF THE PROGRAM IS ALREADY RELOCATED
013362  100423                               BMI     14$
013364  004567  172656                       JSR     R5,RELOC
013370  000000                               .WORD   0
013372  020000                               .WORD   20000
013374  012737  177777  000120               MOV     #-1,@#120
013402  012701  020000                       MOV     #20000,R1
013406  060106                               ADD     R1,SP
013410  060116                               ADD     R1,(SP)
013412  060107                               ADD     R1,PC
013414  060137  000114                       ADD     R1,@#114
013420  060167  000174                       ADD     R1,SAVPC
013424  012767  177777  154466               MOV     #-1,RELFL
013432  016700  000326               14$:    MOV     LOSAV,R0        ;ESTABLISH VALID LIMITS FOR THE TEST
013436  004767  177300                       JSR     PC,ROT          ;THROUGH THE FIRST 4K MEMORY BANK.
013442  020027  000320                       CMP     R0,#320
013446  002002                               BGE     15$
013450  012700  000320                       MOV     #320,R0
013454  010067  001772               15$:    MOV     R0,LO
013460  016700  000302                       MOV     HISAV,R0
013464  022700  000200                       CMP     #200,R0
013470  002002                               BGE     2$
013472  012700  000200                       MOV     #200,R0
013476  020027  000004               2$:     CMP     R0,#4
013502  003002                               BGT     3$
013504  000167  000034                       JMP     LOSEG3
013510  004767  177226               3$:     JSR     PC,ROT
013514  010067  001730                       MOV     R0,HI
                                      ;
                                      ;
013520  012701  000024                       MOV     #LOSEG3-.,R1
013524  060701                               ADD     PC,R1
013526  062701  177772                       ADD     #-6,R1
013532  010137  000034                       MOV     R1,@#34
013536  005037  000036                       CLR     @#36
013542  000002                               RTI                     ;PERFORM THE TEST
                                      ;
                                      ;
                                      ;LOSEG3 RELOCATES BACK INTO THE FIRST 4K MEMORY BANK.
013544  016746  000350               LOSEG3: MOV     SAVPC,-(SP)
013550  005767  000042                       TST     PERRFL
013554  100417                               BMI     1$
013556  004567  172464                       JSR     R5,RELOC
013562  020000                               .WORD   20000
013564  000000                               .WORD   0
013566  005037  000120                       CLR     @#120
013572  012701  020000                       MOV     #20000,R1
013576  160106                               SUB     R1,SP
```

```
013600  160116                           SUB     R1,(SP)
013602  160137  000114                   SUB     R1,@#114
013606  160107                           SUB     R1,PC
013610  160166  000002                   SUB     R1,2(SP)
013614  000207                   1$:     RTS     PC
013616  000000                   PERRFL: .WORD   0
013620  000000                   SAVPC:  .WORD   0
013622  000000                   LOSFL:  .WORD   0
                                 ;
                                 ;
013624  005067  177766           LOSEG4: CLR     PERRFL
013630  026727  000132  000200           CMP     HISAV,#200
013636  101002                           BHI     1$
013640  005367  177752                   DEC     PERRFL
013644  000207                   1$:     RTS     PC
                                 ;
                                 ;
                                 ;
                                 ;LIMITS IS CALLED TO ASK THE USER FOR BOTH
                                 ;THE HIGH AND LOW UNIBUS ADDRESS LIMITS FOR
                                 ;THE IMPENDING TEST. THE TWO LIMITS ARE LEFT IN
                                 ;BLOCK NUMBER FORM AT LOCATIONS HILIM AND
                                 ;LOLIM (THEY ARE ALSO PUT IN LOSAV AND HISAV FOR
                                 ;LATER USE BY THE ROUTINE DONE). A VALIDITY CHECK
                                 ;IS MADE TO MAKE SURE THE INDICATED SPAN
                                 ;IS A VALID TEST SPAN.
                                 ;
013646                           LIMITS:
013646  004567  165106                   JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
013652  015104                           .WORD   HIMESS
013654  004767  000110                   JSR     PC,INUM ;ASSEMBLE THIS NUMBER.
013660  004767  000324                   JSR     PC,THRR
013664  004767  000330                   JSR     PC,THRR
013670  010367  000064                   MOV     R3,HILIM
013674  004567  165060                   JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
013700  015124                           .WORD   LOMESS
013702  004767  000062                   JSR     PC,INUM         ;ASSEMBLE THIS NUMBER
013706  004767  000306                   JSR     PC,THRR
013712  004767  000302                   JSR     PC,THRR
013716  010367  000040                   MOV     R3,LOLIM
013722  026767  000032  000032           CMP     HILIM,LOLIM     ;IF LOLIM IS GREATER
013730  003407                           BLE     LIMERR          ;THAN HILIM THEN GOTO
                                                                 ;LIMERR, ERROR
013732  016767  000022  000026   2$:     MOV     HILIM,HISAV     ;STORE THE LIMITS IN
013740  016767  000016  000016           MOV     LOLIM,LOSAV     ;SAVE REGISTERS.
013746  000207                           RTS     PC              ;RETURN
013750                           LIMERR:
013750  004567  165004                   JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
013754  015171                           .WORD   ERRMESS         ;WRITE AN ERROR MESSAGE
013756  000733                           BR      LIMITS          ;AND TRY AGAIN.
                                 ;
013760  000000                   HILIM:  .WORD   0
013762  000000                   LOLIM:  .WORD   0
                                 ;THESE ARE INTERMEDIATE STORGE REGISTERS:
013764  000000                   LOSAV:  .WORD   0
013766  000000                   HISAV:  .WORD   0
```

```
                                  ;
                                  ;
                                  ;THIS ROUTINE IS CALLED TO ASSEMBLE AN 18-BIT
                                  ;NUMBER FROM THE TTY AND TRUNCATE IT DOWN TO
                                  ;12-BITS
                                  ;A CALL IS MADE THUS:
                                  ;        JMP     PC,INUM
                                  ;RES:    .WORD   0
                                  ;THE NUMBER IS ASSEMBLED AND THE RESULTING 12-BIT
                                  ;TRUNCATED NUMBER IS LEFT IN RES. WHEN AND
                                  ;RTS RETURN IS MADE.
                                  ;NOTE THAT THE NUMBER SHOULD BE SPECIFIED
                                  ;IN OCTAL DIGITS. AN CHARACTERS WHICH DO NOT
                                  ;MEET THIS SPECIFICATION IN THE INPUT STRING
                                  ;WILL CAUSE AN ERROR WHICH WILL BE SIGNALED BY
                                  ;A ? ON THE TTY FOLLOWED BY A RETRY.
                                  ;
013770  005046                    INUM:   CLR     -(SP)           ;PUT A ZERO MARKER ON
013772  005002                            CLR     R2              ;SET UP THE TEMPORARY
013774  005003                            CLR     R3              ;STORAGE AND COUNTER
013776  012705  177771                    MOV     #-7,R5          ;REGISTERS.
014002  005737  177562                    TST     @#TKB
014006  012746  000076                    MOV     #'>,-(SP)
014012  004767  000220                    JSR     PC,TYPIT
014016  005726                            TST     (SP)+
014020  105737  177560            1$:     TSTB    @#TKS           ;WAIT FOR A CHARACTER
014024  100375                            BPL     1$
014026  013746  177562                    MOV     @#TKB,-(SP)     ;GET IT ONTO THE
014032  042716  177600                    BIC     #177600,(SP)    ;STACK
014036  022716  000177                    CMP     #177,(SP)       ;IS IT RUBOUT?
014042  001011                            BNE     2$              ;IF NOT GOTO 2$
014044  005726                            TST     (SP)+           ;IF IT WAS A RUBOUT
014046  005716                            TST     (SP)            ;FIRST SEE IF THEE
                                                                  ;IS A PREVIOUS
                                                                  ;CHARACTER ON THE STACK.
014050  001763                            BEQ     1$              ;IF THERE WAS NO PREVIOUS
                                                                  ;CHARACTER TAKE NO
                                                                  ;RUBOUT ACTION AND
                                                                  ;GO WAIT AT 1$ FOR
                                                                  ;THE NEXT CHARACTER
014052  012716  000134                    MOV     #'\,(SP)        ;IF THERE WAS
014056  004767  000154                    JSR     PC,TYPIT        ;A PREVIOUS CHARACTER
014062  005726                            TST     (SP)+           ;PRINT A SLASH
014064  000755                            BR      1$
014066  004767  000144            2$:     JSR     PC,TYPIT        ;IF THE LAST INPUT
                                                                  ;CHARACTER WAS NOT
                                                                  ;RUBOUT ECHO IT
014072  022716  000015                    CMP     #15,(SP)        ;IS IT CR.
014076  001350                            BNE     1$              ;NO, BRANCH TO 1$ FOR
                                                                  ;NEXT CHARACTER.
014100  012716  000012                    MOV     #12,(SP)        ;YES, PRINT A LF
014104  004767  000126                    JSR     PC,TYPIT
014110  005726                            TST     (SP)+
014112  005716                    3$:     TST     (SP)            ;START TO ASSEMBLE
014114  001415                            BEQ     4$              ;THE NUMBER. IF THE
                                                                  ;STACK IS AT THE
```

K07


```
                                                        ;ZERO MARKER WE ARE
                                                        ;DONE, SO GOTO 4$
014116  012604                      MOV     (SP)+,R4    ;CHECK TO SEE IF
014120  062704  177710              ADD     #-70,R4     ;THE CHARACTER IS A
014124  002022                      BGE     INERR       ;VALID OCTAL DIGIT.
014126  062704  000010              ADD     #10,R4      ;IF NOT GOTO INERR.
014132  002417                      BLT     INERR
014134  005205                      INC     R5          ;ARE THERE TO MANY DIGITS,
014136  001415                      BEQ     INERR       ;MORE THAN 6.
014140  004767  000054              JSR     PC,THRR     ;ROTATE THE DOUBLE
                                                        ;LENGTH "WORD" MADE OF
                                                        ;OF THE DIGITS PROCESSED
                                                        ;THUS FAR
014144  010402                      MOV     R4,R2       ;MAKE THIS NEW DIGIT
                                                        ;PART OF THE NUMBER.
014146  000761                      BR      3$          ;LOOP TO GET THE REST OF
                                                        ;THE NUMBER
014150  005205              4$:     INC     R5          ;MAKE SURE THE NUMBER
014152  001403                      BEQ     5$          ;HAS BEEN RIGHT JUSTIFIED
014154  004767  000040              JSR     PC,THRR     ;PROPERLY.
014160  000773                      BR      4$
014162  004767  000022      5$:     JSR     PC,ONER
014166  005726                      TST     (SP)+       ;AND RETURN CONTROL.
014170  000207                      RTS     PC
                            ;ERROR HANDLER FOR THE INUM ROUTINE.
                            ;RETURNS TO THE CALLING ROUTINE, LIMITS, TO ASK
                            ;FOR THE PARAMETER AGAIN.
014172                      INERR:
014172  004567  164562              JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
014176  015135                      .WORD   INRMES
014200  005726              1$:     TST     (SP)+       ;CLEAR JUNK OFF THE
014202  001376                      BNE     1$          ;STACK
014204  000167  177560              JMP     INUM    ;THE ROUTINE LIMITS.
014210  000241              ONER:   CLC                 ;ROUTINE WHICH ROTATES
014212  006002                      ROR     R2          ;TEMP2 AND TEMP3 ONE
014214  006003                      ROR     R3          ;BIT TO THE RIGHT
014216  000207                      RTS     PC          ;TREATING THEM AS A 32-BIT
                                                        ;OPERAND.
014220  004767  177764      THRR:   JSR     PC,ONER     ;ROUTINE WHICH CALL
014224  004767  177760              JSR     PC,ONER     ;ONER 3 TIMES.
014230  004767  177754              JSR     PC,ONER
014234  000207                      RTS     PC
014236  105737  177564      TYPIT:  TSTB    @#TPS       ;TYPIT TAKES THE
014242  100375                      BPL     TYPIT       ;WORD 2 BYTES UP IN
014244  116637  000002  177566      MOVB    2(SP),@#TPB ;THE STACK AND "PRINTS"
014252  000207                      RTS     PC          ;IT ON THE TTY
                            ;MOSPAR IS CALLED TO CHECK OUT THE POSIBILITY OF TURNING ON MOS
                            ;PARITY DURING THE BRANCH GOBBLE TEST. FIRST MOSPAR SEES WHAT
                            ;MOS PARITY REGISTERS EXIST AND THEN IFOME ARE FOUND THEY
                            ;ARE ENABLED AND THE INTERRUPT VECTOR IS SET TO TRAP TO PARERR A ROUTINE
                            ;WHICH WILL NOTIFY THE USER OF THE ERROR AND ITS LOCATION.
                            ;
        172100              PAREGS=172100
014254  004767  000454      MOSPAR: JSR     PC,MPVECT   ;SET THE PARITY ERROR TRAP VECTOR.
014260  012700  172100              MOV     #PAREGS,R0  ;GET READY TO LOOK AT THE POSSIBLE
                                                        ;PARITY REGISTERS PRESENT.
```

L07


```
014264  012701  177762                    MOV     #-16,R1
014270  012737  014352  000004            MOV     #MP2,@#4
014276  005037  000006                    CLR     @#6
014302  005067  000110                    CLR     MPFL
                                 ;THE REAL REGISTERS ARE LOCATED USING A TIME OUT PLAN WHERE BY THE
                                 ;POSSIBILITIES ARE REFERENCED AND IF A TIME OUT OCCURS THEY ARE NONEXISTENT
                                 ;OR IF NO TIME OUT OCCURS THEN THE REGISTER IS REAL. WHEN A REAL
                                 ;REGISTER IS FOUND IT IS WRITTEN INTO AND READ TO DETERMINE
                                 ;WHAT KIND OF PARITY REGISTER IT IS, CORE OR MOS. IF IT IS MOS THEN THE
                                 ;REGISTER IS SET TO ENABLE PARITY.
014306  005710                   MP1:     TST     (R0)                    ;IF THIS INSTRUCTION TIMES OUT THEN
                                                                          ;THERE IS NO PARITY REGISTER AT THE
                                                                          ;ADDRESS IN R0.
014310  005767  000102                    TST     MPFL                    ;SEE IF THE TABLE HEADING HAS BEEN OUTPUT.
014314  100403                            BMI     1$
014316  004567  164436                    JSR     R5,$PRINT               ;GO TO PRINT ROUTINE
014322  015000                            .WORD   MPMES
                                 ;
014324  010046                   1$:      MOV     R0,-(SP)                        ;OUTPUT AN ENTRY INTO THAT TABLE.
014326  004767  165414                    JSR     PC,O2A
014332  004567  164422                    JSR     R5,$PRINT               ;GO TO PRINT ROUTINE
014336  015020                            .WORD   $CRLF
014340  005367  000052                    DEC     MPFL
014344  012710  000001                    MOV     #1,(R0)
014350  000401                            BR      MP25
                                 ;
                                 ;IF A TIME OCCURS COME HERE. OR IF THE REGISTER WAS NOT MOS PARITY.
014352  022626                   MP2:     CMP     (SP)+,(SP)+
014354  062700  000002           MP25:    ADD     #2,R0
014360  005201                            INC     R1
014362  002751                            BLT     MP1                     ;BRANCH IF NOT DONE.
014364  005767  000026                    TST     MPFL                    ;SEE IF ANY MOS PARITY REGISTERS HAVE
                                                                          ;BEEN FOUND.
014370  100403                            BMI     MP3                     ;IF NOT DON'T OUT PUT THE NO PARITY MESSAGE.
014372  004567  164362                    JSR     R5,$PRINT               ;GO TO PRINT ROUTINE
014376  015027                            .WORD   NOMPAR
014400  012737  001116  000004   MP3:     MOV     #ERRTRP,@#4
014406  012737  000002  000006            MOV     #2,@#6
014414  000207                            RTS     PC
014416  000000                   MPFL:    .WORD   0
                                 ;
                                 ;IF A PARITY ERROR IS DETECTED THEN A TRAP IS MADE THROUGH LOCATION
                                 ;114 TO THIS ROUTINE. HERE THE USER IS TOLD OF THE PARITY ERROR AND THE
                                 ;LOCATION PLUS 2 OF THE INSTRUCTION WHICH CAUSED THE ERROR TO COME OUT.
                                 ;THEN A SCAN IS MADE THROUGH MEMORY. IF THE ERROR DOESN'T COME UP DURING
                                 ;THE SCAN THEN THE USER IS TOLD THAT THE ERROR WAS NOT FOUND ON THE SCAN.
                                 ;IF THE ERROR IS DETECTED ON THE SCAN THEN THE USER IS TOLD WHAT LOCATION.
                                 ;CAUSED THE ERROR AND WHETHER OR NOT MEMORY MANAGEMENT WAS ON DURING THE
                                 ;SCAN. IF IT WAS THEN THE PAR INVOLVED IN RELOCATING THE BAD LOCATION'S
                                 ;ADDRESS IS ALSO GIVEN TO THE USER SO HE CAN TAKE THE GIVEN
                                 ;ADDRESS AND RELOCATE IT USING THE PAR GIVEN TO FIND THE REAL MEMORY
                                 ;ADDRESS CAUSING THE PARITY ERROR. WHEN THIS HAS BEEN DONE THE TEST WHICH
                                 ;WAS IN PROGRESS WHEN THE ERROR WAS ENCOUNTERED IS RESTARTED, USING
                                 ;THE PARAMETERS WHICH THE USER LAST INPUT TO THE PROGRAM.
014420                           PARER2:
014420  004567  164334                    JSR     R5,$PRINT               ;GO TO PRINT ROUTINE
```

```
 4048  014424  006676                          .WORD   PARERR          ;TELL THE USER A PARITY ERROR OCCURRED.
 4049  014426  004567  164326                  JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
 4050  014432  015023                          .WORD   PERMES  ;TELL THE USER WHAT INSTRUCT CAUSED THE ERROR.
 4051  014434  004767  165306                  JSR     PC,02A
 4052  014440  004567  164314                  JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
 4053  014444  015020                          .WORD   $CRLF
 4054                                  ;
 4055  014446  012700  000052                  MOV     #PARR3-.,R0     ;SET THE TIME AND TRAP VECTORS SO
 4056  014452  060700                          ADD     PC,R0           ;THAT THE SCAN CAN BE MADE
 4057  014454  062700  177772                  ADD     #-6,R0
 4058  014460  010037  000114                  MOV     R0,@#PARVEC
 4059  014464  005037  000116                  CLR     @#PARVEC+2
 4060  014470  012700  000176                  MOV     #PARR4-.,R0
 4061  014474  060700                          ADD     PC,R0
 4062  014476  062700  177772                  ADD     #-6,R0
 4063  014502  010037  000004                  MOV     R0,@#ERRVEC
 4064  014506  005037  000006                  CLR     @#ERRVEC+2
 4065  014512  005003                          CLR     R3
 4066                                  ;
 4067  014514                          CONT:
 4068  014514  012301                  1$:     MOV     (R3)+,R1        ;PERFORM THE SCAN. EITHER THIS INSTRUCTION.
 4069  014516  000776                          BR      1$              ;WILL TIME OUT OR THE PARITY ERROR WILL
 4070                                                                  ;ONCE AGAIN OCCUR SO THAT CONTROL WILL
 4071                                                                  ;BE TRANSFERRED OUT OF THIS LOOP ONE WAY
 4072                                                                  ;OR ANOTHER.
 4073                                  ;
 4074                                  ;IF THE PARITY ERROR IS AGAIN ENCOUNTERRED THEN THIS PROGRAM WILL
 4075                                  ;RECIEVE CONTROL THROUGH A TRAP THROUGH VECTOR 114. HERE THE USER
 4076                                  ;IS GIVEN THE ADDRESS OF THE LOCATION WHICH CAUSED THE PARITY ERROR
 4077                                  ;DURING THE SCAN. THE USER IS ALSO TOLD WHETHER OR NOT MEMORY MANAGEMENT
 4078                                  ;IS ENABLED OR NOT. IF IT IS ENABLED THEN THE PAR INVOLVED WITH
 4079                                  ;THE ADDRESS RELOCATION IS ALSO GIVEN TO THE USER.
 4080  014520  062703  177776          PARR3:  ADD     #-2,R3          ;R3 CONTAINS THE ADDRESS PLUS 2
 4081  014524  010316                          MOV     R3,(SP)         ;OF THE LOCATION CAUSING THE ERROR.
 4082  014526  062703  000002                  ADD     #2,R3
 4083  014532  010367  000126                  MOV     R3,SAVPER
 4084  014536  004767  165204                  JSR     PC,02A  ;GIVE THE USER THIS ADDRESS.
 4085  014542  004567  164212                  JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
 4086  014546  015045                          .WORD   LOCBAD          ;TELL THE USER THIS IS THE ADDRESS AND
 4087  014550  004567  164204                  JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
 4088  014554  001506                          .WORD   RECDAT
 4089  014556  010116                          MOV     R1,(SP)         ;GIVE HIM THE BAD DATA.
 4090  014560  004767  165162                  JSR     PC,02A
 4091                                  ;
 4092  014564  033727  177572  000001          BIT     @#SR0,#1                ;SEE IF MEMORY MANAGEMENT WAS USED
 4093  014572  001423                          BEQ     2$              ;DURING THE SCAN THROUGH MEMORY.
 4094  014574  004767  177420          1$:     JSR     PC,THRR         ;IF IT WAS THEN ESTABLISH WHICH KIPAR
 4095  014600  004767  177414                  JSR     PC,THRR         ;IS INVOLVED WITH THE RELOCATION OF
 4096  014604  004767  177410                  JSR     PC,THRR         ;THIS BAD ADDRESS.
 4097  014610  004767  177404                  JSR     PC,THRR
 4098  014614  042703  177761                  BIC     #177761,R3
 4099  014620  062703  172340                  ADD     #KIPAR0,R3
 4100  014624  011316                          MOV     (R3),(SP)
 4101  014626  004567  164126                  JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
 4102  014632  015052                          .WORD   KPARM
 4103  014634  004767  165106                  JSR     PC,02A
```

```
 4104  014640  000403                          BR      RETPR           ;GO RESTART THE TEST WHICH WAS IN PROGRESS.
 4105                                ;
 4106  014642                        2$:
 4107  014642  004567  164112                  JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
 4108  014646  015071                          .WORD   NOKT11          ;THE SCAN SO TELL
 4109  014650  004567  164104        RETPR:    JSR     R5,$PRINT
 4110  014654  015020                          .WORD   $CRLF
 4111  014656  016703  000002                  MOV     SAVPER,R3
 4112  014662  000714                          BR      CONT
 4113                                                                  ;THE ADDRESS GIVEN FOR THE ERROR. THEN
 4114                                                                  ;RESTART THE TEST WHICH WAS IN PROGRESS
 4115                                                                  ;WHEN THE ERROR OCCURRED.
 4116  014664  000000                SAVPER:   .WORD   0
 4117                                ;
 4118  014666                        PARR4:
 4119  014666  004567  164066                  JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
 4120  014672  014760                          .WORD   NOTIND          ;TELL THE USER THAT IS THE CASE AND
 4121                                                                  ;RESTART THE SCAN WHICH WAS IN PROGRESS
 4122                                                                        ;WHEN THE ERROR WAS ORIGINALLY
 4123                                                                  ;ENCOUNTERRED.
 4124                                ;
 4125  014674  012706  000500                  MOV     #500,SP         ;RESTART THE TEST WHICH WAS INPROGRESS
 4126  014700  000005                          RESET                   ;WHEN THE ORIGINAL PARITY ERROR WAS FIRST
 4127  014702  004737  000124                  JSR     PC,@#WHERE      ;ENCOUNTERRED. SEE IF RELOCATION INTO THE
 4128  014706  012707  014712                  MOV     #1$,PC          ;FISRT 4K BANK IS NECESSARY.
 4129                                ;
 4130  014712  004767  177336        1$:       JSR     PC,MOSPAR       ;GO RESET THE PARITY REGISTERS BEFORE
 4131  014716  005767  164030                  TST     MMAVA           ;STARTING EITHER A MEMORY MANAGED
 4132  014722  001402                          BEQ     2$              ;OR NON MEMORY MANAGED TEST.
 4133  014724  000167  176112                  JMP     REPET2
 4134  014730  000167  175654        2$:       JMP     SPOT2
 4135                                ;
 4136                                ;
 4137                                ;
 4138                                ;
 4139                                ;
 4140                                ;ROUTINE USED TO SET THE TRAP VECTOR 114.
 4141  014734  012700  177464        MPVECT:   MOV     #PARER2-.,R0
 4142  014740  060700                          ADD     PC,R0
 4143  014742  062700  177772                  ADD     #-6,R0
 4144  014746  010037  000114                  MOV     R0,@#114
 4145  014752  005037  000116                  CLR     @#116
 4146  014756  000207                          RTS     PC
 4147                                ;
 4148                                ;MESSAGES USED FOR THESE PARITY ROUTINES.
 4149  014760  041523  047101  041440 NOTIND:  .ASCIZ  'SCAN COMPLETE'<15><12>
 4150  014766  046517  046120  052105
 4151  014774  006505  000012
 4152  015000  005015  040520  044522 MPMES:   .ASCII  <15><12>'PARITY ENABLED'
 4153  015006  054524  042440  040516
 4154  015014  046102  042105
 4155  015020  005015     000 $CRLF:   .ASCIZ  <15><12>
 4156  015023     120  036503     000 PERMES:  .ASCIZ  'PC='
 4157  015027     015  047012  020117 NOMPAR:  .ASCIZ  <15><12>'NO PARITY'<15><12>
 4158  015034  040520  044522  054524
 4159  015042  005015     000
```

```
[illegible]
HIMESS: .ASCII  15 12 'HIGH LIMIT' 15 12
LOMESS: .ASCII  'LOW LIMIT'
[illegible]
ERRMES: .ASCII  'NOT VALID!' 15 12
IDMESS: .ASCII  15 12 'BRANCH GOBBLE' 15 12

        .E.EN
;THE FOLLOWING IS THE CODE TAKEN DIRECTLY FROM
;THE PDP-11 FAMILY INSTRUCTION EXERCISER CZQKA-A.
;
;
MARKER: .WORD   125252
FIRST:  BR      .+4
        .WORD   0
        MOV     PC,R3
        SUB     #4,R3
        MOV     R3,R4
        INC     R4
        CLR     (R3)
1$:     SEC
        ADCB    (R3)
        BMI     2$
        INCB    (R4)
        BR      1$
2$:     BVS     .+4
        HALT
        CLV
        INCB    (R4)
        BCS     INCB1
        BVC     INCB1
        BMI     .+4
INCB1:  HALT
RTAD:   JMP     @#RELO
LAST:
STRTAD: .WORD   0
LENGTH: .WORD   1+LAST-FIRST/2
```

```
015314  016700  000130                  FIRST1: MOV     HI,R0
015320  012701  177770                          MOV     #LAST-.,R1
015324  060701                                  ADD     PC,R1
015326  062701  177772                          ADD     #-6,R1
015332  012703  000040                          MOV     #RELO-.,R3
015336  060703                                  ADD     PC,R3
015340  062703  177772                          ADD     #-6,R3
015344  010367  177736                          MOV     R3,RTAD+2
015350  016702  177736                          MOV     LENGTH,R2
015354  014140                          ABC:    MOV     -(R1),-(R0)
015356  005302                                  DEC     R2
015360  001375                                  BNE     ABC
015362  010067  177722                          MOV     R0,STRTAD
015366  000177  177716                          JMP     @STRTAD
015372  016700  177712                  RELO:   MOV     STRTAD,R0
015376  162767  000002  177704                  SUB     #2,STRTAD
015404  026767  177700  000040                  CMP     STRTAD,LO
015412  101414                                  BLOS    RET
015414  016701  177672                          MOV     LENGTH,R1
015420  011060  177776                  RELO1:  MOV     (R0),-2(R0)
015424  005720                                  TST     (R0)+
015426  005301                                  DEC     R1
015430  001373                                  BNE     RELO1
015432  016737  177652  177570                  MOV     STRTAD,@#177570
015440  000177  177644                          JMP     @STRTAD
015444  104400                          RET:    TRAP
015446  000722                                  BR      FIRST1
015450  000000                          HI:     .WORD   0
015452  000000                          LO:     .WORD   0
                                        :
                                        :
015454                                  LOCAR:
        000001                                  .END
```

D08

```
TEST DZDMS-G 0-124K MEMORY EXERCISER   MACY11 27(732)  10-SEP-76  11:59  PAGE 95
DZDMSG.P11      CROSS REFERENCE TABLE -- USER SYMBOLS

ABC       015354             4225#    4227
ADRESS    002253             1629     1794#
ALLFLG    004546             2300#    2376     2397     2417     2465     2485     2494*
ASTERI    012340             3273     3449#
BANKS     012102             2335     2505     3418#
BEGIN1    003276             1997     1999     2007#    2274
BELL      001521             1630     1646#
BIT12  =  010000             1300#    1926     2305
BIT13  =  020000             1299#
BIT14  =  040000             1298#    1364
BIT15  =  000100             1297#
BIT6   =  000100             1303#
BIT8   =  000400             1302#    3387
BIT9   =  001000             1301#    1598     3390
BLKCNT    004102             2127*    2135*    2162#    2169     2193
BPTVEC=   000014             1270#
BRANCH    012440             1413     3578#
C      =  000001             1250#
CASWR     011650             2009     2340     2647     2992     3154     3213     3241     3313     3386#
CONST     012161             2425     2524     3427#
CONT      014514             4067#    4112
COUNT     000756             1509#    2848*    2892*    2896*    2906*    3047*    3091*    3158*    3194*
DIGITS    002275             1575     1605     1652     1785#
DIGTAB    002236             1684     1777#
DISPAR    007042             2706     2724#
DISPLA=   177570             1291#    1592*    1814*    2098*    2129*    2139*    2149*    2203*    3395*    3462*
DONE      004564             2271     2275#
DONE0     002510             1836     1845#
DONE1     002600             1856     1873#
DONE10    003554             2077     2083#
DONE11    003650             2105#
DONE12    003700             2121#
DONE13    004074             2157#    2178
DONE14    004146             2174     2176     2180#
DONE15    004176             2192#
DONE3     002702             1898     1903#
DONE4     002770             1909     1920#
DONE4A    003040             1923     1936#
DONE4B    003110             1939     1953#
DONE4C    003156             1954     1966     1971#
DONE4D    003252             1971     1995#
DONE5     003334             2010     2019#
DONE6     003370             2035#    2060
DONE7     003416             2037     2043#
DONE8     003500             2056     2058     2063#
DONE9     003526             2064     2071#
DONIT     013260             3708     3742#
EMTVEC=   000030             1273#
END       004526             2266#
ENDMSG    012342             2277     3449#
ERCNT     000746             1504#    1585     1587*    1591     1812*    3392
ERRMES    015171             3872     4180#
ERROR     001174             1359     1584#    2862     2870     2878     2886     3013     3020     3027     3034     3065     3072     3169
                             3176     3227     3325     3332     3339     3346
ERRRC     001467             1603     1641#
ERRTRP    001116             1355     1563#    2019     2043     2073     2242     2311     2513     2617     4029
```

```
ERRVEC= 000004          1266#   1736*   1828*   1836*   1856*   1903*   1923*   1939*   1971*   2010*   2012*   2027*   2043*
                        2064*   2073*   2135*   2132*   2208*   2242*   2300*   2311*   2513*   2544*   2512*   2517*   2522*
                        2707    2709*   2723*   4062*   4064*
FINISH  004624          2281    2299#
FIRST   015232          4192#   4215
FIRST1  015314          3616    3691    4216#   4242
FPEVEC= 000244          1279#
FT      015450          3628*   3721*   3726*   3729*   3731    3734*   3803*   4216    4242#
[illegible]  013760     3609    3624    3626    3687*   3707    3715    3726    3764*   3857*   3864    3867    3875#
[illegible]MESS 015104  3853    4169#
[illegible]  013766     3687    3760    3795    3835    3867*   3879#
[illegible]  = 104400   1331#   1964    1918    1949    1982    1990    2229    2569    2592    2663    2691    3254
[illegible]  000742     1502#   1813*   2000*   2269*   2270    2312*   2375*   2396*   2416*   2442*   2454*   2434*   3397
[illegible]COUNT 000744 1503#   2852    2897*   2909    2974    2977    2999    3053    3099*   3118    3190*   3201    3356*
                        3396*   3399*
[illegible]MESS 015206  3591    4183#
[illegible]  015300     4208    4209    4211#
[illegible]ERR  014172  3940    3942    3944    3953#
[illegible]NAMES 015135 3965    4174#
[illegible]NUM  013770  3369    3854    3960    3897#   3966
[illegible]TVEC= 000020 1271#
KIPAR0= 172340          1320#   1457    1693    2736*   3664    4099
KIPAR1= 172342          1321#   1694    1881    1907*   2236*   2547*   2737*   2745*   2753*   3709
KIPAR2= 172344          1322#   1695    2214*   2237*   2734*   2777*   2791*
KIPAR3= 172346          1323#   1696    2779*   2792*
KIPAR4= 172350          1324#   1697
KIPAR5= 172352          1325#   1698
KIPAR6= 172354          1326#   1699
KIPAR7= 172356          1327#   1470    1700    2738*   3678*
KIPDR0= 172300          1311#   1452    2212*   2657*   2731*   3665
KIPDR1= 172302          1312#   2732*
KIPDR2= 172304          1313#   2213*   2733*   2780*
KIPDR3= 172306          1314#   2781*
KIPDR4= 172310          1315#   2782*
KIPDR5= 172312          1316#
KIPDR6= 172314          1317#
KIPDR7= 172316          1319#   1475    2735*
KM    = 000000          1257#
KPARM   015052          4102    4164#
KPDR  = 077406          3567#   3670
LAST    015310          4213#   4215    4217
LDCOMP  010152          2975    2979#
LDDISP  000750          1505#   1590*   1591*   1592    1813*   1814    2097*   2098    2129*   2129    2138*   2139    2146*
                        2149    2202*   2203    2313*   3386*   3392*   3394*   3395    3461*   3462
LDMMO   007044          1838    1859    1925    1941    1973    2211    2546    2656    2728#   2774
LDNORM  010164          2976    2978    2982#
LENGTH  015312          4215#   4224    4234
L[illegible]MERR 013750 3614    3865    3870#
LIMITS  013646          2603    3673    3851#   3873
LO      015452          3634*   3712*   3794*   4232    4244#
L[illegible]BAD 015045  4086    4163#
L[illegible]AF  015454  1743    1757    4248#
L[illegible]FLG 012436  3575#
L[illegible]  002042    1730#   1732    1748*   1752    1766*
L[illegible]A 004614    1262    1999    2284#
L[illegible]  013762    3631    3629*   3707    3710    3729    3735*   3762*   3863*   3864    3869    3876#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

H08

```
TEST DCDMS-D [illegible] MEMORY EXERCISER    MACY11 27(732)   10-SEP-76  11:59  PAGE 1[illegible]
DZDMSD.P11      CROSS REFERENCE TABLE -- USER SYMBOLS

RELE4A    012356        2626    3389    3454#
REPET1    012634        3621#   3643
REPET2    013042        3680#   3744    4133
REPET3    012732        3625    3635    3642#
RESLDR    011752        1911    3403#
RESVEC=   000010        1267#
RET       015444        4233    4241#
RETPR     014650        4104    4109#
ROTATE    011302        3221    3249    3265#
R[illegible]     004430        2245#
R[illegible]     004470        2256#
R[illegible]     015304        4212#   4223*
[illegible]   =%000006        1335#   2213    2731    2732    2735    2790    2791
R0       =%000000       1234#   1452*   1455    1457*   1458    1470*   1473*   1475*   1476*   1535    1536*   1539    1542*
                        1616    1634    1668*   1671*   1677*   1696*   1707*   1710*   1717*   1719*   1721*   1735*   1737
                        1740*   1741    1744    1857*   1959*   1960*   1962    1911*   1912*   1913*   1914*   1916    1920*
                        1929*   1931    1943*   1944*   1947    1960*   1961*   1963    1980    1985*   1989    2044*   2045*
                        2046    2051    2224*   2226    2280*   2294    2341*   2342*   2343*   2344*   2345*   2346    2542*
                        2545*   2552*   2564    2566    2575*   2592    2598    2611*   2613*   2615    2618*   2619    2639
                        2640*   2641*   2642*   2643*   2644*   2645*   2659*   2662    2711*   2718*   2806*   2807    2810*
                        2813    2815    2818    2820    2823    2825    2828    2830    2851*   2854*   2857*   2859    2865*
                        2867    2873*   2875    2881*   2883    2891*   2922*   2923    2928*   2931    2932    2933    2934
                        2941    2942    2943    2944    2953*   2979*   2992*   2998*   3001*   3010    3017    3024    3031
                        3039*   3041*   3052*   3055*   3062    3066*   3069    3073*   3077*   3080*   3092*   3102    3103
                        3104    3105    3106    3107    3108    3109    3131*   3133*   3135*   3136    3155*   3161*   3166
                        3173    3183*   3200*   3217*   3224    3245*   3252    3293*   3295    3296    3297    3298    3314*
                        3322    3329    3336    3343    3454*   3455*   3456    3610*   3611*   3612*   3613    3626*   3629
                        3631*   3634    3647*   3664*   3669*   3709*   3713*   3731*   3733*   3734    3789*   3791    3793*
                        3794    3795*   3796    3798*   3799    3803    3990*   4002    4010    4015*   4020*   4055*   4056*
                        4057*   4058    4060*   4061*   4062*   4063    4141*   4142*   4143*   4144    4216*   4225*   4228
                        4230*   4235*   4236
RCT       012742        3627    3633    3646#   3732    3790    3802
R1       =%000001       1235#   1656*   1661*   1662*   1663*   1664*   1665*   1666*   1837*   1854*   1874*   1904*   1924*
                        1940*   1957*   1972*   2013*   2021*   2036*   2049*   2065*   2072*   2105*   2112*   2121*   2137*
                        2157*   2167*   2180*   2186*   2196*   2207*   2249*   2260*   2267*   2641    3646*   3649*   3655*
                        3670*   3711*   3717*   3782*   3783    3784    3785    3786    3787    3806*   3807*   3808*   3809
                        3922*   3923    3824    3925    3926    3927    3992*   4021*   4068*   4099    4217*   4218*   4219*
                        4225    4234*   4237*
R10      =%000000       1242#
R11      =%000001       1243#
R12      =%000002       1244#
R13      =%000003       1245#
R14      =%000004       1246#
R15      =%000005       1247#
R2       =%000002       1236#   1453*   1454    1456*   1471*   1472    1477*   1570*   1600*   1606*   1634*   1635*   1654*
                        1655    1667*   1669*   1673*   1675*   1679*   1709*   1711*   1712*   1713    1716*   1718*   1720*
                        1722*   1742*   1745*   1758*   1760*   1840*   1841    1842*   1855*   1857    1961    1968    1993*
                        1894    1897*   1899    1910*   1911    1915    1927*   1931*   1942*   1946    1959*   1963*   1965
                        1975*   1979    1987    2215*   2218*   2219*   2225    2228    2230    2239*   2541*   2549*   2549*
                        2551*   2557    2564*   2565    2569    2576*   2577    2582*   2583    2586    2588    2602*   2603*
                        2604*   2642    2653*   2659    2676*   2677*   2712*   2720    2721*   2744*   2752*   2764    2768*
                        2769*   2770*   2771*   2775*   2776*   2777    2778*   2779    2783*   2790*   2904*   2913*   2914*
                        2815*   2816*   2819*   2819*   2820*   2821*   2823*   2924*   2925*   2926*   2928*   2929*   2930*
                        2831*   2849*   2858    2866    2874    2982    2994*   2900*   2901*   2902*   2903*   [illegible]   [illegible]
                        2932*   2933*   2934*   2936*   2937*   2939*   2939*   2941*   2942*   2943*   2944*   2945*   [illegible]
                        2948*   2949*   2967    2969    2972    2995*   3009    3016    3023    3030    3049*   3051    [illegible]
```

```
                          3066   3074*  3084*  3092   3093   3094*  3095*  3096*  3097*  3098*  3099*  3100*  3101*
                          3102*  3103*  3104*  3105*  3106*  3107*  3108*  3109*  3123*  3136*  3156*  3165   3172
                          3187*  3194*  3195*  3215*  3222   3243*  3250   3265*  3266*  3267*  3269*  3263*  3270*
                          3271*  3272*  3273*  3274*  3275*  3276*  3277*  3278*  3279*  3280*  3291*  3292*  3294*
                          3295*  3296*  3297*  3298*  3316*  3321   3328   3335   3342   3370   3938*  3949*  3970*
                          4224*  4226*
R3        =%000003        1237#  1454*  1459*  1472*  1474*  1621   1653*  1670*  1674*  1676*  1680*  1685*  1708*
                          1723*  1743*  1744*  1747   1941*  1842   1843*  1861*  1862   1881*  1882*  1883*  1884*
                          1885*  1886*  1897*  1893*  1894*  1895*  1896*  1915*  1916   1946*  1347   1979*  1980
                          1987*  1998   2023*  2024   2225*  2226   2529*  2531   2539   2541   2544*  2547   2551
                          2565*  2566   2577*  2579*  2581   2583   2585*  2586   2590   2643   2662*  2803*  2805*
                          2858*  2859   2866*  2867   2874*  2875   2882*  2883   2923*  2924*  2936   2937   2938
                          2939   2946   2947   2948   2949   2954*  2980*  2983*  3009*  3010   3016*  3017   3023*
                          3024   3030*  3031   3061*  3062   3068*  3069   3096*  3112*  3134*  3137*  3165*  3166
                          3172*  3173   3222*  3224   3250*  3252   3294*  3299*  3321*  3322   3329*  3329   3335*
                          3336   3342*  3343   3371   3666*  3669   3669*  3957   3863   3939*  3971*  4065*  4068
                          4090*  4091   4092*  4093   4098*  4099*  4100   4111*  4194*  4195*  4196   4198*  4200*
                          4220*  4221*  4222*  4223
R4        =%000004        1238#  1652*  1757*  1759   1930*  1932*  1945*  1950*  1962*  1967*  1976   1983*  1986*
                          1991*  2075*  2076*  2078   2091   2143   2216*  2218   2219   2224   2529*  2543*  2580*
                          2600*  2644   2812*  2833*  2929*  2951*  2995*  3042*  3056*  3079*  3093*  3094   3095
                          3096   3097   3098   3099   3100   3101   3128*  3141*  3160*  3181*  3189*  3198*  3214*
                          3230*  3242*  3258*  3290*  3301*  3315*  3351*  3636*  3637*  3638*  3639   3667*  3671*
                          3938*  3939*  3941*  3949   4196*  4197*  4202*  4207*
R5        =%000005        1239#  1384*  1491*  1515   1520*  1572*  1574*  1602*  1604*  1608*  1614*  1619*  1624*
                          1629*  1655*  1656   1672*  1673   1681*  1693*  1714*  1755*  1759*  1762*  1764*  1910*
                          1848*  1850*  2122*  2140*  2197*  2217*  2220*  2223*  2231*  2272*  2276*  2315*  2323*
                          2331*  2335*  2408*  2424*  2502*  2504*  2516*  2523*  2559*  2575   2576   2595   2603
                          2604   2620*  2623*  2645   2646*  2679*  2907*  2908*  2811*  2914   2915   2819   2821
                          2824   2826   2829   2831   2855*  2889*  2959*  2904*  3002*  3039*  3057*  3075*  3091*
                          3110*  3132*  3139*  3162*  3179*  3193*  3196*  3219*  3229*  3247*  3256*  3319*  3349*
                          3457*  3580*  3585*  3710*  3713   3714*  3715   3721   3724*  3735   3773*  3919*  3852*
                          3858*  3871*  3900*  3943*  3953*  3964*  4007*  4012*  4026*  4047*  4049*  4052*  4095*
                          4097*  4101*  4107*  4109*  4119*
SAVPC      013620         3775*  3787*  3815   3830#
SAVPC2     000122         1367#  1383*  1397
SAVPER     014664         4083*  4111   4116#
SAVR0      006662         2639*  2640   2635#
SAVR1      006664         2665#
SAVR2      006666         2687#
SAVR3      006670         2688#
SAVR4      006672         2689#
SAVR5      006674         2690#
SCOPE    = 104000         1332#
SLR      = 177774         1283#
SM       = 040000         1258#
SP        =%000006        1240#  1383   1387*  1392*  1396*  1399*  1402*  1408*  1411*  1418   1419*  1420*  1421*
                          1422*  1423*  1424*  1430   1431   1432   1433   1434   1435   1436   1446*  1450*  1451*
                          1455*  1458*  1460   1467*  1473   1476   1479   1479   1485   1517*  1535*  1536   1537*
                          1539*  1541   1542   1546   1548*  1551*  1558   1569   1570   1600   1606   1616*  1621*
                          1707   1726*  1739   1808*  1915*  1846*  1973*  1888*  1989*  1890   1936   1953   1995
                          2007*  2014*  2015*  2020*  2022*  2024*  2035*  2039*  2039*  2044   2043*  2050*  2051*
                          2059   2063*  2065*  2067*  2071*  2074*  2078*  2087*  2096*  2108*  2109*  2115*  2116*
                          2136*  2147*  2160*  2161*  2168*  2169*  2177   2191*  2192*  2187*  2188*  2201*  2241*
                          2246*  2247*  2250*  2251*  2257*  2259*  2261*  2262*  2269*  2297*  2359*  2360*  2361*
                          2364*  2365*  2366*  2367*  2368   2372*  2373*  2380*  2391*  2394*  2385*  2386*  2387*
```

```
                       2388   2393*  2394*  2403*  2404*  2405   2413*  2414*  2428*  2429*  2432*  2433*  2434*
                       2435*  2436   2439*  2440*  2449*  2450*  2453*  2454*  2455*  2456*  2457   2461*  2462*
                       2469*  2470*  2473*  2474*  2475*  2476*  2477   2481*  2482*  2490   2494   2499*  2511*
                       2557*  2599   2625*  2626*  2628   2629*  2630*  2632*  2661   2668*  2676   2678   2707*
                       2723   2746*  2755*  2760*  2793*  2803   2804   2837*  2838*  2940   2849   2961*  2969*
                       2977*  2985*  2894   2896   2910*  2911*  2920   2959*  2961*  2962*  2995   2996   3012*
                       3019*  3026*  3033*  3047   3048   3064*  3071*  3084   3086   3118*  3119*  3128   3129
                       3143*  3144*  3156   3158   3168*  3175*  3187   3189   3202*  3204*  3214   3215   3226*
                       3232*  3233*  3242   3243   3260*  3261*  3290   3291   3303*  3304*  3315   3316   3324*
                       3331*  3338*  3345*  3358*  3359*  3373   3376*  3460*  3590*  3591*  3593   3594   3597
                       3599*  3601   3613*  3615*  3616*  3617*  3618*  3621*  3623   3680*  3681*  3682*  3683*
                       3690*  3692   3743   3775   3793*  3784*  3815*  3823*  3824*  3927*  3897*  3902*  3904
                       3907*  3908*  3909   3911   3912   3920*  3922   3927   3930*  3932   3933   3933   3959
                       3966   3980   4010*  4019   4081*  4089*  4100*  4125*
SPOT      012560       3608#  3610
SPOT2     012610       3609   3615#  4134
SR0     = 177572       1306#  1450   1479*  1659   1929*  2306   2739*  2747*  2748*  2756*  2759*  2765*  2794*
                       3663*  3706*  3736*  4092
SR1     = 177574       1307#
SR2     = 177576       1308#  1451   1479*  2746   2755   2793
SR3     = 172516       1309#
START     002310       1398   1807#
START1    002352       1807   1815#  1817   2288
STBANK    012053       2324   3414#
STKPTR=   000500       1294#  1808   1815   1846   1873   2007   2020   2035   2048   2063   2071   2097   2136
                       2201   2241   2268   2297   2499   2511   2663   3460
STMM2     007240       2764#  2805   2850   2895   2921   2997   3049   3085   3130   3157   3198   3216   3244
                       3292   3317
STRTAD    015310       4214#  4228*  4229   4230   4231*  4232   4239   4240
SWR     = 177570       1290#  1563   1588   1593   1597   1627   1631   1326   2305   2521*  2533*  2555*  2705
                       3045   3387   3354   3387   3390   3397
T       = 000020       1254#
TBITVE=   000014       1268#
TEMP1     011642       3373*  3374*  3379#
TEMP2     011644       3370*  3375   3380#
TEMP3     011646       3371*  3374   3381#
THRR      014220       3955   3856   3861   3862   3945   3955   3974#  4094   4095   4096   4097
TKB     = 177562       1287#  2445   3590   3901   3907
TKS     = 177560       1286#  1446   1485*  2443   2554*  3599   3905
TKVEC   = 000060       1275#  2553*
TOP       012436       3565#
TPB     = 177566       1289#  3980*
TPS     = 177564       1298#  2278   3978
TPVEC   = 000064       1276#
TRAPVE=   000034       1274#  1358
TRAP4     001452       1573   1638#
TRTVEC=   000014       1269#
TYPIT     014236       3592   3600   3622   3691   3903   3921   3924   3931   3978#  3979
UBREAK=   177770       1285#
UM      = 140000       1259#
UP      = 000000       1336#  2213   2731   2732   2735   2780   2791
V       = 000002       1251#  2760
WCS.25    003434       2047#  2247   2251   2258   2262
WHERE     000124       1380#  1393   1397   1400   1403   1409   1412   4127
WRTAB     005100       2344   2349#
WRTUP     002450       1836#
```

```
\MTDAT   001473          1515    1642#
YMMSGO   012756          3595    3562#
YMMBGI   013122          3693    3706#
Z      = 000004          1252#
$ADRSC   005776          2519#   2528    2551*
$ALL     005676          2356    2494#
$CCNST   006026          2526#   2552
$CRLF    015020          1625    1851    2560    4013    4053    4110    4155#
$FILL    001011          1529#
$FORMD   001524          1571    1601    1607    1649#   1847
$LDR     002044          1732#   1809    3582
$LDRM    002210          1765    1771#
$LDR1    002206          1741*   1755    1769#
$NULL    001010          1528#   1549
$PRINT   000760          1491    1514#   1572    1574    1602    1604    1608    1614    1619    1624    1629    1714    1764
                         1810    1849    1850    2272    2276    2315    2323    2331    2335    2408    2424    2502    2504
                         2516    2523    2559    2523    2646    2679    3580    3585    3852    3858    3971    3954    4007
                         4012    4026    4047    4049    4052    4095    4087    4101    4107    4109    4119
$RESTR   000330          1430#   1484    1626    1688    1725    2537    3372
$RLDR    002134          1394    1752#   2282
$ROTO    005476          2353    2418    2449#   2467
$ROT1    005572          2354    2466    2469#   2486
$SAVR    000304          1418#   1447    1599    1651    1706    2535    3368
$TKFLG   001013          1532#
$TPB     001016          1534#   1552*
$TPFLG   001012          1530#
$TPS     001014          1533#   1556
$USER    005370          2352    2423#   2446
$1X8     005120          2349    2359#   2378    2487    2495
$3X9     005210          2350    2377    2380#   2491
$3X9P    005666          2355    2490#
$8X13    005276          2351    2398    2401#   2419
.      = 015454          1352#   1359#   1361#   1363#   1365#   1391#   1405#   1441#   1501#   1557    1564    1594    1632
                         1547#   1671    1738    1775#   1790#   1844    1948    1981    1989    2125    2194    2200    2279
                         2515    2598    2699#   2853    2860    2868    2876    2984    3000    3011    3018    3025    3032
                         3054    3063    3070    3167    3174    3191    3253    3323    3330    3337    3344    3398    3610
                         3616    3636    3681    3906    4055    4060    4141    4192    4204    4210    4217    4220
.BANKS   005024          2334#   2360    2381    2404    2414    2429    2450    2470
.CONST   005402          2245*   2256*   2427#   2459*   2479*   3293    3314
.EXTAD   006244          2520*   2529    2562*   2571#
.MAMF    005744          1482    1831    2008    2321    2570    2705#
.PARIT   004750          2318#   2319
.PARSR   006466          2639#   2709
.PAT     005040          2338#   2341
.ROTO    011120          2252    2463    3213#
.ROT1    011214          2263    2483    3241#
.STBAN   004776          2326#   2327    2359    2372    2380    2393    2403    2413    2428    2439    2449    2461    2469
                         2481
.TYPE    001020          1519    1535#
.USER    011350          2248    2259    2438    2460    2480    3290#
.YCR39   010110          2958    2967#   3007    3060
..USER   011422          2441    3313#
..1X8    007522          2025    2117    2374    2846#
..3X9    010176          2052    2170    2395    2991#
..8X13   010724          2079    2199    2415    3153#
..[illegible]  007410    2016    2110    2370    2803#
```

L08


```
.1S17    011640          2520    3375*   3378#
.3IS0    010126          2968    2972#
.3IS1    010116          2969#
.3IS9    010144          2971    2973    2977#
.3NOT9   010134          2970    2974#
.3N9     007756          2040    2164    2390    2920#
.9N13    010646          2068    2183    2412    3129#
```

$TYPE 1200#

```
ADC     1674    2562
ADCB    4200
ADD     1516    1537    1587    1635    1649    1650    1673    1712    1843    1860    1890    1896    1913    2095    2036
        2146    2147    2239    2343    2345    2561    2578    2603    2613    2626    2629    2632    2745    2778    2791
        3792    3376    3611    3612    3617    3618    3637    3638    3669    3682    3683    3714    3733    3783    3784
        3795    3786    3787    3807    3808    3939    3941    4020    4056    4057    4061    4062    4080    4082    4099
        4142    4143    4218    4219    4221    4222
ASL     1669    1675    1679    1882    1883    1884    1885    1886    1887    2134    2135    2195    2342    2361    2364
        2365    2366    2367    2384    2385    2386    2387    2432    2433    2434    2435    2453    2454    2455    2456
        2473    2474    2475    2476    2721    2897    3089    3190    3356    3547
ASR     2543
BCC     2556    2616    2722    3251
BCS     2309    2522    2534    3223    4208
BEQ     1449    1469    1481    1586    1589    1628    1658    1660    1733    1863    1880    1906    1917    1948    1966
        1978    1981    1989    1997    2056    2077    2085    2092    2144    2174    2194    2210    2227    2235    2271
        2281    2320    2406    2486    2567    2587    2589    2591    2655    2730    2754    2765    2860    2868    2876
        2884    2926    2956    2968    2970    2973    3005    3011    3018    3025    3032    3059    3063    3070    3116
        3167    3174    3191    3225    3253    3323    3330    3337    3344    3388    3391    3398    3595    3598    3915
        3934    3944    3954    4093    4132
BGE     3609    3625    3792    3797    3940
BGT     3800
BHI     2540    3836
BHIS    3758    3761
BIC     1590    1661    1672    1689    1690    1889    2076    2128    2138    2202    2548    3386    3455    3461    3591
        3908    4098
BIS     1591    2097    2148    2549    2554    2720    2760    3392    3394
BIT     1588    1597    1627    1659    1826    2305    2405    2705    2967    2969    2972    3045    3087    3354    3387
        3390    3397    4092
BLE     3865
BLOS    2532    3708    3716    4233
BLT     1552    3649    3719    3942    4022
BMI     1381    2853    2976    3773    3777    3817    4006    4025    4201    4210
BNE     1540    1547    1568    1598    1611    1613    1682    1687    1724    1746    1753    1761    1827    1900    1933
        1951    1968    1984    1992    2058    2176    2221    2232    2306    2328    2363    2377    2383    2398    2402
        2418    2431    2452    2466    2472    2530    2584    2675    2706    2767    2934    2836    2890    2893    2905
        2907    2909    2952    2960    3040    3043    3046    3076    3079    3082    3088    3111    3113    3138    3140
        3142    3180    3182    3185    3197    3199    3202    3229    3231    3257    3259    3300    3302    3350    3352
        3355    3357    3672    3910    3928    3967    4227    4238
BPL     1490    1557    1564    1594    1632    1999    2279    2444    2975    2978    3000    3054    3589    3906    3979
BR      1554    1596    1678    1738    1844    1872    1892    1919    1934    1952    1969    1993    2060    2178    2238
        2240    2378    2393    2419    2446    2467    2538    2550    2563    2570    2595    2598    2660    2692    2971
        3393    3602    3725    3765    3873    3923    3951    3956    4016    4069    4104    4112    4192    4203    4242
BVC     2898    3090    3192    4209
BVS     1898    1964    2719    4204
CLC     3220    3969
CLR     1388    1487    1578    1653    1685    1709    1716    1766    1812    1813    1816    1819    1820    1822    1925
        1829    1928    1943    1960    2015    2027    2039    2067    2108    2115    2156    2160    2168    2181    2187
        2216    2256    2295    2298    2299    2304    2312    2313    2314    2479    2512    2580    2648    2653    2667
        2713    2733    2736    2747    2759    2782    2851    2861    2869    2877    2885    2922    2998    3012    3019
        3026    3033    3052    3064    3071    3131    3168    3175    3226    3245    3324    3331    3338    3345    3463
        3465    3578    3579    3615    3635    3663    3666    3680    3694    3706    3756    3810    3821    3834    3897
        3898    3899    3994    3995    4059    4064    4065    4145    4198
CLRB    3339
CLV     4206
CMP     1585    1739    1862    1899    1916    1936    1947    1953    1965    1976    1980    1988    1995    2193    2226
        2270    2531    2539    2566    2583    2586    2598    2599    2859    2867    2875    2883    3010    3017    3024
```

CROSS REFERENCE TABLE -- PERMANENT SYMBOLS