

PDP11

MEMORY EXERCISER
MD-11-DZKMA-C

EP-DZKMA-C-DL-B

JUN 1977

COPYRIGHT © 1977

digital

FICHE 1 OF 1

MADE IN USA

This microfiche card contains a grid of frames. The first column on the left contains 12 frames, each showing a vertical list of data or a small diagram. The remaining frames in the grid (approximately 11 columns wide and 12 rows high) contain various types of data, including tables, diagrams, and text. The data appears to be organized in a structured manner, possibly representing a memory exercise or a technical specification. The frames are arranged in a regular grid pattern, typical of microfiche cards.

B01

EOF1DZLPGSEQ

00010000

770608

PDP10 411

HDR1DZKMACSEQ

00010000

770608

CO1

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 1
DZKMAC.P11 18-FEB-77 12:30

.REPT 0

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZKMA-C-D
PRODUCT NAME: MOS/CORE MEMORY EXERCISER FOR 0 TO 124K
WITH OR WITHOUT PARITY BITS
DATE CREATED: FEBRUARY-1977
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: PERVEZ A. ZAKI
REVISED: FRED STRAIGHT
DAN CASALETTO (AUGUST 1976)

COPYRIGHT (C) 1976, 1977
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY
ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH
THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS
SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED
OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT
FOR USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE
LICENSE TERMS. TITLE TO AND OWNERSHIP OF THE SOFTWARE
SHALL AT ALL TIMES REMAIN IN DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE
WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMIT-
MENT BY DIGITAL EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY
OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY
DEC.

CONTENTS

1.0	ABSTRACT
1.1	GETTING STARTED
2.0	REQUIREMENTS
2.1	EQUIPMENT
2.2	STORAGE
3.0	LOADING PROCEDURE
4.0	STARTING PROCEDURE
4.1	SWITCH SETTINGS
4.2	CONTROL-C OPTION
4.3	STARTING ADDRESS =200 RESTART ADDRESS =250
4.4	PROGRAM AND/OR OPERATOR ACTION
4.5	LONG GALLOP OPTION
5.0	PROGRAM HALTS (NORMAL + ERROR)
6.0	ERRORS
6.1	ERROR MESSAGE FORMAT.
6.2	ERROR DICTIONARY
6.3	ERROR HISTORY
6.4	ERROR RECOVERY
7.0	RESTRICTIONS
8.0	MISCELLANEOUS
8.1	ADDRESS/BANK RANGES IN OCTAL AND DECIMAL
8.2	EXECUTION TIME
8.3	PASS COUNT AND TEST NO. LOCATIONS
8.4	STACK POINTER
8.6	POWER FAIL
9.0	PROGRAM DESCRIPTION
9.1	NARRATIVE FLOW CHART
9.2	TEST TITLES
	TEST 0: TEST FOR PROPER BANK SELECTION
	TEST 1: CHECK DATI/DATO LINES
	TEST 2: TEST MEMORY FOR HOLDING DATA AND BYTE SELECTION
	TEST 3: DUAL ADDRESS TEST A
	TEST 4: DUAL ADDRESS TEST B
	TEST 5: MARCHING 1'S AND 0'S
	TEST 6: CELLS' VOLATILITY TEST
	TEST 7: SHIFTING DIAGONAL
	TEST 10: READ RECOVERY GALLOPING TEST THROUGH EVERY 64TH CELL
	TEST 11: READ RECOVERY LONG GALLOPING/FAST GALLOPING TEST
	TEST 12: WORST CASE TESTING FOR CORE MEMORY
	TEST 13: WRITE RECOVERY TEST
10.0	RXDP & ACT11 & APT OPERATION

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 3
 DZKMAC.P11 18-FEB-77 12:30

[1.0] ABSTRACT

THIS DIAGNOSTIC WILL TEST 0 - 124K OF MOS OR CORE MEMORY ON ANY PDP-11 FAMILY COMPUTER. SOME TESTS ARE WORST CASE FOR MOS AND SOME FOR CORE, BUT ALL TESTS ARE ALWAYS RUN. THE TESTS OCCUPIES LESS THAN 2K OF MEMORY SO IT CAN BE USED TO TEST A SYSTEM WITH ONLY 4K OF MEMORY. IF ONLY 4K EXISTS, HOWEVER, THE ABSOLUTE LOADER IS NOT SAVED.

THIS PROGRAM CAN BE RUN UNDER XXDP, APT AND ACT MONITORS. ON PROCESSORS WITH NO HARDWARE SWITCH REGISTER, SOFTWARE SWITCH REGISTER = LOCATION 176.

[1.1] GETTING STARTED

IF NO HARDWARE SWITCH REGISTER SET LOCATION 176 TO OBTAIN SWITCH OPTIONS.

TO START:

- A. SET SWITCH REGISTER = 00000
- B. START AT 200.
- C. THE MEMORY LIMITS WILL BE PRINTED.
- D. SEE SECTION 4.4 FOR REST OF PRINTOUTS EXPECTED.
- E. "END PASS #01" WILL BE TYPED LAST, AND THE TEST WILL RESTART.
- F. TO HALT THE TEST. TYPE CONTROL-C, THIS WILL INSURE THE PROGRAM IS RELOCATED BACK TO LOWER MEMORY.
 BE PATIENT, THE CONTROL-C IS ONLY RECOGNIZED AT THE END OF THE CURRENT SUBTEST.
- G. IF AN UNEXPECTED HALT OCCURS SEE SECTION 6.0. IF AN ERROR # IS TYPED SEE SECTION 6.2.

!CAUTION! BEFORE "DIGGING" INTO THE LISTING READ SECTION 9.

SWITCH SETTING SUMMARY (SEE SECTION 4.1 FOR DETAILS)

BIT15(100000)	HALT ON ERROR
BIT14(040000)	LOOP IN SUBTEST DEFINED BY BITS <3:0>
BIT13(020000)	INHIBIT ERROR PRINTOUTS
BIT12(010000)	ENABLE TESTING ABOVE 28K (MEMORY MANAGEMENT)
BIT11(004000)	ENABLE PARITY TESTING
BIT10(002000)	HALT AFTER EACH SUBTEST
BIT09(001000)	INHIBIT PROGRAM RELOCATION
BIT08(000400)	TYPE FIRST FAILING BIT ERROR PER 4K.
BIT07(000200)	ENABLE LONG GALLOPING TEST
BIT06(000100)	INHIBIT MEMORY SIZING
BIT05(000040)	INHIBIT "END PASS #XX" PRINTOUTS
BIT04(000020)	INHIBIT PRINTOUTS
BIT03-BIT00	BEGINNING TEST NUMBER.

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 4
 DZKMAC.P11 18-FEB-77 12:30

[2.0] REQUIREMENTS

[2.1] EQUIPMENT

STANDARD 11 FAMILY COMPUTER WITH A CONSOLE OUTPUT DEVICE
 AND FROM 4K TO 124K OF MEMORY.

[2.2] STORAGE

PROGRAM STORAGE - 0000 - 7744. PROGRAM EXPANDS FOR ERROR
 HISTORY AND TO SAVE ABSOLUTE LOADER OR XXDP CHAIN MONITOR.
 (SEE SECTION 9. FOR DETAILS)

[3.0] LOADING PROCEDURE

USE STANDARD PROCEDURE FOR PDP-11 ABSOLUTE BINARY FORMATTED TAPES.

[4.0] STARTING PROCEDURE

[4.1] SWITCH SETTINGS

SOFTWARE SWITCH REGISTER = LOCATION 176

BIT15(100000) HALT ON ERROR

BIT14(040000) LOOP ON TEST DEFINED BY SWITCH REGISTER BITS <3:0>

BIT13(020000) INHIBIT ERROR PRINTOUTS

BIT12(010000) ENABLE TESTING ABOVE 28K. (MEMORY MANAGEMENT)

BIT11(004000) ENABLE PARITY MODULES.

BIT10(002000) !"PARITY" WILL BE TYPED

HALT AFTER EACH SUBTEST

BIT09(001000) !PRESS CONTINUE TO DO NEXT SUBTEST

INHIBIT PROGRAM RELOCATION

!IF SET LOCATIONS 430-7776 WILL NOT BE
 !TESTED.

BIT08(000400) TYPE FIRST FAILING BIT IN EACH 4K BANK ONLY.

!THE TOTAL ERROR COUNT (UP TO 377) WILL

!BE SAVED IN THE ERROR HISTORY.

BIT07(000200) ENABLE LONG GALLOPING TEST.

! "GLP" WILL BE TYPED.

!CAUTION! INCREASES TEST TIME BY FACTOR OF 25.

BIT06(000100) INHIBIT MEMORY SIZING.

!THE MEMORY LIMITS MUST BE SETUP IN THE FOLLOWING LOCATIONS:
 (VALUES TO TEST 0-8K ARE SHOWN)
 (LOWTWO=LOCATION 322)

LOWTWO: 0
 LOWADD: 0

;STORE BITS 17:16 OF LOW TEST ADDRESS
 ;STORE REST OF LOW TEST ADDRESS

HIGHTWO: 0 ;STORE BITS 17:16 OF HIGH TEST ADDRESS
 HIGHADD: 37776 ;STORE REST OF HIGH TEST ADDRESS
 NOTE: HIGHADD MUST BE SET TO A 4K BOUNDARY. (E.G. 37776)

BIT05(000040) INHIBIT "END PASS #XX" PRINTOUTS
 BIT04(000020) A. INHIBIT ERROR HISTORY PRINTOUTS. THE
 ERROR HISTORY CAN STILL BE OBTAINED
 BY TYPING CONTROL-C.
 B. INHIBIT PRINTOUTS "PARITY", "GLP", "TST13 BNK XX".
 BIT03-BIT00 NUMBER OF TEST (0-13) TO RUN FIRST.
 !NORMALLY USED WITH BIT14 (LOOP ON TEST)

[4.2] CONTROL-C OPTION

CONTROL C [+C] AFTER COMPLETION OF THE CURRENT TEST.
 THE ERROR HISTORY (SEE SEC. 6.3) WILL BE
 TYPED. THE PROGRAM WILL HALT IN LOWER MEMORY.
 PRESSING CONTINUE WILL RESTART THE DIAGNOSTIC.

[4.3] STARTING ADDRESS= 200
RESTART ADDRESS = 250 OR 200

RESTART AT 200 CLEARS PASS COUNT (\$PASS) AND PRINTS "DZKMA-C" TITLE.

[4.4] PROGRAM AND/OR OPERATOR ACTION

- 1) LOAD PROGRAM INTO MEMORY USING ABSOLUTE LOADER.
- 2) SET OPTIONS (SEE SEC. 4.1)
- 3) START THE PROGRAM AT 200
- 4) THE FOLLOWING IS AN EXAMPLE WITH EXPLANATIONS
OF THE PRINTOUTS EXPECTED.

"XXXXX-YYYYY" ;ADDRESSES OF TEST BOUNDARIES.
 "PARITY" ;IF PARITY OPTION SELECTED
 "GLP" ;IF LONG GALLOPING OPTION SELECTED.
 ;PRINTED AS TST11 IS ENTERED.
 "TST13 BNK 00" ;ENTERING BANK 00 IN TEST 13.
 "TST13 BNK 01" ;AND BANK 1....
 ETC... ;UNTIL ALL BANKS (UP TO 6) HAVE BEEN TESTED.
 "RELOC" ;THE DIAGNOSTIC RELOCATES TO HIGHEST
 BANK UNDER TEST. AND RUNS TST0-TST13 AGAIN.
 "TST13 BNK 00" ;TESTING BANK 00 IN TEST 13 (RELOCATED STATE.)
 ;NOTE-ONLY BANK 00 IS TESTED IN THE RELOCATED STATE.
 "END PASS #XX" ;WHERE "XX" IS THE PASS NO.

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 6
 DZKMAC.P11 18-FEB-77 12:30

ADDITIONAL PRINTOUTS

"NO PAR"

;PRINTED IF PARITY SELECTED BUT NOT AVAILABLE.

"NO MNG"

;PRINTED IF GREATER THAN 28K AND NO MEMORY
 ;MANAGEMENT AVAILABLE.

4.5 LONG GALLOP OPTION

NORMAL WORST CASE SR SETTING = 0000. FOR LONG GALLOP
 SR = 200. LONG GALLOP OPTION SHOULD ONLY BE USED IF AN
 MOS MEMORY PROBLEM IS SUSPECTED AND NO OTHER SUBTESTS
 WILL FAIL. THE TEST TIME IS INCREASED 25 TIMES.

[5.0] PROGRAM HALTS (NORMAL+ ERROR)

THIS IS A LIST OF EXPECTED HALTS. IF THE TEST HALTS
 IN A LOCATION NOT IN THIS LIST AND ITS LESS THAN 776, IT
 MAY BE DUE TO A DEVICE INTERRUPTING.

NOTE THE HALT AT END OF SUBTEST AND HALT ON ERROR HALT LOCATIONS
 MAY BE RELOCATED. THE ACTUAL LOCATIONS THEY ARE IN CAN BE FOUND
 BY SUBTRACTING 500 FROM 1664(SMHALT) AND ADDING THIS DIFFERENCE TO THE
 CONTENTS OF SAVR6 [LOC. 346].

PC ---	REASON -----	RECOVERY -----
112	TRAP TO LOC. 4	EXAMINE R6, IT CONTAINS THE POINTER TO THE PC WHERE THE TRAP OCCURRED.
146	POWER FAIL	POWER UP WILL RECOVER IF IN CORE MEMORY.
1666	HALT AT END OF TEST SWITCH SET.	PRESS CONTINUE TO GO TO NEXT SUBTEST.
6132	HALT ON ERROR SWITCH SET.	PRESS CONTINUE.
6216	CONTROL-C TYPED OR FATAL ERROR OCCURRED	PRESS CONTINUE TO RE- START TEST.

[6.0] ERRORS

[6.1] ERROR MESSAGE FORMAT

THE ERROR PRINTOUT CONSISTS OF 6 OCTAL WORDS IN THE FOLLOWING
 FORMAT:

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 7
 DZKMAC.P11 18-FEB-77 12:30

"LOCATION GOOD BAD PC ERROR PASFLG"

"ADR ERR" WILL BE PRINTED PRIOR IF AN ADDRESSING ERROR IS SUSPECTED.
 "PAR ERR" WILL BE PRINTED PRIOR IF A PARITY ERROR TRAP OCCURRED
 !CAUTION! IF PARITY ERROR THE GOOD DATA PRINTOUT IS THE
 PARITY MODULE UNIBUS ADDRESS THAT FAILED.

WHERE:

LOCATION=	FAILING MEMORY LOCATION
GOOD =	GOOD DATA [DATA THAT WAS EXPECTED]
BAD =	BAD DATA [DATA THAT WAS FOUND]
PC =	PROGRAM COUNTER AT ERROR CALL.
ERROR =	FAILING ERROR NO. (SEE SEC 6.2 - ERROR DICTIONARY)
PASFLG =	CONTENTS OF LOCATION PASFLG. THIS MAY NOT BE RELEVANT. (SEE SEC. 6.2-ERROR DICTIONARY)

!THE TEST WILL CONTINUE AFTER THE ERROR PRINTOUT.
 !"NO MNG" WILL BE TYPED IF TESTING ABOVE 28K SELECTED AND NO MEMORY
 !MANAGEMENT IS FOUND.

!"NO PAR" WILL BE TYPED IF PARITY OPTION SELECTED
 !AND NO PARITY MODULES WERE FOUND.

(FATAL ERRORS)

"ERROR #XXXXXX" WILL BE TYPED WHERE "XXXXXX" IS
 THE ERROR NUMBER. THE DIAGNOSTIC WILL USUALLY HALT ON THIS TYPE
 OF ERROR. SEE SEC. 6.2 -ERROR DICTIONARY - FOR DESCRIPTIONS
 OF THE ERROR.

(APT MODE ERRORS)

ALL ERRORS ARE TREATED AS FATAL UNDER APT. WHEN AN
 ERROR OCCURS UNDER APT A "1" IS STORED IN LOCATION
 SMSGTY AND THE PROGRAM HALTS AT FATHLT.

\$FATAL CONTAINS THE ERROR NO. IN THE LOW BYTE AND
 THE FAILING BANK NO. UNDER TEST IN THE HIGH BYTE.

[6.2] ERROR DICTIONARY

THIS IS A LIST OF ERROR NUMBERS PRINTED AND POSSIBLE
 CAUSES FOR THE ERROR.
 THE ROUTINE NAME WHERE THE ERROR CALL ORIGINATED IS GIVEN IN
 BRACKETS.
 NOTE- "BAKPAT" REFERS TO THE BACKGROUND PATTERN WRITTEN INTO MEMORY

J01

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 8
DZKMAC.P11 18-FEB-77 12:30

FOR VARIOUS TESTS. IF PARITY SELECTED IT HAS A VALUE = 376 ,ELSE=377
"SWAPPED BAKPAT" = 77000 IF PARITY SELECTED, ELSE=77400

.ENDR

```
;ERROR # 0      ;[BUSER] BUS ERROR TRAP TO LOC. 4 OCCURRED
;               ;      THIS ERROR IS NOT PRINTED AND IS FOR "APT" USE.

;ERROR # 1      ;[TSTTRP]FATAL DATA ERROR
;               ;LOCATIONS 0000-430 FAILED 1'S + 0'S TEST.
;               ;RD = GOOD DATA
;               ;R1 = ADDRESS OF FAILING LOCATION.

;ERROR # 2      ;[APTSIZ] APT FATAL ERROR
;               ;APT MEMORY TABLES NOT SETUP CORRECTLY.
;               ;CHECK LOCATIONS SHAMS1 [430] TO SHADR4[446]
;               ;, FOR CORRECT MEMORY SIZE DATA.

;ERROR # 3      ;[TSTSIZ] OPERATOR FATAL ERROR
;               ;SELECTED MEMORY SIZE GREATER THAN 28K, BUT
;               ;SR BIT12 (10000) NOT SET.
;               ;SET BIT12 AND RESTART AT 200.

;ERROR # 4      ;[TSTSIZ] OPERATOR FATAL ERROR
;               ;LOWEST SELECTED TEST LIMIT IS HIGHER THAN
;               ;HIGHEST TEST LIMIT. SET LOCATIONS "LOWTWO"[322]
;               ;TO "HIGHADD" [330] CORRECTLY AND RESTART
;               ;AT 200.

;ERROR # 5      ;[TSTO] TEST SEQUENCE ERROR
;               ;TSTO HAS BEEN ENTERED OUT OF SEQUENCE
;               ;TESTN SHOULD = 00
;               ;THE DIAGNOSTIC HAS BEEN CORRUPTED.
;               ;IF POSSIBLE SELECT ANOTHER 4K BANK
;               ;BANK 0 AND RERUN THE TEST ON THE FAILING MEMORY.

;ERROR # 6      ;[TSTO] DUAL ADDRESSING ERROR
;               ;FOR THIS ERROR THE GOOD DATA PRINTED IS AN
;               ;ADDRESS. THIS IS THE ADDRESS SELECTED WHEN
;               ;THE SAME DATA WAS WRITTEN INTO THE FAILING
;               ;LOCATION. CHECK BANK SELECT CIRCUITRY

;ERROR # 7      ;[TSTO] ADDRESS AND DATA ERROR
;               ;IDENTICAL TO PREVIOUS ERROR EXCEPT THE DATA
;               ;WRITTEN INTO THE FAILING LOCATION WAS IN
;               ;ERROR ALSO.

;ERROR # 10     ;[TSTO] DATA ERROR
;               ;IF BAD DATA = 0000 COULD BE AN ADDRESSING
;               ;ERROR , ELSE COMPARE GOOD AND BAD DATA FOR FAILING BITS.

;ERROR # 11     ;[TSTO] ADDRESSING ERROR
;               ;THE FAILING ADDRESS RESPONDED BUT IS NON-
;               ;EXISTENT. MAY BE A DUAL ADDRESSING PROBLEM.
```


K01

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 9
DZKMAC.P11 18-FEB-77 12:30

;ERROR # 12 :[TST1] TEST SEQUENCE ERROR
:[TEST (404) SHOULD = 01
: THE DIAGNOSTIC HAS BEEN CORRUPTED.

;ERROR # 13 :[TST1] DATA ERROR
:[COMPARE GOOD AND BAD PRINTED DATA, FAILING
:[DATA BITS MAY SHORTED OR SWAPPED.

;ERROR # 14 :[TST2] TEST SEQUENCE ERROR
:[STESTN (404) SHOULD = 02
: THE DIAGNOSTIC HAS BEEN CORRUPTED.

;ERROR # 15 :[TST2] ADDRESS OR DATA ERROR
:[IF "ADR ERR" NOT PRINTED THEN THE BYTE SELECT
:[CIRCUITRY PROBABLY FAILED.

;ERROR # 16 :[TST3] TEST SEQUENCE ERROR
:[STESTN (404) SHOULD = 03
: THE DIAGNOSTIC HAS BEEN CORRUPTED.

;ERROR # 17 :[TST3] DUAL ADDRESSING ERROR
:[DUAL ADDRESSING PROBLEM FOR BITS THAT DIFFER
:[IN GOOD AND BAD DATA PRINTOUT.

;ERROR # 20 :[TST3] DUAL ADDRESSING ERROR
:[FOR THIS ERROR THE DATA PRINTED IS AN ADDRESS.
:[THIS IS THE ADDRESS THAT WAS SELECTED WHEN THE
:[SAME DATA WAS WRITTEN INTO THE FAILING LOCATION.

;ERROR # 21 :[TST3] DUAL ADDRESSING ERROR
:[SAME AS ERROR #20 EXCEPT DIFFERENT DATA
:[(SWAPPED BAKPAT) WAS WRITTEN.

;ERROR # 22 :[TST4] TEST SEQUENCE ERROR
:[STESTN (404) SHOULD = 04.
: THE DIAGNOSTIC HAS BEEN CORRUPTED.

;ERROR # 23 :[TST4] DUAL ADDRESSING ERROR
:[IF PASFLG = 0 THEN THE FAILING LOCATION
:[AND FAILING DATA ARE DUAL ADDRESSES.

;ERROR # 24 :[TST5] TEST SEQUENCE ERROR
:[STESTN (404) SHOULD = 05
: THE DIAGNOSTIC HAS BEEN CORRUPTED.

;ERROR # 25 :[TST5] DATA ERROR
:[DATA WRITE OR READ ERROR.

;ERROR # 26 :[TST5] MARCHING 1'S AND 0'S DATA ERROR
:[IF PASFLG=0 FAILED MARCHING 1'S + 0'S IN
:[MAX TO MIN DIRECTION.
:[IF PASFLG=1 FAILED MARCHING 1'S + 0'S IN
:[MIN TO MAX DIRECTION
:[IF PASFLG=3 FAILED MARCHING 0'S + 1'S IN
:[MAX TO MIN DIRECTION.

;ERROR # 27 :[TST5] MARCHING 1'S AND 0'S DATA ERROR

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 10
 DZKMAC.P11 18-FEB-77 12:30

```

;IDENTICAL TO PREVIOUS ERROR EXCEPT THE DATA IS
;CHECKED IMMEDIATELY AFTER BEING WRITTEN.

;ERROR # 30 ;[TST6] TEST SEQUENCE ERROR
;           ;STESTN SHOULD = 06
;           ;THE DIAGNOSTIC HAS BEEN CORRUPTED.

;ERROR # 31 ;[TST6] VOLATILITY/REFRESH TEST ERROR
;           ;IF PASFLG=0 BAKPAT WRITE OR READ ERROR.
;           ;IF PASFLG=1 THE FAILING LOCATION CHANGED WHILE
;           ;           ;ANOTHER LOCATION WAS WRITTEN FOR
;           ;           ;2 MS. THE OTHER LOCATION IS SAVED
;           ;           ;IN SAVLOC (352)
;           ;IF PASFLG=2 SWAPPED BAKPAT (77400 OR 77000)
;           ;           ;WRITE OR READ ERROR.
;           ;IF PASFLG=3 SAME AS IF PASFLG=2 EXCEPT
;           ;           ;THE DATA IS SWAPPED BAKPAT.

;ERROR # 32 ;[TST7] TEST SEQUENCE ERROR
;           ;STESTN SHOULD = 07
;           ;THE DIAGNOSTIC HAS BEEN CORRUPTED.

;ERROR # 33 ;[TST7] SHIFTING DIAGONAL DATA ERROR
;           ;IF PASFLG=0 BAKPAT WRITE OR READ ERROR.
;           ;IF PASFLG=1 BAKPAT READ CHECK ERROR
;           ;IF PASFLG= GREATER THAN 1 BUT EVEN VALUE THEN:
;           ;           ;THE FAILING LOCATION COULD NOT BE WRITTEN INTO.
;           ;IF PASFLG= GREATER THAN 1 BUT ODD VALUE THEN:
;           ;           ;THE FAILING LOCATION WAS WRITTEN CORRECTLY
;           ;           ;BUT LOST THE DATA.

;ERROR # 34 ;[TST10] TEST SEQUENCE ERROR
;           ;STESTN SHOULD = 10
;           ;THE DIAGNOSTIC HAS BEEN CORRUPTED.

;ERROR # 35 ;[TST10] BAKPAT DATA ERROR
;           ;BAKPAT WRITE OR READ ERROR INTO THE FAILING LOCATION.

;ERROR # 36 ;[TST10] READ RECOVERY DATA ERROR
;           ;THIS ERROR CAN BE REPORTED BY TST10 AND TST11.
;           ;(THEY SHARE CODE). SEE STESTN (404) FOR WHICH TEST FAILED.
;           ;FOR BOTH TESTS COMPARE THE GOOD AND BAD DATA AT THE FAILING
;           ;LOCATION TO SEE WHICH BITS FAILED.

;ERROR # 37 ;[TST10] READ RECOVERY DATA ERROR
;           ;IDENTICAL TO THE PREVIOUS ERROR EXCEPT SWAPPED BAKPAT IS
;           ;USED AS WRITE AND READ DATA.

;ERROR # 40 ;[TST11] TEST SEQUENCE ERROR
;           ;STESTN SHOULD = 11
;           ;THE DIAGNOSTIC HAS BEEN CORRUPTED.

;ERROR # 41 ;[TST12] TEST SEQUENCE ERROR
;           ;STESTN SHOULD = 12
;           ;THE DIAGNOSTIC HAS BEEN CORRUPTED.

```


MO1

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 11
 DZKMAC.P11 18-FEB-77 12:30

```

;ERROR # 42 :[TST12] WORST CASE CORE TEST DATA ERROR
              :IF PASFLG=1 COMPARE GOOD AND BAD DATA FOR FAILING BITS.
              :IF PASFLG=2 THE FAILING LOCATION WAS WRITTEN AND READ
              :              WITH GOOD DATA, BUT FAILED READ CHECK
              :              READING IN THE MIN. TO MAX DIRECTION.
              :IF PASFLG=3 SAME CONDITIONS AS PASFLG=2 EXCEPT FAILED
              :              DOING THE READ CHECK FROM MAX TO MIN DIRECTION.

;ERROR # 43 :[TST12] WORST CASE CORE TEST DATA ERROR
              :IDENTICAL TO PREVIOUS ERROR EXCEPT THE DATA WRITTEN
              :AND READ IS COMPLEMENTED.

;ERROR # 44 :[TST13] TEST SEQUENCE ERROR
              :STESTN SHOULD = 13
              :THE DIAGNOSTIC HAS BEEN CORRUPTED.

;ERROR # 45 :[TST13] WRITE RECOVERY TEST DATA ERROR
              :IF PASFLG=0 COMPARE GOOD AND BAD DATA FOR FAILING BITS.
              :IF PASFLG=77400 DATA ERROR FOUND WHILE DOING A SECOND READ CHECK.
              :IF PASFLG=77402 DATA ERROR FOUND IN FAILING LOCATION AFTER
              :              SMALL TEST PROGRAM RUN IN FAILING BANK.

;ERROR # 46 :[TST13] WRITE RECOVERY TEST DATA ERROR
              :DATA ERROR FOUND JUST BEFORE THE SMALL TEST
              :WAS TO BE RUN IN THE FAILING BANK. TO AVOID "BLOWING" UP
              :WHEN THE SMALL TEST IS RUN TST13 IS ABORTED.

;ERROR # 47 :[TST13] WRITE RECOVERY TEST DATA ERROR
              :IDENTICAL TO ERROR #XXX EXCEPT THE DATA WRITTEN
              :AND READ IS DIFFERENT. (177667).
              :177667 IS THE COMPLEMENT OF "JMP (R0)" (110) WHICH IS
              :THE ESCAPE FROM THE SMALL TEST PROGRAM RUN IN THE BANK
              :UNDER TEST.

;ERROR # 50 :[PARERR] PARITY TRAP ERROR
              :PARITY TRAP TO 114 OCCURRED.
              :FOR THIS ERROR PRINTOUT THE "GOOD DATA" IS ACTUALLY
              :THE FAILING PARITY MODULE UNIBUS ADDRESS.
              :SAVLOC [352] CONTAINS THE PC WHERE THE TRAP OCCURRED.

;ERROR # 51 :[PARITY] PARITY TRAP FATAL ERROR
              :A PARITY TRAP TO 114 OCCURRED, BUT NO PARITY MODULES COULD BE FOUND
              :WITH AN ERROR BIT (BIT15) SET.

;ERROR # 52 :[NOMM] OPERATOR FATAL ERROR
              :TESTING ABOVE 28K WAS SELECTED, BUT NO MEMORY MANAGEMENT
              :OPTION WAS FOUND.
              :RESET SWITCH OPTIONS AND RESTART AT 200.

;ERROR # 53 :[PARITY] OPERATOR FATAL ERROR
              :PARITY TESTING WAS SELECTED BUT NO PARITY MODULES
              :WERE FOUND.
              :RESET SWITCH OPTIONS AND START AT 200.

```

.REPT 0

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 12
 DZKMAC.P11 18-FEB-77 12:30

[6.3] ERROR HISTORY

LOCATIONS IN MEMORY ARE SET ASIDE TO COLLECT A HISTORY OF THE FAILING BITS IN A PARTICULAR MEMORY BANK. THIS DATA IS COLLECTED FOR EVERY ERROR REGARDLESS OF SWITCH SETTINGS.

NORMALLY THE DATA IS OUTPUT AT THE END OF TESTING, BUT IF CONTROL-C IS TYPED IT IS OUTPUT AT THE END OF THE CURRENT TEST.

THE ERROR HISTORY IS INTENDED TO HIGHLIGHT IF THE ERRORS ARE DUE TO 1 BIT FAILING OR ONLY ADDRESS ERRORS.

ERROR HISTORY FORMAT:

ERROR	BANK	COUNT
----	----	----

WHERE:

ERROR	=	BIT THAT FAILED (NUMBER OF THE FAILING BIT IN DECIMAL I.E. 0-15 WILL BE TYPED OUT OR THE WORDS "ADR ERR" OR "PAR ERR" WILL BE TYPED OUT IF ADDRESS ERROR OR PARITY ERROR WAS SEEN IN THE SPECIFIC BANK OF MEMORY)
BANK	=	4K MEMORY BANK IN WHICH THIS FAILURE WAS SEEN
COUNT	=	A 0 FOR 0 TO 4K, A 1 FOR 4 TO 8K AND SO ON NUMBER OF TIMES THIS MEMORY BANK FAILED. (377 IS MAXIMUM FAILURE COUNT RECORDED.)

[6.4] ERROR RECOVERY

IF THE PROGRAM IS HALTED AFTER REPORTING AN ERROR IT CAN EITHER BE CONTINUED OR RESTARTED AT 200 OR 250 (SEE SEC 4.2). HOWEVER FOR CPU'S THAT DESTROY CONTENTS OF REGISTERS AFTER COMING TO A HALT THE PROGRAM SHOULD ONLY BE RESTARTED.

[7.0] RESTRICTIONS

MEMORY UNDER TEST SHOULD BE CONTIGUOUS. FOR SYSTEMS HAVING NON-CONTIGUOUS MEMORY THE MEMORY BOUNDARIES SHOULD BE DEFINED BY THE OPERATOR. (CONTIGUOUS MEMORY IS DEFINED AS A MEMORY THAT CAN BE BOTH READ AND WRITTEN IN CONSECUTIVE LOCATIONS.)

[8.0] MISCELLANEOUS

[8.1] ADDRESS/BANK RANGES IN OCTAL AND DECIMAL

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 13
 DZKMAC.P11 18-FEB-77 12:30

THIS REFERENCE TABLE CROSS REFERENCES THE MEMORY BANK NO.'S,
 THE RANGE AND THE PAR USED WHEN MEMORY MANAGEMENT IS ENABLED.
 IT IS ALSO USEFUL TO SHOW STARTING ADDRESSES IN A PAR-
 TICULAR 4K BANK.

BANK NO.	DECIMAL RANGE	OCTAL RANGE	[PAGE ADDRESS REGISTER] USED/CONTENT		UNIBUS ADDRESS
0	0 - 4K	000000-017776	0	0000	772340
1	4K - 8K	020000-037776	NOT USED		
2	8K-12K	040000-057776	NOT USED		
3	12K-16K	060000-077776	NOT USED		
4	16K-20K	100000-117776	NOT USED		
5	20K-24K	120000-137776	NOT USED		
6	24K-28K	140000-157776	NOT USED		
7	28K-32K	160000-177776	1	1600	772342
8	32K-36K	200000-217776	2	2000	772344
9	36K-40K	220000-237776	3	2200	772346
10	40K-44K	240000-257776	4	2400	772350
11	44K-48K	260000-277776	5	2600	772352
12	48K-52K	300000-317776	6	3000	772354
13	52K-56K	320000-337776	1	3200	
14	56K-60K	340000-357776	2	3400	
15	60K-64K	360000-377776	3	3600	
16	64K-68K	400000-417776	4	4000	
17	68K-72K	420000-437776	5	4200	
18	72K-76K	440000-457776	6	4400	
19	76K-80K	460000-477776	1	4600	
20	80K-84K	500000-517776	2	5000	
21	84K-88K	520000-537776	3	5200	
22	88K-92K	540000-557776	4	5400	
23	92K-96K	560000-577776	5	5600	
24	96K-100K	600000-617776	6	6000	
25	100K-104K	620000-637776	1	6200	
26	104K-108K	640000-657776	2	6400	
27	108K-112K	660000-677776	3	6600	
28	112K-116K	700000-717776	4	7000	
29	116K-120K	720000-737776	5	7200	
30	120K-124K	740000-757776	6	7400	
31	124K-128K	760000-777776	7	7600	772354

NOTES:

1. THE PAR (PAGE ADDRESS REGISTER) CONTENTS ARE SHOWN IN A TEST THAT SELF SIZES.
 IF THE LIMITS OF TESTING ARE SET BY THE OPERATOR AND
 IF THE BANK IS ABOVE 28K PAR NO. 1 WILL BE SET TO THE
 BEGINNING PAGE. FOR EXAMPLE IF THE TESTING WAS TO
 BEGIN WITH BANK 8 PAR NO. 1 WOULD EQUAL 2000, PAR 2
 WOULD EQUAL 2200 ETC.

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 14
 DZKMAC.P11 18-FEB-77 12:30

[8.2] EXECUTION TIME

HERE ARE SOME TYPICAL EXECUTION TIMES.

LSI-11 AND 4K:= 100 SECS.
 LSI-11 AND 8K:= 5 MINUTES.

[8.2] PASS COUNT AND TEST NO. LOCATIONS

SPASS [406] = PASS COUNT - CLEARED BY START AT 200.

STESTN [404] = CURRENT TEST NO. AND RELOCATION, PARITY FLAGS.

WHERE:
 LOW BYTE = TEST NO.
 IF BIT15 = 1 TEST IS RELOCATED
 IF BIT13 = 1 PARITY UNDER TEST.

[8.4] STACK POINTER

THE STACK STARTS AT 500 WHEN THE PROGRAM IS NOT RELOCATED.
 SAVR6[346] CONTAINS THE STACK STARTING VALUE WHEN THE DIAGNOSTIC IS RELOCATED.

SAVR6 ALSO CONTAINS THE STARTING ADDRESS OF THE PROGRAM WHEN IT IS RELOCATED.

[8.5] POWER FAIL

THE DIAGNOSTIC CAN BE POWER FAILED WITH NO ERRORS. TO USE, START THE TEST AS USUAL AND POWER DOWN THEN UP AT ANY TIME. THE PROGRAM SHOULD TYPE "P" AND CONTINUE TO RUN FROM TEST 0 IN THE SAME STATE (I.E. STATE OF RELOCATION) AS IT WAS BEFORE THE POWER WAS INTERRUPTED. HOWEVER IF THE DIAGNOSTIC WAS IN A MEMORY THAT CAN NOT HOLD DATA WITH THE POWER DOWN THEN THE PROGRAM WILL NOT RECOVER FROM POWER FAIL.

[9.0] PROGRAM DESCRIPTION

[9.1] NARRATIVE FLOW CHART

THE TEST IS LOADED INTO LOCATIONS 0000 - 7744 BUT EXPANDS DEPENDING ON HOW MUCH MEMORY IS UNDER TEST. SEE STEP 6. BELOW FOR A DETAILED EXPLANATION.

THE FOLLOWING NARRATIVE FLOW CHART DESCRIBES MAJOR PROGRAM OPERATION. FOR THE PERSON WHO NEEDS DETAIL THE TAG ASSOCIATED WITH THE OPERATION IS GIVEN IN BRACKETS.

FOR THIS DISCUSSION SWITCH SETTINGS ARE IGNORED AND EVERYTHING IS ASSUMED ENABLED.

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 15
 DZKMAC.P11 18-FEB-77 12:30

1. [START] PRINT "DZKMA-A" TITLE
2. [TSTRP] SAVE DATA FROM LOCATIONS 0-376 INTO 7744-10314.
3. [TSTRP] TEST LOCATIONS 0-376 BY WRITING AND READING 1'S AND 0'S. NOTE THIS IS THE ONLY EXPLICIT TESTING OF THESE LOCATIONS.
4. [SLFSIZ] SIZE MEMORY BY WRITING INTO SUCCEEDING MEMORY LOCATIONS UNTIL TIMEOUT TRAP TO 4 OCCURS. ENABLE MEMORY MANAGEMENT AND SIZE MEMORY ABOVE 28K.
5. [TYPsiz] TYPE MEMORY TEST LIMITS.
6. [SETSTK] SPACE IS SAVED AT THE END OF THE TEST FOR AN ERROR HISTORY. FOR EACH 4K BANK 18 BYTES ARE SAVED IN THE FOLLOWING FORMAT:

```
!ADR ERR!PAR ERR!
!BIT15 !ERR CNT!
!BIT13 !BIT14
!BIT11 !BIT12
!BIT09 !BIT10
!BIT07 !BIT07
!BIT05 !BIT06
!BIT03 !BIT04
!BIT01 !BIT02
!UNUSED !BIT00 !
```

IF GREATER THAN 4K UNDER TEST THE ABSOLUTE LOADER (300 ADDRESSES) IS APPENDED. IF GREATER THAN 4K AND UNDER XXDP CHAIN MODE 5674 (OCTAL) ADDRESSES ARE APPENDED TO THE TEST. THIS SAVES THE XXDP MONITOR, AND ALLOWS THE LOCATIONS OCCUPIED BY XXDP TO BE TESTED.

7. [CLRMEM] CALL "PARITY" ROUTINE AND IF SELECTED, ENABLE ALL PARITY MODULES. "PARMAP" [LOC. 352] CONTAINS A MAP OF PARITY MODULES FOUND. IF MODULE 172336 BIT 15 IS SET, IF #172334 FOUND BIT 14 IS SET ETC..
8. [CLRMEM] CLEAR MEMORY CURRENTLY UNDER TEST
9. [CONT] DISPATCH TO TST0
10. [TST0] EXECUTE TEST 0. SEE SECTION 10 FOR TEST DESCRIPTIONS.
11. [TSTSCP] COMES HERE AFTER EACH TEST AND IF CNTRL-C TYPED THEN GO TO ERROR HISTORY PRINTOUT. IF SR=2000 THEN HALT
 IF SR=40000 THEN LOOP ON TEST DEFINED BY <3:0>

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 16
 DZKMAC.P11 18-FEB-77 12:30

ELSE CONTINUE TO NEXT TEST.

12. [TST1-TST12] EXECUTE TST1-TST12 EACH TIME GOING TO STEP 9.
13. [TST13] TEST 13 IS DIFFERENT FROM TESTS 0-12, BECAUSE IT IS A SMALL PROGRAM ACTUALLY RUNNING IN THE MEMORY UNDER TEST. BEFORE THIS SMALL PROGRAM IS STARTED "TST13 BNK XX" IS TYPED. THIS IS DONE IN CASE THE PROGRAM FAILS. THE USER CAN THEN AT LEAST TELL WHICH BANK OF MEMORY FAILED.
14. [RELOC] THE PROGRAM RELOCATES TO HIGH MEMORY TO TEST THE LOCATIONS IT OCCUPIES. (430-ENDPRG). WHERE "ENDPRG" IS THE CONTENTS OF ENDSTK(306). I.E THE LAST PROGRAM ADDRESS. NOTE "RELOC" IS PRINTED JUST PRIOR TO THE ACTUAL RELOCATION.
15. TESTS 0-13 ARE RUN AS DESCRIBED ABOVE EXCEPT ONLY BANK 0 LOCATIONS 430-ENDPRG ARE TESTED.
16. [RELOER] RELOCATE THE PROGRAM BACK TO LOWER MEMORY.
17. [LOWER] IF CONTROL-C TYPED GO PRINT ERROR HISTORY.
18. [TSTM] IF MEMORY MANAGEMENT SELECTED AND AVAILABLE, RUN TESTS 0-13 ON THE FIRST 24K SLICE ABOVE 28K.
19. [CONTHM] CALL "UPMH" TO UPDATE MEMORY MANAGEMENT PAR REGISTERS TO POINT TO THE NEXT 24K SLICE OF UPPER MEMORY.
20. [MAXADR] REPEAT STEPS 18 + 19 UNTIL ALL MEMORY ABOVE 28K IS TESTED.
21. [ENDPAS] PRINT ERROR HISTORY OF FAILING BITS
22. [SEOP] DISABLE PARITY MODULES.
 PRINT "END PASS #XX"

[9.2] TEST TITLES

SEE THE TEST HEADINGS IN THE LISTING FOR DETAILS ON EACH TEST.

TEST 0: TEST FOR PROPER BANK SELECTION
 TEST 1: CHECK DAT1/DATO LINES
 TEST 2: TEST MEMORY FOR HOLDING DATA AND BYTE SELECTION
 TEST 3: DUAL ADDRESS TEST A
 TEST 4: DUAL ADDRESS TEST B
 TEST 5: MARCHING 1'S AND 0'S
 TEST 6: CELLS' VOLATILITY TEST

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 17
 DZKMAC.P11 18-FEB-77 12:30

TEST 7: SHIFTING DIAGONAL
 TEST 10: READ RECOVERY GALLOPING TEST THROUGH EVERY 64TH CELL
 TEST 11: READ RECOVERY LONG GALLOPING/FAST GALLOPING TEST
 TEST 12: WORST CASE TESTING FOR CORE MEMORY
 TEST 13: WRITE RECOVERY TEST

[110.0] RXDP & ACT11 & APT OPERATION

RXDP CHAIN MODE

OPERATION IS IDENTICAL TO STAND ALONE EXCEPT:

1. NO "DZKMA-C" TITLE IS PRINTED.
2. NO TEST 13 PRINTOUTS SUCH AS "TST13 BNK 00".
3. THE PROGRAM ALWAYS HALTS ON ERROR.
4. AT THE END OF TEST (SENDAD) CONTROL IS RETURNED TO THE RXDP CHAIN MONITOR VIA LOCATION 42.

ACT11

OPERATION IS IDENTICAL TO STAND ALONE EXCEPT:

1. NO PRINTOUTS EXCEPT ERROR PRINTOUTS.
2. THE PROGRAM ALWAYS HALTS ON ERROR.
3. AT THE END OF TEST (SENDAD) CONTROL IS RETURNED TO THE ACT11 MONITOR VIA LOCATION 42.

APT

OPERATION IS SIMILAR TO STAND ALONE EXCEPT:

1. THE SOFTWARE SWITCH REGISTER BECOMES LOCATION 422 (\$SWREG).
2. AUTO SIZING CAN BE INHIBITED BY SETTING BIT 7 OF BYTE LOCATION 421 (\$ENV).
3. ALL PRINTOUTS CAN BE INHIBITED BY SETTING BIT 5 OF BYTE LOCATION 421 (\$ENV).
4. ALL ERRORS CAUSE LOCATION 400 (\$MSGTY) TO BE SET = 0001 AND THE PROGRAM HALTS AT LOCATION 6214 (\$FATHLT). LOCATION 402 (\$FATAL) CONTAINS THE ERROR NO. IN THE LOW BYTE AND THE FAILING MEMORY BANK NO. IN THE HIGH BYTE.

APT MANAGER INFORMATION

THE FOLLOWING IS AN EXAMPLE SCRIPT TO TEST A 4K MEMORY.
 IT IS RECOMMENDED THAT DIFFERENT SCRIPTS BE USED FOR DIFFERENT MEMORY SIZES TO SAVE AUTO SIZING TIME.

THE EXAMPLE ASSUMES YOU ARE LOGGED INTO THE APT MONITOR.

READY

G02

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 18
DZKMAC.P11 18-FEB-77 12:30

RUN APPLU
APT 11 PAPER TAPE PROGRAM LOAD UTILITY

THE FOLLOWING COMMANDS ARE VALID

ED EDIT A PROGRAM
LI LIST A PROGRAM

COMMAND: ED
PROGRAM NAME TO EDIT: EXAMPL
DO YOU WANT TO LOAD A NEW REV OF THE PROGRAM(Y/N)? N
FIRST PASS RUN TIME IN SECONDS <110>:
LONGEST TEST TIME IN SECONDS <10>:
ADDITIONAL RUN TIME IN SECONDS <0>:
WHICH ETABLE DO YOU WISH TO EDIT? A
SOFTWARE ENVIRONMENT<000>: 1
ENVIRONMENTAL MODE<000>: 240
SWITCH 1 <000000>:
SWITCH 2 <000000>:
CPU OPTIONS<0000>:
MEMORY TYPE 1 <000>:
MAXIMUM ADDRESS<00000000>:
MEMORY TYPE 2 <000>:
MAXIMUM ADDRESS<00000000>:
MEMORY TYPE 3 <000>:
MAXIMUM ADDRESS<00000000>:
MEMORY TYPE 4 <000>: 1
MAXIMUM ADDRESS<00000000>: 17776
WHICH ETABLE DO YOU WISH TO EDIT?
COMMAND: OFF

.ENDR

H02

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 19
DZKMAC.P11 18-FEB-77 12:30

.ABS
.NLIST MD,MC,CND

985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025

```
.LIST ME,BIN,SEQ,LOC
.TITLE DZKMA
; *COPYRIGHT (C) JANUARY 1976
; *DIGITAL EQUIPMENT CORP.
; *MAYNARD, MASS. 01754
; *
; *PROGRAM BY PERVEZ ZAKI
; *
; *THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
; *PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
; *
$SWR=160000      ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYP0UT
```

;;TRAP CATCHER OF .+2 AND HALT FOR 0-776 LOCATIONS

SCOPE =NOP

000240
000042 000042
000000

.=42
.WORD 0 ;FOR ACT/XXDP

.SBTTL ACT11 HOOKS

;;*****
;HOOKS REQUIRED BY ACT11

000044
000046
000046 000156
000052
000052 040000
000044

```
$SVPC=.      ;SAVE PC
.=46
$ENDAD      ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
.=52
;WORD 40000  ;;2)SET LOC.52 TO 40000
;=$SVPC      ;; RESTORE PC
```

000070 000070
000070 012737 000136 000024 PWRDN: MOV #PWRUP,0#24
000000 HALT

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 20
 DZKMAC.P11 18-FEB-77 12:30 ACT11 HOOKS

```

1026
1027
1028      000104      . = 104
1029      : GET HERE IF AN ILLEGAL TRAP TO LOC. 4 OCCURRED.
1030 000104 005237 000400  BUSER: INC      2#MSGTY      : TELL APT FATAL ERROR#000
1031 000110 000000      : HALT      : *ERROR* TRAP TO LOC. 4 OCCURRED.
1032 000112 000000      : HALT      : IN CASE CONTINUE PRESSED.
1033      : 114 AND 116 ARE RESERVED FOR PARITY TRAP VECTORS.  SETUP IN
1034      : ROUTINE "BEGIN".
1035      . = 120
1036
1037
1038
1039      : * WRITE MEMORY BACKGROUND
1040      : -----
1041      :
1042      : THIS ROUTINE IS USED TO WRITE THE MEMORY BACKGROUND TO
1043      : THE VALUE STORED AT LOCATION BAKPAT.  THE ROUTINE ASSUMES
1044      : THAT R4 IS POINTING TO THE LOWEST LOCATION AND R5 TO THE
1045      : HIGHEST LOCATION TO BE WRITTEN.  THE PROGRAM LEAVES THE
1046      : SUBROUTINE WITH R0 CONTAINING THE CONTENTS OF BAKPAT.
1047      :
1048
1049 000120 010401      WRTMEM: MOV      R4,R1      ; SET R1 TO LOWEST LOCATION UNDER TEST
1050 000122 013700 000316  : MOV      2#BAKPAT,R0      ; LOAD R0 WITH THE CONTENTS OF LOCATION BAKPAT
1051 000126 010021      2S:  MOV      R0,(R1)+    ; STARTING FROM THE LOWEST LOCATION WRITE THE
1052 000130 020105      : CMP      R1,R5      ; MEMORY TO BACK GROUND PATTERN
1053 000132 103775      : BLO      2S
1054 000134 000207      : RTS      PC      ; RETURN FROM THE SUBROUTINE
1055
1056
1057 000136 013706 000350  PWRUP: MOV      2#SAVR6,SP      ; RESTORE STACK POINTER
1058 000142 012700 006066  : MOV      #PNTMES-BEGIN,R0
1059 000146 060600      : ADD      SP,R0      ; GET THE INDIRECT ADDRESS OF LOCATION TPCRLF
1060      : JSR      PC,(R0)      ; RELATIVE TO LOCATION OF DIAGNOSTIC IN THE CORE
1061 000150 004710      : .ASCIZ  /P/      ; GO TO THE TYPE ROUTINE AND TYPE CR, LF AND A "P"
1062 000152 000120      : .EVEN
1063
1064
1065 000154 000411      BR      START
1066
1067      : * SERVICE XXDP/ACT11
1068 000156 004710  SENDAD: JSR      PC,(R0)      ; RETURN TO ACT11/XXDP MONITOR
1069 000160 000240      : NOP
1070 000162 000240      : IF QUICK VERIFY=RESET ELSE NOP
1071 000164 000240      : IF QUICK VERIFY=CLR #-1 ELSE INC #0
1072 000166 000430      : IF QUICK VERIFY=BR -4 ELSE NOP
1073      : BR      RESTRT      ; REPEAT TEST UNDER ACT11/XXDP
1074
1075 000176 000176      . = 176
1076 000176 000000  SWREG: .WORD  0
1077
1078      : *****
1079      : SBTTL  START AND RESTART ROUTINES
1080      : *      RESTART AT 200 TO CLEAR APT TABLES
1081      : *****

```


J02

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 21
 DZKMAC.P11 18-FEB-77 12:30 START AND RESTART ROUTINES

1082	000200	013706	000350	START:	MOV	2:SAVR6, SP	; SETUP STACK POINTER.
1083	000204	012703	000412		MOV	8:UNIT, R3	; CLEAR THE APT MAILBOX FROM SMAIL TO SDEVCT
1084	000210	005043		1S:	CLR	-(R3)	; CLEAR A MAILBOX LOCATION
1085	000212	022703	000400		CMF	8:MAIL, R3	; DONE?
1086	000216	001374			BNE	1S	; BRANCH IF NO
1087	000220	105737	000042		TSTB	2:42	; ACT11 MODE?
1088	000224	001011			BNE	RESTR	; BRANCH IF YES
1089	000226	105737	000405		TSTB	2:STESTN+1	; ARE WE RELOCATED?
1090	000232	100406			BMI	RESTR	; BR IF YES- SINCE TPCRLF IS RELOCATED ALSO-
1091	000234	004767	006320		JSR	PC, TPCRLF	; PRINT TITLE
1092	000240	055104	046513	026501	.ASCIZ	/DZKMA-C/	
1093	000246	000103					
1094					.EVEN		
1095							
1096	000250	012704	007744	RESTR:	MOV	8:ENDPRG, R4	; LOAD R4 WITH THE ADDRESS OF THE END OF THE PROGRAM
1097	000254	012703	000346		MOV	8:SAVR5, R3	; CAUSE R3 TO POINT TO THE LOCATION SAVR5
1098	000260	012305			MOV	(R3)+, R5	; RESTORE R5
1099	000262	012306			MOV	(R3)+, SP	; AND RESTORE R6 JUST IN CASE IT IS A RESTART
1100	000264	010600			MOV	SP, R0	; PLACE THE STARTING ADDRESS OF THE TEST IN R0
1101	000266	012746	000340		MOV	8:40, -(SP)	; SET HIGH PRIORITY FOR RTI
1102	000272	010046			MOV	R0, -(SP)	
1103	000274	000002			RTI		; GO TO "START"-MAY BE RELOCATED.
1104							; IF RELOCATED SEE LOCATION SAVR6 FOR START.
1105							
1106							
1107							
1108							
1109							
1110							
1111							
1112							
1113							
1114							
1115							
1116							
1117							
1118		000276					
1119		000024					
1120	000024	000200					
1121		000044					
1122	000044	000276					
1123		000276					
1124							
1125							
1126							
1127							
1128	000276						
1129	000276	000000					
1130	000300	000400					
1131	000302	000012					
1132	000304	000156					
1133	000306	000000					
1134	000310	000024					
1135							
1136							
1137		000405					

REL=\$TESTN+1 ; IT WILL BE 0 IF THE PROGRAM IS IN THE LOWER

.SBTTL APT PARAMETER BLOCK

```

;*****
;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
;*****
.SX=.      ;SAVE CURRENT LOCATION
.=24      ;SET POWER FAIL TO POINT TO START OF PROGRAM
200       ;FOR APT START UP
.=44      ;POINT TO APT INDIRECT ADDRESS PNTR.
$APTHDR   ;POINT TO APT HEADER BLOCK
.=.SX     ;RESET LOCATION COUNTER
;*****
;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
;INTERFACE SPEC.

```

```

$APTHD:
$HIBTS: .WORD 0      ;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
$MBADR: .WORD $MAIL  ;ADDRESS OF APT MAILBOX (BITS 0-15)
$STSM:  .WORD 10     ;RUN TIM OF LONGEST TEST
$PASTM: .WORD 110    ;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
$UNITM: .WORD        ;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
        .WORD SETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)

```

```

;CORE. BIT 7 OF THE BYTE WILL BE SET IF THE
;PROGRAM IS IN A RELOCATED STATE AND BIT 5
;WILL BE SET IF PARITY BITS ARE BEING TESTED

;THIS BYTE IS USED TO DETERMINE IF MEMORY
;MANAGEMENT IS AVAILABLE OR NOT

;THIS BYTE IS USED TO DETERMINE IF THE
;TYPE OUT OF ERROR HAS BEEN ENABLED OR NOT

;THIS BYTE DETERMINES IF THE PROGRAM HAS FOUND
;A PARITY ERROR

;THIS BYTE IS USED TO DETERMINE IF THE
;PROGRAM HAS ENCOUNTERED ADDRESS ERROR

;LOWER BYTE OF THIS WORD GIVES THE PASS NUMBER FOR
;THE SPECIFIC TEST WHEREAS THE UPPER BYTE
;HAS BEEN USED BY DIFFERENT TEST FOR DIFFERENT PURPOSES

BANK UNDER TEST FOR "TST BNK XX" PRINTOUT.

;THIS BYTE DETERMINES THE NUMBER OF WORDS
;TO BE TYPED
;THIS LOCATION IS USED TO SAVE THE CHARACTER
;HIT BY THE OPERATOR
;ALSO IS USED AS TEMP IN ROUTINE $GTSIZ.

;BACKGROUND PATTERN WRITTEN TO MEMORY.

;HOLDS LOWEST TEST ADDRESS WHEN RELOCATED.

*****
MITS SET BY OPERATOR
;HOLDS BITS 17:16 OF LOW TEST ADDRESS
;HOLDS BITS 15:0 OF LOW TEST ADDRESS

```


L02

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 23
 DZKMAC.P11 18-FEB-77 12:30 APT PARAMETER BLOCK

1194				
1195	000330	000000	HIGHTWO: 0	; HOLDS BITS 17:16 OF HIGH TEST ADDRESS
1196	000332	037776	HIGHADD: 37776	; HOLDS BITS 15:0 OF HIGH TEST ADDRESS
1197			;*****	
1198				
1199	000334	000000	SHIMAX: 0	; HOLDS BITS 17:16 OF MAXIMUM AVAILABLE MEMORY
1200	000336	017776	SIMAX: 17776	; HOLDS BITS 15:0 OF MAXIMUM AVAILABLE MEMORY
1201				
1202	000340	000000	MAXMEM: .WORD	; MAXIMUM CURRENT VIRTUAL MEMORY UNDER TEST
1203				
1204	000342	000000	SAVMAX: .WORD	
1205	000344	000000	SAVR4: .WORD	
1206	000346	000000	SAVR5: .WORD	
1207				
1208			;* SAVR6 POINTS TO WHERE THE PROGRAM STARTS EVEN WHEN RELOCATED.	
1209	000350	000500	SAVR6: .WORD	BEGIN ; CONTAINS START ADDRESS WHEN RELOCATED ALSO.
1210	000352	000000	PARMAP: 0	; MAP OF PARITY MODULES UNDER TEST.
1211	000354	000000	SAVLOC: 0	; TEST 6 STORES ERROR INFO HERE
1212	000356	000000	PARSP: 0	; SAVE SP DURING PARITY ERROR TRAP.
1213	000360	000000	PARPS: 0	; SAVE PSW DURING PARITY ERROR TRAP.
1214				; NOTE-PARSP +PARPS ARE NEEDED SINCE THERE IS
1215				; IS NOT ENOUGH ROOM ON THE STACK (500-452) AND
1216				; SO THE STACK MUST BE RESET IN THE PARERR ROUTINE.
1217				; IN THIS CRUDE FASHION.
1218				
1219				
1220				; *364-400 IS USED AS A STACK AREA BY ERRCHK ROUTINE FOR ERROR HISTORY PRINTOUT

MO2

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 24
DZKMAC.P11 18-FEB-77 12:30 APT PARAMETER BLOCK

```

1221      000400
1222      .SBTTL      =400
1223      .SBTTL      APT MAILBOX-ETABLE
1224      ;:*****
1225      .EVEN
1226      $MAIL:      APT MAILBOX
1227      $MSGTY: .WORD  AMSCTY  ;:MESSAGE TYPE CODE
1228      $FATAL: .WORD  AFATAL  ;:FATAL ERROR NUMBER
1229      $TESTN: .WORD  ATESTN  ;:TEST NUMBER
1230      $PASS:  .WORD  APASS   ;:PASS COUNT
1231      $DEVCT: .WORD  ADEVCT  ;:DEVICE COUNT
1232      $UNIT:  .WORD  AUNIT   ;:I/O UNIT NUMBER
1233      $MSGAD: .WORD  AMSGAD  ;:MESSAGE ADDRESS
1234      $MSGLG: .WORD  AMSGLG  ;:MESSAGE LENGTH
1235      $ETABLE:  ;:APT ENVIRONMENT TABLE
1236      $ENV:     .BYTE  AENV   ;:ENVIRONMENT BYTE
1237      $ENVN:    .BYTE  AENVN  ;:ENVIRONMENT MODE BITS
1238      $SWREG:   .WORD  ASWREG  ;:APT SWITCH REGISTER
1239      $USWR:    .WORD  AUSWR  ;:USER SWITCHES
1240      $CPUOP:   .WORD  ACPUOP  ;:CPU TYPE, OPTIONS
1241      ;:BITS 15-11=CPU TYPE
1242      ;:      11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05
1243      ;:      11/70=06, P00=07, Q=10
1244      ;:BIT 10=REAL TIME CLOCK
1245      ;:BIT 9=FLOATING POINT PROCESSOR
1246      ;:BIT 8=MEMORY MANAGEMENT
1247      $MAMS1: .BYTE  AMAMS1  ;:HIGH ADDRESS, M.S. BYTE
1248      $MTYP1: .BYTE  AMTYP1  ;:MEM. TYPE, BLK#1
1249      ;:MEM. TYPE BYTE -- (HIGH BYTE)
1250      ;:      900 NSEC CORE=001
1251      ;:      300 NSEC BIPOLAR=002
1252      ;:      500 NSEC MOS=003
1253      $MADR1: .WORD  AMADR1  ;:HIGH ADDRESS, BLK#1
1254      ;:MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE
1255      $MAMS2: .BYTE  AMAMS2  ;:HIGH ADDRESS, M.S. BYTE
1256      $MTYP2: .BYTE  AMTYP2  ;:MEM. TYPE, BLK#2
1257      $MADR2: .WORD  AMADR2  ;:MEM. LAST ADDRESS, BLK#2
1258      $MAMS3: .BYTE  AMAMS3  ;:HIGH ADDRESS, M.S. BYTE
1259      $MTYP3: .BYTE  AMTYP3  ;:MEM. TYPE, BLK#3
1260      $MADR3: .WORD  AMADR3  ;:MEM. LAST ADDRESS, BLK#3
1261      $MAMS4: .BYTE  AMAMS4  ;:HIGH ADDRESS, M.S. BYTE
1262      $MTYP4: .BYTE  AMTYP4  ;:MEM. TYPE, BLK#4
1263      $MADR4: .WORD  AMADR4  ;:MEM. LAST ADDRESS, BLK#4
1264      $SETEND:
1265      .MEXIT
1266      ;:*****
1267      .SBTTL      BEGIN OF AREA TESTED (+20) WHEN PROGRAM RELOCATES.
1268

```


N02

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 25

DZKMAC.P11 18-FEB-77 12:30

BEGIN OF AREA TESTED (+20) WHEN PROGRAM RELOCATES.

```

1269
1270 000450 177570
1271
1272 000500 000500
1273 000500 010706
1274
1275 000502 005746
1276 000504 010637 000350
1277 000510 012737 000070 000024
1278 000516 005037 000300
1279 000522 005037 000314
1280 000526 012700 000114
1281 000532 012710 005456
1282 000536 060720
1283 000540 012710 000340
1284 000544 105737 000405
1285 000550 100002
1286 000552 000167 000542
1287
1288 000556 005737 000406
1289 000562 001402
1290 000564 000167 000374
1291 000570 012704 007744
1292 000574 012700 000377
1293 000600 010037 000316
1294 000604 005001
1295 000606 012124
1296 000610 020127 000400
1297 000614 103774
1298 000616 005741
1299 000620 010011
1300
1301 000622 020011
1302 000624 001403
1303
1304 000626 004767 005304
1305 000632 000001
1306
1307 000634 000300
1308 000636 001370
1309 000640 005701
1310 000642 001365
1311 000644 012701 000400
1312 000650 014441
1313 000652 005701
1314 000654 001375
1315 000656 012700 000006
1316 000662 012710 000340
1317 000666 012740 000700
1318 000672 005777 177552
1319 000676 000404
1320 000700 022626
1321 000702 012737 000176 000450
1322
1323
1324 000710 105737 000420

```

SWR: 177570 ;CHANGES TO SWREG IF NO HARDWARE SWITCH REGISTER

BEGIN: . =500
MOV PC, SP ;SET UP STACK POINTER TO EQUAL BEGIN ADDRESS

TST -(SP)
MOV SP, 2#SAVR6 ;SAVE SP FOR FUTURE USE
MOV 2#PSW, 2#24 ;PREPARE FOR ANY FUTURE POWER DOWN
CLR 2#SPERR
CLR 2#TYPCNT
MOV 2#114, RO ;PREPARE TO SETUP PARITY TRAP VECTOR
MOV 2#PARERR, -6, (RO)
ADD PC, (RO)+ ;TO PARERR
MOV 2#340, (RO) ;AND PSW OF 340
TSTB 2#REL ;IS THIS CODE RELOCATED?
BPL ONEPAS ;BRANCH IF NO
JMP TSTREL ;THIS CODE IS RELOCATED SO GET TEST SIZE.

ONEPAS: TST 2#SPASS ;IS THIS THE FIRST PASS?
BEQ TSTRP ;BRANCH IF YES (TEST TRAP CATCHER ADDRESSES)
JMP SETSTK ;GET THE TEST SIZE
TSTRP: MOV 2#ENDPRG, R4 ;LOAD R4 WITH THE ADDRESS OF THE END OF THE PROGRAM
MOV 2#377, RO
MOV RO, 2#BAKPAT
CLR R1
2\$: MOV (R1)+, (R4)+ ;SAVE FROM 0000 TO BEGIN-30 AT END OF PROGRAM FOR NOW
CMP R1, 2#MAIL
BLO 2\$
3\$: TST -(R1) ;PREPARE TO TEST THE TRAP VECTORS
4\$: MOV RO, (R1) ;CHECK THE TRAP VECTORS FOR THE CAPABILITY
;OF HOLDING 0'S & 1'S
CMP RO, (R1) ;IS THE DATA OK?
BEQ 6\$;BRANCH IF YES

JSR PC, FATERR ;*ERROR* REPORT ERROR MESSAGE AND HALT AT FATHLT
1 ;*****ERROR NUMBER 1*****

6\$: SWAB RO
BNE 4\$
TST R1 ;IF WE HAVE NOT REACHED THE LOWEST MEMORY LOCATION
BNE 3\$;THEN REPEAT FROM 3\$
MOV 2#MAIL, R1
8\$: MOV -(R4), -(R1) ;RESTORE TRAP CATCHER ETC.
TST R1
BNE 8\$
SETSWR: MOV 2#6, RO
MOV 2#340, (RO) ;SET UP TIME OUT TRAP PSW
MOV 2#45, -(RO) ;AND THE RETURN ADDRESS
2\$: TST 2#SWR ;DOES THE SWITCH REGISTER POINTED BY SWR EXIST ?
BR 5\$;BRANCH IF YES
4\$: CMP (SP)+, (SP)+ ;RESTORE THE STACK POINTER
MOV 2#SWREG, 2#SWR ;AND PLACE THE ADDRESS OF THE SWITCH REGISTER
;DESIGNED FOR THE COMPUTERS NOT HAVING HARDWARE
;SWITCH REGISTER AND RUNNING STAND ALONE
5\$: TSTB 2#SENV ;RUNNING UNDER APT?

B03

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 26
 DZKMAC.P11 18-FEB-77 12:30

BEGIN OF AREA TESTED (+20) WHEN PROGRAM RELOCATES.

```

1325 000714 001403          BEQ  APTSIZ      ;BRANCH IF NO
1326 000716 012737 000422 000450      MOV  #SSWREG,2#SWR ;SET SWR EQUAL TO APT SWITCH REGISTER.
1327
1328
1329
1330
1331      ;APTSIZ- THIS ROUTINE WILL SEARCH THE APT MEMORY ETABLE AND WHEN
1332      ;A NON ZERO TYPE IS FOUND WILL SETUP TO TEST TO GIVEN HIGH ADDRESS.
1333      ;IF APT DEFINES SIZE THE LOW TEST ADDRESS MUST=00000.(DUE TO ETABLE FORMAT)
1334      ;FLOW;
1335      ;IF BLOCK 4 (OR 3,2,1) TYPE NON ZERO THEN GET APT HIGH ADDRESS AND EXIT.
1336      ;ELSE SEND ERROR #3
1337      ;NOTE; THE MEMORY TYPE IS IGNORED SINCE ALL TESTS ARE RUN REGARDLESS OF MEMORY TYPE.
1338
1339 000724 012703 000340      APTSIZ: MOV  #MAXMEM,R3      ;POINT R3 TO MAXMEM.
1340 000730 013737 000330 000334      MOV  2#HIGHTWO,2#SHIMAX ;IN CASE NO SELF SIZING DONE.
1341 000736 013737 000332 000336      MOV  2#HIGHADD,2#SMAXH ;IN CASE NO SELF SIZING DONE.
1342 000744 105737 000421      TSTB  2#SENVH ;DOES APT ALLOW SELF SIZING?
1343 000750 100021      BPL  TRYSR ;BRANCH IF YES
1344
1345 000752 012701 000451      1S:  MOV  #SHTYP4+4,R1 ;POINT R1 TO BLOCK TYPE 4(+4)
1346 000756 162701 000004      SUB  #4,R1 ;POINT R1 TO NEXT BLOCK TYPE.
1347 000762 105711      TSTB  (R1) ;IS THE BLOCK TYPE NON ZERO?
1348 000764 001006      BNE  2S ;BRANCH IF YES (MEMORY EXISTS)
1349 000766 020127 000431      CMP  R1,#SHTYP1 ;ALL APT BLOCK TYPES BEEN CHECKED?
1350 000772 101371      BHI  1S ;BRANCH IF NO
1351
1352 000774 004767 005136      JSR  PC,FATERR ;#ERROR# REPORT ERROR MESSAGE AND HALT AT FATHLT
1353 001000 000002      2S:  JSR  2 ;*****ERROR NUMBER 2*****
1354
1355 001002 004767 006302      2S:  JSR  PC,GETADR ;GO SET MAXIMUM APT ADDRESS INTO SMAXH + SHIMAX
1356 001006 004767 006276      JSR  PC,GETADR ;GO SET MAXIMUM APT ADDRESS INTO HIGHADD+HIGHTWO
1357 001012 000446      BRTPSZ: BR  TYPsiz ;TYPE THE SIZE OF MEMORY UNDER TEST
1358
1359 001014 032777 000100 177426 TRYSR: BIT  #100,2#SWR ;USER DEFINED MEMORY TEST BOUNDARIES??
1360 001022 001042      BNE  TYPsiz ;BRANCH IF YES (DON'T SIZE MEMORY)
1361
1362
1363
1364
1365
1366 001024 010401          SLFSIZ: MOV  R4,R1 ;SETUP R1 AND R4 TO THE LOWEST ADDRESS OF MEMORY
1367 001026 012710 001042      MOV  #4S,(R0) ;SET UP RETURN ADDRESS FROM TIME OUT TRAP TO 4S
1368 001032 011111      2S:  MOV  (R1),(R1) ;WRITE A MEMORY LOCATION INTO ITSELF AND TRAP IF NONEXIS
1369
1370 001034 062701 000002      ADD  #2,R1 ;ADD 2 TO THE ADDRESS POINTER
1371 001040 000774      BR  2S ;KEEP ON SIZING UP THE MEMORY UNTIL
1372 ;NOM TRAP (TIME OUT TRAP) IS ENCOUNTERED
1373
1374 001042 022626      4S:  CMP  (SP)+,(SP)+ ;RESTORE THE STACK POINTER
1375 001044 004767 005770      JSR  PC,MEHMG ;SERVICE MEMORY MANAGEMENT IF IT IS AVAILABLE
1376 ;AND IF IT HAS TO BE TESTED
1377 001050 105737 000276      TSTB  2#MMAVA ;SEE IF MEMORY MANAGEMENT HAS TO BE TESTED
1378 001054 001414      BEQ  12S ;IF NO MEM. MANG. THEN GO TO 12S
1379 001056 012710 001070      6S:  MOV  #8S,(R0) ;SET UP THE RETURN ADDRESS FROM TRAP TO 8S
1380 001062 012701 020000      MOV  #20000,R1 ;BEGIN CHECKING MEMORY ABOVE 28K

```


DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 27
 DZKMAC.P11 18-FEB-77 12:30

BEGIN OF AREA TESTED (+20) WHEN PROGRAM RELOCATES.

1381	001066	000761			BR	25			
1382	001070	022626		85:	CMP	(SP)+ (SP)+			:RESTORE STACK POINTER
1383	001072	022701	160000		CMP	#160000,R1			:IF R1 DID NOT READ ALL THE LOCATIONS POINTED BY
1384									:PAGE ADDRESS REGISTER 6 THEN IT HAS REACHED THE
1385									:MAXIMUM AVAILABLE MEMORY
1386	001076	001003			BNE	125			:IN WHICH CASE GO TO 125
1387	001100	004767	006106		JSR	PC,UPMM			:OTHERWISE GO TO UPDATE MEM. MANG. REGISTERS
1388	001104	000764			BR	65			
1389	001106	024341		125:	CMP	-(R3),-(R1)			:CAUSE R3 TO POINT TO LOCATION SHAXH AND R1
1390									:TO THE MAXIMUM AVAILABLE MEMORY
1391	001110	004767	006104		JSR	PC,PUTADR			:GO TO THE SUBROUTINE TO PLACE THE ADDRESS IN R1
1392									:AT LOCATIONS SHAXH AND SHIMAX
1393	001114	024343			CMP	-(R3),-(R3)			:MAKE R3 POINT TO HIGHADD
1394	001116	004767	006076		JSR	PC,PUTADR			:PLACE THE ADDRESS IN R1 AT LOCATIONS HIGHADD
1395									:AND HIGHTWO
1396	001122	005743			TST	-(R3)			
1397	001124	005043			CLR	-(R3)			:CLEAR THE LOCATION LOWADD
1398	001126	005043			CLR	-(R3)			:AND LOWTWO
1399	001130	012720	000104	TYPSTZ:	MOV	#BUSER,(R0)+			:SET UP VECTOR FOR ANY FUTURE TRAP
1400	001134	010403			MOV	R4,R3			:SET R3 TO POINT TO THE LOWEST AVAILABLE MEMORY
1401									:LOCATION
1402	001136	012701	000324		MOV	#LOWTWO,R1			
1403	001142	004767	005400		JSR	PC,PCRLF			:TYPE CR/LF
1404	001146	004767	005546		JSR	PC,OCTTYP			:TYPE LOW TEST ADDRESSE (LOWTWO+LOWADD)
1405	001152	004767	005302	TYPHEM:	JSR	PC,STYPE			:TYPE "--"
1406	001156	000055			.ASCIZ	/- /			
1407					.EVEN				
1408	001160	004767	005534		JSR	PC,OCTTYP			:TYPE HIGHEST TEST ADDRESS (HIGHTWO+HIGHADD)
1409	001164	012703	000330	SETSTK:	MOV	#HIGHTWO,R3			:MAKE R3 POINT TO THE HIGH ORDER BITS OF TOP ADDRESS
1410	001170	004767	006130		JSR	PC,\$GTSIZ			:GET THE BITS 13-17 OF THE TOP ADDRESS
1411									:PLACED IN BITS 0-4 OF R2
1412	001174	010401			MOV	R4,R1			:SET R1 TO LOWEST TEST ADDRESS
1413									
1414	001176	062704	000022	45:	ADD	#18.,R4			:APPEND THE ERROR STACK FOR THE MEMORY UNDER
1415									:TEST TO THE END OF THE PROGRAM
1416	001202	005302			DEC	R2			
1417	001204	002374			BGE	45			
1418	001206	010437	000310		MOV	R4,#ENDSTK			:SAVE THE ADDRESS OF THE END OF THE ERROR STACK
1419	001212	005021		65:	CLR	(R1)+			:CLEAR THE ERROR STACK
1420	001214	020104			CMP	R1,R4			
1421	001216	101775			BLOS	65			
1422	001220	012737	157776 000340		MOV	#157776,#MAXMEM			:SET MAXMEM TO MAXIMUM VIRTUAL ADDRESS
1423	001226	005723			TST	(R3)+			:TESTING MEMORY MANAGEMENT?
1424	001230	001004			BNE	SAVLDR			:BRANCH IF YES (GO SAVE LOADERS AT TOP OF VIRTUAL MEMORY
1425	001232	021300			CMP	(R3),R0			:IS THE VIRTUAL ADDRESS ABOVE 157776?
1426	001234	103002			BHIS	SAVLDR			:BRANCH IF YES (GO SAVE LOADERS)
1427	001236	011363	000002		MOV	(R3),2(R3)			:OTHERWISE MAKE THE CONTENTS OF LOCATION MAXMEM
1428									:EQUAL TO THE MAXIMUM AVAILABLE MEMORY
1429									:AND FALL INTO SAVE LOADERS.
1430									
1431	001242	004767	006136	SAVLDR:	JSR	PC,CLAMM			:DISABLE THE MEMORY MANAGEMENT UNIT
1432	001246	005723			TST	(R3)+			:MAKE R3 TO POINT TO THE LOCATION MAXMEM
1433	001250	011305			MOV	(R3),R5			:R5 CONTAINS THE ADDRESS OF MAXIMUM AVAILABLE MEM.
1434									
1435									
1436									

;IF ONLY 4K BEING TESTED DON'T SAVE LOADERS

1437	001252	020527	017776		CMP	R5, #17776	: ONLY TESTING 4K MAX?	
1438	001256	103416			BLO	4S	: BRANCH IF YES (DON'T SAVE LOADERS)	
1439								
1440	001260	162705	000276	3S:	SUB	#276, R5	: PREPARE TO SAVE 300 BYTES OF THE LOADERS	
1441	001264	005737	000042		TST	2#42	: IS THE PROGRAM RUNNING UNDER ACT OR XXDP ?	
1442	001270	001406			BEQ	2S	: IF NOT THEN GO TO 2S	
1443	001272	023737	000042	000046	CMP	2#42, 2#46	: ARE WE RUNNING UNDER XXDP CHAIN MODE?	
1444	001300	001402			BEQ	2S	: BRANCH IF NO	
1445	001302	162705	005674		SUB	#(1502.*2), R5	: SAVE 1500. WORDS FOR XXDP CHAIN MODE	
1446	001306	012524		2S:	MOV	(R5)+, (R4)+	: SAVE LOADER	
1447	001310	020513			CMP	R5, (R3)		
1448	001312	101775			BLOS	2S		
1449	001314	012323		4S:	MOV	(R3)+, (R3)+	: SAVE THE CONTENTS OF LOCATION MAXMEM IN SAVMAX	
1450	001316	010423			MOV	R4, (R3)+	: AND THE CONTENTS OF R4 AT SAVR4	
1451								
1452	001320	010537	000346		TSTREL: MOV	R5, 2#SAVR5	: SAVE HIGHEST VIRTUAL ADDRESS+2	
1453	001324	004767	006054		TSTSIZ: JSR	PC, CLMMH	: GO TO DISABLE MEMORY MANAGEMENT UNIT	
1454	001330	005745			TST	-(R5)	: SET R5 BACK TO HIGHEST VIRTUAL ADDRESS	
1455	001332	012703	000324	1S:	MOV	#LOWTWO, R3	: PREPARE TO LOAD R4 AND R5 WITH THE MEMORY BOUNDRIES	
1456	001336	005723			TST	(R3)+	: IF THE BITS 16, 17 OF THE LOWEST LOCATION UNDER	
1457							: TEST ARE NON ZERO	
1458	001340	001003			BNE	2S	: THEN GO TO 2S	
1459	001342	021327	157776		CMP	(R3), #157776	: IF THE LOWEST LOCATION UNDER TEST IS HIGHER THAN	
1460							: 157776 THEN GO TO TEST MEMORY MANAGEMENT	
1461	001346	103411			BLO	4S		
1462	001350	032777	010000	177072	2S:	BIT	#10000, 2SWR	: IS MEMORY MANAGEMENT SELECTED?
1463	001356	001003			BNE	3S	: YES ALL IS WELL	
1464	001360	004767	004552		JSR	PC, FATERR	: *ERROR* REPORT ERROR MESSAGE AND HALT AT FATHLT	
1465	001364	000003				3	: *****ERROR NUMBER 3*****	
1466								
1467	001366	000167	003526	3S:	JMP	TSTMH	: GO TO TEST MEMORY MANAGEMENT	
1468	001372	020423		4S:	CMP	R4, (R3)+		
1469	001374	103002			BHIS	6S		
1470	001376	016304	177776		MOV	-2(R3), R4	: ADJUST R4 TO POINT TO THE LOWEST LOCATION UNDER TEST	
1471	001402	005723		6S:	TST	(R3)+	: IF BITS 16-17 OF HIGHEST LOCATION TO BE TESTED	
1472	001404	001003			BNE	8S	: ARE NON ZERO THEN GO TO 8S	
1473	001406	021305			CMP	(R3), R5	: OTHERWISE SEE IF THE HIGHEST LOCATION TO BE	
1474							: TESTED IS HIGHER THAN 157776	
1475	001410	101001			BHI	8S	: IF SO THEN GO TO 8S	
1476	001412	011305			MOV	(R3), R5	: MODIFY R5	
1477	001414	105737	000405	8S:	TSTB	2#REL	: ARE WE RELOCATED.?	
1478	001420	100014			BPL	10S	: BRANCH IF NO	
1479	001422	013704	000322		MOV	2#RELBOT, R4	: SET BOTTOM TEST ADDRESS WHEN RELOCATED.	
1480	001426	020527	017776		CMP	R5, #17776	: ARE WE RELOCATED IN BANK 0?	
1481	001432	103402			BLO	9S	: BRANCH IF YES	
1482	001434	012705	017776		MOV	#17776, R5	: ELSE SET HIGH MEMORY UNDER TEST=4K	
1483								
1484	001440	020405		9S:	CMP	R4, R5	: IS LOW LIMIT LOWER THAN HIGH LIMIT?	
1485	001442	103403			BLO	10S	: BRANCH IF YES	
1486	001444	004767	004466		JSR	PC, FATERR	: *ERROR* REPORT ERROR MESSAGE AND HALT AT FATHLT	
1487	001450	000004				4	: *****ERROR NUMBER 4*****	
1488								
1489	001452	012703	000342	10S:	MOV	#SAVMAX, R3	: RESTORE THE CONTENTS OF MAXMEM	
1490	001456	011343			MOV	(R3), -(R3)	: MAKE THE CONTENTS OF MAXMEM = MAXIMUM AVAILABLE	
1491	001460	062713	000002	MENTST:	ADD	#2, (R3)	: MEMORY +2	
1492								

E03

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 29

DZKMAC.P11 18-FEB-77 12:30

BEGIN OF AREA TESTED (+20) WHEN PROGRAM RELOCATES.

```

1493 001464 005725          TST      (R5)+          ;AND SET R5=MAX MEMORY+2
1494
1495          ;CLEAR MEMORY UNDER TEST
1496
1497 001466 012702 000001  CLRMEM: MOV      #1,R2          ;SET R2 TO ENABLE PARITY MODULE CODE.
1498 001472 004767 006006          JSR      PC,PARITY      ;ENABLE PARITY IF WANTED AND AVAILABLE.
1499 001476 010500          MOV      R5,R0
1500 001500 005040 2$:      CLR      -(R0)          ;BEGIN CLEARING THE MEMORY FROM THE TOP
1501 001502 020004          CMP      R0,R4          ;UNTIL THE BOTTOM IS REACHED
1502 001504 101375          BHI      2$
1503 001506 012702 000316          MOV      @BAKPAT,R2
1504 001512 012212          MOV      (R2)+,(R2)      ;WRITE SWAPPED BAKPAT IN LOCATION SWAPAT
1505 001514 000312          SWAB      (R2)
1506 001516 017702 176726          MOV      @SWR,R2          ;LOAD R2 WITH THE OPTIONS STORED AT $SWREG
1507 001522 042702 177760          BIC      #177760,R2      ;ONLY LEAVE THE LOWER 4 BITS OF $SWREG IN R2 TO GO TO
1508                                     ;THE TEST # SPECIFIED [DEFAULT IS TEST#0]
1509
1510
1511          ;ENTER HERE FROM TSTSCP ROUTINE AT END OF SUBTEST
1512
1513
1514 001526 005037 000306  CONT:   CLR      @PASFLG      ;INIT SUBTEST PASS FLAG.
1515 001532 110237 000404          MOV      R2,@$TESTN      ;SET UP $TESTN WITH THE TEST NUMBER GOING
1516                                     ;TO BE EXECUTED
1517 001536 010401          LOOP:  MOV      R4,R1          ;LOAD R1 WITH THE LOWEST LOCATION UNDER TEST
1518 001540 010246          MOV      R2,-(SP)          ;SAVE R2 ON THE STACK
1519 001542 012703 000376          MOV      #376,R3          ;POINT R3 TO SCRATCH STACK
1520 001546 004767 005446          JSR      PC,PUTADR      ;GO TO GENERATE 18 BIT ADDRESS OUT OF THE ADDRESS
1521                                     ;STORED IN R1 AND STORE IT IN LOCATIONS (R3)
1522                                     ;AND (R3-2)
1523 001552 005743          TST      -(R3)          ;CAUSE R3 TO POINT TO THE HIGH ORDER BITS OF THE
1524                                     ;18 BIT ADDRESS
1525 001554 004767 005544          JSR      PC,$GTSIZ      ;PLACE BITS 13-17 OF THE ADDRESS IN BITS
1526                                     ;0-4 OF R2
1527 001560 010400          MOV      R4,R0          ;PLACE THE ADDRESS OF THE LOWEST LOCATION UNDER
1528                                     ;TEST IN R0
1529 001562 010401          MOV      R4,R1          ;IN R1
1530 001564 010403          MOV      R4,R3          ;AND IN R3
1531 001566 012602          MOV      (SP)+,R2          ;RESTORE R2
1532 001570 006302          ASL      R2
1533 001572 060702          ADD      PC,R2
1534 001574 066207 000004          ADD      TBL-,(R2),PC      ;GO TO THE TEST #
1535                                     ;STORED IN BITS 0-3 OF SWITCH REGISTER
1536
1537
1538 001600 000102          TBL:    TST0-TBL      ;RELATIVE ADDRESS OF TEST # 0
1539 001602 000334          TST1-TBL      ;RELATIVE ADDRESS OF TEST # 1
1540 001604 000434          TST2-TBL      ;RELATIVE ADDRESS OF TEST # 2
1541 001606 000544          TST3-TBL      ;RELATIVE ADDRESS OF TEST # 3
1542 001610 001012          TST4-TBL      ;RELATIVE ADDRESS OF TEST # 4
1543 001612 001122          TST5-TBL      ;RELATIVE ADDRESS OF TEST # 5
1544 001614 001270          TST6-TBL      ;RELATIVE ADDRESS OF TEST # 6
1545 001616 001424          TST7-TBL      ;RELATIVE ADDRESS OF TEST # 7
1546 001620 001646          TST10-TBL     ;RELATIVE ADDRESS OF TEST # 10
1547 001622 002174          TST11-TBL     ;RELATIVE ADDRESS OF TEST # 11
1548 001624 002246          TST12-TBL     ;RELATIVE ADDRESS OF TEST # 12

```

F03

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 30

DZKMAC.P11 18-FEB-77 12:30

BEGIN OF AREA TESTED (+20) WHEN PROGRAM RELOCATES.

1549 001626 002520
1550 001630 003146

TST13-TBL
RELOC-TBL

;RELATIVE ADDRESS OF TEST # 13
;RELATIVE ADDRESS OF ROUTINE 'RELOC'

1551
1552
1553
1554
1555

;R5 IS POINTING TO THE TOP OF THE MEMORY TO BE TESTED+2
;R4 & R0 ARE POINTING TO THE LOWEST ADDRESS OF MEMORY TO BE TESTED

G03

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 31
 DZKMAC.P11 18-FEB-77 12:30 SCOPE ROUTINE

```

1556          *      SCOPE ROUTINE
1557          *      -----
1558          *
1559          *
1560          *      PROGRAM COMES TO THIS ROUTINE AFTER COMPLETION OF EACH TEST AND
1561          *      IF CNTRL-C TYPED GOTO ERROR HISTORY TYPE ROUTINE.
1562          *      IF SR= 2000 (BIT10) THEN HALT
1563          *      IF SR= 40000 (BIT14) THEN LOOP ON TEST DEFINED BY SR BITS<3:0>
1564          *      ELSE CONTINUE TO NEXT TEST.
1565          *
1566          *
1567          *
1568 001632 105737 000420      TSTSCP: TSTB      2#SENV      ;ARE WE RUNNING UNDER APT?
1569 001636 001002          BNE      CNTSCP      ;IF SO THEN GO TO CNTSCP
1570 001640 004767 006020      JSR      PC,CHECKC ;TEST FOR CONTROL-C AND IF TYPED GO
1571          *      PRINT ERROR HISTORY AND HALT AT FATHLT
1572 001644 113702 000404      CNTSCP: MOVB      2#STESTN,R2 ;PLACE THE TEST NUMBER IN THE LOWER BYTE OF R2
1573          *      SINCE THERE ARE LESS THAN 377 TESTS UPPER BYTE
1574          *      OF R2 WILL BE 0
1575 001650 005237 000410          INC      2#SDEVCT ;TELL APT WE ARE STILL RUNNING OKAY
1576 001654 032777 002000 176566      BIT      2000,2SWR ;IS THE PROGRAM GOING TO HALT AFTER EACH TEST?
1577 001662 001401          BEQ      TSTGO      ;IF NOT THEN GO TO 25
1578 001664 000000      SWHALT: HALT      ;HALT AT END OF TEST SWITCH SET.
1579          *
1580 001666 032777 040000 176554      TSTGO: BIT      40000,2SWR ;IS THE PROGRAM GOING TO LOOP ON TEST
1581 001674 001320          BNE      LOOP      ;IF SO THEN GO TO THE STARTING OF THE SAME TEST
1582 001676 105202          INCB      R2
1583 001700 000712          BR      CONT      ;GO TO CONT AND CONTINUE EXECUTING THE NEXT TEST

```

H03

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 32
 DZKMAC.P11 18-FEB-77 12:30 TO TEST FOR PROPER BANK SELECTION

```

1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596 001702 105737 000404
1597 001706 001403
1598 001710 004767 004222
1599 001714 000005
1600
1601 001716 012703 177777
1602 001722 010401
1603 001724 010310
1604 001726 020001
1605 001730 001417
1606 001732 005711
1607 001734 001430
1608 001736 020311
1609
1610
1611 001740 001004
1612 001742 012767 000006 000042
1613
1614 001750 000403
1615 001752
1616 001752 012767 000007 000032
1617
1618 001760 010046
1619 001762 105237 000301
1620 001766 000407
1621 001770 020311
1622 001772 001411
1623 001774 012767 000010 000010
1624
1625 002002 010046
1626 002004 010300
1627 002006 004767 003566
1628 002012 000000
1629 002014 012600
1630
1631 002016 013706 000350
1632 002022 062701 020000
1633
1634
1635 002026 020105
1636
1637 002030 103736
1638
1639 002032 105737 000421

;*****
;TEST 0 TEST FOR PROPER BANK SELECTION
;*(1) THIS TEST ASSUMES THAT THE MEMORY IS IN A STATE
; OF ALL 0'S AND R0 HAS THE ADDRESS OF THE LOWEST
; LOCATION UNDER TEST
;*(2) IT CHECKS FOR PROPER BANK SELECTION BY WRITING
; 1'S IN A LOCATION AND CHECKING FOR 0'S IN THE SAME
; LOCATIONS OF OTHER 4K BANKS OF THE MEMORY
; [I.E. LOCATIONS LIKE 7766 AND 27766 ETC.]
;*(3) THIS TEST ALSO CHECKS TO SEE THAT NONE OF THE NON EXIST-
; ING BANK RESPOND WHEN THEY ARE ADDRESSED
;*****
;STO: TSTB 0#STESTN ;CHECK FOR PROPER TEST SEQUENCE
; BEQ +10
; JSR PC,SEGERA ;*ERROR* REPORT ERROR MESSAGE AND HALT AT FATHLT
; 5 ;*****ERROR NUMBER 5*****

1S: MOV #177777,R3 ;R1 = ADDRESS OF LOWEST LOCATION OF MEMORY UNDER TEST
; MOV R4,R1 ;SET ALL THE BITS AT (R0)
; MOV R3,(R0) ;IS R0 POINTING TO THE SAME MEMORY LOCATION AS R1
2S: CMP R0,R1 ;IN WHICH CASE CHECK FOR ALL 1'S AT (R1)
; BEQ 4S ;OTHERWISE CHECK (R1) FOR ALL 0'S
; TST (R1)
; BEQ 5S
; CMP R3,(R1) ;IF R1 IS NOT EQUAL TO R0 AND (R1)
; DOES NOT CONTAIN ALL 0'S THEN
; CHECK TO SEE IF (R0) = (R1)
; BNE 3S
; MOV #6,12S ;*ERROR* SETUP ERROR NO. IN 12S
; ;*****ERROR NUMBER #6*****

3S: BR 10S
; MOV #7,12S ;*ERROR* SETUP ERROR NO. IN 12S
; ;*****ERROR NUMBER #7*****

10S: MOV R0,-(SP) ;SAVE R0 ON STACK
; INCB 0#SADERR ;AN ADDRESSING ERROR IS SUSPECTED
; BR 11S
; 4S: CMP R3,(R1) ;CHECK (R1) FOR ALL 1'S
; BEQ 5S
; MOV #10,12S ;*ERROR* SETUP ERROR NO. IN 12S
; ;*****ERROR NUMBER #10*****

; MOV R0,-(SP) ;SAVE R0 ON STACK
; MOV R3,R0
; 11S: JSR PC,ERROR ;GO TO THE ERROR SUBROUTINE
; 12S: .WORD ;ERROR NUMBER TO BE REPORTED WILL BE PLACED HERE
; MOV (SP)+,R0 ;RESTORE R0

5S: MOV 0#SAVR6,SP ;RESTORE THE STACK POINTER
; ADD #20000,R1 ;CAUSE R1 TO POINT TO THE SAME CHIP
; ;LOCATION IN THE NEXT 4K BANK OF MEMORY
; ;BY ADDING 1 TO THE 14TH BIT OF ADDRESS IN R1
; CMP R1,R5 ;COMPARE R1 WITH THE HIGHEST MEMORY
; ;LOCATION WHICH IS STORED IN R5
; BLO 2S ;IF R1 LESS THAN R5 THEN REPEAT THE TEST FROM 2S

; TSTB 0#SENVN ;HAS APT INHIBITED SIZING?

```


DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 33
 DZKMAC.P11 18-FEB-77 12:30 TO

TEST FOR PROPER BANK SELECTION

1640	002036	100430		BMI	BS	;BRANCH IF YES (DON'T TEST NON-EXISTENT MEMORY)
1641	002040	032777	000100 176402	BIT	#100,2SWR	;HAS USER INHIBITED SIZING?
1642	002046	001024		BNE	BS	;BRANCH IF YES (DON'T TEST NON-EXISTENT MEMORY)
1643						
1644	002050	020137	000340	CMP	R1,2#MAXMEM	;IS R1 LOWER THAN THE MAXIMUM AVAILABLE
1645						;MEMORY ?
1646	002054	103760		BLO	5S	;IF SO THEN GO TO 5S
1647	002056	012702	000006	MOV	#6,R2	;MAKE R2 POINT TO TRAP VECTOR+2 FOR NXH
1648	002062	012712	000340	MOV	#340,(R2)	;SET PSW TO 340
1649	002066	012742	177722	MOV	#5S-,-6,-(R2)	;SET UP RETURN ADDRESS FROM TRAP TO 5S
1650	002072	060712		ADD	PC,(R2)	
1651	002074	020127	157776	CMP	R1,#157776	;SEE IF R1 HAS CROSSED 28K BOUNDARY OF VIRTUAL ADDRESS
1652	002100	101004		BHI	6S	;IN WHICH CASE GO TO 6S
1653	002102	011111		MOV	(R1),(R1)	;TRY TO WRITE TO NON-EXISTENT MEMORY (SHOULD TRAP)
1654	002104	004767	004026	JSR	PC,FATERR	;*ERROR* REPORT ERROR MESSAGE AND HALT AT FATHLT
1655	002110	000011		11		;*****ERROR NUMBER 11*****
1656						
1657						
1658	002112	012722	000006	6S: MOV	#6,(R2)+	;RESTORE TRAP VECTOR
1659	002116	005012		CLR	(R2)	
1660	002120	005010		8S: CLR	(R0)	
1661						
1662	002122	062700	020000	ADD	#20000,R0	;CAUSE R0 TO POINT TO THE SAME CHIP
1663						;LOCATION IN THE NEXT 4K MEMORY BANK
1664						;BY ADDING 1 TO THE 14TH BIT OF ADDRESS IN R0
1665	002126	020005		CMP	R0,R5	;COMPARE R0 WITH THE HIGHEST MEMORY
1666						;LOCATION WHICH IS STORED IN R5.
1667	002130	103674		BLO	1S	;IF R0 LESS THEN REPEAT THE TEST
1668	002132	000637		BR	TSTSCP	
1669						
1670						

J03

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 34
 DZKMAC.P11 18-FEB-77 12:30 T1 CHECK DI/DO LINES

```

1671 ;*****
1672 ;*TEST 1 CHECK DI/DO LINES
1673 ;*(1) THIS TEST CHECKS THE DATI/DATO LINES BY SHIFTING
1674 ;* A 1 IN THE WORD DIRECTION
1675 ;*****
1676 002134 122737 000001 000404 TST1: CMPB #1,2#STESTN ;CHECK FOR PROPER TEST SEQUENCE
1677
1678
1679 002142 001403 BEQ .+10
1680 002144 004767 003766 JSR PC,SEQERR ;*ERROR* REPORT ERROR MESSAGE AND HALT AT FATHLT
1681 002150 000012 12 *****ERROR NUMBER 12*****
1682
1683 002152 012700 000001 1$: MOV #1,R0
1684 002156 010002 MOV R0,R2 ;SET R2=1
1685 002160 010011 2$: MOV R0,(R1) ;MOV 1 AT LOCATION (R1)
1686 002162 020011 3$: CMP R0,(R1) ;COMPARE R1 WITH THE CONTENTS OF LOCATION (R1)
1687 002164 001403 BEQ 4$
1688 002166 004767 003406 JSR PC,ERROR ;*ERROR* REPORT ERROR MESSAGE
1689 002172 000013 13 *****ERROR NUMBER 13*****
1690
1691
1692 002174 005702 4$: TST R2 ;ARE WE SHIFTING A 0 IN DATA DIRECTION?
1693 002176 001406 BEQ 5$ ;IF SO THEN GO TO 5$
1694 002200 006300 ASL R0 ;SHIFT THE 1 BROUGHT IN AT 1$ IN
1695 ;DATA DIRECTION
1696 002202 103366 BCC 2$ ;IF THE 1 HAS NOT BEEN SHIFTED THRU
1697 ;THE 16 DATA BITS THEN REPEAT FROM 2$
1698 002204 005002 CLR R2 ;INITIATE SHIFTING OF 0 IN DATA DIRECTION
1699 002206 012700 177776 MOV #177776,R0
1700 002212 000762 BR 2$
1701
1702 002214 000261 5$: SEC ;SET C BIT
1703 002216 006100 ROL R0 ;SHIFT A 0 16 TIMES IN DATA DIRECTION
1704 002220 103757 BCS 2$ ;IF THE 0 HAS NOT BEEN SHIFTED THRU
1705 ;THE 16 DATA BITS THEN REPEAT FROM 2$
1706 002222 062701 020000 ADD #20000,R1 ;OTHERWISE GO TO THE NEXT BANK OF
1707 ;4K MEMORY AND REPEAT THE TEST
1708 002226 020105 CMP R1,R5
1709 002230 103750 BLO 1$
1710 002232 000737 END1: BR ENDO
1711

```


K03

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 35
 DZKMAC.P11 18-FEB-77 12:30 T2

TEST MEMORY FOR HOLDING DATA AND BYTE SELECTION

```

1712      ;*****
1713      ;#TEST 2      TEST MEMORY FOR HOLDING DATA AND BYTE SELECTION
1714      ;#(1)      THIS TEST CHECKS THE MEMORY FOR THE CAPABILITY
1715      ;#      OF HOLDING 1'S AND 0'S BY WRITING A BACKGROUND
1716      ;#      OF BAKPAT AND READING IT
1717      ;#(2)      MEMORY IS WRITTEN USING A BYTE AT A TIME
1718      ;#(3)      STEPS 1 & 2 ARE REPEATED WITH A SWAPPED BACKGROUND PATTERN
1719      ;*****
1720      002234 122737 000002 000404 15T2:  CMPB  #2,#STESTN      ;CHECK FOR PROPER TEST SEQUENCE
1721
1722      002242 001403      BEQ  +10
1723      002244 004767 003666      JSR  PC,SEGERR      ;*ERROR* REPORT ERROR MESSAGE AND HALT AT FATHLT
1724      002250 000014      14      ;*****ERROR NUMBER 14*****
1725
1726      002252 013700 000316      15:  MOV  #BAKPAT,R0
1727      002256 110021      MOVB  R0,(R1)+
1728      002260 113721 000317      MOVB  #BAKPAT+1,(R1)+;WRITE THE MEMORY WITH THE WORD STORED IN BAKPAT
1729      002264 020105      CMP  R1,R5
1730      002266 103771      BLO  15
1731
1732      002270 020041      25:  CMP  R0,-(R1)      ;TEST THE MEMORY TO SEE IF IT CONTAINS
1733      ;THE WORD STORED IN BAKPAT
1734      002272 001416      BEQ  85
1735      002274 062701 000002      ADD  #2,R1
1736      002300 123741 000317      CMPB  #BAKPAT+1,-(R1);CHECK FOR BYTE SELECTION PROBLEM
1737      002304 001402      BEQ  45
1738      002306 120041      CMPB  R0,-(R1)      ;AGAIN CHECK FOR BYTE SELECTION PROBLEM
1739      002310 001002      BNE  65
1740      002312 105237 000301      45:  INCB  #SADERR      ;PREPARE TO INFORM THAT IT IS ADDRESSING ERROR
1741      002316 042701 000001      65:  BIC  #1,R1      ;MAKE THE ADDRESS IN R1 EVEN
1742      002322 004767 003252      JSR  PC,ERROR      ;*ERROR* REPORT ERROR MESSAGE
1743      002326 000015      15      ;*****ERROR NUMBER 15*****
1744
1745      002330 020104      85:  CMP  R1,R4      ;KEEP ON TESTING THE MEMORY UNTIL
1746      002332 101356      BHI  25      ;R1 EQUALS THE LOWEST ADDRESS
1747      002334 000337 000316      SWAB  #BAKPAT      ;CHANGE THE DATA PATTERN
1748      002340 001744      BEQ  15      ;IF THE DATA PATTERN DOES NOT HAVE LOW
1749      ;BYTE =0 THEN FALL THRU
1750      002342 000733      END2: BR  END1
1751
1752      ;THE TEST LEAVES BAKPAT LOCATION THE SAME AS IT WAS IN THE BEGINNING
1753

```

L03

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 36
DZKMAC.P11 18-FEB-77 12:30 T3 DUAL ADDRESS TEST A

```

1754 ::*****
1755 ;*TEST 3 DUAL ADDRESS TEST A
1756
1757 ;*(1) THIS TEST CHECKS FOR DUAL ADDRESSING PROBLEMS BY WRITING A
1758 ;* BACK GROUND OF BAKPAT.
1759 ;*(2) STARTING FROM THE LOWEST LOCATION IN THE BANK THE TEST WRITES A
1760 ;* LOCATION WITH SWAPPED BAKPAT
1761 ;*(3) READS THE MEMORY FOR PROPER CONTENTS
1762 ;*(4) SHIFTS A 1 ALONG THE ADDRESS DIRECTION AND REPEATS STEPS 1-3
1763 ;*(5) REPEATS STEP 1-4 FOR EACH 4K BANK
1764 ::*****
1765 002344 122737 000003 000404 1753: CMPB #3,2#STESTN ;CHECK FOR PROPER TEST SEQUENCE
1766 002352 001403 BEQ .+10
1767 002354 004767 003556 JSR PC,SEGERR ;*ERROR* REPORT ERROR MESSAGE AND HALT AT FATHLT
1768 002360 000016 16 *****ERROR NUMBER 16*****
1769
1770 002362 005003 CLR R3
1771 002364 004737 000120 25: JSR PC,2#WRTMEM ; WRITE MEMORY WITH THE BACKGROUND STORED
1772 ;AT LOCATION BAKPAT
1773 002370 005002 45: CLR R2
1774 002372 050302 65: BIS R3,R2 ;MAKE R2 POINT TO THE MEMORY BANK POINTED BY R3
1775 002374 020204 CMP R2,R4 ;IF R2 IS LESS THAN R4
1776 002376 103465 BLO 16$ ;THEN DO NOTHING
1777 002400 020205 CMP R2,R5 ;IF R2 IS HIGHER THAN THE HIGHEST LOCATION TO BE
1778 002402 103077 BHIS 20$ ;TESTED THEN EXIT THE TEST
1779 002404 000312 SWAB (R2) ;OTHERWISE WRITE THE COMPLEMENT OF BAKPAT IN
1780 ;THE LOCATION POINTED BY R2
1781 002406 005001 CLR R1
1782 002410 050301 75: BIS R3,R1
1783 002412 020104 CMP R1,R4 ;IF R1 IS POINTING TO A LOCATION LOWER THAN R4
1784 002414 103445 BLO 12$ ;THEN GO TO 12$
1785 002416 020105 CMP R1,R5
1786 002420 103053 BHIS 15$
1787 002422 020102 CMP R1,R2 ;CHECK THE MEMORY FOR CORRECT DATA
1788 002424 001431 BEQ 10$
1789 002426 020011 CMP R0,(R1) ;IF R1 IS NOT = TO R2 THEN (R1) SHOULD HAVE
1790 ;THE SAME WORD AS BAKPAT
1791 002430 001437 BEQ 12$ ;IN WHICH CASE GO BACK TO 12$
1792 002432 012767 000017 000032 MOV #17,22$ ;*ERROR* SETUP ERROR NO. IN 22$
1793 *****ERROR NUMBER #17*****
1794 002440 010046 85: MOV R0,-(SP) ;PLACE R0 ON THE STACK
1795 002442 000316 SWAB (SP)
1796 002444 022611 CMP (SP)+,(R1) ;IF (R1) IS NOT = R0 THEN SEE IF IT IS SAME
1797 ;AS A SWAPPED R0
1798 002446 001003 BNE 95 ;IF NOT THEN A SUSPECTED DUAL ADDRESSING PROBLEM
1799 ;FOR THE BITS THAT ARE DIFFERENT IN R0 AND (R1)
1800 ;OTHERWISE THERE IS DUAL ADDRESSING FOR THE
1801 ;ENTIRE WORD
1802 002450 012767 000020 000014 MOV #20,22$ ;*ERROR* SETUP ERROR NO. IN 22$
1803 *****ERROR NUMBER #20*****
1804 002456 105237 000301 95: INCB 2#SADERR ;ADDRESSING PROBLEM IS DETECTED
1805 002462 010046 MOV R0,-(SP) ;SAVE R0
1806 002464 010200 MOV R2,R0 ;SET R0=GOOD ADDRESS FOR ERROR REPORT
1807 002466 004767 003106 JSR PC,ERROR ;GO TO THE ERROR SUBROUTINE
1808 002472 000000 22$: .WORD ;ERROR NUMBER TO BE REPORTED WILL BE PLACED HERE
1809 002474 012600 MOV (SP)+,R0 ;RESTORE R0

```


M03

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 37
 DZKMAC.P11 18-FEB-77 12:30 T3 DUAL ADDRESS TEST A

1810	002476	010011		MOV	RO,(R1)	:RESTORE (R1)
1811	002500	020037	000316	CMP	RO,2#BAKPAT	:IF THE CONTROL CAME HERE FROM 15S-2 THEN
1812	002504	001411		BEG	12S	
1813	002506	000407		BR	11S	:RETURN TO 11S
1814	002510	000300		10S: SWAB	RO	:MAKE RO SAME AS SWAPPED BAKPAT
1815	002512	020011		CMP	RO,(R1)	:IF R1 = R2 THEN (R1) SHOULD CONTAIN A WORD
1816						:EQUAL TO SWAPPED RO
1817	002514	001404		BEG	11S	:IN WHICH CASE GO BACK TO 11S
1818	002516	012767	000021 177746	MOV	#21,22S	:*ERROR* SETUP ERROR NO. IN 22S
1819						:*****ERROR NUMBER #21*****
1820	002524	000745		BR	8S	:AND GO TO 8S
1821	002526	000300		11S: SWAB	RO	:RESTORE RO TO BAKPAT
1822	002530	040301		12S: BIC	R3,R1	:TAKE OUT THE BANK ADDRESS FROM THE ADDRESS IN R1
1823	002532	005701		TST	R1	:IF R1 IS 0 THEN PLACE A 1 IN R1
1824	002534	001001		BNE	13S	:OTHERWISE GO TO 13S
1825	002536	005201		INC	R1	
1826	002540	006101		13S: ROL	R1	
1827	002542	020127	020000	CMP	R1,#20000	:IF R1 IS LESS THAN A 4K BOUNDARY
1828	002546	103720		BLO	7S	:THEN REPEAT FROM 7S
1829	002550	000312		15S: SWAB	(R2)	:RESTORE (R2) TO BAKPAT
1830	002552	040302		16S: BIC	R3,R2	:TAKE OUT THE BANK ADDRESS FROM THE ADDRESS
1831						:STORED IN R2
1832	002554	005702		TST	R2	:IF R2 = 0 THEN MOVE A 1 TO R2
1833	002556	001001		BNE	18S	:OTHERWISE GO TO 18S
1834	002560	005202		INC	R2	
1835	002562	006102		18S: ROL	R2	:SHIFT A ONE IN THE ADDRESS WORD
1836	002564	020227	020000	CMP	R2,#20000	:IS THE ADDRESS IN R2 MORE THAN THE BOUNDARY
1837						:OF 4K
1838	002570	103700		BLO	6S	:IF NOT THEN GO TO 6S
1839	002572	060203		ADD	R2,R3	:OTHERWISE MAKE R3 POINT TO THE NEXT 4K BANK
1840	002574	020337	000340	CMP	R3,2#MAXMEM	:IF R3 IS POINTING TO A BANK THAT IS LOWER
1841						:THAN MAXMEM
1842	002600	103673		BLO	4S	:THEN REPEAT FROM 4S
1843	002602	000337	000316	20S: SWAB	2#BAKPAT	
1844	002606	001656		BEG	TST3	:REPEAT THE TEST WITH SWAPPED BAKPAT ONLY IF
1845						:THE LOWER BYTE OF BAKPAT IS 0
1846	002610	000654		END3: BR	END2	

N03

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 38
DZKNAC.P11 18-FEB-77 12:30 T4 DUAL ADDRESS TEST B

```

1847 .....
1848 .....
1849 .....
1850 .....
1851 .....
1852 .....
1853 002612 122737 000004 000404 TST4: CMPB 04,2#STESTN ;CHECK FOR PROPER TEST SEQUENCE
1854 002620 001403 BEQ +10
1855 002622 004767 003310 JSR PC,SEERR ;*ERROR* REPORT ERROR MESSAGE AND HALT AT FATHLT
1856 002626 000022 22 ;*****ERROR NUMBER 22*****
1857 .....
1858 002630 005003 CLR R3
1859 002632 010100 1S: MOV R1,R0
1860 002634 005703 TST R3 ;IF R3 IS NOT 0 THEN STORE THE ADDRESS
1861 002636 001401 BEQ 2S ;IN THE LOCATION
1862 002640 005100 COM R0 ;OTHERWISE STORE COMPLEMENT
1863 002642 010021 2S: MOV R0,(R1)+ ;OF THE ADDRESS
1864 002644 020105 CMP R1,R5 ;UNTIL THE HIGHEST MEMORY LOCATION IS REACHED
1865 002646 103771 BLO 1S
1866 .....
1867 002650 020041 3S: CMP R0,-(R1) ;CHECK THE LOCATION FOR THE CORRECT CONTENTS
1868 002652 001405 BEQ 4S
1869 002654 105237 000301 INCB 2#SADERR ;THIS IS PROBABLY ADDRESS PROBLEM RATHER THAN
1870 ..... ;BIT PROBLEM
1871 002660 004767 002714 JSR PC,ERROR ;*ERROR* REPORT ERROR MESSAGE
1872 002664 000023 23 ;*****ERROR NUMBER 23*****
1873 .....
1874 002666 010100 4S: MOV R1,R0
1875 002670 162700 000002 SUB R2,R0 ;CHECK THAT THE ADDRESS IS STORED AT
1876 002674 005703 TST R3 ;LOCATION IF R3 IS NOT 0
1877 002676 001401 BEQ 5S ;OTHERWISE CHECK FOR
1878 002700 005100 COM R0 ;ADDRESS COMPLEMENT
1879 002702 020104 5S: CMP R1,R4
1880 002704 101361 BHI 3S
1881 002706 112737 000001 000306 MOVB R1,2#PASFLG ;SET PASFLG FOR ERROR REPORT.
1882 002714 005103 COM R3 ;COMPLEMENT THE CONTENTS OF R3
1883 002716 001345 BNE 1S ;REPEAT TST3 IF R3, IS NON 0, ENABLING ADDRESS
1884 ..... ;COMPLEMENT TO BE WRITTEN AND READ, OTHERWISE FALL THRU
1885 002720 000733 END4: BR END3
1886 .....

```


B04

DZKNA MACY11 27(1006) 18-FEB-77 12:35 PAGE 39
 DZKNAC.P11 18-FEB-77 12:30 TS MARCHING 1'S AND 0'S

```

1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901 002722 122737 000005 000404
1902
1903 002730 001403
1904 002732 004767 003200
1905 002736 000024
1906
1907 002740 004737 000120
1908
1909 002744 020041
1910 002746 001403
1911 002750 004767 002624
1912 002754 000025
1913
1914 002756 000300
1915 002760 010011
1916 002762 021100
1917 002764 001403
1918 002766 004767 002606
1919 002772 000006
1920
1921
1922 002774 000300
1923
1924 002776 001023
1925
1926
1927
1928
1929
1930
1931
1932 003000 005703
1933
1934
1935
1936
1937 003002 001023
1938 003004 062701 000002
1939 003010 020105
1940
1941 003012 103006
1942 003014 020011

;*****
;#TEST 5 MARCHING 1'S AND 0'S
;#(1) THIS TEST WRITES A BACK GROUND OF THE WORD STORED
;# AT BAKPAT.
;#(2) READS EVERY LOCATION FOR CORRECT DATA, SWAPS BYTES
;# AT THE LOCATION AND PROCEEDS IN MAX. TO MIN
;# DIRECTION OF MEMORY LOCATIONS.
;#(3) READS EVERY LOCATION FOR SWAPPED BAKPAT PATTERN
;# WRITES BAKPAT BACKGROUND IN THE LOCATION AND PROCEEDS
;# IN MIN. TO MAX. DIRECTION
;#(4) REPEATS STEP 2 GOING IN MIN. TO MAX. DIRECTION
;#(5) REPEATS STEP 3 GOING IN MAX. TO MIN. DIRECTION
;*****
1TS: CMPB #5,#$TESTN ;CHECK FOR PROPER TEST SEQUENCE
      BEQ .+10
      JSR PC,SEGERR ;#ERROR# REPORT ERROR MESSAGE AND HALT AT FATHLT
      24 ;*****ERROR NUMBER 24*****
1S: JSR PC,#$WRTMEN ;GO TO WRITE THE MEMORY WITH A BACKGROUND OF THE
      ;WORD STORED IN BAKPAT
2S: CMP RO,-(R1) ;READ THE CONTENTS OF LOCATION POINTED BY R1
      BEQ 3S ;TO SEE IF IT HAS THE SAME VALUE AS R0
      JSR PC,ERROR ;#ERROR# REPORT ERROR MESSAGE
      25 ;*****ERROR NUMBER 25*****
3S: SWAB RO ;SWAP THE BYTES AT (R1)
      MOV RO,(R1) ;READ (R1) FOR CORRECT VALUE
      CMP (R1),RO
      BEQ 4S ;#ERROR# REPORT ERROR MESSAGE
      JSR PC,ERROR ;*****ERROR NUMBER 26*****
      26
4S: SWAB RO ;SWAP THE BYTES OF THE REGISTER
      ;CONTAINING BACKGROUND PATTERN
      BNE 9S ;IF THE LOWER BYTE OF THE REGISTER
      ;IS NOT 0 THEN THE PROGRAM IS READING
      ;THE MEMORY TO CONTAIN A BACK GROUND OF
      ;BAKPAT AND WRITING THE SWAPPED WORD
      ;IN WHICH CASE GO TO 9S
5S: TST R3 ;R3 WAS 0 WHEN THE PROGRAM ENTERED
      ;THIS TEST, AND IT IS NOT ALTERED UNTIL PASFLG=3
      ;IF R3 EQUAL 0 THEN THE PROGRAM IS
      ;READING/WRITING MIN. TO MAX. OTHERWISE
      ;IT IS GOING IN MAX. TO MIN. DIRECTION
      BNE 10S ;IF R3 IS NOT CLEAR THEN GO TO 10S
      ADD #2,R1 ;OTHERWISE ADD 2 TO THE CONTENTS OF R1
      CMP R1,R5 ;COMPARE R1 WITH THE MAX. MEMORY LOCATION TO
      ;BE TESTED
      BHS 8S ;IF R1>R5 THEN GO TO 8S OTHERWISE
      8S ;READ (R1) FOR THE CORRECT DATA
      CMP RO,(R1)

```

C04

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 40
 DZKMAC.F11 18-FEB-77 12:30 TS MARCHING 1'S AND 0'S

1943	003016	001757		BEQ	3S		: WRITE COMPLEMENT OF THE DATA FOUND AT (R1)
1944							: AND REPEAT UNTIL R1 > R5
1945	003020	004767	002554	JSR	PC,ERROR		: *ERROR* REPORT ERROR MESSAGE
1946	003024	000027		27			: *****ERROR NUMBER 27*****
1947							
1948	003026	000753		BR	3S		
1949	003030	105237	000306	8S: INCB	2#PASFLG		
1950	003034	000300		SWAB	RO		
1951	003036	001742		BEQ	2S		: IF THE LOWER BYTE OF RO IS ALL 0'S
1952							: THEN BEGIN READING BAKPAT SWAPPED WRITING BAKPAT
1953							: AND READING BAKPAT GOING FROM MAX. TO MIN. (PASFLG=4)
1954	003040	005103		COM	R3		: OTHERWISE CLEAR RO
1955	003042	010401		MOV	R4,R1		: PUT THE LOWEST TESTING ADDRESS IN R1
1956	003044	000763		BR	7S		: AND BEGIN READING 0'S, WRITING 1'S AND
1957							: READING 1'S IN MIN. TO MAX. DIRECTION (PASFLG=3)
1958							
1959	003046	005703		9S: TST	R3		: IF R3 IS NON 0, I.E. PASFLG=3
1960	003050	001353		BNE	5S		: THEN READ BAKPAT, WRITE
1961							: SWAPPED BAKPAT AND READ SWAPPED BAKPAT
1962							: IN MIN. TO MAX. DIRECTION
1963	003052	020104		10S: CMP	R1,R4		: OTHERWISE TEST IS PROCEEDING IN MAX. TO
1964							: MIN. DIRECTION.
1965	003054	101333		BHI	2S		: KEEP ON LOOPING UNTIL R1=R4
1966	003056	105237	000306	INCB	2#PASFLG		
1967	003062	000300		SWAB	RO		
1968	003064	001753		BEQ	7S		: IF RO SWAPPED HAS LOWER BYTE=0
1969							: THEN READ BAKPAT SWAPPED, WRITE BAKPAT,
1970							: AND READ BAKPAT GOING FROM MIN. TO MAX.
1971	003066	000714		ENDS: BR	END4		
1972							

004

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 41
DZKMAC.P11 18-FEB-77 12:30 T6 CELLS' VOLATILITY TEST

```

1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989 003070 122737 000006 000404
1990
1991
1992 003076 001403
1993 003100 004767 003032
1994 003104 000030
1995
1996 003106 004737 000120
1997
1998 003112 005037 000306
1999 003116 010403
2000 003120 010401
2001 003122 020011
2002 003124 001403
2003 003126 004767 002446
2004 003132 000031
2005
2006 003134 062701 000002
2007 003140 020105
2008 003142 103767
2009 003144 132737 000001 000306
2010 003152 001002
2011 003154 105237 000306
2012
2013 003160 020305
2014 003162 103012
2015 003164 012702 037776
2016 003170 000313
2017 003172 005302
2018 003174 001375
2019 003176 010337 000354
2020 003202 062703 020000
2021
2022
2023 003206 000744
2024 003210 105237 000306
2025 003214 000337 000316
2026 003220 001732
2027 003222 000721
2028

;*****
;TEST 6 CELLS' VOLATILITY TEST
;
;(1) THIS TEST WRITES THE MEMORY WITH A BACK GROUND OF BAKPAT
;(2) WITH PASFLG=0 THE TEST READS THE MEMORY FOR BAKPAT
; AND THEN INCREMENTS PASFLG
;(3) IT THEN READS/WRAPS BYTES/WRITES A LOCATION X FOR
; OVER 2 MSEC AND THEN READS THE MEMORY FOR BAKPAT
;(4) REPEATS STEP 3 WITH X=X+4K UNTIL END OF MEMORY IS ENCOUNTERED
;(5) IT THEN INCREMENTS PASFLG AND WRITES THE MEMORY TO
; BAKPAT AND WITH PASFLG=2 IT READS MEMORY FOR ALL
; SWAPPED BAKPAT AFTER WHICH PASFLG IS INCREMENTED TO 3
;(6) REPEATS STEPS 3 AND 4 READING THE MEMORY FOR SWAPPED
; BAKPAT INSTEAD OF BAKPAT.
;*****

TST6: CMPB #6,0#STESTN ;CHECK FOR PROPER TEST SEQUENCE
;*****
BEQ .+10
JSR PC,SEGERR ;#ERROR* REPORT ERROR MESSAGE AND HALT AT FATHLT
30 ;*****ERROR NUMBER 30*****

RPT6: JSR PC,0#WRTHEM ;GO TO WRITE THE MEMORY WITH A BACKGROUND OF THE
;WORD STORED AT LOCATION BAKPAT

1$: CLR 0#PASFLG
2$: MOV R4,R3 ;SET R3
3$: MOV R4,R1 ;AND R1 TO THE STARTING ADDRESS OF MEMORY UNDER TEST
CMP R0,(R1) ;CHECK (R1) FOR CORRECT DATA
BEQ 4$
JSR PC,ERROR ;#ERROR* REPORT ERROR MESSAGE
31 ;*****ERROR NUMBER 31*****

4$: ADD #2,R1 ;INCREMENT R1 BY 2
CMP R1,R5 ;SEE IF R1 HAS REACHED THE MAX. OF MEMORY
BLO 3$
BITB #1,0#PASFLG ;CHECK TO SEE IF PASFLG=0 OR 2
BNE 5$
INCB 0#PASFLG ;IN WHICH CASE INCREMENT PASFLG COUNTER BY 1

5$: CMP R3,R5 ;SEE IF R3 HAS REACHED THE MAX. OF THE MEMORY
BHIS 7$
MOV #37776,R2 ;WRITE INTO 1 LOC FOR >2MS (ABOUT 100MS)
6$: SWAB (R3)
DEC R2
BNE 6$
MOV R3,0#SAVLOC ;SAVE LOCATION WRITTEN FOR 2MS FOR ERROR REPORT.
ADD #20000,R3 ;BY ADDING 1 TO THE 14TH ADDRESS BIT CAUSE
;R3 TO POINT TO A LOCATION IN THE NEXT
;4K BANK OF MEMORY

7$: BR 2$
INCB 0#PASFLG ;MAKE PASFLG=2
SWAB 0#BAKPAT ;IF BAKPAT IS NOT BEING SWAPPED FOR THE 2ND
BEQ RPT6 ;THEN GO BACK TO THE LOCATION RPT6
BR ENDS
END6:

```

E04

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 42
DZKMAC.P11 18-FEB-77 12:30 T6 CELLS' VOLATILITY TEST

2029

F04

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 43
 DZKMAC.P11 18-FEB-77 12:30 T7 SHIFTING DIAGONAL

```

2030      ;*****
2031      ;TEST 7      SHIFTING DIAGONAL
2032
2033      ;*(1)  THIS TEST WRITES THE MEMORY WITH A BACKGROUND OF BAKPAT
2034      ;*(2)  IT WRITES A DIAGONAL OF SWAPPED BAKPAT THROUGH EACH MEMORY BANK
2035      ;*(3)  READS THE MEMORY FOR CORRECT DATA
2036      ;*(4)  SHIFTS THE DIAGONAL AND REPEATS STEP 3 UNTIL THE
2037      ;      DIAGONAL HAS BEEN SHIFTED 64 TIMES
2038      ;*(5)  WRITES A BACKGROUND OF SWAPPED BAKPAT, A DIAGONAL OF
2039      ;      BAKPAT AND REPEATS FROM STEP 3
2040      ;*****
2041 003224 122737 000007 000404 1ST7: CMPB  #7,2#STESTN ;CHECK FOR PROPER TEST SEQUENCE
2042
2043 003232 001403      BEQ  +10
2044 003234 004767 002676      JSR  PC,SEERR ;*ERROR* REPORT ERROR MESSAGE AND HALT AT FATHLT
2045 003240 000032      32 ;*****ERROR NUMBER 32*****
2046
2047 003242 005037 000306 2S: CLR  2#PASFLG
2048 003246 010337 000304      MOV  R3,2#LOWBNK ;LOWBNK CONTAINS ADDRESS OF THE LOWEST LOCATION
2049 ;IN THE 4K BANK THAT CAN BE TESTED
2050 003252 010302      MOV  R3,R2
2051 003254 052702 017776      BIS  #17776,R2 ;R2 CONTAINS THE ADDRESS OF THE TOP OF THE BANK
2052 003260 005722      TST  (R2)+ ;ADD 2 TO R2
2053 003262 020502      CMP  R5,R2
2054 003264 103001      BHS  4S ;IF R2 IS GREATER THAN R5 THEN GO TO 4S
2055 003266 010502      MOV  R5,R2 ;NOW R2 CONTAINS THE ADDRESS OF THE HIGHEST LOCATION
2056 ;THAT CAN BE TESTED
2057 003270 010337 000302 4S: MOV  R3,2#STRTDI ;LOAD STRTDI WITH THE STARTING ADDRESS OF THE
2058 ;DIAGONAL
2059 003274 013701 000304      MOV  2#LOWBNK,R1 ;R1 IS NOW POINTING TO THE LOWEST LOCATION IN THE 4K
2060 ;BANK
2061 003300 013700 000316 6S: MOV  2#BAKPAT,R0 ;STORE THE CONTENTS OF BAKPAT IN R0
2062 003304 020103      CMP  R1,R3 ;IS R1 POINTING TO A LOCATION IN THE DIAGONAL ?
2063 003306 001010      BNE  10S ;IF NOT THEN GO TO 10S
2064 003310 062703 000002      ADD  #2,R3 ;THE FOLLOWING CODE IS USED TO PLACE THE
2065 003314 032703 000176      BIT  #176,R3 ;ADDRESS OF THE NEXT LOCATION IN THE DIAGONAL
2066 003320 001402      BEQ  8S ;IN R3
2067 003322 062703 000200      ADD  #200,R3
2068 003326 000300 8S: SWAB  R0 ;DIAGONAL WILL CONTAIN SWAPPED BACKGROUND PATTERN
2069 003330 132737 000001 000306 10S: BITB  #1,2#PASFLG ;CONTENTS OF LOCATION PASFLG WILL BE EVEN IF THE
2070 ;MEMORY IS BEING WRITTEN AND IT WILL BE ODD
2071 ;IF IT IS ONLY BEING READ
2072 003336 001001      BNE  12S ;IF IT IS BEING READ ONLY THEN GO TO 12S
2073 003340 010011      MOV  R0,(R1) ;OTHERWISE WRITE THE MEMORY WITH THE CONTENTS
2074 ;OF R0
2075 003342 020011 12S: CMP  R0,(R1) ;CHECK THE LOCATION POINTED BY R1 TO CONTAIN
2076 ;PROPER DATA
2077 003344 001403      BEQ  14S ;IF IT IS OK THEN GO TO 14S
2078 003346 004767 002226      JSR  PC,ERROR ;*ERROR* REPORT ERROR MESSAGE
2079 003352 000033      33 ;*****ERROR NUMBER 33*****
2080
2081 003354 062701 000002 14S: ADD  #2,R1 ;CAUSE R1 TO POINT TO THE NEXT MEMORY LOCATION
2082 003360 020102      CMP  R1,R2 ;IS IT THE END OF THE BANK ?
2083 003362 103746      BLO  6S ;IF NOT THEN GO TO 6S
2084 003364 005237 000410 16S: INC  2#$DEVCT ;TELL APT WE ARE STIL RUNNING OKAY
2085 003370 105237 000306      INCB 2#PASFLG

```

G04

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 44
 DZKMAC.P11 18-FEB-77 12:30 T7 SHIFTING DIAGONAL

2086	003374	013703	000302		MOV	2#STRDI,R3	:LOAD R3 WITH THE STARTING ADDRESS OF THE DIAGONAL
2087	003400	132737	000001	000306	BITB	2#PASFLG	:HAS THE READ OF THE MEMORY BEEN DONE ?
2088	003406	001330			BNE	4\$	:IF NOT THEN GO TO 4\$
2089	003410	005723			TST	(R3)+	:ADD 2 TO THE STARTING ADDRESS OF THE DIAGONAL
2090	003412	020302			CMP	R3,R2	:AND UNLESS THE END OF THE BANK IS REACHED
2091	003414	103003			BHIS	18\$	
2092	003416	105737	000306		TSTB	2#PASFLG	:OR THE DIAGONAL HAS BEEN ROTATED 64 TIMES
2093	003422	100322			BPL	4\$	:REPEAT FROM 4\$
2094	003424	013703	000304	18\$:	MOV	2#LOWBNK,R3	:MAKE R3 POINT TO THE LOWEST LOCATION IN THE
2095							:IN THE BANK UNDER TEST
2096	003430	000337	000316		SWAB	2#BAKPAT	
2097	003434	001715			BEQ	4\$	:AND IF THE TEST HAS NOT BEEN PERFORMED WITH THE
2098							:SWAPPED BACK GROUND PATTERN THEN GO TO 4\$
2099	003436	010203			MOV	R2,R3	:MAKE THE PRESENT HIGH BOUNDARY AS THE NEXT
2100							:LOW BOUNDARY
2101	003440	020205			CMP	R2,R5	:UNLESS THE PRESENT HIGH BOUNDARY IS ALSO THE
2102							:HIGH BOUNDARY FOR THE MEMORY UNDER TEST
2103	003442	103677			BLO	2\$	
2104	003444	000666		END7:	BR	END6	

H04

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 45
DZKMAC.P11 18-FEB-77 12:30 T10

READ RECOVERY GALLOPING TEST THROUGH EVERY 64TH CELL

```

2105      ;*****
2106      ;TEST 10      READ RECOVERY GALLOPING TEST THROUGH EVERY 64TH CELL
2107
2108      ;*(1)      THIS TEST WRITES THE MEMORY WITH A BACK GROUND PATTERN
2109      ;*          STORED AT LOCATION BAKPAT
2110      ;*(2)      TEST BEGINS AT LOWEST LOCATION BEING TESTED
2111      ;*          (LETS NAME IT 'A')
2112      ;*(3)      LETS NAME THE 1ST LOCATION IN THE ROW/COLUMN UNDER TEST AS 'B'.
2113      ;*(4)      SWAPS BYTES FOR LOCATION 'A'.
2114      ;*(5)      READS 'A', READS 'B'
2115      ;*(6)      'B' = 'B'+200 (MAKES 'B'=64TH CELL I.E. 200TH OCTAL
2116      ;*          LOCATION FROM THE PRESENT LOCATION OF 'B')
2117      ;*(7)      REPEATS STEPS 5 AND 6 UNTIL 'B' IS GREATER THAN THE
2118      ;*          END OF THE 4K BANK OF THE MEMORY IN WHICH 'A' IS RESIDING
2119      ;*(8)      A = A+2
2120      ;*(9)      REPEATS STEPS 3-8 UNTILL 'A' REACHES THE END OF THE BANK
2121      ;*(10)     GOES TO THE NEXT 4K BANK OF MEMORY AND REPEATS STEPS
2122      ;*          3-9 UNTIL THE END OF THE MEMORY
2123      ;*(11)     AFTER EXECUTING THE TEST BYTES ARE SWAPPED AT
2124      ;*          LOCATION BAKPAT AND STEPS 1-10 ARE REPEATED
2125      ;*(12)     IN THIS TEST R0 IS POINTING TO LOCATION 'A', R1 TO
2126      ;*          LOCATION 'B', R2 TO THE END OF THE 4K BANK IN WHICH THE
2127      ;*          TEST IS TAKING PLACE AND R3 TO THE LOWEST LOCATION IN THE
2128      ;*          COLUMN/ROW CONTAINING 'A' AND 'B'
2129      ;*(13)     MOST OF THE CODE USED BY THIS TEST IS ALSO USED BY TEST 11
2130
2131      ;*****
2132      003446 122737 000010 000404 1ST10: CMPB      #10,2#STESTN      ;CHECK FOR PROPER TEST SEQUENCE
2133
2134      003454 001403          BEQ          +10
2135      003456 004767 002454      JSR          PC,SEERR          ;*ERROR* REPORT ERROR MESSAGE AND HALT AT FATHLT
2136      003462 000034          34          ;*****ERROR NUMBER 34*****
2137
2138      003464 010402          MOV          R4,R2          ;SET R2 TO THE LOWEST MEMORY UNDER TEST
2139      003466 052702 017776      RPT10: BIS          #17776,R2      ;MAKE R2 POINT TO THE HIGHEST LOCATION IN THE 4K
2140      ;*          BANK FOR WHICH GALLOPING WILL BE PERFORMED
2141      003472 062702 000002      GALLOP: ADD          #2,R2          ;INCREMENT R2 BY 2
2142      003476 020205          CMP          R2,R5          ;IF THE HIGH BOUNDARY OF THE TEST IS HIGHER THAN
2143      003500 101401          BLOS         2$          ;THE MAXIMUM ALLOWED ADDRESS THEN ADJUST R2
2144      003502 010502          MOV          R5,R2
2145      003504 005046          2$:      CLR          -(SP)
2146      003506 010200          MOV          R2,R0
2147      003510 013740 000316      4$:      MOV          2#BAKPAT,-(R0) ;WRITE THE MEMORY UNDER TEST WITH A BACKGROUND OF
2148      ;*          BAKPAT
2149      003514 020003          CMP          R0,R3
2150      003516 101374          BHI          4$
2151      003520 010301          6$:      MOV          R3,R1          ;R3 AND R1 ARE POINTING TO THE LOWEST LOCATION THAT
2152      ;*          CAN BE TESTED IN THIS BLOCK
2153      003522 023710 000316      CMP          2#BAKPAT,(R0)      ;BEFORE STARTING THE GALLOPING TEST FOR LOCATION
2154      ;*          (R0) CHECK IT
2155      003526 001410          BEQ          8$          ;CONTINUE IF OK
2156      003530 010001          MOV          R0,R1          ;OTHERWISE PREPARE TO REPORT THE ERROR
2157      003532 013700 000316      MOV          2#BAKPAT,R0
2158      003536 004767 002036      JSR          PC,ERROR      ;*ERROR* REPORT ERROR MESSAGE
2159      003542 000035          35          ;*****ERROR NUMBER 35*****
2160

```

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 46
 DZKMAC.P11 18-FEB-77 12:30 T10

READ RECOVERY GALLOPING TEST THROUGH EVERY 64TH CELL

2161	003544	010011				MOV	RO, (R1)	;RESTORE THE CONTENTS OF (R1)
2162	003546	010100				MOV	R1, RO	;RESTORE RO
2163								
2164	003550	000310			8\$:	SWAB	(RO)	
2165	003552	031011			10\$:	BIT	(RO), (R1)	;CHECK TO SEE THAT NONE OF THE BITS SET ;IN (RO) ARE SET IN (R1) AND VICE VERSA
2166								;THE ONLY EXCEPTION TO THIS WILL BE WHEN RO=R1
2167	003554	020001				CMP	RO, R1	
2168								
2169	003556	001412				BEQ	12\$	
2170	003560	021137	000316			CMP	(R1), 2#BAKPAT	;CHECK THAT (R1) HAS BAKPAT IN IT
2171	003564	001407				BEQ	12\$	
2172	003566	010046				MOV	RO, -(SP)	;SAVE RO ON STACK
2173	003570	013700	000316			MOV	2#BAKPAT, RO	;PLACE THE PATTERN WORD IN RO
2174	003574	004767	002000			JSR	PC, ERROR	;*****ERROR REPORT ERROR MESSAGE
2175	003600	000036				36		;*****ERROR NUMBER 36*****
2176								
2177	003602	012600				MOV	(SP)+, RO	;RESTORE RO
2178	003604	021037	000320		12\$:	CMP	(RO), 2#SWAPAT	;CHECK THAT (RO) HAS SWAPPED BAKPAT IN IT
2179	003610	001412				BEQ	14\$	
2180	003612	010146				MOV	R1, -(SP)	;SAVE R1 ON THE STACK
2181	003614	010001				MOV	RO, R1	;MAKE R1 POINT TO THE FAILING LOCATION
2182	003616	013700	000320			MOV	2#SWAPAT, RO	;LOAD RO WITH THE EXPECTED RESULT IN (R1)
2183	003622	004767	001752			JSR	PC, ERROR	;*****ERROR REPORT ERROR MESSAGE
2184	003626	000037				37		;*****ERROR NUMBER 37*****
2185								
2186	003630	010011				MOV	RO, (R1)	;RECOVER (R1) FROM THE ERROR
2187	003632	010100				MOV	R1, RO	;RESTORE RO
2188	003634	012601				MOV	(SP)+, R1	;AND RESTORE R1
2189	003636	122737	000011	000404	14\$:	CMPB	#11, 2#STESTN	;IS THE PROGRAM EXECUTING TEST # 11 ?
2190	003644	001402				BEQ	16\$	;IF SO THEN GO TO 16\$
2191	003646	062701	000176			ADD	#176, R1	
2192	003652	062701	000002		16\$:	ADD	#2, R1	;MAKE R1 POINT TO THE NEXT ADJACENT CELL
2193	003656	020102				CMP	R1, R2	;AND IF R1 HAS NOT REACHED THE END OF THE BOUNDARY
2194	003660	103734				BLO	10\$	;THEN REPEAT FROM 10\$
2195	003662	000320				SWAB	(RO)+	;RESTORE THE LOCATION FOR WHICH THE GALLOPING TEST ;WAS BEING PERFORMED
2196								
2197	003664	122737	000011	000404		CMPB	#11, 2#STESTN	;IS IT TEST 11 ?
2198	003672	001407				BEQ	17\$	;IF SO THEN GO TO 17\$
2199	003674	005723				TST	(R3)+	;OTHERWISE INCREMENT R3 BY 2
2200	003676	062716	000002			ADD	#2, (SP)	;FOR EVERY ROW/COLUMN TESTED ADD 2
2201	003702	105716				TSTB	(SP)	
2202	003704	100002				BPL	17\$	;UNTIL (SP) IS 200
2203	003706	161603				SUB	(SP), R3	;SUBTRACT 200 FROM R3
2204	003710	005016				CLR	(SP)	
2205	003712	032700	000177		17\$:	BIT	#177, RO	;AT A 64TH CALL BOUNDARY?
2206	003716	001002				BNE	18\$	;BRANCH IF NO
2207	003720	005237	000410			INC	2#SDEVCT	;TELL APT WE ARE STILL RUNNING
2208	003724	020002			18\$:	CMP	RO, R2	;IF RO HAS NOT REACHED THE END OF THE BOUNDARY
2209	003726	103674				BLO	6\$	;THEN REPEAT FROM 6\$
2210	003730	162603				SUB	(SP)+, R3	;RESTORE SP AND R3
2211	003732	000337	000320			SWAB	2#SWAPAT	
2212	003736	000337	000316			SWAB	2#BAKPAT	
2213	003742	001660				BEQ	2\$	;IF THE LOWER BYTE OF BAKPAT IS 0 THEN REPEAT FROM 2\$
2214	003744	010203				MOV	R2, R3	;OTHERWISE MAKE THE PRESENT HIGH BOUNDARY AS THE ;NEXT LOW BOUNDARY
2215								
2216	003746	020205				CMP	R2, R5	

J04

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 47
 DZKMAC.P11 18-FEB-77 12:30 T10

READ RECOVERY GALLOPING TEST THROUGH EVERY 64TH CELL

2217	003750	001410		BEQ	END10	: IF PREVIOUS HIGH BOUNDARY WAS THE END OF THE
2218						: TEST BOUNDARY THEN EXIT THE TEST
2219	003752	032702	017776	BIT	#17776,R2	: WAS IT A 4K BOUNDARY ?
2220	003756	001025		BNE	RPT11	: IF NOT THEN WE WERE PERFORMING TEST 11 WITH LONG
2221						: GALLOPING TEST DISABLED
2222	003760	122737	000011 000404	CMPB	#11,2#STESTN	: IF IT IS TEST # 11 THEN GO TO REPEAT TEST 11
2223	003766	001421		BEQ	RPT11	
2224	003770	000636		BR	RPT10	: OTHERWISE REPEAT TEST 10
2225	003772	000624		BR	END?	
2226						
2227						
2228						

END10:

K04

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 48
DZKMAC.P11 18-FEB-77 12:30 T11

READ RECOVERY LONG GALLOPING/FAST GALLOPING TEST

```

2229          ;*****
2230          ;#TEST 11      READ RECOVERY LONG GALLOPING/FAST GALLOPING TEST
2231
2232          ;*(1)      THIS TEST WRITES MEMORY WITH BAKPAT
2233          ;*(2)      THE TEST BEGINS AT THE LOWEST LOCATION BEING TESTED
2234          ;*(3)      (LETS NAME IT 'B')
2235          ;*(4)      'A'+ 'B' (MOVE THE ADDRESS OF 'B' TO THE POINTER FOR LOCATION 'A')
2236          ;*(5)      SWAPS BYTES FOR LOCATION 'A'
2237          ;*(6)      READS 'A', READS 'B'
2238          ;*(7)      'B'='B'+2
2239          ;*(8)      IF GALLOPING OPTION BIT AT SSWREG IS HIGH THEN STEPS 4 AND 5
2240          ;*(9)      ARE REPEATED UNTIL 'B' REACHES THE HIGHEST MEMORY LOCATION
2241          ;*(10)     OF THE 4K BANK IN WHICH 'A' IS RESIDING, THEN 'A' IS
2242          ;*(11)     DECREMENTED BY 2 AND AFTER MAKING 'B' TO POINT TO THE LOWEST
2243          ;*(12)     LOCATION OF THE 4K MEMORY BANK CONTAINING 'A' STEPS 3,4,5 AND
2244          ;*(13)     6 ARE REPEATED UNTIL 'A' EQUALS THE END OF THE ENTIRE MEMORY
2245          ;*(14)     IF GALLOPING OPTION BIT IS NOT HIGH THEN STEPS 4 AND 5 ARE
2246          ;*(15)     REPEATED UNTIL 'B' IS POINTING TO A CELL IN THE NEXT COLUMN
2247          ;*(16)     IF SEQUENTIAL CELLS LIE ALONG THE ROW, OR THE NEXT ROW
2248          ;*(17)     IF SEQUENTIAL CELLS LIE ALONG THE COLUMN, AT WHICH TIME
2249          ;*(18)     STEPS 2,3,4,5 AND 7 ARE REPEATED UNTIL THE END OF THE MEMORY
2250          ;*(19)     TEST IS REPEATED FOR THE OPPOSITE BACKGROUND DATA
2251          ;*(20)     IN THIS TEST R0 POINTS TO LOCATION 'A', R1 TO LOCATION
2252          ;*(21)     'B', R2 TO THE HIGHEST LOCATION AND R3 TO THE LOWEST
2253          ;*(22)     LOCATION IN A 64/4K CELL BOUNDARY
2254          ;*(23)     MOST OF THE CODE USED BY TEST 10 IS ALSO USED BY THIS TEST
2255
2256          ;*****
2257          ;#TEST 11:      CMPB      #11,0#STESTN      ;CHECK FOR PROPER TEST SEQUENCE
2258
2259          ;*(24)     BEQ          .+10
2260          ;*(25)     JSR          PC,SEQERR      ;*ERROR* REPORT ERROR MESSAGE AND HALT AT FATHLT
2261          ;*(26)     40          ;*****ERROR NUMBER 40*****
2262
2263          ;*(27)     MOV          R4,R2      ;MAKE R2 TO POINT TO THE LOWEST LOCATION
2264          ;*(28)     TSTB         @SWR      ;UNDER TEST
2265          ;*(29)     BPL          RPT11      ;LONG GALLOP ENABLED?
2266          ;*(30)     JSR          PC,PNTMES   ;BRANCH IF NO
2267          ;*(31)     .ASCIZ       /GLP/      ;TYPE "GLP"
2268          ;*(32)     RPT11:      TSTB         @SWR      ;LONG GALLOPING ENABLED?
2269          ;*(33)     BMI          RPT10      ;BRANCH IF YES
2270          ;*(34)     BIS          #176,R2     ;TO RPT10
2271          ;*(35)     ;OTHERWISE SET THE LOW ORDER BITS OF THE ADDRESS
2272          ;*(36)     ;TO GET THE HIGH BOUNDARY
2273          ;*(37)     BR           GALLOP      ; PERFORM GALLOPING TEST
2274
2275          003774 122737 000011 000404
2276          004002 001403
2277          004004 004767 002126
2278          004010 000040
2279          004012 010402
2280          004014 105777 174430
2281          004020 100004
2282          004022 004767 002540
2283          004026 046107 000120
2284          004032 105777 174412
2285          004036 100613
2286          004040 052702 000176
2287          004044 000612

```


L04

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 49
DZKMAC.P11 18-FEB-77 12:30 T12 WORST CASE TESTING FOR CORE MEMORY

```

2276 *****
2277 ;*TEST 12 WORST CASE TESTING FOR CORE MEMORY
2278 ;*(1) STARTING FROM THE LOWEST LOCATION UNDER TEST THE MEMORY
2279 ;* IS WRITTEN WITH A BACKGROUND OF BAKPAT, HOWEVER LOCATIONS
2280 ;* HAVING ADDRESS SUCH THAT EXCLUSIVE OR OF ADDRESS BITS 1 &
2281 ;* 8 = 1 ARE WRITTEN TO A VALUE OF SWAPPED BAKPAT
2282 ;*(2) STARTING FROM THE LOWEST LOCATION THE MEMORY IS CHECKED
2283 ;* TO CONTAIN THE CORRECT DATA AS EXPLAINED IN STEPS 3 & 4,
2284 ;* UNTILL THE HIGHEST LOCATION UNDER TEST IS REACHED
2285 ;*(3) READ EACH LOCATION FOR THE CORRECT CONTENT
2286 ;*(4) COMPLEMENT THE LOCATION AND READ IT; COMPLEMENT THE LOCATION
2287 ;* BACK TO ITS ORIGINAL VALUE AND READ IT AGAIN
2288 ;*(5) STARTING FROM THE HIGHEST LOCATION UNDER TEST REPEAT STEPS
2289 ;* 3 & 4 UNTIL THE LOWEST LOCATION UNDER TEST IS REACHED
2290 ;*(6) REPEAT STEPS 1-5, HOWEVER THIS TIME LOCATIONS WITH XOR
2291 ;* OF ADDRESS BITS 8 & 13 = 1 ARE WRITTEN TO SWAPPED BAKPAT
2292 ;*(7) REPEAT STEPS 1-5, HOWEVER THIS TIME LOCATIONS WITH XOR
2293 ;* OF ADDRESS BITS 3 & 9 = 1 ARE WRITTEN TO SWAPPED BAKPAT
2294 ;*(8) REPEAT STEPS 1-7 WITH A BACKGROUND OF SWAPPED BAKPAT AND
2295 ;* THE LOCATIONS TO BE WRITTEN TO SWAPPED BAKPAT WRITTEN TO
2296 ;* BAKPAT.
2297 *****
2298 004046 122737 000012 000404 TST12: CMPB #12,2#TESTN ;CHECK FOR PROPER TEST SEQUENCE
2299 004054 001403 BEQ +10
2300 004056 004767 002054 JSR PC,SEGERR ;*ERROR* REPORT ERROR MESSAGE AND HALT AT FATHLT
2301 004062 000041 41 *****ERROR NUMBER 41*****
2302
2303
2304 004064 012702 000002 MOV #2,R2 ;PREPARE TO TAKE THE EXCLUSIVE OR OF ADDRESS BITS 1
2305 004070 012703 000400 MOV #400,R3 ;AND 8
2306 004074 112737 000001 000306 1$: MOVB #1,2#PASFLG ;INITIALIZE THE COUNTER FOR THE SUBTEST
2307 004102 010401 2$: MOV R4,R1 ;PLACE THE STARTING ADDRESS OF MEMORY UNDER
2308 ;TEST IN R1
2309 004104 013700 000316 4$: MOV 2#BAKPAT,R0
2310 004110 030201 BIT R2,R1 ;CHECK TO SEE IF ADDRESS BIT STORED IN R2 IS SET
2311 004112 001004 BNE 8$ ;IF IT IS SET THEN GO TO 8$
2312 004114 030301 BIT R3,R1 ;CHECK TO SEE IF ADDRESS BIT POINTED BY R3 IS SET
2313 004116 001404 BEQ 12$ ;IF IT IS NOT SET THEN GO TO 12$
2314 004120 005100 6$: COM R0 ;COME HERE ONLY IF EXCLUSIVE OR OF ADDRESS BITS
2315 ;POINTED BY R2 & POINTED BY R3 = 1 IN WHICH
2316 ;CASE PREPARE TO WRITE THE LOCATION
2317 ;WITH A COMPLEMENT OF LOCATIONS NOT MEETING
2318 ;THIS CONDITION
2319 004122 000402 BR 12$
2320 004124 030301 8$: BIT R3,R1 ;COME HERE IF ADDRESS BIT POINTED BY R2 IS 1 AND
2321 ;CHECK ADDRESS BIT POINTED BY R3
2322 004126 001774 BEQ 6$ ;IF ADDRESS BIT POINTED BY R3 IS 0 THEN GO TO 6$
2323 004130 132737 000002 000306 12$: BITB #2,2#PASFLG ;IS IT 2ND OR 3RD PASS OF THE SUBTEST ?
2324 004136 001001 BNE 14$ ;IF SO THEN READ THE MEMORY

```

M04

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 50
 DZKMAC.P11 18-FEB-77 12:30 T12

WORST CASE TESTING FOR CORE MEMORY

2325	004140	010011			MOV	RO, (R1)	; OTHERWISE WRITE THE MEMORY BEFORE READING IT
2326	004142	020011			14\$: CMP	RO, (R1)	; READ THE MEMORY FOR CORRECT CONTENTS
2327	004144	001403			BEQ	16\$	
2328	004146	004767	001426		JSR	PC, ERROR	; *ERROR* REPORT ERROR MESSAGE
2329	004152	000042			42		; *****ERROR NUMBER 42*****
2330							
2331	004154	012746	000002		16\$: MOV	#2, -(SP)	
2332	004160	005100			18\$: COM	RO	
2333	004162	005111			COM	(R1)	
2334	004164	020011			CMP	RO, (R1)	; READ THE MEMORY AGAIN
2335	004166	001404			BEQ	19\$	
2336	004170	004767	001404		JSR	PC, ERROR	; *ERROR* REPORT ERROR MESSAGE
2337	004174	000043			43		; *****ERROR NUMBER 43*****
2338							
2339	004176	010011			MOV	RO, (R1)	; RESTORE THE LOCATION (R1)
2340	004200	005316			19\$: DEC	(SP)	
2341	004202	001366			BNE	18\$	; EXECUTE THE CODE FROM 18\$ TWICE
2342	004204	005726			TST	(SP)+	; RESTORE THE STACK POINTER
2343	004206	122737	000003	000306	CMPB	#3, #PASFLG	; IS IT THE 3RD PASS OF THE SUBTEST ?
2344	004214	001412			BEQ	20\$	; IF SO THEN GO TO 20\$
2345	004216	062701	000002		ADD	#2, R1	; IN FIRST 2 PASSES THE PROGRAM PROCEEDS IN
2346							; MIN. TO MAX. DIRECTION
2347	004222	020105			CMP	R1, R5	; HAVE WE REACHED THE MAX. ADDRESS UNDER TEST ?
2348	004224	103727			BLO	4\$	; IF NOT THEN REPEAT FROM 4\$
2349	004226	105237	000306		INCB	#PASFLG	
2350	004232	122737	000002	000306	CMPB	#2, #PASFLG	; IF IT IS THE 2ND PASS OF THE SUBTEST
2351	004240	001720			BEQ	2\$	; THEN REPEAT FROM 2\$
2352	004242	162701	000002		20\$: SUB	#2, R1	; OTHERWISE EXECUTE THE TEST IN MAX. TO MIN.
2353							; DIRECTION
2354	004246	020104			CMP	R1, R4	; HAVE WE REACHED THE MIN. ADDRESS UNDER TEST ?
2355	004250	103315			BHIS	4\$	; IF NOT THEN REPEAT FROM 4\$
2356	004252	012702	020000		MOV	#20000, R2	; PREPARE TO CHECK THE MEMORY WITH THE XOR OF
2357							; ADDRESS BITS 8 AND 13
2358	004256	105237	000307		INCB	#PASFLG+1	; THE SUB TEST HAS CHECKED THE XOR ONE KIND
2359	004262	123727	000307	000002	CMPB	#PASFLG+1, #2	; HAS TWO XOR COMBINATIONS BEEN CHECKED ?
2360	004270	103701			BLO	1\$	; IF NOT THEN GO TO 1\$
2361	004272	101004			BHI	22\$	; IF ALL THREE HAVE BEEN CHECKED THEN GO TO 22\$
2362	004274	012702	000010		MOV	#10, R2	; IF IT IS THE 2ND XOR COMBINATION THEN CHECK
2363	004300	006303			ASL	R3	; FOR ADDRESS BITS 3 & 8
2364	004302	000674			BR	1\$	
2365	004304	005137	000316		22\$: COM	#BAKPAT	; IF THE TEST WAS NOT PERFORMED WITH THE SWAPPED
2366	004310	105737	000316		TSTB	#BAKPAT	
2367	004314	001654			BEQ	TST12	; BAKPAT THEN RE-EXECUTE THE TEST
2368	004316	000625			END12: BR	END10	

N04

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 51
DZKMAC.P11 18-FEB-77 12:30 T13 WRITE RECOVERY TEST

2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424

004320 122737 000013 000404
004326 001403
004330 004767 001602
004334 000044
004336 012702 010247
004342 012700 177667
004346 010546
004350 010446
004352 000241
004354 006005
004356 006004
004360 160405
004362 006005
004364 103002
004366 062716 000002
004372 012604
004374 012605
004376 010403
004400 000405

```
*****
TEST 13 WRITE RECOVERY TEST
THIS TEST DIFFERS FROM 0-12 IN THAT IT CONSISTS OF A SMALL TEST PROGRAM
ACTUALLY RUNNING IN THE 4K BANK UNDER TEST.
THE PROGRAM IS SELF MODIFYING AND MAY BE DIFFICULT TO DEBUG.
TO AID IN THE DEBUG, BEFORE A BANK IS ENTERED "TST13 BANK XX"
IS TYPED. THIS WILL ALLOW THE USER TO AT LEAST SEE WHICH MEMORY
BANK FAILED.
THE TEST CONSISTS OF 1/2 OF THE BANK STORED WITH "MOV R2, -(PC)"
AND THE OTHER 1/2 CONTAINING "177667". "177667" IS THE COMPLEMENT
OF "JMP (R0)" INSTRUCTION.
R2 CONTAINS "COM -(R1)" INSTRUCTION ON ENTRY TO THE BANK AND R1 CONTAINS
THE HIGHEST TEST ADDRESS IN THAT BANK. THE HIGHEST TEST ADDRESS IS
USUALLY ON 4K BOUNDARIES. WHEN TESTING BANK 0 RELOCATED, HOWEVER
R1 CONTAINS THE FIRST FREE TEST ADDRESS BELOW THE DIAGNOSTIC.
IF YOU UNDERSTAND THIS SO FAR THE REST IS EASY.
THE TEST EXECUTION IS AS FOLLOWS:
1. THE "MOV R2, -(PC)" INSTRUCTION EXECUTES STORING
THE CONTENTS OF R2 IN THE ADDRESS IT VACATED (DUE TO -(PC)).
2. SINCE R2 CONTAINS A "COM -(R1)" INSTRUCTION IT COMPLEMENTS
THE HIGHEST ADDRESS UNDER TEST. THIS ADDRESS CONTAINED
"177667" SO AFTER THE COM -(R1) IT EQUALS 110
CLEVERLY THIS IS THE "JMP (R0)" INSTRUCTION.
3. THIS SEQUENCE CONTINUES UNTIL THE "MOV R2, -(PC)" INSTRUCTIONS
REACH THE MIDDLE OF THE TEST BANK. THEN THE "JMP (R0)" INSTRUCTION IS
AND EXECUTED. R0 CONTAINED THE RETURN ADDRESS BACK
TO TEST 13.
4. THESE STEPS ARE REPEATED FOR EACH BANK UNDER TEST.
*****
TST13: CMPB #13, #TESTN ;CHECK FOR PROPER TEST SEQUENCE
        BEQ .+10
        JSR PC, SEQERR ;:ERROR* REPORT ERROR MESSAGE AND HALT AT FATHLT
        44 ;*****ERROR NUMBER 44*****
1$: MOV #10247, R2 ;PLACE THE OP CODE OF INSTRUCTION MOV R2, -(PC)
    MOV #177667, R0 ;IN R2.
    JMP (R0) IN R0 ;PLACE THE COMPLEMENT OF THE INSTRUCTION
    ;INSURE LOWEST TEST ADDRESS TO END OF 4K SEGMENT IS MULTIPLE OF 2
    ;SINCE THE TEST STORES "MOV R2, -(PC)" IN 1/2 AND 177667 IN THE OTHER 1/2
2$: MOV R5, -(SP) ;SAVE R5
    MOV R4, -(SP) ;STORE LOWEST ADDRESS ON STACK
29$: CLC
    ROR R5 ;MAKE POSITIVE BYTE COUNT OF HIGH ADDRESS
    ROR R4 ;DO SAME FOR LOWEST ADDRESS
    SUB R4, R5 ;GET DIFFERENCE OF LOWEST ADDRESS AND HIGHEST
    ROR R5 ;IF DIFFERENCE IS ODD THEN R4 IS AT LOWEST ADDRESS
    BCC 30$ ;BRANCH IF R4 IS AT LOWEST TEST ADDRESS.
    ADD #2, (SP) ;INCREASE LOWEST TEST ADDRESS BY 2
    MOV (SP)+, R4 ;RESTORE R4 (POSSIBLY INCREASED BY 2 FROM ENTRY)
    MOV (SP)+, R5 ;RESTORE HIGHEST TEST ADDRESS
    MOV R4, R3 ;PLACE THE LOWEST LOCATION UNDER TEST
    IN R3
30$: BR 28$ ;LEAVE LOW BITS OF R3 ALONE FIRST TIME IN CASE BANK 0
```

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 52
 DZKMAC.P11 18-FEB-77 12:30 T13

WRITE RECOVERY TEST

2425	004402	042703	017776	3S:	BIC	#17776,R3	;CAUSE R3 TO POINT TO THE LOWEST LOCATION
2426							;IN THE 4K BANK UNDER TEST
2427	004406	105737	000405		TSTB	#REL	;ARE WE RELOCATED?
2428	004412	100504			BHI	14S	;BRANCH IF YES-TEST BANK0 ONLY-
2429	004414	020305		28S:	CMP	R3,R5	;IF R3 IS HIGHER THAN THE HIGHEST LOCATION
2430	004416	103102			BHIS	14S	;UNDER TEST THEN EXIT
2431							;IF R5 LESS THAN 20000 THEN WE ARE TESTING BANK0 RELOCATED IN BANK0
2432	004420	020527	020000		CMP	R5,#20000	;IS HIGHEST TEST ADDRESS BELOW 4K?
2433	004424	103002			BHIS	31S	;BRANCH IF NO
2434	004426	010501			MOV	R5,R1	;SET R1 TO HIGHEST TEST ADDRESS IN BANK0
2435	004430	000405			BR	32S	
2436							
2437	004432	010301		31S:	MOV	R3,R1	;SET R1 TO LOWEST CURRENT TEST ADDRESS
2438	004434	042701	017776		BIC	#17776,R1	;CLEAR LOW ORDER ADDRESS BITS
2439	004440	062701	020000		ADD	#20000,R1	;CAUSE R1 TO POINT TO THE HIGHEST LOCATION+2
2440							;OF THE 4K BANK BEING POINTED BY R3
2441	004444	020137	000340	32S:	CMP	R1,#MAXMEM	;IF R1 IS HIGHER THAN MAX. OF THE
2442	004450	101065			BHI	14S	;MEMORY+2 ALTHOUGH R3 IS LESS THAN R5
2443							;THEN THE HIGHEST LOCATION UNDER
2444							;TEST IS NOT IN A 4K BANK EXIT
2445							
2446	004452	132737	000001 000306		BITB	#1,#PASFLG	;IS THE LOWEST BIT OF LOCATION PASFLG
2447	004460	001101			BNE	16S	;SET? IN WHICH CASE BACK GROUND HAS
2448							;ALREADY BEEN WRITTEN AND WRITE RECOVERY
2449							;TEST IS BEING PERFORMED
2450							
2451	004462	020304		4S:	CMP	R3,R4	;OTHERWISE WRITE THE BACKGROUND
2452	004464	103430			BLO	8S	;DEFINED AT STEP 3.
2453	004466	105737	000307		TSTB	#PASFLG+1	;IS THE TEST JUST DOING READ, I.E.
2454	004472	001002			BNE	6S	;IS THE PASFLG+1 LOCATION NON ZERO? IF SO
2455							;THEN GO TO 6S
2456	004474	012713	010247		MOV	#10247,(R3)	;WRITE THE LOCATION WITH THE OP CODE FOR MOV R2,-(PC)
2457	004500	020213		6S:	CMP	R2,(R3)	;READ (R3) TO CONTAIN CORRECT DATA
2458	004502	001421			BEQ	8S	
2459	004504	010046			MOV	R0,-(SP)	;SAVE R0
2460	004506	010146			MOV	R1,-(SP)	;AND R1 ON THE STACK
2461	004510	010301			MOV	R3,R1	
2462	004512	010200			MOV	R2,R0	;SET R0= GOOD DATA FOR ERROR PRINTOUT
2463	004514	004767	001060		JSR	PC,ERROR	;#ERROR# REPORT ERROR MESSAGE
2464	004520	000045			4S		;*****ERROR NUMBER 45*****
2465							
2466	004522	012601			MOV	(SP)+,R1	;RESTORE R1
2467	004524	012600			MOV	(SP)+,R0	;AND R0
2468	004526	105737	000306		TSTB	#PASFLG	;IF PASFLG IS 0 AND THE MEMORY DOES NOT HAVE
2469							;THE PROPER DATA THEN WE DON'T WANT TO GO AND
2470							;EXECUTE THE INSTRUCTIONS STORED IN MEMORY UNDER
2471							;TEST
2472	004532	001005			BNE	8S	;BRANCH IF PASFLG NOT =0
2473							
2474	004534	010200			MOV	R2,R0	;SAVE FOR ERROR REPORT
2475	004536	004767	001036		JSR	PC,ERROR	;#ERROR# REPORT ERROR MESSAGE
2476	004542	000046			46		;*****ERROR NUMBER 46*****
2477							
2478	004544	000664			BR	END12	;ABORT TST 13.
2479							
2480	004546	062703	000002	8S:	ADD	#2,R3	;INCREMENT R3 BY 2

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 53
 DZKMAC.P11 18-FEB-77 12:30 T13 WRITE RECOVERY TEST

2481	004552	162701	000002		SUB	#2,R1	;DECREMENT R1 BY 2
2482	004556	020105			CMP	R1,R5	;WRITE THE BACKGROUND DEFINED AT STEP 4.
2483	004560	103014			BHIS	12\$	
2484	004562	020103			CMP	R1,R3	;HAS STORING THE 177667 REACHED WHERE "MOV R2,-(PC) IS?
2485	004564	103405			BLO	10\$	;BRANCH IF YES DON'T DESTROY THE MOV R2,-(PC) IS.
2486	004566	105737	000307		TSTB	2#PASFLG+1	;IS THE THE READ ONLY CHECK PASS?
2487	004572	001002			BNE	10\$	;BRANCH IF YES
2488	004574	012711	177667		MOV	#177667,(R1)	;WRITE THE LOCATION WITH THE COMPLEMENT OF THE
2489							;OP CODE JMP (R0)
2490	004600	020011		10\$:	CMP	R0,(R1)	;READ R1 TO CONTAIN CORRECT DATA
2491	004602	001403			BEQ	12\$	
2492	004604	004767	000770		JSR	PC,ERROR	;#ERROR# REPORT ERROR MESSAGE
2493	004610	000047			47		;*****ERROR NUMBER 47*****
2494							
2495	004612	020301		12\$:	CMP	R3,R1	;IF WE HAVE NOT REACHED THE MIDDLE OF 4K BANK
2496	004614	103722			BLO	4\$	;THEN REPEAT FROM 4\$
2497							
2498							;RETURN HERE AFTER PROGRAM RUN IN BANK UNDER TEST
2499							
2500	004616	062703	020000	13\$:	ADD	#20000,R3	;OTHERWISE GO TO THE NEXT 4K BANK
2501	004622	000667			BR	3\$	
2502							
2503	004624	122737	000001 000306	14\$:	CMPB	#1,2#PASFLG	;THE PROGRAM CONTROL COMES HERE AS FOLLOWS
2504							1-PASFLG=0, PROGRAM HAS JUST COMPLETED A
2505							WRITE/READ CYCLE FOR THE BACK GROUND
2506							AND WANTS TO BEGIN THE WRITE RECOVERY TEST
2507	004632	001440			BEQ	24\$	;2-PASFLG=1, PROGRAM HAS JUST COMPLETED
2508							THE WRITE RECOVERY TEST AND WANTS TO
2509							READ MEMORY FOR CORRECT DATA
2510	004634	103630			BLO	END12	;3-PASFLG=2, PROGRAM HAS CORRECTLY READ THE
2511							MEMORY AND WANTS TO GO THE NEXT TEST.
2512							
2513	004636	105137	000307		COMB	2#PASFLG+1	;ENTER HERE WITH PASFLG=0, ON THE FIRST ENTRY
2514							;ENABLE READ ONLY FOR THE MEMORY AND ON THE SECOND
2515							;ENTRY DISABLE READ ONLY
2516	004642	001241			BNE	2\$	
2517	004644	012702	005141		MOV	#5141,R2	;PLACE THE OP CODE FOR INSTRUCTION COM -(R1)
2518							;IN R2
2519	004650	012700	177740		MOV	#13\$-,-6,R0	;PLACE THE RETURN ADDRESS IN R0 AS 13\$
2520	004654	060700			ADD	PC,R0	;THUS WHEN THE READ RECOVERY TEST REACHES
2521							THE MIDDLE OF THE 4K MEMORY THEN THE
2522							INSTRUCTION EXECUTED WILL BE JMP (R0)
2523							;BRANCHING THE PROGRAM TO 13\$
2524	004656	105237	000306	15\$:	INCB	2#PASFLG	;INCREMENT PASFLG BY 1.
2525	004662	000631			BR	2\$	
2526							
2527	004664	032777	000020 173556	16\$:	BIT	#20,2\$WR	;HAS THE PRINTOUTS BEEN SUPRESSED ?
2528	004672	001017			BNE	18\$	;IF SO THEN GO TO 18\$
2529	004674	105737	000042		TSTB	2#42	;IS THE PROGRAM RUNNING UNDER ACT?
2530	004700	001014			BNE	18\$	;BRANCH IF YES
2531	004702	004767	001660		JSR	PC,PNTMES	;TYPE THE BANK UNDER TEST
2532	004706	051524	030524 020063		.ASCIZ	/TST13 BANK/	
2533	004714	040502	045516 000				
2534		004722			.EVEN		
2535	004722	004767	002476		JSR	PC,GETBANK	;GET BANK NO. UNDER TEST INTO DECHRD FOR PRINT.
2536	004726	004767	001662		JSR	PC,STPDEC	;TYPE BANK NO. UNDER TEST

D05

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 54
 DZKMAC.P11 18-FEB-77 12:30 T13 WRITE RECOVERY TEST

2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547

004732 000113

18S: JMP (R3)

;BEGIN EXECUTING MOV R2,-(PC) ,COM -(R1) SEQUENCE IN TES

004734 105137 000307
 004740 012700 000110
 004744 000744

24S: COMB 2#PASFLG+1
 MOV #110,R0
 BR 15S

;PLACE THE OP CODE FOR JMP (R0) IN R0
 ;READ THE MEMORY FOR CORRECT DATA AFTER
 ;INCREMENTING PASFLG TO 2

;TST13 EXITS VIA END12.

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 55
 DZKMAC.P11 18-FEB-77 12:30 PARITY OR RELOCATE?

```

2548 004746 012737 000377 000316 RELOC: MOV      #377,2#BAKPAT
2549 004754 105737 000276          TSTB     2#MMAVA
2550 004760 001065          BNE      CONTMM
2551
2552 004762 032777 001000 173460      BIT      #1000,2#SWR
2553 004770 001046          BNE      CKDONE
2554 004772 105737 000405          TSTB     2#REL
2555 004776 100420          BMI      RELOER
2556 005000 112737 000200 000405      MOVB     #200,2#REL
2557
2558          ;RELOCATE THE DIAGNOSTIC TO HIGHEST AVAILABLE MEMORY
2559
2560
2561 005006 004767 001554          JSR      PC,PNTMES
2562 005012 042522 047514 000103      .ASCIZ  /RELOC/
2563          .EVEN
2564 005020 013705 000340          MOV      2#MAXMEM,R5
2565
2566 005024 014445          2S:      MOV      -(R4),-(R5)
2567 005026 020427 000430          CMP      R4,#BEGIN-50
2568 005032 101374          BMI      2S
2569 005034 000165 000050          JMP      50(R5)
2570
2571          ;#RELOCATE THE DIAGNOSTIC BACK TO LOWER MEMORY
2572
2573
2574 005040 013705 000346      RELOER: MOV      2#SAVRS,R5
2575 005044 105737 000405          TSTB     2#REL
2576 005050 100016          BPL      CKDONE
2577
2578 005052 012704 000430          2S:      MOV      #BEGIN-50,R4
2579 005056 012524          MOV      (R5)+,(R4)+
2580 005060 020537 000340          CMP      R5,2#MAXMEM
2581 005064 103774          BLO      2S
2582 005066 105037 000405          CLRB     2#REL
2583 005072 010537 000346          MOV      R5,2#SAVRS
2584 005076 012706 000500          MOV      #BEGIN,SP
2585 005102 010637 000350          MOV      SP,2#SAVR6
2586 005106 000137 005112      CKDONE: JMP      2#LOWER
2587
2588
2589
2590 005112 105737 000315      LOWER: TSTB     2#SAVKBB
2591 005116 001073          BNE      STPSTK
2592 005120 004767 001714      TSTMM: JSR      PC,MMING
2593
2594 005124 105737 000276          TSTB     2#MMAVA
2595 005130 001462          BEQ      ENDPAS
2596 005132 000402          BR      SCNTMM
2597 005134 004767 002052      CONTMM: JSR      PC,UPMM
2598 005140 012703 000324      SCNTMM: MOV      #LOWTWO,R3
2599 005144 004767 002160          JSR      PC,GETSIZ
2600
2601 005150 012704 020000          MOV      #20000,R4
2602
2603 005154 020237 172342          CMP      R2,2#172342
  
```

; IS THE MEMORY MANAGEMENT BEING TESTED ?
 ; IF SO THEN GO TO CONTMM AND CONTINUE TESTING
 ; MEMORY MANAGEMENT
 ; RELOCATION WANTED?
 ; BRANCH IF NO
 ; IF THE PROGRAM HAS ALREADY BEEN RELOCATED THEN ALSO
 ; PLACE THE PROGRAM BACK IN LOWER CORE
 ; OTHERWISE PREPARE TO RELOCATE
 ; TYPE "RELOC"
 ; PREPARE TO LOAD THE PROGRAM IN THE HIGHEST
 ; AVAILABLE MEMORY
 ; RELOCATE THE PROGRAM
 ; NEITHER RELOCATE NOR TEST LOCATIONS LOWER THAN BEGIN-50
 ; RESTORE R5
 ; IS DIAGNOSTIC IN RELOCATED STATE?
 ; BRANCH IF NO
 ; PREPARE TO RELOCATE THE PROGRAM TO LOWER CORE
 ; SAVE R5
 ; RESET STACK TO LOWER MEMORY
 ; "BEGIN" USES THIS TO RESET THE STACK.
 ; TRANSFER THE PROGRAM CONTROL TO THE LOWER CORE
 ; HERE DUE TO IC TYPED?
 ; BRANCH IF YES (TYPE ERROR STACK)
 ; SET THE REGISTERS IF THE MEMORY MANAGEMENT
 ; IS AVAILABLE
 ; IS MEM. MANAG. AVAILABLE ?
 ; BRANCH IF NO
 ; BEGIN TESTING ABOVE 28K
 ; GO TO UPDATE MEM. MANAG. REGISTERS
 ; MAKE R3 POINT TO THE LOCATION LOWTWO
 ; LOAD BITS 6-10 OF R2 WITH THE BITS 13-17
 ; OF THE LOWEST ADDRESS UNDER TEST
 ; MAKE R4 POINT TO THE LOWEST LOCATION IN THE BANK
 ; POINTED BY PAGE ADDRESS REGISTER 1 (PAR1)
 ; IS THE CONTENT OF R2 LOWER THAN THE CONTENT OF

Address	Op Code	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418
---------	---------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

H05

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 58
 DZKMAC.P11 18-FEB-77 12:30 END OF PASS

2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743

;* END OF PASS
 ;*-----

TYPE "END PASS" AND DISABLE PARITY.
 ALSO SERVICE ACT11.
 AND EVERY CONSECUTIVE PASSES UNLESS BIT 4 OF SSWREG IS HIGH

```

SEOP:  CLR      R2                ;SET R2= PARITY MODULE DISABLE CODE
      JSR      PC,PARITY          ;GO DISABLE PARITY MODULES IF SELECTED.
      TSTB     @#SAVKBB           ;CONTROL-C TYPED?
      BNE      CTLC               ;BRANCH IF YES-RESTORE LOADERS AND HALT-
      INC      @#SPASS             ;INCREMENT PASS COUNT
      BIT      @40,SSWR           ;"END PASS #XX" PRINTOUT WANTED?
      BNE      ACT11              ;BRANCH IF NO
      JSR      PC,TPCRLF          ; TYPE CR, LF, AND "END PASS #"
      .ASCIZ   /END PASS #/

      .EVEN
      MOV      @#SPASS,@#DECHRD   ;GET PASS COUNT
      JSR      PC,STPDEC          ;TYPE IT
      MOV      @#42,R0            ;GET THE MONITOR ADDRESS
      BEQ      SDOAGN             ;IF NONE
      JSR      PC,RLODER          ;RESTORE XXDP MONITOR
      RESET                      ;RETURN TO ACT11 MONITOR.

;* SERVICE XXDP/ACT11
      JMP      @#SENDAD           ;JUMP TO ACT SERVICE
SDOAGN: JMP      @#RESTRT         ;REPEAT TEST IF NOT UNDER ACT11/XXDP
RLODER: JSR      PC,CLRMM          ;STOP MEMORY MANAGEMENT SO CAN RESTORE LOADERS
      MOV      @#SAVR4,R4         ;RESTORE R4 WITH SAVR4
4$:     MOV      -(R4),-(R5)       ;RESTORE LOADERS
      CMP      R4,@#ENDSTK
      BHI      4$
      RTS      PC                 ;RETURN FROM RLODER CALL

;CONTROL C HANDLER
CTLC:   JSR      PC,RLODER        ;RESTORE ABS LOADER
      JMP      APTHLT             ;IF NOT APT HALT AT FATHLT
  
```


DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 59
 DZKMAC.P11 18-FEB-77 12:30 ERROR HANDLING ROUTINE

```

2744      * ERROR HANDLING ROUTINE
2745      * -----
2746      *
2747      * PROGRAM COMES HERE EACH TIME AN ERROR IS ENCOUNTERED THIS
2748      * ROUTINE TYPES OUT THE ERROR MESSAGE IN THE FORMAT GIVEN EARLIER
2749      *
2750
2751 005600 017637 000000 000402 ERROR: MOV 2(SP),2$SFATAL ;LOAD THE LOCATION SFATAL WITH THE ERROR NUMBER
2752 005606 010346 1$: MOV R3,-(SP) ;SAVE R3
2753 005610 010046 MOV R0,-(SP) ;AND R0 ON THE STACK
2754
2755 ;SETUP BANK NO. IN FATAL FOR APT
2756
2757 005612 010103 MOV R1,R3 ;GET VIRTUAL ADDRESS UNDER TEST FOR GETBNK
2758 005614 004767 JSR PC,GETBNK ;GET BANK NO. UNDER TEST INTO PBK
2759 005620 013703 MOV 2$PBK,R3 ;GET BANK UNDER TEST
2760 005624 110337 MOVB R3,2$SFATAL+1 ;STORE FAILING BANK NO. FOR APT
2761
2762 ;
2763
2764 005630 010346 MOV R3,-(SP) ;TEMPORARILY STORE R3
2765 005632 012703 MOV 2$R3,R3 ;MAKE R3 AS THE STACK POINTER
2766 005636 013743 MOV 2$PASFLG,-(R3) ;OUTPUT THE WORD STORED AT
2767 005642 005043 2$: CLR -(R3)
2768 005644 113713 MOVB 2$SFATAL,(R3) ;PUT ERROR NO. ON ERROR STACK
2769 005650 016643 MOV 6(SP),-(R3) ;PLACE THE RETURN PC AT (R3)
2770 005654 011143 MOV (R1),-(R3) ;PLACE BAD DATA
2771 005656 010043 MOV R0,-(R3) ;AND GOOD DATA ON THE STACK
2772 005660 005043 CLR -(R3)
2773 005662 016313 MOV 4(R3),(R3) ;TAKE THE
2774 005666 040013 BIC R0,(R3) ;EXCLUSIVE OR OF GOOD AND BAD DATA
2775 005670 046300 BIC 4(R3),R0 ;TO FIND THE BITS THAT FAILED
2776 005674 050013 BIS R0,(R3) ;AND PLACE IT ON THE STACK
2777 005676 012700 MOV 2$ENDPRG--24.,R0 ;THIS CODE BRINGS THE RELATIVE ADDRESS
2778 005702 060700 ADD PC,R0 ;OF THE STARTING OF THE ERROR STACK
2779 005704 062700 6$: ADD 2$18.,R0 ;FOR THE SPECIFIC 4K BANK
2780 005710 005316 DEC (SP)
2781 005712 002374 BGE 6$
2782 005714 005726 TST (SP)+ ;RESTORE THE STACK POINTER
2783
2784 005716 105037 ERRTP: CLRB 2$TYPENB ;DISABLE ANY TYPE OUT
2785 005722 105737 1$: TSTB 2$SPRERR ;IF THIS IS PARITY PROBLEM
2786 005726 001007 BNE 3$ ;THEN GO TO 3$
2787 005730 105720 TSTB (R0)+ ;OTHERWISE INCREMENT THE ERROR STACK POINTER BY 1
2788 005732 105737 000301 TSTB 2$SADERR ;IF THIS IS ADDRESSING PROBLEM
2789 005736 001003 BNE 3$ ;THEN GO TO 3$
2790 005740 105720 TSTB (R0)+ ;INCREMENT THE POINTER R0 BY 1
2791 005742 005713 2$: TST (R3) ;IS BIT 15 OF (R3) SET?
2792 005744 100015 BPL 4$ ;IF NOT THEN GO TO 4$
2793 005746 122710 3$: CMPB 2$377,(R0) ;OTHERWISE SEE IF THIS ERROR HAS OCCURED 377 TIMES
2794 005752 001401 BEQ 5$ ;IF SO DON'T BUMP ERROR COUNT
2795 005754 105210 INCB (R0) ;INCREMENT THE ERROR COUNTER BY 1
2796 005756 122710 5$: CMPB 2$1,(R0) ;MORE THAN 1 ERROR OCCURRED ON THIS BIT?
2797 005762 001404 BEQ 7$ ;BRANCH IF NO
2798 005764 032777 BIT 2$400,2$SWR ;STOP ERROR PRINTOUT AFTER 1 WANTED?
2799 005772 001002 BNE 4$ ;BRANCH IF YES (DON'T TYPE ERROR)

```

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 60
 DZKMAC.P11 18-FEB-77 12:30 ERROR HANDLING ROUTINE

2800	005774	105237	000277		7S:	INCB	3#TYPENB	:ENABLE THE TYPE OUT ROUTINE
2801	006000	105737	000300		4S:	TSTB	3#SPRERR	:PARITY ERROR?
2802	006004	001411				BEQ	6S	:BRANCH IF NO
2803	006006	004767	000406			JSR	PC,TPPRER	:ELSE TYPE "PAR ERR"
2804	006012	000411				BR	8S	:AND DON'T TEST INDIVIDUAL BIT FAILURES.
2805	006014	105737	000301			TSTB	3#SADERR	:ADDRESS ERROR?
2806	006020	001403				BEQ	6S	:BRANCH IF NO
2807	006022	004767	000346			JSR	PC,TPADERR	:PRINT "ADR ERR"
2808	006026	000403				BR	8S	
2809	006030	105720			6S:	TSTB	(R0)+	:POINT TO NEXT ENTRY IN ERROR STACK
2810	006032	006313				ASL	(R3)	:IS THERE STILL AN ERROR BIT SET IN ERROR.
2811	006034	001342				BNE	2S	:OR IF YES - KEEP FILLING ERROR STACK
2812	006036	112737	000006	000314	8S:	MOVB	86,3#TYPCNT	:TELL TYPCNT TO TYPE 6 WORDS OF ERROR STACK.
2813								:THE STACK POINTED BY R3
2814	006044	004767	001150			JSR	PC,PUTADR	:GO TO THE SUBROUTINE TO PLACE THE ADDRESS IN R1
2815								:AT LOCATIONS (R3) AND (R3-2)
2816	006050	004767	000622			JSR	PC,TYPEERR	:TYPE ERROR STACK (7 WORDS)
2817								
2818	006054	005037	000300		10S:	CLR	3#SPRERR	:CLEAR ADDRESS/PARITY ERROR FLAGS
2819	006060	012600				MOV	(SP)+,R0	:RESTORE R0
2820	006062	012603				MOV	(SP)+,R3	:AND R3
2821	006064	105737	000420		FNDERR:	TSTB	3#SENV	:ARE WE RUNNING UNDER APT?
2822	006070	001404				BEQ	2S	:IF NOT THEN TEST FOR HALT
2823	006072	012737	000001	000400		MOV	81,3#MSGTY	:OTHERWISE INFORM THE APT
2824	006100	000443				BR	FATHLT	:GOTO FATHLT AND WAIT FOR APT.
2825								
2826	006102	010246			2S:	MOV	R2,-(SP)	:SAVE R2 TEMP
2827	006104	005777	172340			TST	3#SWR	:DOES THE OPERATOR REQUIRE THE PROGRAM TO HALT
2828								:ON ERROR
2829	006110	100405				BMI	4S	:IF SO THEN HALT ON ERROR
2830								:CHECK FOR CONTROL-C KEY
2831								
2832	006112	004767	001546			JSR	PC,CHECKC	:IF CONTROL-C TYPED THEN PRINT ERROR HISTORY
2833								:AND HALT AT FATHLT.
2834	006116	105737	000042		7S:	TSTB	3#42	:ARE WE RUNNING UNDER ACT?
2835	006122	001401				BEQ	6S	:BRANCH IF NO
2836								
2837	006124	000000			4S:	HALT		:PROGRAM HAS HALTED ON ERROR, R1 IS POINTING
2838								:TO A LOCATION WHICH SHOULD HAVE CONTAINED
2839								:THE WORD STORED IN R0
2840	006126	012602			6S:	MOV	(SP)+,R2	:RESTORE R2
2841	006130	062716	000002			ADD	82,(SP)	:RESTORE THE RETURN ADDRESS
2842	006134	000207				RTS	PC	:RETURN FROM THE SUBROUTINE
2843								
2844								
2845								
2846	006136				FATERR:			
2847	006136	004767	000416		SEQERR:	JSR	PC,TPCRLF	:TYPE "ERROR #"
2848	006142	051105	047522	020122		.ASCIZ	/ERROR #/	
2849	006150	000043						
2850						.EVEN		
2851								
2852	006152	017637	000000	000402		MOV	3(SP),3#SFATAL	:LOAD THE LOCATION SFATAL WITH THE ERROR NUMBER
2853	006160	105237	000314			INCB	3#TYPCNT	:TELL STNUM TO TYPE 1 WORD
2854	006164	012703	000376			MOV	8376,R3	:STNUM USES R3 AS STACK
2855	006170	013743	000402			MOV	3#SFATAL,-(R3)	:PUT ERROR NO. ON STACK

K05

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 61
 DZKMAC.P11 18-FEB-77 12:30 ERROR HANDLING ROUTINE

2856	006174	005743		TST	-(R3)	:STPNM REQUIRES THIS
2857	006176	004767	000562	ISR	PC,FATYP	:TYPE ERROR NO.
2858	006202	105737	000420	APTHLT: TSTB	3#ENV ;RUNNING	:UNDER APT?
2859	006206	001326		BNE	FNDERR	:BRANCH IF YES
2860	006210	000000		FATHLT: HALT	3#RESTR	:FATAL ERROR OR IC HALT
2861	006212	000137	000250	JMP		:RESTART TST BUT DON'T CLEAR PASS COUNT
2862						:IN CASE IC RESTART.
2863						
2864						
2865						
2866						
2867						
2868						
2869						
2870						
2871						
2872						
2873						
2874						
2875						
2876						
2877	006216	012637	000356	PARERR: MOV	(SP)+,3#PARSP	:SET PARSP TO RETURN ADDRESS
2878	006222	011637	000360	MOV	(SP),3#PARPS	:SAVE PSW FOR RETURN
2879	006226	013706	000350	MOV	3#SAVR6,SP	:AND RESET THE SP SINCE NOT ENOUGH STACK ROOM
2880						:TO COMPLETE THE ERROR SERVICE ROUTINE.
2881	006232	010067	000132	MOV	RO,SAVR0	:SAVE RO DURING PARITY SERVICE
2882	006236	010167	000130	MOV	R1,SAVR1	:SAVE R1 DURING PARITY SERVICE
2883	006242	013701	000352	MOV	3#PARMAP,R1	:GET PARITY AVAILABLE MAP
2884	006246	012700	172100	MOV	#172100,RO	:RO= FIRST PARITY ADDRESS.
2885						
2886	006252	005701		TST	R1	:ANY PARITY MODULES AVAILABLE?
2887	006254	001442		BEQ	4\$	:BR IF NO -FATAL ERROR-
2888	006256	000241		CLC		
2889	006260	006001		15: ROR	R1	:SHIFT PARITY MAP BIT INTO C BIT.
2890	006262	103005		BCC	2\$	:BRANCH IF THIS PARITY MODULE NOT AVAILABLE.
2891	006264	005710		TST	(RO)	:PARITY MODULE ERROR BIT SET?
2892	006266	100406		BMI	3\$	:BRANCH IF YES -CALL "ERROR" ROUTINE
2893	006270	020027	172136	CMF	RO,#172136	:DONE ALL PARITY MODULES?
2894	006274	002032		BGE	4\$	:BR IF YES- GO TO FATAL ERROR CALL-
2895	006276	062700	000002	2\$: ADD	#2,RO ;POINT TO NEXT PARITY ADDRESS	
2896	006302	000766		BR	1\$	:AND KEEP TRYING
2897	006304	042710	100000	3\$: BIC	#100000,(RO)	:CLEAR PARITY ERROR BIT.
2898	006310	011001		MOV	(RO),R1	:GET PARITY MODULE CSR
2899	006312	006101		ROL	R1	:SHIFT ERROR ADDRESS BITS 11-5 INTO 15-9
2900	006314	006101		ROL	R1	
2901	006316	006101		ROL	R1	
2902	006320	006101		ROL	R1	
2903	006322	042701	000777	BIC	#777,R1 ;SAVE ERROR ADDRESS ONLY	
2904	006326	105237	000300	INCB	3#SPRERR	:TELL "ERROR" PARITY ERROR CALL.
2905	006332	004767	177242	JSR	PC,ERROR	:#ERROR# REPORT ERROR MESSAGE
2906	006336	000050		SO		:*****ERROR NUMBER 50*****
2907						
2908	006340	016700	000024	MOV	SAVR0,RO	:RESTORE RO
2909	006344	016701	000022	MOV	SAVR1,R1	:RESTORE R1
2910	006350	013746	000360	MOV	3#PARPS,-(SP)	:SET RETURN PSW ON STACK
2911	006354	013746	000356	MOV	3#PARSP,-(SP)	:AND SET RETURN ADDRESS ON STACK

L05

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 62
DZKMAC.P11 18-FEB-77 12:30 ERROR HANDLING ROUTINE

```

2912 006360 000002          RTI          ;RETURN TO TEST WHERE PARITY TRAP OCCURRED.
2913
2914          ;COME HERE IF NO PARITY ERROR FLAG FOUND SET
2915 006362
2916 006362 004767 177550    4S:          JSR      PC,FATERR      ;*ERROR* REPORT ERROR MESSAGE AND HALT AT FATHLT
2917 006366 000051          51          ;*****ERROR NUMBER 51*****
2918
2919
2920          ;RO+R1 ARE SAVED HERE FOR PARITY TRAP DUE TO INSUFFICIENT
2921          ;STACK SPACE BETWEEN 500-450.
2922 006370 000000    SAVRO: 0          ;SAVE RO DURING PARITY TRAP SERVICE
2923 006372 000000    SAVR1: 0          ;SAVE R1 DURING PARITY TRAP SERVICE
2924
2925
2926 006374 105737 000277    TPADER: TSTB    2#TYPENB      ;TYPE ERROR?
2927 006400 001406          BEQ      1$          ;BRANCH IF NO
2928 006402 004767 000160    JSR      PC,PNTMES          ; TYPE CR, LF AND "ADR ER"
2929 006406 042101 020122 051105    .ASCIZ  /ADR ERR/
2930 006414 000122
2931
2932 006416 000207          1$:      .EVEN
2933          RTS      PC
2934
2935 006420 105737 000277    TPPRER: TSTB    2#TYPENB      ;ERROR PRINTOUTS ALLOWED?
2936 006424 001406          BEQ      1$          ;BRANCH IF NO
2937 006426 004767 000134    JSR      PC,PNTMES          ;GO TO TYPE CR, LF AND "PAR ERR"
2938 006432 040520 020122 051105    .ASCIZ  /PAR ERR/
2939 006440 000122
2940 006442 000207          1$:      .EVEN
          RTS      PC

```


M05

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 63
DZKMAC.P11 18-FEB-77 12:30 TYPE OUT ROUTINE

```

2941
2942
2943      * TYPE OUT ROUTINE
2944      *-----*
2945      *
2946      * THIS ROUTINE IS USED BY THE PROGRAM TO TYPE OUT ANY CHARACTER
2947      *
2948
2949 006444 010146 000002 NOTYP: MOV R1, -(SP)
2950 006446 016601 4S: MOV 2(SP), R1
2951 006452 105721 TSTB (R1)+
2952 006454 001376 BNE 4S
2953 006456 000412 BR RETTYP
2954 006460 010146 STYPE: MOV R1, -(SP)
2955 006462 010046 MOV RO, -(SP)
2956 006464 016601 000004 MOV 4(SP), R1
2957 006470 112100 2S: MOVB (R1)+, RO
2958 006472 001403 BEQ 4S
2959 006474 004767 000022 JSR PC, STPCHR
2960 006500 000773 BR 2S
2961 006502 012600 4S: MOV (SP)+, RO
2962 006504 005201 RETTYP: INC R1
2963 006506 042701 000001 BIC R1, R1
2964 006512 010166 000002 MOV R1, 2(SP)
2965 006516 012601 MOV (SP)+, R1
2966 006520 000416 BR EXTYP
2967
2968 006522 132737 000040 000421 STPCHR: BITB #40, 2#SENVH
2969 006530 001005 BNE 4S
2970 006532 105737 177564 2S: TSTB 2#STPS
2971 006536 100375 BPL 2S
2972 006540 110037 177566 MOVB RO, 2#STPB
2973 006544 000404 4S: BR EXTYP
2974
2975 006546 004767 177706 PCRLF: JSR PC, STYPE
2976 006552 005015 000 .ASCIZ <15><12>
2977 006556 000207 EVEN
2978 EXTYP: RTS PC ; RETURN
2979
2980 006560 004767 177762 TPCRLF: JSR PC, PCRLF
2981 006564 000735 BR STYPE
2982
2983
2984 006566 032777 000020 171654 PNTMES: BIT #20, 2#SWR
2985 006574 001323 BNE NOTYP
2986 006576 123737 000042 000046 CMPB 2#42, 2#46
2987 006604 001717 BEQ NOTYP
2988 006606 000764 BR TPCRLF

```

```

; IF THIS TYPE OUT HAS BEEN SUPRESSED THEN
; PREPARE TO RETURN
; SAVE R1
; AND RO ON THE STACK
; PLACE THE ADDRESS OF MESSAGE TO BE TYPED IN R1
; PLACE THE BYTE TO BE TYPED IN RO
; IF IT IS END OF MESSAGE THEN GO TO 4S
; OTHERWISE GO TO TYPE THE CONTENTS OF RO
; RESTORE RO
; CAUSE R1 TO
; POINT TO EVEN ADDRESS
; MODIFY THE RETURN ADDRESS
; RESTORE R1
; AND RETURN VIA RTS PC
; HAVE TYPE OUTS BEEN DISABLED?
; IF SO THEN RETURN FROM THE SUBROUTINE
; WAIT HERE
; UNTIL THE PRINTER IS READY
; LOAD DATA TO BE TYPED INTO DATA REG.
; RETURN
; CR/LF
; TYPE CR/LF
; NOW GO TO TYPE THE REST OF THE MESSAGE
; PRINTOUTS ALLOWED?
; BRANCH IF NO
; RUNNING UNDER ACT 11?
; BRANCH IF YES -NOT PRINTOUT-
; SEND CR/LF AND TYPE MESSAGE.

```

N05

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 64
DZKMAC.P11 18-FEB-77 12:30 ROUTINE TO TYPE OUT A DECIMAL NUMBER

```

2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999 006610 004767 177732 TYPDEC: JSR PC,PCRLF ;TYPE CR/LF
3000 006614 005046 STPDEC: CLR -(SP)
3001 006616 013746 000312 MOV 2#DECHRD,-(SP) ;GET THE WORD THAT HAS TO BE CONVERTED TO A
3002 006622 162716 000012 2S: SUB #10.,(SP) ;DECIMAL NUMBER
3003 006626 002403 BLT 4S ;IF THE NUMBER IN (SP) WAS LESS THAN 10. THEN
3004 ;GO TO 4S
3005 006630 005266 000002 INC 2(SP) ;OTHERWISE ADD 1 TO THE LOCATION STORING 10'S DIGIT
3006 006634 000772 BR 2S ;AND RETURN TO 2S
3007 006636 062716 000012 4S: ADD #10.,(SP)
3008 006642 052716 000060 BIS #60,(SP) ;MAKE THE CONTENTS OF (SP) A DECIMAL NUMBER
3009 006646 112667 000020 MOVB (SP)+,6S-2 ;PLACE THE 1'S DIGIT TO BE TYPED
3010 006652 052716 000060 BIS #60,(SP) ;MAKE THE CONTENTS OF (SP) A DECIMAL NUMBER
3011 006656 112667 000007 MOVB (SP)+,6S-3 ;PLACE THE 10'S DIGIT TO BE TYPED
3012 006662 004767 177572 JSR PC,STYPE ;GO TO TYPE THE NUMBER IN DECIMAL FOLLOWED BY
3013 ;3 SPACES
3014 006666 020040 030040 000060 .ASCIZ / 00/
3015 .EVEN
3016 006674 000207 6S: RTS PC ;RETURN FROM THE SUBROUTINE

```


DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 65
 DZKMAC.P11 18-FEB-77 12:30 OCTAL TYPE OUT ROUTINE

```

3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032 006676 032777 020000 171544 TYPERR: BIT #20000, JSNR ;ERROR PRINTOUT WANTED?
3033 006704 001054 BNE OCTXT ;BRANCH IF NO
3034 006706 004767 177634 JSR PC, PCRLF ;TYPE CR/LF
3035 006712 004767 000012 JSR PC, TYPOCT ;TYPE OCTAL NO.
3036 006716 000447 BR OCTXT ;RETURN VIA RTS PC
3037 006720 012123 OCTTYP: MOV (R1)+, (R3)+ ;PLACE THE HIGH ORDER BITS AT LOCATION POINTED
3038 ;BY R3
3039 006722 012113 MOV (R1)+, (R3) ;AND NOW PLACE THE LOW ORDER BITS
3040 006724 105237 000314 INCB #2, TYPCNT ;ENABLE THE TYPE OUT OF ONE OCTAL WORD
3041 006730 052743 000004 TYPOCT: BIS #4, -(R3)
3042 006734 106113 2S: ROLB (R3)
3043 006736 103376 BCC 2S
3044 006740 005000 CLR RO
3045 006742 106113 ROLB (R3) ;GET BITS 17 & 16 INTO RO
3046 006744 006100 ROL RO
3047 006746 106113 ROLB (R3)
3048 006750 006100 ROL RO
3049 006752 000405 BR STPNUM
3050 006754 004767 177500 RPTOCT: JSR PC, $TYPE ;TYPE 3 SPACES
3051 006760 020040 .ASCIZ / /
3052 .EVEN
3053 006764 005000 FATYP: CLR RO
3054 006766 012723 000006 $TPNUM: MOV #6, (R3)+ ;ENABLE THE TYPE OUT OF 6 OCTAL DIGITS
3055 006772 000241 4S: CLC
3056 006774 006113 ROL (R3)
3057 006776 006100 ROL RO ;PLACE THE CARRY FROM (R3) IN RO
3058 007000 052700 000060 BIS #60, RO ;OR THE CONTENTS OF RO WITH AN ASCII 0
3059 007004 004767 177512 JSR PC, $TPCHR ;TYPE THE OCTAL NUMBER STORED IN RO
3060 007010 005000 CLR RO
3061 007012 006113 ROL (R3)
3062 007014 006100 ROL RO ;PLACE THE CARRY FROM (R3) IN RO
3063 007016 006113 ROL (R3)
3064 007020 006100 ROL RO ;PLACE THE CARRY FROM (R3) IN RO
3065 007022 105363 177776 DECB -2(R3) ;IF WE HAVEN'T TYPED THE 6 OCTAL DIGITS
3066 007026 001361 BNE 4S ;THEN REPEAT FROM 4S
3067 007030 105337 000314 DECB #2, TYPCNT ;IF ALL THE WORDS REQUIRED HAVE NOT BEEN
3068 007034 001347 BNE RPTOCT ;TYPED THEN REPEAT FROM RPTOCT
3069 007036 000207 OCTXT: RTS PC

```

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 66
 DZKMAC.P11 18-FEB-77 12:30 SUBROUTINE FOR MEMORY MANAGEMENT

```

3070
3071
3072
3073
3074
3075
3076
3077
3078
3079 007040 012702 001400 MEMING: MOV #1400,R2
3080 007044 105037 000276 MMREG: CLRB #MMAVA
3081
3082 007050 032777 010000 171372 BIT #10000,2SMR
3083 007056 001441 BEQ RETMM
3084 007060 012700 000004 MOV #4,R0
3085 007064 012720 007164 MOV #NOMM,(R0)+
3086 007070 012710 000340 MOV #340,(R0)
3087 007074 005037 177572 CLR #SRO
3088 007100 105237 000276 INCB #MMAVA
3089
3090 007104 012701 172340 MOV #172340,R1
3091 007110 005021 CLR (R1)+
3092 007112 062702 000200 25: ADD #200,R2
3093 007116 010221 MOV R2,(R1)+
3094 007120 020127 172356 CMP R1,#172356
3095 007124 103772 BLO 25
3096 007126 012711 007600 MOV #7600,(R1)
3097 007132 012701 172300 MOV #172300,R1
3098 007136 012721 077406 45: MOV #77406,(R1)+
3099 007142 020127 172316 CMP R1,#172316
3100 007146 101773 BLOS 45
3101 007150 005237 177572 INC #SRO
3102 007154 005010 SRETM: CLR (R0)
3103 007156 012740 000104 MOV #BUSER,-(R0)
3104 007162 000207 RETMM: RTS PC
3105
3106 007164 022626 NOMM: CMP (SP)+,(SP)+
3107 007166 004767 177366 JSR PC,TPCRLF
3108 007172 047516 046440 043516 .ASCIZ /NO MNG/
3109 007200 000 .EVEN
3110 007202 004767 176730 JSR PC,FATERR
3111 007206 000052 52
3112
3113
3114 007210 000761 BR SRETM
3115
3116 007212 013702 172354 UPMM: MOV #172354,R2
3117 007216 000712 BR MMREG

```

ROUTINE TO SET UP MEMORY MANAGEMENT REGISTERS

PROGRAM CONTROL COMES HERE TO DETERMINE IF THE MEMORY MANAGEMENT IS AVAILABLE OR NOT, AND IF IT IS AVAILABLE THEN WHETHER THE MEMORY ABOVE 28K IS REQUIRED TO BE TESTED OR NOT.

; CLEAR THE BYTE THAT IS SUPPOSED TO INDICATE
 ; THAT MEM. MANAG. IS AVAILABLE FOR TESTING
 ; HAS THE OPERATOR ASKED TO CHECK MEMORY MANAG. ?
 ; IF NOT THEN RETURN FROM THE SUBROUTINE
 ; PREPARE TO SETUP TIME OUT VECTOR
 ; RETURN ADDRESS TO NOMM
 ; AND WITH A PSM OF 340
 ; TRY TO REACH MEM. MANAG. SRO
 ; IF IT IS AVAILABLE THEN SET MEM. MANAG. AVAILABLE
 ; BYTE
 ; R1 IS POINTING TO PAR0
 ; PAR0 WILL POINT TO BANK 0
 ;
 ; SETUP PAR1-PAR6
 ;
 ; PAR7 IS POINTING TO THE I/O PAGE
 ;
 ; SETUP PDR0-PDR7
 ;
 ; ENABLE MEM. MANAG.
 ; RESTORE TIME OUT TRAP VECTOR FOR ANY FUTURE TRAP
 ;
 ; RESTORE STACK POINTER
 ; TYPE "NO MEMORY MANAGEMENT MESSAGE"
 ;
 ; *ERROR* REPORT ERROR MESSAGE AND HALT AT FATHLT
 ; *****ERROR NUMBER 52*****
 ;
 ; RESTORE TIME OUT TRAP VECTOR
 ;
 ; PREPARE TO UPDATE MEMORY MANAG. REGISTERS

:# 18 BIT ADDRESS GENERATOR

THIS SUBROUTINE IS USED TO PLACE THE ADDRESS STORED IN R1
IN THE LOCATION POINTED BY R3. THE ADDRESS IN R1 IS CONVERTED
TO AN 18 BIT ADDRESS ONLY IF MEM. MANAG. IS AVAILABLE IN WHICH
CASE THE HIGH ORDER BITS OF THE ADDRESS ARE PLACED IN LOCATION
POINTED BY R3-2

PUTADR:	CLR	-2(R3)			
	MOV	R1, (R3)			: PLACE THE ADDRESS STORED IN R1 IN LOCATION (R3)
	TSTB	25HAVA			: IS THE MEM. MANAG. AVAILABLE ?
	BEQ	6S			: IF NOT THEN RETURN FROM THE SUBROUTINE
	MOV	R1, -(SP)			: SAVE R1
	BIC	#17777, R1			: CLEAR BITS 0-12 OF THE ADDRESS IN R1
	BIC	R1, (R3)			: LEAVE BITS 0-12 OF THE ADDRESS IN (R3)
	BIS	#4000, R1			: PREPARE TO SHIFT R1 BY 12 PLACES
2S:	ROR	R1			
	BCC	2S			: GET THE NUMBER OF PAR IN R1
	ADD	#172340, R1			: GET THE ADDRESS OF PAR IN R1
	MOV	(R1), R1			: LOAD R1 WITH THE CONTENTS OF PAR
	BIS	#10000, R1			
4S:	ROL	R1			
	BCC	4S			: PLACE THE ADDRESS BITS 13-17 IN BITS 11-15 OF R1
	ROL	R1			
	ROL	-(R3)			: PLACE BIT 17 IN LOCATION POINTED BY R3-2
	ROL	R1			
	ROL	(R3)+			: PLACE BIT 16 OF THE ADDRESS
	BIS	R1, (R3)			: PLACE BITS 13-15 OF THE ADDRESS IN LOCATION (R3)
	MOV	(SP)+, R1			: RESTORE R1
6S:	RTS	PC			: RETURN FROM THE SUBROUTINE

GET ADDRESS FROM THE APT MAILBOX

THIS SUBROUTINE IS USED TO GET ADDRESS FROM APT MAILBOX AND PLACE IT IN THE LOCATION USED BY THE PROGRAM TO DEFINE THE MEMORY BOUNDRIES.
PROGRAM CONTROL SHOULD COME TO THIS SUBROUTINE WITH R1 POINTING TO THE MEMORY TYPE IN THE APT MAILBOX AND R3 POINTING TO THE LOCATION+2 WHERE THE LOW ORDER BITS OF THE ADDRESS HAVE TO BE PLACED

```

GETADR:  MOV     1(R1),-(R3)      ;PLACE THE LOW ORDER BITS OF THE ADDRESS
        CLR     -(R3)           ;CLEAR THE LOCATION WHERE THE HIGH ORDER BITS
                                ;HAVE TO BE PLACED
        MOVB    -1(R1),(R3)      ;PLACE BITS 16 & 17
2$:      RTS     PC              ;RETURN FROM THE SUBROUTINE

```

E06

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 68
 DZKNAC.P11 18-FEB-77 12:30 CONVERT 18 BIT ADDRESS TO THE PAR FORM

```

3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179 007324 105237 000315      $GTSIZ: INCB      2#SAVKBB      ;PREPARE TO PLACE ADDRESS BITS 13-17 IN BITS
3180                                     ;0-4 OF R2
3181
3182 007330 012301      GETSIZ: MOV      (R3)+,R1
3183 007332 011302      MOV      (R3),R2      ;LOAD R2 WITH THE LOW ORDER BITS OF THE ADDRESS
3184 007334 042702 017777      BIC      #17777,R2      ;CLEAR ADDRESS BITS 0-12
3185 007340 052702 000040      2$: BIS      #40,R2
3186 007344 006001      4$: ROR      R1
3187 007346 006002      ROR      R2      ;ROTATE R1 AND R2 7 TIMES
3188 007350 103375      BCC      4$
3189 007352 105737 000315      TSTB     2#SAVKBB
3190 007356 001405      BEQ      6$
3191 007360 105037 000315      CLRB     2#SAVKBB
3192 007364 052702 000100      BIS      #100,R2
3193 007370 000765      BR       4$
3194 007372 012301      6$: MOV      (R3)+,R1      ;PLACE THE LOW ORDER ADDRESS BITS IN R1
3195 007374 012700 160000      MOV      #160000,R0
3196 007400 040001      BIC      R0,R1      ;LEAVE BITS 0-12 OF THE ADDRESS IN R1
3197 007402 000207      RTS       PC      ;RETURN FROM THE SUBROUTINE
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207 007404 105737 000276      CLRMM: TSTB     2#MMAVA      ;WAS THE MEMORY MANAGEMENT ENABLED ?
3208 007410 001404      BEQ      1$      ;IF NOT THEN GO TO 1$
3209 007412 005037 177572      CLR      2#SRO      ;DISABLE THE MEMORY MANAGEMENT
3210 007416 105037 000276      CLRB     2#MMAVA      ;AND DO NOT ATTEMPT TO TEST MEM. MANAG.
3211 007422 000207      1$: RTS       PC      ;RETURN FROM THE SUBROUTINE
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221 007424 010046      ;* GET BANK NO. UNDER TEST
3222 007426 010346      ;CALLED BY ERTYP AND TST13 TO GET BANK NO. UNDER TEST INTO PBANK.
3223 007430 042703 017777      ;REGISTERS
3224 007434 052703 010000      ;R0=POINTER TO PAR UNDER TEST
3225                                     ;R3=VIRTUAL ADDRESS ON ENTRY
3226                                     ;R0+R3 ARE RESTORED ON EXIT.
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4159
4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279
4280
4281
4282
4283
4284
4285
4286
4287
4288
4289
4290
4291
4292
4293
4294
4295
4296
4297
4298
4299
4300
4301
4302
4303
4304
4305
4306
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4390
4391
4392
4393
4394
4395
4396
4397
4398
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444
4445
4446
4447
4448
4449
4450
4451
4452
4453
4454
4455
4456
4457
4458
4459
4460
4461
4462
4463
4464
4465
4466
4467
4468
4469
4470
4471
4472
4473
4474
4475
4476
4477
4478
4479
4480
4481
4482
4483
4484
4485
4486
4487
4488
4489
4490
4491
4492
4493
4494
4495
4496
4497
4498
4499
4500
4501
4502
4503
4504
4505
4506
4507
4508
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4520
4521
4522
4523
4524
4525
4526
4527
4528
4529
4530
4531
4532
4533
4534
4535
4536
4537
4538
4539
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4550
4551
4552
4553
4554
4555
4556
4557
4558
4559
4560
4561
4562
4563
4564
4565
4566
4567
4568
4569
4570
4571
4572
4573
4574
4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585
4586
4587
4588
4589
4590
4591
4592
4593
4594
4595
4596
4597
4598
4599
4600
4601
4602
4603
4604
4605
4606
4607
4608
4609
4610
4611
4612
4613
4614
4615
4616
4617
4618
4619
4620
4621
4622
4623
4624
4625
4626
4627
4628
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662
4663
4664
4665
4666
4667
4668
4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685
4686
4687
4688
4689
4690
4691
4692
4693
4694
4695
4696
4697
4698
4699
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720
4721
4722
4723
4724
4725
4726
4727
4728
4729
4730
4731
4732
4733
4734
4735
4736
4737
4738
4739
4740
4741
4742
4743
4744
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4779
4780
4781
4782
4783
4784
4785
4786
4787
4788
4789
4790
4791
4792
4793
4794
4795
4796
4797
4798
4799
4800
4801
4802
4803
4804
4805
4806
4807
4808
4809
4810
4811
4812
4813
4814
4815
4816
4817
4818
4819
4820
4821
4822
4823
4824
4825
4826
4827
4828
4829
4830
4831
4832
4833
4834
4835
4836
4837
4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4850
4851
4852
4853
4854
4855
4856
4857
4858
4859
4860
4861
4862
4863
4864
4865
4866
4867
4868
4869
4870
4871
4872
4873
4874
4875
4876
4877
4878
4879
4880
4881
4882
4883
4884
4885
4886
4887
4888
4889
4890
4891
4892
4893
4894
4895
4896
4897
4898
4899
4900
4901
4902
4903
4904
4905
4906
4907
4908
4909
4910
4911
4912
4913
4914
4915
4916
4917
4918
4919
4920
4921
4922
4923
4924
4925
4926
4927
4928
4929
4930
4931
4932
4933
4934
4935
4936
4937
4938
4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977
4978
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
5000
5001
5002
5003
5004
5005
5006
5007
5008
5009
5010
5011
5012
5013
5014
5015
5016
5017
5018
5019
5020
5021
5022
5023
5024
5025
5026
5027
5028
5029
5030
5031
5032
5033
5034
5035
5036
5037
5038
5039
5040
5041
5042
5043
5044
5045
5046
5047
5048
5049
5050
5051
5052
5053
5054
5055
5056
5057
5058
5059
5060
5061
5062
5063
5064
5065
5066
5067
5068
5069
5070
5071
5072
5073
5074
5075
5076
5077
5078
5079
5080
5081
5082
5083
5084
5085
5086
5087
5088
5089
5090
5091
5092
5093
5094
5095
5096
5097
5098
5099
5100
5101
5102
5103
5104
5105
5106
5107
5108
5109
5110
5111
5112
5113
5114
5115
5116
5117
5118
5119
5120
5121
5122
5123
5124
5125
5126
5127
5128
5129
5130
5131
5132
5133
5134
5135
5136
5137
5138
5139
5140
5141
5142
5143
5144
5145
5146
5147
5148
5149
5150
5151
5152
5153
5154
5155
5156
5157
5158
5159
5160
5161
5162
5163
5164
5165
5166
5167
5168
5169
5170
5171
5172
5173
5174
5175
5176
5177
5178
5179
5180
5181
5182
5183
5184
5185
5186
5187
5188
5189
5190
5191
5192
5193
5194
5195
5196
5197
5198
5199
5200
5201
5202
5203
5204
5205
5206
5207
5208
5209
5210
5211
5212
5213
5214
5215
5216
5217
5218
5219
5220
5221
5222
5223
5224
5225
5226
5227
5228
5229
5230
5231
5232
5233
5234
5235
5236
5237
5238
5239
5240
5241
5242
5243
5244
5245
5246
5247
5248
5249
5250
5251
5252
5253
5254
5255
5256
5257
5258
5259
5260
5261
5262
5263
5264
5265
5266
5267
5268
5269
5270
5271
5272
5273
5274
5275
5276
5277
5278
5279
5280
5281
5282
5283
5284
5285
5286
5287
5288
5289
5290
5291
5292
5293
5294
5295
5296
5297
5298
5299
5300
5301
5302
5303
5304
5305
5306
5307
5308
5309
5310
5311
5312
5313
5314
5315
5316
5317
5318
5319
5320
5321
5322
5323
5324
5325
5326
5327
5328
5329
5330
5331
5332
5333
5334
5335
5336
5337
5338
5339
5340
5341
5342
5343
5344
5345
5346
5347
5348
5349
5350
5351
5352
5353
5354
5355
5356
5357
5358
5359
5360
5361
5362
5363
5364
5365
5366
5367
5368
5369
5370
5371
5372
5373
5374
5375
5376
5377
5378
5379
5380
5381
5382
5383
5384
5385
5386
5387
5388
5389
5390
5391
5392
5393
5394
5395
5396
5397
5398
5399
5400
5401
5402
5403
5404
5405
5406
5407
5408
5409
5410
5411
5412
5413
5414
5415
5416
5417
5418
5419
5420
5421
5422
5423
5424
5425
5426
5427
5428
5429
5430
5431
5432
5433
5434
5435
5436
5437
5438
5
```


F06

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 69
DZKMAC.P11 18-FEB-77 12:30

SUBROUTINE TO DISABLE MEMORY MANAGEMENT

```

3225 007440 000241          CLC
3226 007442 006003          1$: ROR    R3          ;SHIFT A BANK BIT
3227 007444 103376          BCC    1$          ;UNTIL IN BITS <2:0> OF R3
3228 007446 105737 000276   TSTB   2$MMAVA      ;MEMORY MANAGEMENT UNDER TEST?
3229 007452 001407          BEQ    2$          ;NO EXIT
3230
3231          ;GET PAR ADDRESS AND PHYSICAL BANK NO.
3232 007454 006303          ASL     R3          ;MAKE R3 PAR ADDRESS OFFSET.
3233 007456 062703 172340   ADD     #172340,R3  ;MAKE FULL PAR ADDRESS.
3234 007462 011300          MOV     (R3),R0     ;GET PAR CONTENTS
3235 007464 006300          ASL     R0
3236 007466 000300          SWAB   R0
3237 007470 110003          MOV     R0,R3      ;SHIFT BANK BITS TO BITS <7:0>
3238 007472 010337 000312   MOV     R3,2$PBK    ;SET R3 TO PHYSICAL BANK NO.
3239 007476 012603          MOV     (SP)+,R3   ;STORE PHYSICAL BANK NO.
3240 007500 012600          MOV     (SP)+,R0   ;RESTORE R3
3241 007502 000207          RTS    PC          ;RESTORE R0
3242                                     ;RETURN TO CALLER
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263

```

: PARITY ENABLE/DISABLE ROUTINE

THIS ROUTINE ENABLES OR DISABLES PARITY MODULES AND PRINTS ASSOCIATED MEESSAGES.
IF PARITY AVAILABLE THEN BIT13 OF "REL" IS SET AND "PAR"ITY IS PRINTED.
ALSO THE BACKGROUND TEST PATTERN (LOC. BAKPAT) IS SET=376

: REGISTER USAGE.

:R0= POINTS TO BUS TIMEOUT TRAP VECTOR (LOC. 4)

:R1= HOLDS PARITY MODULE UNIBUS ADDRESS.

:R2= ON ENTRY HOLDS ENABLE/DISABLE CODE .

IF R2=0 THEN DISABLE

IF R2=1 THEN ENABLE

:R3= SCRATCH TO SETUP LOC. PARMAP WITH A MAP OF PARITY MODULES PRESENT.

:CALL IS

```

MOV     #1,R2 ;ENABLE CODE
JSR     PC,PARITY

```

```

3264 007504 032777 004000 170736 PARITY: BIT    #4000,2$WR ;PARITY TEST WANTED?
3265 007512 001460          BEQ    6$          ;BRANCH IF NO
3266
3267 007514 012700 000004          MOV     #4,R0 ;POINT R0 TO BUS TIMEOUT ADDRESS.
3268 007520 012710 000122          MOV     #5$-6,(R0) ;SET RETURN FROM TIMEOUT TRAP TO 5$
3269 007524 060710          ADD     PC,(R0) ;IN THE CURRENT BANK.
3270 007526 005037 000352          1$: CLR     2$PARMAP ;CLEAR PARITY MAP HOLDER.
3271 007532 012701 172140          MOV     #172140,R1 ;SET R1 TO LAST PARITY MODULE ADDRESS+2
3272 007536 012703 100000          MOV     #100000,R3 ;SET R3 TO PARMAP AVAILABLE CODE BEGIN.
3273 007542 010241          2$: MOV     R2,-(R1) ;ENABLE A PARITY MODULE+TRAP IF NOT AVAILABLE.
3274 007544 050337 000352          BIS     R3,2$PARMAP ;NO TRAP TO 5$, SO SET PARITY AVAILABLE.
3275 007550 000241          CLC
3276 007552 006003          3$: ROR     R3          ;SETUP NEXT PARMAP BIT
3277 007554 103372          BCC    2$          ;BRANCH IF NOT DONE ALL PARITY ADDRESSES.
3278 007556 012710 000104          MOV     #BUSER,(R0) ;RESET BUS TIMEOUT TRAP VECTOR
3279 007562 005702          TST     R2          ;IS THIS A DISABLE CALL?
3280 007564 001433          BEQ    6$          ;BRANCH IF YES (EXIT)

```

G06

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 70
 DZKMAC.P11 18-FEB-77 12:30 SUBROUTINE TO DISABLE MEMORY MANAGEMENT

```

3281 007566 005737 000352      TST      3#PARMAP      ;WERE ANY PARITY MODULES FOUND?
3282 007572 001011      BNE      4$      ;BRANCH IF YES
3283 007574 004767 176760      JSR      PC,TPCRLF      ;PRINT "NO PAR"
3284 007600 047516 050040 051101 .ASCIZ  /NO PAR/
3285 007606      000
3286      007610
3287 007610 004767 176322      JSR      PC,FATERR      ;*ERROR* REPORT ERROR MESSAGE AND HALT AT FATHLT
3288 007614 000053      53      ;*****ERROR NUMBER 53*****
3289
3290
3291 007616 152737 000040 000405 4$: BISB      #40,3#REL      ;SET PARITY UNDER TEST FLAG
3292 007624 012737 000376 000316      MOV      #376,3#BAKPAT      ;SET BACKGROUND PATTERN TO
3293      ;WORST CASE PARITY CODE.
3294 007632 004767 176722      JSR      PC,TPCRLF      ;PRINT "TST PARITY"
3295 007636 040520 044522 054524 .ASCIZ  /PARITY/
3296 007644      000
3297      007646
3298 007646 000405      BR      EXITC      ;AND EXIT VIA RTS PC
3299
3300      ;GET HERE IF PARITY ADDRESS TIMED OUT TO LOC. 4
3301

```


H06

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 71

DZKMAC.P11 18-FEB-77 12:30

SUBROUTINE TO DISABLE MEMORY MANAGEMENT

```

3302 007650 022626      55:    CMP    (SP)+,(SP)+    ;RESET STACK FROM TRAP
3303 007652 000737      BR      35                  ;KEEP TRYING PARITY ADDRESSES.
3304
3305 007654 142737 000040 000405 65:    BICB    #40,2#REL    ;CLEAR PARITY TESTING FLAG
3306 007662      EXITC:      ;
3307 007662 000207      75:    RTS     PC              ;RETURN TO CALLER
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318

```

;CHECKC

THIS ROUTINE CHECKS IF CONTROL-C WAS TYPED AT THE END OF EACH
TEST OR IN THE ERROR TYPE ROUTINE.
IF CONTROL-C TYPED THE PROGRAM IS RETURNED TO LOWER MEMORY IF IT WAS
RELOCATED AND THE ERROR HISTORY IS PRINTED OUT.
FINALLY IT HALTS AT FATHLT.

```

3319 007664 105037 000315 CHECKC: CLRB    2#SAVKBB    ;INIT CONTROL-C FLAG.
3320 007670 105737 177560      TSTB    2#TKS        ;ANY CHAR. TYPED?
3321 007674 100372      BPL      EXITC        ;BR IF NO-EXIT VIA RTS PC-
3322 007676 113702 177562      MOVB    2#SKBB,R2    ;GET THE CHAR TYPED.
3323 007702 042702 000200      BIC      #200,R2 ;CLEAR THE PARITY BIT.
3324 007706 122702 000003      CMPB    #3,R2      ;IS IT CONTROL-C?
3325 007712 001363      BNE      EXITC        ;BRANCH IF NO -EXIT VIA RTS PC-
3326 007714 110237 000315      MOVB    R2,2#SAVKBB    ;ELSE STORE THE CHAR. FOR USE AS A FLAG.
3327 007720 004767 176634      JSR      PC,TPCRLF    ;PRINT "IC"
3328 007724 041536      .ASCIZ  /IC/
3329      .EVEN
3330 007730 000167 175104      JMP      RELOER    ;GO RETURN PROGRAM TO LOWER CORE IF RELOCATED.
3331
3332
3333
3334
3335
3336
3337
3338

```

. =7744
ENDPRG: 0

THIS BEGINS THE STORAGE FOR THE ERROR HISTORY
STACK FOR EACH 4K BANK 18. BYTES ARE SAVED.
ALSO THE ABSOLUTE LOADER AND XXDP CODE IS SAVED
AFTER THE ERROR STACK.
FOR 4K MEMORY SIZE THEN PROGRAM=7744+22=7776

.END

000001

[illegible]

J06

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 74
 DZKMAC.P11 18-FEB-77 12:30 CROSS REFERENCE TABLE -- USER SYMBOLS

CHECKC	007664	1570	2832	3319#															
CKDONE	005106	2553	2576	2586#															
CLAMEM	001466	1497#	2618																
CLAMM	007404	1431	1453	2731	3207#														
CNTSCP	001644	1569	1572#																
CONT	001526	1514#	1583																
CONTHM	005134	2550	2597#																
CTLC	005570	2710	2740#																
DECHRD	000312	1170#	1171	2679#	2682#	2718#	3000												
ENDPAS	005276	2595	2655#																
ENDPRG	007744	1096	1291	2659	2777	3333#													
ENDSTK	000310	1167#	1168	1418#	2663	2734													
END0	002132	1668#	1710																
END1	002232	1710#	1750																
END10	003772	2217	2225#	2368															
END12	004316	2368#	2478	2510															
END2	002342	1750#	1846																
END3	002610	1846#	1885																
END4	002720	1885#	1971																
END5	003066	1971#	2027																
END6	003222	2027#	2104																
END7	003444	2104#	2225																
ERROR	005600	1627	1688	1742	1807	1871	1911	1918	1945	2003	2078	2158	2174	2183					
		2328	2336	2463	2475	2492	2751#	2905											
ERRTYP	005716	2784#																	
EXITC	007662	3298	3306#	3321	3325														
EXTYP	006556	2966	2973	2978#															
FAILNM	005374	2678	2682#	2695															
FATERR	006136	1304	1352	1464	1486	1654	2846#	2916	3111	3287									
FATHLT	006210	2824	2860#																
FATYP	006764	2857	3053#																
FINDER	006064	2821#	2859																
GALLOP	003472	2141#	2275																
GETADR	007310	1355	1356	3164#															
GETBNK	007424	2535	2758	3221#															
GETSIZ	007330	2599	2610	2615	3182#														
HIGHAD	000332	1196#	1341																
HIGHTM	000330	1195#	1340	1409															
LOOP	001536	1517#	1581																
LOWADD	000326	1193#																	
LOWBNK	000304	1160#	1161	2048#	2059	2094													
LOWER	005112	2586	2590#																
LOWTWO	000324	1192#	1402	1455	2598														
M	= 000200	1005#	2492																
MAXADR	005224	2613	2616	2627#															
MAXMEM	000340	1202#	1339	1422#	1644	1840	2441	2564	2580										
MEMING	007040	1375	2592	3079#															
MEMTST	001460	1491#																	
MMAVA	000276	1142#	1145	1377	2549	2594	3080#	3088#	3131	3207	3210#	3228							
MMREG	007044	2609	3080#	3117															
N	= 000054	1005#	1304	1307#	1352	1355#	1464	1467#	1486	1489#	1598	1601#	1612	1614#					
		1615	1618#	1623	1625#	1654	1657#	1680	1683#	1688	1691#	1723	1726#	1742					
		1745#	1767	1770#	1792	1794#	1802	1804#	1818	1820#	1855	1858#	1871	1874#					
		1904	1907#	1911	1914#	1918	1921#	1945	1948#	1993	1996#	2003	2006#	2044					
		2047#	2078	2081#	2135	2138#	2158	2161#	2174	2177#	2183	2186#	2260	2263#					
		2300	2303#	2328	2331#	2336	2339#	2401	2404#	2463	2466#	2475	2478#	2492					

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 75
DZKMAC.P11 18-FEB-77 12:30 CROSS REFERENCE TABLE -- USER SYMBOLS

[illegible]

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 76
DZKMAC.P11 18-FEB-77 12:30 CROSS REFERENCE TABLE -- USER SYMBOLS

SW11 =	004000	1005#												
TBL	001600	1534	1538#	1539	1540	1541	1542	1543	1544	1545	1546	1547	1548	1549
		1550												
TKS =	177560	1180#	3320											
TPADER	006374	2677	2807	2926#										
TPCRLF	006560	1091	2714	2847	2980#	2988	3107	3283	3294	3327				
TPPRER	006420	2694	2803	2934#										
TRYSR	001014	1343	1359#											
TSTG0	001666	1577	1580#											
TSTMH	005120	1467	2592#											
TSTREL	001320	1286	1452#											
TSTRP	000770	1289	1291#											
TSTSCP	001632	1568#	1668											
TSTSIZ	001324	1453#												
TST0	001702	1538	1596#											
TST1	002134	1539	1676#											
TST10	003446	1546	2132#											
TST11	003774	1547	2257#											
TST12	004046	1548	2298#	2367										
TST13	004320	1549	2399#											
TST2	002234	1540	1720#											
TST3	002344	1541	1765#	1844										
TST4	002612	1542	1853#											
TST5	002722	1543	1901#											
TST6	003070	1544	1989#											
TST7	003224	1545	2041#											
TYPCNT	000314	1172#	1279#	2689#	2812#	2853#	3040#	3067#						
TYPDEC	006610	2681	2997#											
TYPEINB	000277	1146#	1149	2784#	2800#	2926	2934							
TYPEOP	005470	2714#												
TYPEERR	006676	2816	3032#											
TYPMEM	001152	1405#												
TYPOCT	006730	3035	3041#											
TYPSIZ	001130	1357	1360	1399#										
TYPSTK	005316	2661#	2667											
UPMH	007212	1387	2597	3116#										
WRTMEM	000120	1049#	1771	1907	1996									
SA =	000001	393#	395	396#	398	399#	403	404#	408	409#	413	414#	419	420#
		426	427#	432	433#	437	438#	441	442#	445	446#	449	450#	453
		454#	457	458#	461	462#	465	466#	469	470#	474	475#	478	479#
		482	483#	486	487#	490	491#	492	493#	500	501#	504	505#	508
		509#	519	520#	523	524#	532	533#	536	537#	539	540#	545	546#
		549	550#	553	554#	557	558#	565	566#	569	570#	573	574#	578
		579#	583	584#	590	591#	596	597#	600	601#	605	606#	609#	
SADERR	000301	1154#	1157	1619#	1740#	1804#	1869#	2788	2805					
SAPTHD	000276	1122	1128#	1141										
SCNTMH	005140	2596	2598#											
SCPUOP	000426	1240#												
SOEVCT	000410	1231#	1575#	2084#	2207#									
SOOAGN	005542	2721	2729#											
SENDAD	000156	1017	1068#	2727										
SENV	000420	1236#	1324	1568	2821	2858								
SENVH	000421	1237#	1342	1639	2968									
SEOP	005440	2656	2664	2707#										
SETABL	000420	1235#												
SETEND	000450	1134	1264#											

M06

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 77
 DZKMAC.P11 18-FEB-77 12:30 CROSS REFERENCE TABLE -- USER SYMBOLS

SFATAL	000402	1228#	2751#	2760#	2768	2852#	2855								
SGTSIZ	007324	1410	1525	3179#											
SHD	= 000002	996													
SHIBTS	000276	1129#													
SHIMAX	000334	1189#	1340#												
SHIBB	= 177562	1181#	3322												
SHADR1	000432	1253#													
SHADR2	000436	1257#													
SHADR3	000442	1260#													
SHADR4	000446	1263#													
SHATL	000400	1085	1130	1134	1226#	1296	1311								
SHAMS1	000430	1247#													
SHAMS2	000434	1255#													
SHAMS3	000440	1258#													
SHAMS4	000444	1261#													
SHAXM	000336	1200#	1341#												
SHBADR	000300	1130#													
SHSGAD	000414	1233#													
SHSGLC	000416	1234#													
SHSGTY	000400	1030#	1227#	2823#											
SHYTP1	000431	1248#	1349												
SHYTP2	000435	1256#													
SHYTP3	000441	1259#													
SHYTP4	000445	1262#	1345												
SHMTST=	000001	1584#	1586	1671#	1673	1712#	1714	1754#	1756	1847#	1849	1887#	1889	1973#	
		1975	2030#	2032	2105#	2107#	2229#	2231#	2276#	2278#	2369#	2371			
SPASS	000406	1230#	1288	2711#	2718										
SPASTH	000304	1132#													
SPRERR	000300	1150#	1153	1278#	2785	2801	2818#	2904#							
SPETHM	007154	3102#	3114												
SSVPC =	000044	1015#	1020												
SSMR =	000000	996#	1005#	1597	1677	1721	1766	1854	1902	1990	2042	2133	2258	2299	
		2400													
SSUREG	000422	1238#	1326												
STESTN	000404	1089	1137	1229#	1515#	1572	1596	1676	1720	1765	1853	1901	1989	2041	
		2132	2189	2197	2222	2257	2298	2399							
STN =	000014	986#	996	1584	1597#	1671	1677#	1712	1721#	1754	1766#	1847	1854#	1887	
		1902#	1973	1990#	2030	2042#	2105	2133#	2229	2258#	2276	2299#	2369	2400#	
STPB =	177566	1183#	2972#												
STPCHR	006522	2959	2968#	3059											
STPDEC	006614	2536	2683	2719	2999#										
STPNUM	006766	3049	3054#												
STPS =	177564	1182#	2970												
STPSTK	005306	2591	2657#												
STSTH	000302	1131#													
STYPE	006460	1405	2954#	2975	2981	3012	3050								
SUNIT	000412	1083	1232#												
SUNITM	000306	1133#													
SUSMR	000424	1239#													
SZ =	000362	1218#													
SZZ =	007734	3332#													
SSH =	000200	2492#													
.	= 007746	1003#	1008#	1015	1016#	1018#	1020#	1022#	1028#	1035#	1074#	1118	1119#	1121#	
		1123#	1141#	1145#	1149#	1153#	1157#	1159#	1161#	1166#	1168#	1171#	1218	1221#	
		1272#	1281	1534	1597	1649	1679	1722	1766	1854	1903	1992	2043	2134	
		2259	2299	2400	2519	2534#	2717#	2777	2977#	3110#	3268	3286#	3297#	3329#	

N06

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 78
DZKMAC.P11 18-FEB-77 12:30 CROSS REFERENCE TABLE -- USER SYMBOLS

.SX = 000276 3332#
1118# 1123

B07

DZKMA MACY11 27(1006) 18-FEB-77 12:35 PAGE 80
 DZKMAC.P11 18-FEB-77 12:30 CROSS REFERENCE TABLE -- MACRO NAMES

ERRLST	393#	395	398	403	408	413	419	426	432	437	441	445	449	453	457
	461	465	469	474	478	482	486	490	492	500	504	508	519	523	532
	536	539	545	549	553	557	565	569	573	578	583	590	596	600	605
MSG	1584#	1586	1671#	1673	1712#	1714	1754#	1756	1847#	1849	1887#	1889	1973#	1975	2030#
	2032	2105#	2107	2229#	2231	2276#	2278	2369#	2371	2105	2229	2276	2369		
NEWTST	986#	1584	1671	1712	1754	1847	1887	1973	2030	2105	2229	2276	2369		
PLCERR	1001#	1612	1615	1623	1792	1802	1818								
STARS	986#	1013	1078	1081	1115	1117	1124	1190	1197	1224	1267	1269	1584	1595	1671
	1675	1712	1719	1754	1764	1847	1852	1887	1900	1973	1988	2030	2040	2105	2131
	2229	2256	2276	2297	2369	2398									
SERRM	1001#	1688	1742	1871	1911	1918	1945	2003	2079	2158	2174	2183	2328	2336	2463
	2475	2492	2905												
SFATAL	394#	395	398	403	408	413	419	426	432	437	441	445	449	453	457
	461	465	469	474	478	482	486	490	492	500	504	508	519	523	532
	536	539	545	549	553	557	565	569	573	578	583	590	596	600	605
SFTERR	1001#	1304	1352	1464	1486	1654	2915	3111	3287						
SSQERR	1001#	1598	1680	1723	1767	1855	1904	1993	2044	2135	2260	2300	2401		
SSNEWT	986#	1584	1671	1712	1754	1847	1887	1973	2030	2105	2229	2276	2369		
.HEADE	986#														
.SACT1	986#	1011													
.SAPT8	986#	1222													
.SAPTH	986#	1113													

. ABS. 007746 000

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

DSKZ:DZKMAC.BIN,DSKZ:DZKMAC.SEG/CRF/SOL=DSKZ:DZKMAC.P11
 RUN-TIME: 7 8 .5 SECONDS
 RUN-TIME RATIO: 82/16=4.8
 CORE USED: 11K (22 PAGES)