Data General

TECHNICAL STATEMENT

TEXT LISTING

068-001066-01

PROGRAM

ARRAY PROCESSOR EXERCISER-B

TAPE

097-001066-01

ABSTRACT

THIS PROGRAM IS A FUNCTIONAL TEST FOR THE ARRAY PROCESSOR (AP). IT IS EXECUTED BY A CENTRAL PROCESSOR (OR IOP) CONTROLLING THE AP AND TESTS SPECIFIC AP INSTRUCTIONS.

```
 0001 AP8.T    MACRO REV 06.30          09:35:56 09/19/79
01                      ;*****************************************************************
02                      ;
03                      ; NAME: AP8.TX                              PART NUMBER 097-1066
04                      ;
05                      ;
06                      ; DESCRIPTION: ARRAY PROCESSOR EXERCISER-8.
07                      ;
08                      ;
09                      ; REVISION HISTORY:
10                      ;
11                      ;       REV.            DATE
12                      ;
13                      ;       00              03/29/78
14                      ;       01              12/29/78
15                      ;
16                      ;
17                      ; COPYRIGHT (C) DATA GENERAL CORPORATION, 1978
18                      ; ALL RIGHTS RESERVED.
19                      ;*****************************************************************
```

```
!0002 AP8.T
01
02                      ;
03                      ;       AP8.TX                  PART NUMBER: 097-1066
04                      ;
05                      ;
06                      ;
07                      ;1.0    PROGRAM NAME
08                      ;
09                      ;       SOURCE FILE:    AP8.SR
10                      ;       DIOS FILE:      AP8 EXER
11                      ;
12                      ;2.0    REVISION HISTORY
13                      ;
14                      ;       DATE            REVISION
15                      ;       ----            --------
16                      ;       XX/XX/78           00
17                      ;
18                      ;3.0    MACHINE REQUIREMENTS
19                      ;
20                      ;       1. ECLIPSE FAMILY CENTRAL PROCESSOR  (HOST OR IOP)
21                      ;          WITH AT LEAST 16.K READ/WRITE MEMORY
22                      ;       2. ARRAY PROCESSOR BOARDS AP1, AP2, AP3
23                      ;       3. BASIC I/O TELETYPE INTERFACE AND CONTROL
24                      ;
25                      ;4.0    TEST REQUIREMENTS
26                      ;
27                      ;       1. SAME AS MACHINE REQUIREMENTS
28                      ;       2. I/O TESTER OR DISC
29                      ;       3. MMPU1
30                      ;
31                      ;5.0    SUMMARY
32                      ;
33                      ;       THIS PROGRAM IS A FUNCTIONAL TEST FOR
34                      ;       THE ARRAY PROCESSOR (AP).  IT IS EXECUTED BY
35                      ;       A CENTRAL PROCESSOR (OR IOP) CONTROLLING THE AP
36                      ;       AND TESTS SPECIFIC AP INSTRUCTIONS.
37                      ;
38                      ;6.0    RESTRICTIONS
39                      ;
40                      ;       THE PROGRAM ASSUMES
41                      ;       - THE SYSTEM EXCLUDING THE AP IS ERROR FREE
42                      ;       - THE HOST COMPUTER AND MEMORY/MAP SYSTEMS ARE
43                      ;         WORKING PROPERLY.
44                      ;       - THE 2 AP MAINTAINANCE DIAGNOSTICS USING
45                      ;         THE MAINTAINANCE INSTRUCTION SET (MIS) HAVE
46                      ;         SUCCESSFULLY RUN.
```

```
01              ;7.0    PROGRAM DESCRIPTION/THEORY OF OPERATION
02              ;7.1    APXB TEST
03              ;       INSTRUCTIONS TESTED:
04              ;
05              ;       ARS - ADD REAL SCALAR TO REAL ARRAY
06              ;       SRS - SUBTRACT REAL SCALAR FROM REAL ARRAY
07              ;       CMS - COMPARE REAL SCALAR TO REAL ARRAY
08              ;       MRS - MULTIPLY REAL ARRAY BY REAL SCALAR
09              ;       MCS - MULTIPLY COMPLEX ARRAY BY COMPLEX SCALAR
10              ;       SRW - STORE REAL SCALAR FROM W-REG.
11              ;       SCW - STORE COMPLEX SCALAR FROM W-REG.
12              ;
13              ;7.2    TEST DESCRIPTION
14              ;
15              ;7.2.1  THE ARS AND SRS INSTRUCTIONS ARE TESTED USING THE
16              ;       IDEA THAT WHEN A SCALAR WITH NO FRACTIONAL PART
17              ;       IS ADDED TO A SEQUENTIAL ARRAY THE RESULT IS AN-
18              ;       OTHER SEQUENTIAL ARRAY.  THUS IF WE ADD 1.0 TO
19              ;       A SEQUENTIAL ARRAY OF SA X.  THEN THE RESULT WOULD
20              ;       BE THE SEQUENTIAL ARRAY OF STARTING ADDRESS X + 2.
21              ;       IN MAIN MEMORY, THIS CONCEPT IS USED SO THAT SIM-
22              ;       ULATION CAN BE AVOIDED.
23              ;
24              ;       IN ARS AND SRS TESTS:
25              ;NOTE:  AP RAM SA IN MM IS IN LOC RAMPT
26              ;       A) N IS VARIED => 20, 60, 200, 700, 1777
27              ;       INDEX VALUE = 1.
28              ;       A => SCALAR = 1.0
29              ;       THE DATA BASE IS SEQUENTIAL AT FDATA.
30              ;       THE RESULT IN RAM IS COMPARED TO ARRAY
31              ;       STARTING AT FDATA+2
32              ;       B) THE SEQUENTIAL DATA, WHICH IS ALWAYS
33              ;       LOADED TO RAM, IS RANDOMLY SLICED IN LOWER
34              ;       RAM 1/2 AND THE OUTPUT DATA IS ALSO SENT
35              ;       TO A RANDOM SLICE IN TOP HALF OF RAM.  THIS
36              ;       WILL TEST VARIOUS RANDOM INPUT/OUTPUT STARTING
37              ;       ADDRESSES AND RANDOM N'S.  THE RESULT IS COM-
38              ;       PARED TO THE START ADDR+2 OF SOME RANDOM SLICE
39              ;       OF FDATA ARRAY.
40              ;
41
42              ;       C) USING SAME DATA N IS SET AT 101.
43              ;       IJ = 10., IK = 1.
44              ;       THEN INDEXING IS REVERSED N=101
45              ;       IK =10., IJ=1.
46              ;       RESULTS ARE CHECKED AS IN PREVIOUS MANNER
47              ;       D) FINALLY N, IJ, IK ARE VARIED RANDOMLY.
48              ;       N RANGE IS 1-77, IJ => 1-17, IK=>1-17.
49              ;       RESULTS ARE CHECKED AS IN PREVIOUS MANNER.
50              ;
```



```
01
02              ;7.2.2  CMS IS TESTED USING SIMULATION OF INSTRUCTION,
03              ;       DO INSTRUCTION AND THEN COMPARE RESULTS METHOD.
04              ;       IN THIS SET OF TESTS EACH CMS OPERATOR GOES
05              ;       THROUGH THE SAME TEST.  A DATA BASE OF SEQUENTIAL
06              ;       FLOATING POINT #'S IS GENERATED AT FDATA.
07              ;
08              ;       A) IN THIS TEST N IS VARIED FROM 2-2000 BY 2.
09              ;       THE SCALAR USED IS 100.0.  THE SIMULATED OUTPUT
10              ;       ARRAY IS AT BDATA.  THE AP RESULT GOES TO IDATA.
11              ;       THESE ARE COMPARED.  NOTE THAT INDEX 0 OF THESE
12              ;       ARRAYS IS THE # OF ELEMENTS IN THE INPUT ARRAY
13              ;       THAT MET THE OP CODE REQUIREMENTS WITH THE GIVEN
14              ;       SCALAR.  EACH CMS OP CODE IS TESTED IN THIS WAY,
15              ;       SO THAT NOT ONLY IS N SENSITIVITY TESTED BUT BASIC
16              ;       FUNCTIONALITY IS TESTED.
17              ;       B) IN THE SECOND SET OF CMS TESTS.  THE INPUT DATA,
18              ;       WHICH IS ALWAYS RELOADED TO AP RAM, IS SLICED
19              ;       RANDOMLY.  THIS IS SENT THROUGH THE SIMULATOR AND
20              ;       THEN CHECKED AGAINST AP RESULTS.  OUTPUTS ARE IN
21              ;       THE SAME PLACE AS IN PREVIOUS TEST.  ALL OP CODES
22              ;       GO THROUGH THIS TEST.
23              ;       ALSO "A" = 400.0
24              ;       C) IN THIS ST OF TESTS N = 100., A = 400.0 BUT INDEX
25              ;       IS VARIED FROM 1 - 10.  PROCEDURE, OTHERWISE, IS
26              ;       SAME AS ABOVE TESTS.
27              ;       D) IN LAST TEST THE N IS VARIED RANDOMLY FROM 1-77,
28              ;       THE INDEX VARIES RANDOMLY FROM 1-17, AND THE OP-
29              ;       ERATION IS VARIED RANDOMLY.  THE SCALAR IS 400.0
30              ;       ALL RANDOM #'S ARE GENERATED, THE INSTRUCTION SIM-
31              ;       ULATED AND THEN THE AP IS EXERCISED.  RESULTS ARE
32              ;       COMPARED IN THE STANDARD MANNER.
33              ;
34              ;
35
36              ;7.2.3  THE NRS INSTRUCTION IS TESTED BY FIRST CREATING BOTH
37              ;       INPUT AND OUTPUT DATA.  THEN THE INSTRUCTION IS DONE.
38              ;       RESULTS ARE COMPARED
39              ;
40              ;       A DATA BASE OF SEQUENTIAL #'S IS GENERATED AT IDATA
41              ;       AND MULTIPLIED BY 2.5. THIS NEW ARRAY IS LOADED
42              ;       AT FDATA.
43              ;
44              ;       A) TEST #1 VARIES N FROM 2-1024.  THE SCALAR A = 2.5.
45              ;       THE INPUT DATA IS LOADED AT LOC 0 IN AP RAM.  OUTPUT
46              ;       DATA IS LOADED TO LOC 2000 AP RAM.  THIS OUTPUT DATA
47              ;       IS COMPARED TO FDATA.
48              ;
49              ;       B) TEST #2 IS SIMILAR TO PREVIOUS RANDOM SLICE TEST.
50              ;       LOWER RAM WHERE INPUT DATA IS, IS SLICED RANDOMLY
51              ;       (I.E. RANDOM SA, RANDOM N)  THE UPPER RAM, WHERE
52              ;       OUTPUT DATA IS, IS ALSO SLICED RANDOMLY.  THIS
53              ;       RANDOM SLICE IS COMPARED TO THE CORRESPONDING CHECK
54              ;       DATA AT FDATA IN MM.
55              ;
```

```
01
02                ;       C) TEST #3, #4 HAVE N=101., IJ=10, IK=1, OR IK=10
03                ;       IJ=1.  ALL OTHER ASPECTS OF TESTS ARE IDENTICAL
04                ;       TO PREVIOUS ONES.
05                ;
06                ;       D) FINAL TESTS VARIES N, IJ, IK RANDOMLY.  INPUT
07                ;       DATA AT IDATA IS ONCE AGAIN LOADED TO RAM SA=0 OUT-
08                ;       PUT IS ALWAYS TO SA=2000 IN AP RAM.  N RANGE
09                ;       1-77(8) IJ, IK 1-17(8)
10                ;       "A" = 2.5
11                ;
12                ;7.2.4  MCS IS TEST IN SAME MANNER AS MRS EXCEPT ALL #'S
13                ;       ARE COMPLEX.  THE SCALAR USED IS (2.5 + 1.0 I).
14                ;       SRW, SCW ARE TESTED USING OTHER AP INSTRUCTIONS
15                ;       IN APXB TEST.
16                ;       THE SRW, SCW INSTRUCTIONS READ THE WREG AND STORE
17                ;       TO MM.
18                ;       A) SRW TEST
19                ;       FIRST AN ARRAY OF RANDOM #'S IS BUILD AT FDATA.
20                ;       EACH OF THESE #'S IS USED AS THE SCALAR TO BE
21                ;       LOADED TO WREG VIA THE ARA INSTRUCTION.
22                ;       SRW IS THEN EXECUTED.  THESE RESULTING # READ
23                ;       FROM WREG IS COMPARED TO THE ORIGINAL VIA RNCOMP.
24                ;
25                ;       B) SCW TEST IS SIMILAR TO THE ABOVE TEST EXCEPT
26                ;       MSC IS USED TO LOAD A COMPLEX (64 BIT) # TO WREG.
27                ;       CNCOMP IS USED FOR COMPARISON.
28
```



```
01
02                      S?WPD 8
03
04
05                ;8.   SWITCH SETTINGS
06                ;
07                ;     LOCATION "SWREG" IS USED TO SELECT THE PROGRAM OPTIONS
08                ;     (NOT SYSTEM CONFIGURATION). WHILE RUNNING UNDER DTOS,
09                ;     THIS LOCATION WILL BE LOADED BY THE MONITOR.
10                ;     HOWEVER UNDER STAND ALONE AND PROGRAM LOAD MODES THIS
11                ;     LOCATION WILL BE SET ACCORDING TO THE ANSWERS SUPPLIED
12                ;     BY THE OPERATOR. IN ANY CASE THE OPTIONS CAN BE CHANGED
13                ;     OR VERIFIED BY USING ONE OF THE COMMANDS GIVEN IN SEC.
14                ;     8.2
15                ;
16                ;
17                ;8.1  SWITCH OPTIONS
18                ;     DIFFERENT BITS AND THEIR INTERPRETATION AT LOCATION
19                ;     "SWREG" IS AS FOLLOWS:
20                ;
21                ;     BIT     OCTAL   BINARY  INERPRETATION
22                ;             VALUE   VALUE
23                ;
24                ;     1               0       LOOP ON ERROR
25                ;             40000   1       SKIP LOOPING ON ERROR
26                ;
27                ;     2               0       PRINT TO CONSOLE
28                ;             20000   1       ABORT PRINT OUT TO CONSOLE
29                ;
30                ;     3               0       DO NOT PRINT % FAILURE
31                ;             10000   1       PRINT % FAILURE
32                ;
33                ;     4               0       ALLOW END OF PASS PRINT OUT
34                ;             04000   1       SUPPRESS END OF PASS PRINT OUT
35                ;
36                ;     5               0       DO NOT PRINT ON THE LINE PRINTER
37                ;             02000   1       PRINT ON THE LINE PRINTER
38                ;
39                ;     6               0       DO NOT HALT ON ERROR
40                ;             01000   1       HALT ON ERROR
41                ;
42                ;     7               0       DO NOT PRINT SUMMARY AND/OR
43                ;                             PASSING OF EACH SUBTEST
44                ;             00400   1       PRINT SUMMARY AND/OR
45                ;                             PASSING OF EACH SUBTEST
46                ;     8               0       PRINT ONLY THE FIRST ERROR
47                ;             00200   1       PRINT EVERY ERROR
48
49                ;8.2  SWITCH COMMANDS
50                ;     ONCE THE PROGRAM STARTS EXECUTING THE STATE OF ANY OF
51                ;     THE BITS CAN BE CHANGED BY HITTING KEYS 1-9, A-F. THE
52                ;     PROGRAM WILL CONTINUE RUNNING AFTER UPDATING THE OPTIONS.
53                ;     EACH KEY WILL COMPLEMENT THE STATE OF THE BIT AFFILIAT-
54                ;     ED WITH IT, THUS BIT 4 CAN BE ALTERED BY HITTING KEY 4.
55                ;     SETTING OF ANY BIT OF LOCATION "SWREG" WILL SET BIT 0.
56                ;     (DEFAULT MODE IS DEFINED AS ALL BITS OF SWREG SET TO 0)
57                ;     THE PROGRAM CAN BE LOCKED INTO SWITCH MODIFICATION MODE
58                ;     BY TYPING A 0, IN WHICH CASE MORE THAN ONE BIT CAN BE
59                ;     CHANGED BEFORE  CONTROL IS ALLOWED TO RETURN TO THE
60                ;     MAIN PROGRAM.
```

```
01              ;
02              ;8.2.1  OTHER COMMANDS
03              ;
04              ;       "CR"    A "RETURN" CAN BE TYPED TO CONTINUE THE PROGRAM
05              ;               AFTER ITS LOCKED IN A SWITCH MODIFICATION MODE
06              ;
07              ;       ^D      THIS COMMAND GIVEN AT ANY TIME WILL RESET "SWREG"
08              ;               TO DEFAULT MODE AND RESTART THE PROGRAM.
09              ;
10              ;       ^R      THIS COMMAND GIVEN AT ANY TIME WILL RESTART THE
11              ;               PROGRAM. SWITCHES ARE LEFT WITH THE VALUES THEY
12              ;               HAD BEFORE THE COMMAND WAS ISSUED.
13              ;
14              ;       ^O      THIS COMMAND GIVEN AT ANY TIME WILL CAUSE THE
15              ;               PROGRAM CONTROL TO GO TO ODT (NOTE: THIS IS AN
16              ;               OPTIONAL COMMAND AND IS AVAILBLE ONLY IF
17              ;               ODTPK IS PRESENT)
18              ;
19              ;       M       THIS COMMAND GIVEN AT ANY TIME WILL PRINT THE
20              ;               CURRENT OPERATING MODES.
21              ;
22
23              ;
```



```
01              ;9.0    OPERATING PROCEDURE/OPERATOR INPUT
02              ;
03              ;       9.1     OPERATING PROCEDURE
04              ;               1. LOAD THE PROGRAM IN THE COMPUTER
05              ;                  USING THE BINARY LOADER OR DTOS TAPE.
06              ;               2. SET SWITCHES TO ONE OF THE FOLLOWING:
07              ;                  500,200 -    RUN ALL TESTS
08              ;                  501 -        ARA,SRS INSTRUCTION TESTS
09              ;                  502 -        CMS INSTRUCTION TESTS
10              ;                  503 -        MRS,MCS INSTRUCTION TESTS
11              ;                  504 -        SRW,SCW INSTRUCTION TESTS
12
13              ;
14              ;
15              ;
16              ;
17              ;
18              ;               3. PRESS START.
19              ;               4. THE PROGRAM PRINTS "PASS" FOLLOWING
20              ;                  THE SUCCESSFUL COMPLETION OF THE TEST.
21              ;
22              ;
```

```
10009 APB.T
01
02              ;10.   ERRORS
03              ;      WHEN AN ERROR IS DETECTED A VALUE FOR AN
04              ;      ERROR PC WILL BE PRINTED ALONG WITH AC'S 0,1 AND 2
05              ;      OF THE HOST COMPUTER.
06              ;
07              ;      THE PC VALUE PRINTED MAY POINT TO ONE OF THE
08              ;      FOLLOWING:
09              ;
10              ;              1. A CALL TO EHALT
11              ;              2. A CALL TO THE ROUTINE WHICH DETECTED
12              ;                 THE ERROR AND SUBSEQUENTLY EXITED
13              ;                 TO EHALT SUCH AS THE VECTOR COMPARE
14              ;                 ROUTINES TCOMP,CCOMP,RCOMP, AND RCCMP.
15
16              ;      SUBROUTINES OF THIS TYPE HAVE SPECIAL ERROR PRINTOUTS
17              ;      CALLED IN THE ERROR ROUTINE. IN THE CASE OF THE REAL
18              ;      VECTOR COMPARE ROUTINES, TCOMP AND RCOMP, UP TO 3
19              ;      PAIRS OF GOOD/BAD DATA IS PRINTED IN 2-16. BIT HEX
20              ;      OR OCTAL FORMAT (SEE SW10), ALONG WITH THE MM ADDRESS
21              ;      OF THE DATA AND THE RELATIVE INDEX WITHIN THE VECTOR.
22              ;      A TOTAL ERROR COUNT IS PRINTED ALONG WITH
23              ;      THE VECTOR/VECTOR N,IJ,IK DATA. IN THE CASE OF THE
24              ;      COMPLEX VECTOR COMPARE ROUTINES, CCOMP AND RCCMP, THE
25              ;      GOOD/BAD DATA IS PRINTED IN 4-16. BIT HEX OR OCTAL
26              ;      FORMAT.
27
28              ;      THE SIGNIFICANCE OF ERROR PRINTOUTS IS EXPLAINED IN THE
29              ;      LISTING AT THE SOURCE OF THE ERROR.  IN GENERAL
30              ;      AC0 CONTAINS THE CHECKWORD AND AC1 THE DATA IN
31              ;      QUESTION.
32
33              ;      AFTER PRINTING THE ABOVE THE PROGRAM HALTS ALLOWING
34              ;      THE OPERATOR TO SET THE SWITCHES.  UPON PRESSING
35              ;      CONTINUE THE PROGRAM MAY EITHER GO ON TO THE NEXT
36              ;      TEST OR GO INTO A SCOPE LOOP BETWEEN THE APPROPRIATE
37              ;      CALLS TO .SETUP AND .LOOP (SEE 8.0 SWITCH SETTINGS)
```

```
10010 APB.T
01
02              ;11.0  DEBUG HELP
03              ;
04              ;      11.1   MONITOR LOCATIONS
05              ;
06              ;             LOCATION        LABEL   MEANING
07              ;               200                   DTOS START
08              ;               201                   FIRST LOCATION OF CURRENT T
09              ;               202                   START ADDRESS
10              ;               203                   XXXX
11              ;               204                   PASS COUNT
12              ;
13              ;      11.2   PROGRAMMING DESCRIPTION
14              ;
15              ;              SEE PROGRAMMERS REFERENCE
16              ;             FOR ARRAY PROCESSOR.
17              ;
```

```
!0011 APB.T
01
02                ;       11.3    COMMON SUBROUTINES
03                ;
04                ;       A) ADDSET,SETP,LOOP -
05                ;       THESE ALL WORK IN CONJUNCTION TO
06                ;       SET UP THE TEST LOOP.
07                ;       ADDSET TELLS SETP WHERE THE LOOP ENDS.
08                ;       SETP MARKS THE START OF THE TEST LOOP.
09                ;       LOOP MARKS THE END OF THE TEST LOOP
10                ;       THE AP INSTRUCTION IS EXECUTED AND THE RESULTS ARE
11                ;       CHECKED FOR ERROR WITHIN THIS TEST LOOP. IF AN ERROR
12                ;       OCCURS, THEN THE PROGRAM CAN LOOP INDEFINITELY BETWEEN
13                ;       THE SETP AND THE LOOP.(SEE SECTION 8 ON VARIOUS SWITCH
14                ;       SETTINGS).
15                ;
16                ;       NORMALLY, WITH NO ERRORS, THE PROGRAM WILL GO THROUGH
17                ;       THIS LOOP ONCE ON THE FIRST PASS OF THE PROGRAM. ON
18                ;       SUBSEQUENT PASSES THE PROGRAM WILL STAY IN THE LOOP
19                ;       FOR 5 LOOPS.(OR 100. DEPENDING ON THE PROGRAM).
20                ;
21                ;
22                ;       B) PARAMETER BLOCK LOADER ROUTINES -
23                ;       THESE CALLS ALL HAVE DIFFERENT NAMES DEPENDING ON THE
24                ;       INSTRUCTION. HOWEVER THEY OPERATE IN THE SAME MANNER.
25                ;       THE COMMENT FIELD ON THE CALL WILL ALWAYS INDICATE
26                ;       THAT THIS IS A PARAMETER BLOCK LOAD ROUTINE AND WILL
27                ;       NAME THE AP INSTRUCTION THAT THE LOAD IS COMPATIBLE WITH.
28                ;       THE ARGUMENTS OF THE CALL ARE THE PARAMETERS OF THE AP
29                ;       INSTRUCTION. THESE PARAMETERS ARE TO BE LOADED
30                ;       BY THESE ROUTINES INTO MAIN MEMEORY AT A LOCATION POINTED
31                ;       TO BY AC2. AC2 MUST ALWAYS POINT TO THE PARAMETER BLOCK
32                ;       DURING AP INSTRUCTION EXECTION.
33
```

```
!0012 APB.T
01
02                ;       C) SLICING ROUTINES -
03                ;       THESE TYPES OF ROUTINES EXIST IN SEVERAL DIFFERENT FORMS.
04                ;       THEY ALL HAVE VERY SIMILAR NAMES.FOR EXAMPLE - SLICE,
05                ;       SLICC,SLICA,SLICB.THESE ARE ALL JSR'S TO SOME SLICING
06                ;       ROUTINE.
07                ;       THE SLICING ROUTINES WILL SET UP VARIOUS RANDOM
08                ;       SLICES IN AP RAM FOR THE VARIOUS INPUT AND OUTPUT
09                ;       ARRAYS USED IN THE AP INSTRUCTIONS.
10                ;       TO CREATE THESE SLICES, RANDOM N AND STARTING ADDRESS
11                ;       ARE CREATED. THE COMMENTS ON THE CALL WILL INDICATE
12                ;       THE DESTINATION OF THESE RANDOM PARAMETERS WITHIN THE
13                ;       TEST CODE. THE ROUTINES ALL SLICE DIFFERENTLY.
14                ;       ONE WILL DIVIDE THE RAM IN HALF AND THEN FIND
15                ;       A RANDOM SLICE WITHIN EACH HALF. ANOTHER MIGHT
16                ;       FIND A RANDOM SLICE FOR EACH THIRD OF RAM.
```

```
01              ;       11.4    SEQUENCE OF TESTING
02              ;
03              ;               IT IS IMPORTANT THAT THE PROGRAMS BE
04              ;               EXECUTED IN A SPECIFIC SEQUENCE:
05              ;
06              ;
07              ;
08              ;               APIS DIAG (FOR CPU/AP ONLY)
09              ;               APFP DIAG (FOR CPU/AP ONLY)
10              ;               APA EXER
11              ;               APB EXER
12              ;               APC EXER
13              ;               APD EXER
14              ;               APE EXER
15              ;               APF EXER
16              ;               APG EXER
17              ;               APH EXER
18              ;               API EXER
19              ;               APJ EXER
20              ;               APK EXER
```



```
01                      0?DTD 12
02
03              ;12.    OCTAL DEBUG TOOL (ODT)
04              ;
05              ;       THE DIAGNOSTIC IS EQUIPED WITH A BUILT IN ODT WHICH CAN
06              ;       BE ACCESSED BY HITTING CONTROL O (^O) AT ANY TIME DURING
07              ;       THE EXECUTION OF THE PROGRAM (AFTER SETTING THE PARA-
08              ;       METERS).
09              ;       ON ENTERING ODT THE ADDRESS OF THE LOCATION HAVING THE
10              ;       NEXT INSTRUCTION TO BE EXECUTED WILL BE TYPED-OUT.
11              ;
12              ;12.1   CONVENTIONS AND SYMBOLS
13              ;       THE FOLLOWING CONVENTIONS ARE USED BY THE ODT:
14              ;       ?       PRESSING ANY ILLEGAL KEY CAUSES THE ODT TO RES-
15              ;               POND WITH A "?".
16              ;       @       ODT IS READY AND AT YOUR SERVICE.
17              ;
18              ;12.2   COMMAND STRUCTURE
19              ;       AN ODT COMMAND HAS THE FOLLOWING FORMAT:
20              ;               [ARGUMENT][COMMAND]
21              ;       AN ARGUMENT MAY BE ONE OF THE FOLLOWING:
22              ;       "EXP"   AN OCTAL EXPRESSION CONSISTING OF OCTAL NUMBERS
23              ;               SEPARATED BY PLUS (+) OR MINUS (-) SIGNS. LEAD-
24              ;               ING ZEROS NEED NOT BE TYPED.
25              ;       "ADR"   AN ADDRESS IS THE SAME AS AN EXPRESSION EXCEPT
26              ;               THAT BIT 0 IS NEGLECTED.
27              ;       A COMMAND IS A SINGLE TELETYPE CHARACTER
28              ;
29              ;12.3   ODT COMMANDS
30              ;       THE LOCATIONS THAT CAN BE EXAMINED AND MODIFIED BY THE
31              ;       USER ARE CALLED CELLS. THESE CELLS ARE OF TWO TYPES:
32              ;       INTERNAL CPU CELLS AND MEMORY LOCATIONS.
33              ;
34              ;12.3.1 OPENING INTERNAL CELLS
35              ;       THE COMMAND TO OPEN ONE OF THE INTERNAL REGISTERS IS OF
36              ;       THE FORM "NA" WHERE N IS ANY OCTAL EXPRESSION BETWEEN
37              ;       0 AND 7
38              ;       0-3     FOR ACCUMULATORS 0-3
39              ;       4       FOR PC OF THE NEXT INSTRUCTION TO BE EXECUTED IN
40              ;               THE EVENT OF A "P" COMMAND.
41              ;       5       CPU AND TTO STATUS
42              ;               BIT     INTERPRETATION
43              ;               15      STATUS OF TTO DONE FLAG
44              ;               14      STATUS OF INTERRUPTS (ION FLAG)
45              ;               13      STATUS OF CARRY BIT
46              ;       6       ADDRESS OF THE LOCATION HAVING THE BREAK POINT (IF
47              ;               ANY)
48              ;       7       INSTRUCTION AT THE BREAK POINT LOCATION
49
50              ;       OTHER COMMANDS TO OPEN CELLS ARE:
51              ;
52              ;       "ADR"/  OPEN THE CELL AND PRINT ITS CONTENTS
53              ;       ./      OPEN THE CELL CURRENTLY POINTED TO BY THE POINTER
54              ;               AND PRINT ITS CONTENTS.
55              ;       .+"ADR"/        ADD "ADR" TO THE POINTER, OPEN THE CELL
56              ;               AND PRINT ITS CONTENTS.
57              ;       .-"ADR"/        SUBTRACT "ADR" FROM THE POINTER, OPEN
58              ;               THE CELL AND PRINT ITS CONTENTS.
59              ;       "CR"    THE RETURN KEY IS USED TO CLOSE THE OPEN CELL
60              ;               WITH OR WITHOUT MODIFICATION.
```

```
 0015 APB.T
01              ;       "LF"    LINE FEED IS USED TO CLOSE THE OPEN CELL WITH OR
02              ;               WITHOUT MODIFICATION AND TO OPEN THE SUCCEEDING
03              ;               CELL.
04              ;       ^       CLOSE THE OPEN CELL WITH OR WITHOUT MODIFICATION
05              ;               AND OPEN THE PRECEDING CELL
06              ;       /       CLOSE THE OPEN CELL WITHOUT MODIFICATION, AND
07              ;               OPEN THE CELL POINTED TO BY ITS CONTENTS.
08              ;       +"ADR"/ CLOSE THE OPEN CELL WITHOUT MODIFICATION, AND
09              ;               OPEN THE CELL POINTED TO BY ITS CONTENTS + "ADDR".
10              ;       -"ADR"/ CLOSE THE OPEN CELL WITHOUT MODIFICATION, AND
11              ;               OPEN THE CELL POINTED TO BY ITS CONTENTS - "ADR".
12              ;
13              ;12.3.2 MODIFICATION OF A CELL
14              ;       ONCE A CELL HAS BEEN OPENED ITS CONTENTS CAN BE MODIFIED
15              ;       BY TYPING THE NEW VALUE THE CELL IS TO CONTAIN IN THE
16              ;       FORM OF AN OCTAL EXPRESSION FOLLOWED BY "CR" OR "LF".
17              ;       IF A + OR - IS TYPED AS THE FIRST CHARACTER OF THE EX-
18              ;       PRESSION THEN THE VALUE OF THE EXPRESSION IS ADDED TO OR
19              ;       SUBTRACTED FROM THE OLD CONTENTS OF THE CELL. THE
20              ;       ADDRESS ITSELF OR AN EXPRESION RELATIVE TO THE ADDRESS
21              ;       CAN BE DEPOSITED BY TYPING A "." OR ".+/-OCTAL EXPRESS-
22              ;       ION". A RUBOUT COMMAND GIVEN RIGHT AFTER OPENING A CELL
23              ;       ALLOWS THE MODIFICATION OF ITS CONTENTS AS IF THEY WERE
24              ;       TYPED IN JUST BEFORE THE COMMAND WAS ISSUED.
25              ;
26              ;12.3.3 OTHER ODT COMMANDS
27              ;       RUBOUT  THIS KEY IS USED TO DELETE ERRONEOUSLY TYPED
28              ;               DIGITS. EACH TIME THE KEY IS PRESSED THE RIGHT MOST
29              ;               DIGIT IS DELETED AND ECHOED ON THE TERMINAL. IF
30              ;               THE RUBOUT KEY IS PRESSED RIGHT AFTER OPENING A
31              ;               CELL THEN IT DELETES THE RIGHT MOST DIGIT OF THE CELLS
32              ;               CONTENTS. THIS ALLOWS THE MODIFICATION OF THE CELL
33              ;               AS IF ITS CONTENTS WERE TYPED IN JUST BEFORE THE
34              ;               KEY WAS PRESSED.
35              ;       "ADR"B  INSERT A BREAK POINT AT LOCATION "ADR".
36              ;               ONLY ONE BREAK POINT CAN BE INSERTED AND ANY
37              ;               ENTRY TO ODT AFTER EXECUTING A BREAK POINT WILL
38              ;               CAUSE IT TO BE DELETED.
39              ;       D       DELETE THE BREAK POINT IF ANY.
40              ;       P       RESTART THE EXECUTION OF THE PROGRAM AT LOCATION
41              ;               POINTED BY 4A.
42              ;       "ADR"R  START EXECUTING THE PROGRAM AT "ADR" AFTER AN
43              ;               IO-RESET.
44              ;       K       KILL THE STRING TYPED SO FAR. THE ODT RESPONDS
45              ;               WITH A "?" AND THE OPEN CELL IS CLOSED WITHOUT
46              ;               MODIFICATION.
47              ;       =       PRINT THE OCTAL VALUE OF THE INPUT ONLY.
48              ;               THIS WILL CLOSE ANY OPEN CELLS WITHOUT
49              ;               MODIFICATION AND WILL NOT OPEN A CELL
50              ;
51              ;
52              ;       NOTE:   IN PROGRAMS WHICH RELOCATE THEMSELVES THE
53              ;               THE USER SHOULD PLACE BREAK POINTS ONLY IN THE
54              ;               THE ORIGINAL PROGRAM AREA. IF A BREAK POINT IS
55              ;               PLACED OUTSIDE THIS AREA THE RESULTS WILL
56              ;               BE UNPREDICTABLE.
57              ;
58              ;
```

```
!0016 APB.T
01              ;13.    SPECIAL NOTES/SPECIAL FEATURES
02              ;
03              ;       13.1    FOR A COMPLETE TEST ALL PROGRAMS
04              ;               SHOULD BE EXECUTED WITH CAT/KITTEN.
05              ;
06              ;       13.2    A NOTE ABOUT AP ADDRESSING
07              ;
08              ;               ADDRESSES IN THE AP CAN BE OF SEVERAL
09              ;               MODES:
10              ;
11              ;               1) ONE WORD MODE (1E) SAME AS STANDARD
12              ;                  ADDRESSING
13              ;               2) TWO WORD MODE (2WM) - EACH 32 BITS IS
14              ;                  NOW ONE ADDRESS SPACE.  THIS IS USED
15              ;                  IN THE AP TO SIMPLIFY REAL NUMBER
16              ;                  ADDRESSING.
17              ;               3) FOUR WORD MODE (4WM) - EACH 64 BITS IS
18              ;                  NOW ONE ADDRESS SPACE.  THIS IS USED
19              ;                  IN THE AP TO SIMPLIFY COMPLEX NUMBER
20              ;                  ADDRESSING.
21              ;
22              ;               THE AP ACCESSES AN ADDRESS RELATIVE TO THE START OF
23              ;               THE AP RAM.  THUS AP RAM LOC 0 WOULD BE THE
24              ;               FIRST LOCATION THAT IS IN THE AP.  HOWEVER,
25              ;               AS FAR AS THE ECLIPSE CPU IS CONCERNED, AP
26              ;               LOC 0 IS CONTAINED AT LOCATION LABEL
27              ;               "RAMPT". (IN PAGE ZERO.  SO, IF RAMPT CON-
28              ;               TAINS 64000, THEN 2000 2WM (AP RAM) IS REALLY
29              ;               64000+2000+2000=70000.
30              ;
31              ;               NOTE: "STOP ON STORE" OR "STOP ON ADDRESS" IN AP RAM
32              ;               SPACE WILL NOT WORK IF THE AP IS USING
33              ;               THE INTERNAL AP ADDRESS LINES TO ACCESS AP RAM.
```

```
!0017 APB.T
01                ;14.0    RUN TIME
02                ;
03                ;        14.1     PASS 1                    27 SECONDS
04                ;
05                ;        14.2     SUBSEQUENT PASSES         42 SECONDS
06                ;
                           .TITL APB.TX
08                         .END

**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS
```

```
 0018 APB.T

O?DTD 001551  MC    14/01
S?WPD 001075  MC     6/02
```