

DataGeneral

**TECHNICAL
STATEMENT**

TEXT LISTING

068-001065-01

PROGRAM

ARRAY PROCESSOR EXERCISER-A

TEXT TAPE

097-001065-01

ABSTRACT

THIS PROGRAM IS A COMBINATION DIAGNOSTIC AND EXERCISER FOR THE ARRAY PROCESSOR. THE PROGRAM IS EXECUTED BY A CENTRAL PROCESSOR (OR IOP) CONTROLLING THE ARRAY PROCESSOR (AP).

```

0001 .MAIN          MACRO REV 06.30          09:34:00 09/19/79          10002 .MAIN
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
*****
; NAME: APA.TX          PART NUMBER: 097-1065
; DESCRIPTION: ARRAY PROCESSOR EXERCISER-A.
; REVISION HISTORY:
; REV.      DATE
; 00      06/01/78
; 01      12/29/78
; COPYRIGHT © DATA GENERAL CORPORATION, 1978
; ALL RIGHTS RESERVED.
*****

APA.TX          PART NUMBER:097-1466

PROGRAM NAME
SOURCE FILE:   APA.SR
OTOS FILE:    APA EXER

REVISION HISTORY
DATE          REVISION
-----
6/1/78      00

MACHINE REQUIREMENTS
1. ECLIPSE FAMILY CENTRAL PROCESSOR (HOST) WITH
   AT LEAST 16-K READ/WRITE MEMORY
2. API,AP2,AP3 PROCESSOR BOARDS
3. BASIC I/O TELETYPE INTERFACE AND CONTROL

TEST REQUIREMENTS
1. SAME AS MACHINE REQUIREMENTS
2. I/O TESTER OR DISC
3. MMPUI

SUMMARY
THIS PROGRAM IS A COMBINATION DIAGNOSTIC AND EXERCISER
FOR THE ARRAY PROCESSOR. THE PROGRAM IS EXECUTED BY
A CENTRAL PROCESSOR (OR IOP) CONTROLLING THE ARRAY
PROCESSOR (AP).

RESTRICTIONS
THE PROGRAM ASSUMES THAT THE HOST COMPUTER AND
MEMORY/MAP SYSTEMS ARE WORKING PROPERLY.

***THE PROGRAM ALSO ASSUMES THAT THE 2 AP MAINTENANCE
DIAGNOSTICS USING THE MTS (MAINTENANCE INSTRUCTION
SET) HAVE BEEN SUCCESSFULLY RUN.

```

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47

```

```

10003 .MAIN
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

10004 .MAIN
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

7.0 PROGRAM DESCRIPTION/THEORY OF OPERATION
7.1 APXA TEST
INSTRUCTIONS TESTED
LDR= LOAD A VECTOR TO RAM
STR= READ A VECTOR FROM AP RAM
FLL= FLOAT AND LOAD AN INTEGER ARRAY TO AP RAM
FXS= FIX AND STORE A VECTOR FROM AP RAM
LSR= LOAD CRAM
SSR= READ CRAM
SCB= READ BIT-REVERSED VECTOR (COMPLEX)
BRC= IN PLACE BIT-REVERSAL:COMPLEX VECTOR

7.2 TEST DESCRIPTION
7.2.1 LDR-STR
A BLOCK OF DATA IS SET UP (ALL 125252) IN A BLOCK NAMED "CATCH". THIS BLOCK IS THEN MOVED TO AP-RAM VIA THE LDR INSTRUCTION. THIS IS DONE FOR 2 REAL NUMBERS, THEN 4, 10(BASE 8) UP TO 4000. AFTER EACH LDR IS PERFORMED AN STR INSTRUCTION IS EXECUTED TO RETRIEVE THE DATA BACK FROM AP-RAM, WHICH IS THEN CHECKED AGAINST THE ORIGINAL DATA. WHEN 4000 NUMBERS HAVE BEEN CHECKED, THE BITS IN "CATCH" ARE COMPLEMENTED (52525) AND THE PROCESS IS REPEATED. FINALLY, THE LDR AND STR INSTRUCTIONS ARE TESTED WITH RANDOM RAM ADDRESSING, CONSTANT INDEXING AND RANDOM INDEXING

7.2.2 FLL-FXS
A BLOCK OF DATA RANGING IN SIZE FROM 1-41 INTEGER NUMBERS (ALL OF WHICH ARE 2'S) IS FLOATED AND LOADED INTO RAM (FLL INSTRUCTION) WITH FIXED INDEXING. THE RESULTANT VECTOR IS THEN READ BACK FROM AP-RAM (MANUALLY) AND CHECKED AGAINST A FLOATING POINT 2 (1.E-2.0E0). SIMILAR TESTS ARE THEN PERFORMED WITH RANDOM RAM ADDRESS; WORD COUNT = 15; SOURCE = 6; DESTINATION INDEX = 1; AND RANDOM WORD COUNT (1-17); RANDOM INDEXING (1-17)

THE FXS INSTRUCTION IS TESTED VIA FIRST EXECUTING AN FLL INSTRUCTION AND FIXING THE NUMBERS AND BRINGING THEM BACK TO MAIN MEMORY (MM) WITH THE FXS INSTRUCTION. THESE NUMBERS ARE THEN CHECKED AGAINST THE ORIGINAL DATA. THE TEST IS PERFORMED ON WORD COUNTS FROM DECIMAL 1-.33 WITH FIXED SOURCE AND DESTINATION INDEXES =1; A SOURCE INDEX OF 6, DESTINATION INDEX OF 1, WORD COUNT 15; AND FINALLY RANDOM INDEXING (1-17) AND RANDOM WORD COUNT (1-17).

7.2.3 LSR-SSR
THESE INSTRUCTIONS ARE TESTED ALMOST EXACTLY LIKE LDR AND STR, RESPECTIVELY, EXCEPT THAT THESE TEST LOADING AND STORING FROM SCRATCH PAD MEMORY (CRAM)

7.2.4 SCB-BRC
THESE INSTRUCTIONS ARE TESTED SIMILAR TO THE PREVIOUS SIX EXCEPT THAT THE BIT REVERSAL IS SIMULATED IN A SUBROUTINE AND THESE RESULTS ARE CHECKED AGAINST THE BIT REVERSAL OF THE INSTRUCTIONS (SCB,BRC).

SWMPD 8
18. SWITCH SETTINGS
LOCATION "SWREG" IS USED TO SELECT THE PROGRAM OPTIONS (NOT SYSTEM CONFIGURATION). WHILE RUNNING UNDER DTOS, THIS LOCATION WILL BE LOADED BY THE MONITOR, HOWEVER UNDER STAND ALONE AND PROGRAM LOAD MODES THIS LOCATION WILL BE SET ACCORDING TO THE ANSWERS SUPPLIED BY THE OPERATOR. IN ANY CASE THE OPTIONS CAN BE CHANGED OR VERIFIED BY USING ONE OF THE COMMANDS GIVEN IN SEC. 8.2

SWITCH OPTIONS AND THEIR INTERPRETATION AT LOCATION "SWREG" IS AS FOLLOWS:
BIT OCTAL BINARY INERPRETATION
VALUE VALUE
1 40000 1 LOOP ON ERROR
2 20000 0 PRINT TO CONSOLE
3 10000 1 DO NOT PRINT % FAILURE
4 04000 1 ALLOW END OF PASS PRINT OUT
5 02000 1 DO NOT PRINT ON THE LINE PRINTER
6 01000 1 HALT ON ERROR
7 00400 1 PASSING OF EACH SUBTEST
8 00200 1 PRINT ONLY THE FIRST ERROR

SWITCH COMMANDS
ONCE THE PROGRAM STARTS EXECUTING THE STATE OF ANY OF THE BITS CAN BE CHANGED BY HITTING KEYS 1-9, A-F. THE PROGRAM WILL CONTINUE RUNNING AFTER UPDATING THE OPTIONS. EACH KEY WILL COMPLEMENT THE STATE OF THE BIT AFFILIATED WITH IT, THUS BIT 4 CAN BE ALTERED BY HITTING KEY 4. (DEFAULT MODE IS DEFINED AS ALL BITS OF SWREG SET TO 0) THE PROGRAM CAN BE LOCKED INTO SWITCH MODIFICATION MODE BY TYPING A 0, IN WHICH CASE MORE THAN ONE BIT CAN BE CHANGED BEFORE CONTROL IS ALLOWED TO RETURN TO THE MAIN PROGRAM.

```



```

10009 .MAIN
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

0?0TD
12
OCTAL DEBUG TOOL (ODT)

THE DIAGNOSTIC IS EQUIPPED WITH A BUILT IN ODT WHICH CAN
BE ACCESSED BY HITTING CONTROL 0 (^O) AT ANY TIME DURING
THE EXECUTION OF THE PROGRAM (AFTER SETTING THE PARA-
METERS).
ON ENTERING ODT THE ADDRESS OF THE LOCATION HAVING THE
NEXT INSTRUCTION TO BE EXECUTED WILL BE TYPED-OUT.

CONVENTIONS AND SYMBOLS
THE FOLLOWING CONVENTIONS ARE USED BY THE ODT:
? PESSING ANY ILLEGAL KEY CAUSES THE ODT TO RES-
POND WITH A "2".
@ ODT IS READY AND AT YOUR SERVICE.

COMMAND STRUCTURE
AN ODT COMMAND HAS THE FOLLOWING FORMAT:
[ARGUMENT] [COMMAND]
AN ARGUMENT MAY BE ONE OF THE FOLLOWING:
"EXP" AN OCTAL EXPRESSION CONSISTING OF OCTAL NUMBERS
SEPARATED BY PLUS (+) OR MINUS (-) SIGNS. LEAD-
ING ZEROS NEED NOT BE TYPED.
"ADR" AN ADDRESS IS THE SAME AS AN EXPRESSION EXCEPT
THAT BIT 0 IS NEGLECTED.
A COMMAND IS A SINGLE TELETYPE CHARACTER

ODT COMMANDS
THE LOCATIONS THAT CAN BE EXAMINED AND MODIFIED BY THE
USER ARE CALLED CELLS. THESE CELLS ARE OF TWO TYPES:
INTERNAL CPU CELLS AND MEMORY LOCATIONS.

12.3.1 OPENING INTERNAL CELLS
THE COMMAND TO OPEN ONE OF THE INTERNAL REGISTERS IS OF
THE FORM "NA" WHERE N IS ANY OCTAL EXPRESSION BETWEEN
0 AND 7
0-3 FOR ACCUMULATORS 0-3
4 FOR PC OF THE NEXT INSTRUCTION TO BE EXECUTED IN
THE EVENT OF A "P" COMMAND.
5 CPU AND I/O STATUS
8BIT INTERPRETATION
15 STATUS OF I/O DONE FLAG
14 STATUS OF INTERRUPTS (I/O FLAG)
6 STATUS OF CARRY BIT
ADDRESS OF THE LOCATION HAVING THE BREAK POINT (IF
ANY)
7 INSTRUCTION AT THE BREAK POINT LOCATION
OTHER COMMANDS TO OPEN CELLS ARE:

"ADR"/ OPEN THE CELL AND PRINT ITS CONTENTS
./ OPEN THE CELL CURRENTLY POINTED TO BY THE POINTER
AND PRINT ITS CONTENTS.
+"ADR"/ ADD "ADR" TO THE POINTER, OPEN THE CELL
AND PRINT ITS CONTENTS.
-"ADR"/ SUBTRACT "ADR" FROM THE POINTER, OPEN
THE CELL AND PRINT ITS CONTENTS.
"CR" THE RETURN KEY IS USED TO CLOSE THE OPEN CELL

0010 .MAIN
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59

"LF" WITH OR WITHOUT MODIFICATION,
LINE FEED IS USED TO CLOSE THE OPEN CELL WITH OR
WITHOUT MODIFICATION AND TO OPEN THE SUCCEEDING
CELL.
^ CLOSE THE OPEN CELL WITH OR WITHOUT MODIFICATION
AND OPEN THE PRECEDING CELL
/ CLOSE THE OPEN CELL WITHOUT MODIFICATION, AND
OPEN THE CELL POINTED TO BY ITS CONTENTS.
+"ADR"/ CLOSE THE OPEN CELL WITHOUT MODIFICATION, AND
OPEN THE CELL POINTED TO BY ITS CONTENTS + "ADR".
-"ADR"/ CLOSE THE OPEN CELL WITHOUT MODIFICATION, AND
OPEN THE CELL POINTED TO BY ITS CONTENTS - "ADR".

12.3.2 MODIFICATION OF A CELL
ONCE A CELL HAS BEEN OPENED ITS CONTENTS CAN BE MODIFIED
BY TYPING THE NEW VALUE THE CELL IS TO CONTAIN IN THE
FORM OF AN OCTAL EXPRESSION FOLLOWED BY "CR" OR "LF".
IF A + OR - IS TYPED AS THE FIRST CHARACTER OF THE EX-
PRESSION THEN THE VALUE OF THE EXPRESSION IS ADDED TO OR
SUBTRACTED FROM THE OLD CONTENTS OF THE CELL. THE
ADDRESS ITSELF OR AN EXPRESSION RELATIVE TO THE ADDRESS
CAN BE DEPOSITED BY TYPING A "N" OR "N"/-OCTAL EXPRESS-
ION". A RUBOUT COMMAND GIVEN RIGHT AFTER OPENING A CELL
ALLOWS THE MODIFICATION OF ITS CONTENTS AS IF THEY WERE
TYPED IN JUST BEFORE THE COMMAND WAS ISSUED.

12.3.3 OTHER ODT COMMANDS
RUBOUT THIS KEY IS USED TO DELETE ERRONEOUSLY TYPED
DIGITS. EACH TIME THE KEY IS PRESSED THE RIGHT MOST
DIGIT IS DELETED AND ECHOED ON THE TERMINAL. IF
THE RUBOUT KEY IS PRESSED RIGHT AFTER OPENING A
CELL THEN IT DELETES THE RIGHT MOST DIGIT OF THE CELLS
CONTENTS. THIS ALLOWS THE MODIFICATION OF THE CELL
AS IF ITS CONTENTS WERE TYPED IN JUST BEFORE THE
KEY WAS PRESSED.
"ADR"b INSERT A BREAK POINT AT LOCATION "ADR".
ONLY ONE BREAK POINT CAN BE INSERTED AND ANY
ENTRY TO ODT AFTER EXECUTING A BREAK POINT WILL
CAUSE IT TO BE DELETED.
D DELETE THE BREAK POINT IF ANY.
P RESTART THE EXECUTION OF THE PROGRAM AT LOCATION
POINTED BY 4.
"ADR"R START EXECUTING THE PROGRAM AT "ADR" AFTER AN
I/O-RESET.
K KILL THE STRING TYPED SO FAR. THE ODT RESPONDS
WITH A "?" AND THE OPEN CELL IS CLOSED WITHOUT
MODIFICATION.
= PRINT THE OCTAL VALUE OF THE INPUT ONLY.
THIS WILL CLOSE ANY OPEN CELLS WITHOUT
MODIFICATION AND WILL NOT OPEN A CELL

NOTE: IN PROGRAMS WHICH RELOCATE THEMSELVES THE
THE USER SHOULD PLACE BREAK POINTS ONLY IN THE
ORIGINAL PROGRAM AREA. IF A BREAK POINT IS
PLACED OUTSIDE THIS AREA THE RESULTS WILL
BE UNPREDICTABLE.

```

10011 .MAIN

01 ;13. SPECIAL NOTES/SPECIAL FEATURES

02 ?
03 ?
04 ? 13.1 FOR A COMPLETE TEST ALL PROGRAMS
05 ? SHOULD BE EXECUTED WITH CAT/KITTEN.

06 ?
07 ? 13.2 A NOTE ABOUT AP ADDRESSING

08 ? ADDRESSES IN THE AP CAN BE OF SEVERAL
09 ? MODES:

10 ? 1) ONE WORD MODE (1E) SAME AS STANDARD

11 ? ADDRESSING

12 ? 2) TWO WORD MODE (2MM) - EACH 32 BITS IS

13 ? NOW ONE ADDRESS SPACE. THIS IS USED

14 ? IN THE AP TO SIMPLIFY REAL NUMBER

15 ? ADDRESSING.

16 ? 3) FOUR WORD MODE (4MM) - EACH 64 BITS IS

17 ? NOW ONE ADDRESS SPACE. THIS IS USED

18 ? IN THE AP TO SIMPLIFY COMPLEX NUMBER

19 ? ADDRESSING.

20 ? THE AP ACCESSES AN ADDRESS RELATIVE TO THE START OF

21 ? THE AP RAM. THUS AP RAM LOC 0 WOULD BE THE

22 ? FIRST LOCATION THAT IS IN THE AP. HOWEVER,

23 ? AS FAR AS THE ECLIPSE CPU IS CONCERNED, AP

24 ? LOC 0 IS CONTAINED AT LOCATION LABEL

25 ? "RAMPT". (IN PAGE ZERO, SO, IF RAMPT CON-

26 ? TAINS 64000+2000+2000=70000.

27 ?

28 ? NOTE: "STOP ON STORE" OR "STOP ON ADDRESS" IN AP RAM

29 ? SPACE WILL NOT WORK IF THE AP IS USING

30 ? THE INTERNAL AP ADDRESS LINES TO ACCESS AP RAM.

31 ?

32 ?

33 ?

34 ?

35 ? 14.0 RUN TIME

36 ?

37 ? 14.1 PASS 1 8 SEC

38 ?

39 ? 14.2 SUBSEQUENT PASSES 15 SEC

40 ?

41 ?

42 ?

43 ?

44 ?

45 ?

0012 .MAIN

0?0TD 001551 MC 9/02
S?MPD 001075 MC 4/01

**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS