DataGeneral

TECHNICAL STATEMENT

TEXT LISTING

068-001070-01

PROGRAM

ARRAY PROCESSOR EXERCISER-F

TEXT TAPE

097-001070-01

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.

COPYRIGHT © DATA GENERAL CORPORATION, 1978
ALL RIGHTS RESERVED. PRINTED IN U.S.A.

```
;*****************************************************
;
; NAME: APF.TX                    PART NUMBER: 097-1070
;
;
; DESCRIPTION: ARRAY PROCESSOR EXERCISER-F.
;
;
; REVISION HISTORY:
;
;       REV.            DATE
;
;       00              06/01/78
;       01              12/29/78
;
;
; COPYRIGHT © DATA GENERAL CORPORATION, 1978
; ALL RIGHTS RESERVED.
;
;*****************************************************
```



```
;
;               APF.TX          PART NUMBER:094-1471
;
;1.0    PROGRAM NAME
;
;       SOURCE FILE:    APF.SR
;       DTOS FILE:      APF EXER
;
;2.0    REVISION HISTORY
;
;       DATE    REVISION
;       ----    --------
;       3/1/78     0
;
;3.0    MACHINE REQUIREMENTS
;
;       1. ECLIPSE FAMILY CENTRAL PROCESSOR  (HOST OR IOP)
;          WITH AT LEAST 16.K READ/WRITE MEMORY
;       2. ARRAY PROCESSOR BOARDS AP1, AP2, AP3
;       3. BASIC I/O TELETYPE INTERFACE AND CONTROL
;
;4.0    TEST REQUIREMENTS
;
;       1. SAME AS MACHINE REQUIREMENTS
;       2. I/O TESTER OR DISC
;       3. MMPU1
;
;5.0    SUMMARY
;
;       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.
;
;6.0    RESTRICTIONS
;
;       THE PROGRAM ASSUMES
;       - THE SYSTEM EXCLUDING THE AP IS ERROR FREE
;       - THE HOST COMPUTER AND MEMORY/MAP SYSTEMS
;         ARE WORKING PROPERLY.
;       - THE 2 AP MAINTAINANCE DIAGNOSTICS USING
;         THE MAINTAINANCE INSTRUCTION SET (MIS)
;         HAVE BEEN SUCCESSFULLY RUN.
```

```
;7.0     PROGRAM DESCRIPTION/THEORY OF OPERATION
;        7.1     APXF TEST
;                INSTRUCTIONS TEXTED:
;
;                CONR - CONVOLUTION OF REAL ARRAYS
;                CONC - CONVOLUTION OF COMPLEX ARRAYS
;                CORR - CORRELATION OF REAL ARRAYS
;                CORC - CORRELATION OF COMPLEX ARRAYS
;                CONRZ - CONVOLUTION OF REAL ARRAYS (ALL INIT COND.=0)
;                CONCZ - CONVOLUTION OF CMPLX ARRAYS (ALL INIT COND.=0)
;
;        7.2.1   CONR-CONC
;                TO TEST THE 'CONR' INSTRUCTION THE SOURCE ARRAY,
;                'DTY', IS CONVOLUTED WITH ANOTHER SOURCE, 'DTX',
;                AND THE RESULT IS STORED AT LOCATION 1024 IN RAM.
;                THE KNOWN RESULT IS STORED AT 'DTZN' OF MAIN MEMORY.
;                'CONR' IS ALSO TESTED USING RANDOM NUMBERS AND COMPARED
;                WITH SIMULATED RESULTS.
;                THE 'CONC' INSTRUCTION HAS SOURCE ARRAYS 'CTY' AND
;                'CTX', AND DESTINATION ARRAY 'CTZN'.
;
;        7.2.2   CORR-CORC
;                THESE INSTRUCTIONS ARE TESTED USING THE SAME DATA
;                AND METHODS AS THE CONR-CONC PAIR.
```



```
        S?WPD 8


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

;8.2     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 AFFILIAT-
;        ED WITH IT, THUS BIT 4 CAN BE ALTERED BY HITTING KEY 4.
;        SETTING OF ANY BIT OF LOCATION "SWREG" WILL SET BIT 0.
;        (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.
;
```

```
;8.2.1  OTHER COMMANDS
;
;       "CR"    A "RETURN" CAN BE TYPED TO CONTINUE THE PROGRAM
;               AFTER ITS LOCKED IN A SWITCH MODIFICATION MODE
;
;       ^D      THIS COMMAND GIVEN AT ANY TIME WILL RESET "SWREG"
;               TO DEFAULT MODE AND RESTART THE PROGRAM.
;
;       ^R      THIS COMMAND GIVEN AT ANY TIME WILL RESTART THE
;               PROGRAM. SWITCHES ARE LEFT WITH THE VALUES THEY
;               HAD BEFORE THE COMMAND WAS ISSUED.
;
;       ^O      THIS COMMAND GIVEN AT ANY TIME WILL CAUSE THE
;               PROGRAM CONTROL TO GO TO ODT (NOTE: THIS IS AN
;               OPTIONAL COMMAND AND IS AVAILBLE ONLY IF
;               ODTPK IS PRESENT)
;
;       M       THIS COMMAND GIVEN AT ANY TIME WILL PRINT THE
;               CURRENT OPERATING MODES.
;
;
```



```
;9.0    OPERATING PROCEDURE/OPERATOR INPUT
;
;       9.1     OPERATING PROCEDURE
;               1. LOAD THE PROGRAM IN THE COMPUTER
;                  USING THE BINARY LOADER OR DTOS TAPE.
;               2. SET SWITCHES TO ONE OF THE FOLLOWING
;                  STARTING ADDRESSES:
;
;                  500 OR 200 -          RUN ALL TESTS
;                  501 -         TEST CONR,CONRZ
;                  502 -         TEST CONC,CONCZ
;                  503 -         TEST CORR
;                  504 -         TEST CORC
;                  507 -         ENTER MEMORY DUMP
;
;               3. PRESS START.
;               4. THE PROGRAM PRINTS "PASS"
;                  AFTER SUCCESSFUL COMPLETION
;                  OF THE TESTS.
;
```

```
;10.   ERRORS
;      WHEN AN ERROR IS DETECTED A VALUE FOR AN
;      ERROR PC WILL BE PRINTED ALONG WITH AC'S 0,1 AND 2
;      OF THE HOST COMPUTER.
;
;      THE PC VALUE PRINTED MAY POINT TO ONE OF THE
;      FOLLOWING:
;
;              1. A CALL TO EHALT
;              2. A CALL TO THE ROUTINE WHICH DETECTED
;                 THE ERROR AND SUBSEQUENTLY EXITED
;                 TO EHALT SUCH AS THE VECTOR COMPARE
;                 ROUTINES TCOMP,CCOMP,RCOMP, AND RCCMP.
;
;      SUBROUTINES OF THIS TYPE HAVE SPECIAL ERROR PRINTOUTS
;      CALLED IN THE ERROR ROUTINE. IN THE CASE OF THE REAL
;      VECTOR COMPARE ROUTINES, TCOMP AND RCOMP, UP TO 3
;      PAIRS OF GOOD/BAD DATA IS PRINTED IN 2-16. BIT HEX
;      OR OCTAL FORMAT (SEE SW10), ALONG WITH THE MM ADDRESS
;      OF THE DATA AND THE RELATIVE INDEX WITHIN THE VECTOR.
;      A TOTAL ERROR COUNT IS PRINTED ALONG WITH
;      THE VECTOR/VECTOR N,IJ,IK DATA. IN THE CASE OF THE
;      COMPLEX VECTOR COMPARE ROUTINES, CCOMP AND RCCMP, THE
;      GOOD/BAD DATA IS PRINTED IN 4-16. BIT HEX OR OCTAL
;      FORMAT.

;      THE SIGNIFICANCE OF ERROR PRINTOUTS IS EXPLAINED IN THE
;      LISTING AT THE SOURCE OF THE ERROR.  IN GENERAL
;      AC0 CONTAINS THE CHECKWORD AND AC1 THE DATA IN
;      QUESTION.

;      AFTER PRINTING THE ABOVE THE PROGRAM HALTS ALLOWING
;      THE OPERATOR TO SET THE SWITCHES.  UPON PRESSING
;      CONTINUE THE PROGRAM MAY EITHER GO ON TO THE NEXT
;      TEST OR GO INTO A SCOPE LOOP BETWEEN THE APPROPRIATE
;      CALLS TO .SETUP AND .LOOP (SEE 8.0 SWITCH SETTINGS)
```



```
;11.0   DEBUG HELP
;
;      11.1    MONITOR LOCATIONS
;
;              LOCATION        LABEL   MEANING
;                200                   DTOS START
;                201                   FIRST LOCATION OF CURRENT TEST
;                202                   START ADDRESS
;                203                   XXXX
;                204                   PASS COUNT
;
;      11.2    PROGRAMMING DESCRIPTION
;
;               SEE PROGRAMMERS REFERENCE
;              FOR ARRAY PROCESSOR.
;
;      11.3    COMMON SUBROUTINES
;
;              A) ADDSET, SETP, LOOP -
;              THESE ALL WORK IN CONJUCTION TO SET UP THE TEST LOOP.
;              ADDSET TELLS SETP WHERE THE LOOP ENDS.
;              SETP MARKS THE START OF THE TEST LOOP.
;              LOOP MARKS THE END OF THE TEST LOOP. THE AP INSTRUCTION
;              IS EXECUTED AND THE RESULTS ARE CHECKED FOR ERROR
;              WITHIN THIS TEST LOOP.  IF AN ERROR OCCURS, THEN THE
;              PROGRAM CAN LOOP INDEFINITELY BETWEEN THE SETP AND
;              THE LOOP.  (SEE SECTION 8 ON VARIOUS SWITCH SETTINGS).
;
;              NORMALLY, WITH NO ERRORS, THE PROGRAM WILL GO THROUGH
;              THIS LOOP ONCE ON THE FIRST PASS OF THE PROGRAM.  ON
;              SUBSEQUENT PASSES THE PROGRAM WILL STAY IN THE LOOP
;              FOR 5 LOOPS (OR 100. DEPENDING ON THE PROGRAM).
;
;              B) PARAMETER BLOCK LOADER ROUTINES -
;              THESE CALLS ALL HAVE DIFFERENT NAMES DEPENDING
;              ON THE INSTRUCTION.  HOWEVER, THEY OPERATE IN THE SAME
;              MANNER.  THE COMMENT FIELD ON THE CALL WILL ALWAYS
;              INDICATE THAT THIS IS A PARAMETER BLOCK LOAD ROUTINE
;              AND WILL NAME THE AP INSTRUCTION THAT THE LOAD IS
;              COMPATIBLE WITH.
;              THE ARGUMENTS OF THE CALL ARE THE PARAMETERS OF THE AP
;              INSTRUCTION.  THESE PARAMETERS ARE TO LOADED BY THESE
;              ROUTINES INTO MAIN MEMORY AT A LOCATION POINTED TO BY
;              AC2.  AC2 MUST ALWAYS POINT TO THE PARAMETER BLOCK
;              DURING AP INSTRUCTION EXECUTION.
```

```
;        11.4    SEQUENCE OF TESTING
;
                 IT IS IMPORTANT THAT THE PROGRAMS BE
                 EXECUTED IN A SPECIFIC SEQUENCE:
;
;
;                APIS DIAG (CPU/AP ONLY)
;                APFP DIAG (CPU/AP ONLY)
;                APA EXER
;                APB EXER
;                APC EXER
;                APD EXER
;                APE EXER
;                APF EXER
;                APG EXER
;                APH EXER
;                API EXER
;                APJ EXER
;                APK EXER
```



```
                 O?DTD 12

;12.     OCTAL DEBUG TOOL (ODT)
;
;        THE DIAGNOSTIC IS EQUIPED WITH A BUILT IN ODT WHICH CAN
;        BE ACCESSED BY HITTING CONTROL O (^O) AT ANY TIME DURIN
;        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.
;
;12.1    CONVENTIONS AND SYMBOLS
;        THE FOLLOWING CONVENTIONS ARE USED BY THE ODT:
;        ?        PRESSING ANY ILLEGAL KEY CAUSES THE ODT TO RES-
;                 POND WITH A "?".
;        @        ODT IS READY AND AT YOUR SERVICE.
;
;12.2    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
;
;12.3    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 TTO STATUS
;                 BIT      INTERPRETATION
;                 15       STATUS OF TTO DONE FLAG
;                 14       STATUS OF INTERRUPTS (ION FLAG)
;                 13       STATUS OF CARRY BIT
;        6        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
;                 WITH OR WITHOUT MODIFICATION.
```

```
;       "LF"    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 + "ADDR".
;       -"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 EXPRESION RELATIVE TO THE ADDRESS
;       CAN BE DEPOSITED BY TYPING A "." OR ".+/-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 4A.
;       "ADR"R  START EXECUTING THE PROGRAM AT "ADR" AFTER AN
;               IO-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
;               THE ORIGINAL PROGRAM AREA. IF A BREAK POINT IS
;               PLACED OUTSIDE THIS AREA THE RESULTS WILL
;               BE UNPREDICTABLE.
;
;
```



```
;13.    SPECIAL NOTES/SPECIAL FEATURES
;
;       13.1    FOR A COMPLETE TEST ALL PROGRAMS
;               SHOULD BE EXECUTED WITH CAT/KITTEN.
;
;       13.2    A NOTE ABOUT AP ADDRESSING
;
;               ADDRESSES IN THE AP CAN BE OF SEVERAL
;               MODES:
;
;               1) ONE WORD MODE (IE) SAME AS STANDARD
;                  ADDRESSING
;               2) TWO WORD MODE (2WM) - EACH 32 BITS IS
;                  NOW ONE ADDRESS SPACE.  THIS IS USED
;                  IN THE AP TO SIMPLIFY REAL NUMBER
;                  ADDRESSING.
;               3) FOUR WORD MODE (4WM) - EACH 64 BITS IS
;                  NOW ONE ADDRESS SPACE.  THIS IS USED
;                  IN THE AP TO SIMPLIFY COMPLEX NUMBER
;                  ADDRESSING.
;
;               THE AP ACCESSES AN ADDRESS RELATIVE TO THE START OF
;               THE AP RAM.  THUS AP RAM LOC 0 WOULD BE THE
;               FIRST LOCATION THAT IS IN THE AP.  HOWEVER,
;               AS FAR AS THE ECLIPSE CPU IS CONCERNED, AP
;               LOC 0 IS CONTAINED AT LOCATION LABEL
;               "RAMPT". (IN PAGE ZERO.  SO, IF RAMPT CON-
;               TAINS 64000, THEN 2000 2WM (AP RAM) IS REALLY
;               64000+2000+2000=70000.
;
;               NOTE: "STOP ON STORE" OR "STOP ON ADDRESS" IN AP RAM
;               SPACE WILL NOT WORK IF THE AP IS USING
;               THE INTERNAL AP ADDRESS LINES TO ACCESS AP RAM.
```

```
!0013 .MAIN
01              ;14.0   RUN TIME
02              ;
03              ;       14.1    PASS 1                  4.5 MIN
04              ;
05              ;       14.2    SUBSEQUENT PASSES       8.5 MIN
06              ;

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

```
0014 .MAIN

O?DTD 001551  MC   10/01
S?WPD 001075  MC    4/01
```