DataGeneral

TECHNICAL STATEMENT

TEXT LISTING

068-001071-01

PROGRAM

ARRAY PROCESSOR EXERCISER-G

TEXT TAPE

097-001071-01

ABSTRACT

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

```
;*****************************************************************
;
; NAME: APG.TX                                PART NUMBER: 097-1071
;
;
; DESCRIPTION: ARRAY PROCESSOR EXERCISER-G.
;
;
; REVISION HISTORY:
;
;       REV.            DATE
;
;       00              03/01/78
;       01              12/29/78
;
;
; COPYRIGHT (C) DATA GENERAL CORPORATION, 1978
; ALL RIGHTS RESERVED.
;
;*****************************************************************
```



```
;
;               APG.TX          PART NUMBER:097-1071
;
;1.0    PROGRAM NAME
;
;       SOURCE FILE:    APG.SR
;       DTOS FILE:      APG EXER
;
;2.0    REVISION HISTORY
;
;       DATE    REVISION
;       ----    --------
;       3/1/78     0
;
;3.0    MACHINE REQUIREMENTS
;
;       1. ECLIPSE FAMILY CENTRAL PROCESSOR (S130 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.
;
;       THE PROGRAM EXERCISES THE ARRAY PROCESSOR COMPLEX AND
;       REAL FAST FOURIER TRANSFORM INSTRUCTIONS. IT IS INTENDED
;       TO AID IN THE INITIAL CHECKOUT OF THE AP AND MAY BE
;       USED TO HELP ISOLATE PROBLEMS IN THE FOURIER TRANSFORM
;       FUNCTIONS.
;
;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 FOURIER TRANSFORM THEORY

THE FOURIER TRANSFORM AS COMMONLY USED IN DIGITAL SIGNAL PROCESSING PROVIDES A TECHNIQUE FOR ANALYSIS OF THE FREQUENCY CONTENT OF SIGNALS INITIALLY SAMPLED IN THE TIME DOMAIN. THE DISCRETE FOURIER TRANSFORM TAKES THE GENERAL FORM OF:

```
                N-1
                --
                \          (-2*PI*I*J*K/N)
       Z(J) =   /_   X(K)*E                 ,J=0,1,....,N-1
                K=0
```

COMPUTATION OF THE ELEMENTS BY THE ABOVE TECHNIQUE WOULD BE EXTREMELY TIME-CONSUMING HOWEVER, AS IT WOULD NECESSITATE N**2 COMPLEX MULTIPLICATIONS AS WELL AS N**2 ADDITIONS. SINCE THE COMPLEX MULTIPLICATIONS TAKE A LOT MORE TIME ON A DIGITAL COMPUTER THAN THE ADDITIONS, ANY REDUCTION IN THE NUMBER OF MULTIPLICATIONS REQUIRED WOULD SPEED THE OPERATION SIGNIFICANTLY.

LET US ESTABLISH THE RESTRICTION THAT THE NUMBER OF ELEMENTS, N, MUST BE A POSITIVE POWER OF 2. WE MAY NOW SPLIT THE ORIGINAL ARRAY OF N ELEMENTS INTO TWO SMALLER ARRAYS OF N/2 ELEMENTS WE WILL CALL A AND B SUCH THAT:

```
        A(N)=X(2N) FOR N=0,1,....,N/2-1
        B(N)=X(2N+1) FOR N=0,1,....,N/2-1
```

WE NOW COMPUTE THE DFT'S OF THE ABOVE TWO ARRAYS, WHICH ONLY REQUIRE (N/2)**2 COMPLEX MULTIPLIES EACH, YIELDING TWO RESULT ARRAYS WHICH WE WILL CALL C AND D RESPECTIVELY. WE MAY THEN COMPUTE OUR FINAL SET OF ELEMENTS Z FROM THESE TWO INTERMEDIATE ARRAYS AS FOLLOWS:

```
                      (-2*PI*I*J/N)
       Z(J) = C(J)+D(J)*E               FOR J=0,1,2,....,N/2-1
```

THIS REQUIRES AN ADDITIONAL N COMPLEX MULTIPLIES FOR A TOTAL OF ((N/2)**2)*2+N OR N**2/2+N, A VERY SIGNIFICANT SAVINGS OVER THE INITIAL METHOD. WE CAN GO FURTHER THOUGH, SINCE AN N/2 ELEMENT DFT CAN BE COMPUTED BY USING TWO N/4 ELEMENT DFT'S. CARRYING THIS PROCESS TO ITS LOGICAL CONCLUSION WE ARRIVE AT AN ALGORITHM FOR DECIMATION IN TIME AS INITIALLY PROPOSED BY COOLEY AND TUKEY WHICH USES PAIRS OF ELEMENTS FOR EACH ITERATION. THE BASIC BUTTERFLY COMPUTATIONS, AS THEY ARE CALLED, REQUIRE ONLY ONE COMPLEX MULTIPLY PER BUTTERFLY RESULTING IN (N/2)*(LOG N BASE 2) COMPLEX MULTIPLYS FOR THE ENTIRE DFT. IF THE OPERATION IS PERFORMED IN-PLACE NO ADDITIONAL MEMORY IS REQUIRED TO STORE THE RESULTS, BUT THE ORDER OF THE ELEMENTS IS BIT-REVERSED FROM THAT OF THE STARTING ARRAY.

7.1 TEST PROGRAM DESIGN

THE DIAGNOSTIC TEST FOR THE FOURIER TRANSFORM FUNCTIONS CONSISTS OF A SERIES OF PROGRESSIVELY MORE COMPLEX TESTS USING SIMPLE CONSTANTS AND BOUNDED RANDOM NUMBERS AS INPUT DATA. TWO SUBROUTINES PROVIDE SOFTWARE SIMULATION OF THE FFTC AND FFTR INSTRUCTIONS, AND THE DATA RESULTING FROM THE SIMULATION IS COMPARED WITH THE CORRESPONDING DATA FROM THE A.P. HARDWARE INSTRUCTIONS.

TESTS A0-A7 PERFORM COMPLEX FFT'S OF IMPULSE FUNCTIONS, WHICH SHOULD RESULT IN D.C. CONSTANTS OF THE SAME MAGNITUDE AS THE ORIGINAL IMPULSE. WE CHECK ALL LEGAL ARRAY SIZES AS WELL AS A DISPLACED STARTING ADDRESS FOR THE ARRAY.
IN TESTS B- WE CHECK FFTC OF D.C. CONSTANTS, WHICH SHOULD GIVE US AN IMPULSE OF MAGNITUDE N*(D.C. VALUE).

TESTS C- PERFORM THE IMPULSE TESTS USING AN INVERSE FFTC. RESULTS SHOULD BE THE SAME AS IN TESTS A- ABOVE.

TESTS D- PERFORM THE FFTC INVERSE ON D.C. CONSTANTS AS IN TESTS B- ABOVE. RESULTS SHOULD BE THE SAME.

TESTS E- PERFORM A FORWARD FFTC OF COSINE FUNCTIONS, WHICH SHOULD YIELD AN IMPULSE AT THE FREQUENCY OF THE FUNCTION PLUS SOME LOWER LEVEL COMPUTATIONAL NOISE.

TESTS F- USE VECTORS OF ASSORTED COMPLEX BOUNDED RANDOM NUMBERS AND PERFORM FORWARD FFTC INSTRUCTIONS ON THIS DATA. OUTPUT DATA SHOULD MATCH THAT OF THE SUMULATOR, BUT IS UNLIKELY TO PROVIDE EASY ANALYSIS.

TESTS G- USE DATA AS IN F- ABOVE, BUT PERFORM INVERSE FFTC INSTRUCTIONS ON THE DATA. OUTPUT DATA FROM THESE TESTS SHOULD ALSO MATCH SIMULATOR RESULTS.

THE FFTR INSTRUCTION DOES NOT ACTUALLY PERFORM A REAL FOURIER TRANSFORM, BUT MERELY EXECUTES THE ARRAY ELEMENT ADDS AND MULTIPLIES REQUIRED TO CONVERT THE RESULT OF AN FFTC INSTRUCTION ON PACKED REAL DATA INTO THE CORRESPONDING COMPLEX DATA ELEMENTS THAT WOULD HAVE RESULTED FROM A FOURIER TRANSFORM OF REAL INPUT DATA. THE TESTS FOLLOW A FORMAT IDENTICAL WITH THAT OF THE FFTC TESTS EXCEPT THE FFTR INSTRUCTION ALSO PROVIDES FOR USE OF AN EXTERNAL SIN-COS TABLE FOR 2048 ELEMENT OPERATIONS. THE TESTS USING THIS EXTERNAL TABLE ARE THE LAST IN EACH SECTION H- THROUGH L-

```
            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 PROG
            ;               AFTER ITS LOCKED IN A SWITCH MODIFICATION MO
            ;
            ;       ^D      THIS COMMAND GIVEN AT ANY TIME WILL RESET "S
            ;               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 OF DTOS TAPE.
         ;                2. SET SWITCHES TO ONE OF THE FOLLOWING
         ;                   STARTING ADDRESSES:
         ;
         ;                500 OR 200 -    RUN ALL TESTS
         ;                501 -   RUN FFTC FORWARD OF IMPULSE
         ;                502 -   RUN FFTC FORWARD OF D.C. CONSTANTS
         ;                503 -   RUN FFTC INVERSE OF IMPULSE
         ;                504 -   RUN FFTC INVERSE OF D.C. CONSTANTS
         ;                505 -   RUN FFTC FORWARD OF COSINES
         ;                506 -   RUN FFTC FORWARD OF RANDOM VECTORS
         ;                507 -   RUN FFTC INVERSE OF RANDOM VECTORS
         ;                510 -   RUN FFTR FORWARD OF IMPULSE
         ;                511 -   RUN FFTR FORWARD OF D.C. CONSTANTS
         ;                512 -   RUN FFTR INVERSE OF D.C. CONSTANTS
         ;                513 -   RUN FFTR FORWARD OF RANDOM VECTORS
         ;                514 -   RUN FFTR INVERSE OF RANDOM VECTORS
         ;                515 -   START MEMORY DUMP UTILITY PROGRAM
         ;
         ;                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
         ;        AS THEY EXISTED AT THE END OF THE LAST AP
         ;        INSTRUCTIONS BEING TESTED.
         ;
         ;        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 COMPARF ROUTINES, TCOMP AND RCOMP, UP TO 3
         ;        PAIRS OF 32-BIT GOOD/BAD DATA ARE PRINTED GROUPED AS
         ;        TWO 16-BIT HEX OR OCTAL FORMAT WORDS (SEE SW10), ALONG
         ;        WITH THE MM ADDRESS OF THE BAD (HARDWARE) RESULT DATA
         ;        AND THE RELATIVE INDEX WITHIN THE VECTOR. A TOTAL ERROR
         ;        COUNT IS PRINTED ALONG WITH THE VECTOR/VECTOR N,IJ, AND
         ;        IK DATA WHERE N IS THE NUMBER OF ELEMENTS BEING CHECKED, AND
         ;        IJ AND IK ARE THE INDEX INCREMENTS FOR THE GOOD AND BAD
         ;        DATA ARRAYS, RESPECTIVELY. IN THE CASE OF THE COMPLEX
         ;        VECTOR COMPARE ROUTINES, CCOMP AND RCCMP, THE 64-BIT
         ;        GOOD/BAD DATA IS PRINTED GROUPED AS FOUR 16-BIT HEX
         ;        OR OCTAL FORMAT WORDS.
         ;
         ;        THE SIGNIFICANCE OF ERROR PRINTOUTS IS EXPLAINED IN THE
         ;        LISTING AT THE SOURCE OF THE ERROR. ADDRESSES OF THE
         ;        ANSWER BUFFERS AND INFORMATION ABOUT WHAT IS BEING
         ;        TESTED MAY TYPICALLY BE FOUND THERE. FOR THOSE
         ;        TESTS WHICH USE THE EHALT CALLS FROM THE MAIN TEST
         ;        PROGRAM, 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                   OTOS 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 CONJUNCTION 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 BE 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.
;
;
;               SUBROUTINES USED BY APG INCLUDE THOSE COMMON TO
;               ALL THE AP DIAGNOSTICS SUCH AS BLOCK MOVE, BLOCK
;               FILL, AND RANDOM NUMBER GENERATION AS WELL AS A
;               COUPLE ROUTINES UNIQUE TO THIS TEST. MOST OF THE
;               SUBROUTINES HAVE BEEN DESCRIBED IN EARLIER TESTS
;               AND WILL NOT BE REPEATED HERE.
;
;               SCFT-   SIMULATE COMPLEX FOURIER TRANSFORM
;                 THIS SUBROUTINE PERFORMS A COMPLEX FOURIER TRANSFORM
;               ON AN ARRAY IN MAIN MEMORY USING THE SAME ALGORITHM
;               AND SEQUENCE OF OPERATIONS AS THE ARRAY PROCESSOR
```



```
;               HARDWARE. IT MAKES CALLS TO THE ADD AND MULTIPLY
;               SUBROUTINES USED BY ALL AP DIAGNOSTICS TO PERFORM
;               SINGLE PRECISION FLOATING POINT CALCULATIONS. THREE
;               ARGUMENTS ARE REQUIRED FOLLOWING THE JSR INSTRUCTION
;               CALLING THE SUBROUTINE. THEY ARE:
;                       XADR    STARTING ADDRESS OF INPUT ARRAY
;                       N       NUMBER OF ELEMENTS (INTEGER POWER OF 2)
;                       INVER   FLAG WITH BIT 15=1 FOR FWD, 0 FOR INV
;
;               SRFT-   SIMULATE REAL FOURIER TRANSFORM
;               THIS SUBROUTINE SIMULATES THE FFTR INSTRUCTION OF
;               THE AP. IT ALSO REQUIRES THREE ARGUMENTS FOLLOWING
;               THE JSR TO THE SUBROUTINE AS FOLLOWS:
;                       XADR    STARTING ADDRESS OF REAL ARRAY
;                       N       NO. COMPLEX ELEMENTS (1/2*NO. OF REAL)
;                       SOT     BITS<14-15>=01 FOR FWD, 11 FOR INV
```

```
;
;       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
```



```
 FU00000 000000 0?DTD 12
```

```
!0013 .MAIN
          ;13.     SPECIAL NOTES/SPECIAL FEATURES
          ;
          ;        13.1     FOR A COMPLETE TEST ALL PROGRAMS
          ;                 SHOULD BE EXECUTED WITH CAT/KITTEN, USING EITHER
          ;                 AND I/O TESTER OR A DISK.
          ;
          ;        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.
```

```
!0014 .MAIN
          ;14.0   RUN TIME
          ;
          ;       14.1    PASS 1                 1.5 MIN
          ;
          ;       14.2    SUBSEQUENT PASSES      2.5 MIN
          ;

**00002 TOTAL ERRORS, 00000 PASS 1 ERRORS
```

0015 .MAIN

S?WPD 001075 MC 5/02
?OTD 000000U 12/02