

DataGeneral

**TECHNICAL
STATEMENT**

TEXT LISTING

068-001075-01

PROGRAM

ARRAY PROCESSOR EXERCISER-K

TEXT TAPE

097-001075-01

ABSTRACT

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

```

0001 APK.T MACRO REV 06.30 08:32:35 05/25/79
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 *****

*****
NAME: APK.TX
PART NUMBER 097-1075

DESCRIPTION: ARRAY PROCESSOR EXERCISER-K.

REVISION HISTORY:
REV. DATE
00 03/29/78
01 12/29/78

COPYRIGHT (C) DATA GENERAL CORPORATION, 1978
ALL RIGHTS RESERVED.
*****

APK.TX
PROGRAM NAME
:1.0
SOURCE FILE: APK.SR
DTOS FILE: APK EXER

REVISION HISTORY
:2.0
DATE REVISION
:00
XX/XX/78 00

MACHINE REQUIREMENTS
:3.0
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

TEST REQUIREMENTS
:4.0
1. SAME AS MACHINE REQUIREMENTS
2. I/O TESTER OR DISC
3. MHPUI

SUMMARY
:5.0
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.

RESTRICTIONS
:6.0
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.

```

10003 APK.T

```

17.0 PROGRAM DESCRIPTION/THEORY OF OPERATION
17.2 APK WILL CHECK TO SEE IF AP INSTRUCTIONS TRAP
ON STATUS AND WHETHER SINGLE STATUS CONDITIONS
ARE RETURNED.
AT THIS POINT PLEASE REFER TO AP REFERENCE
MANUAL ON AP TRAP ON STATUS CAPABILITIES.
THE TESTS ARE REALLY SELF DOCUMENTING.
A P-BLOCK IS LOADED TO CREATE SOME ERROR CONDITIONS.
TRAP CONDITIONS ARE SET AND CHECKS ARE MADE TO SEE
IF PROPER STATUS WAS RETURNED AND DID PROPER TRAP-
PING OCCUR.
EACH PARAMETER BLOCK LOAD CALL EXPANDS TO SHOW
WHICH PARAMETERS ARE BEING LOADED. IF YOU EXAMINE
THE VERY FIRST TEST YOU WILL SEE THE "BLOR" EXPANSION.
IT LISTS ALL PARAMETERS BEING LOADED BY NAME. THEN,
FOLLOWING "ARCT/2 LINE, THE POSITION IN THE P-BLOCK
THEN THE ARGUMENT IS LISTED (ONE PAIR AFTER ANOTHER).
AC2 POINTS TO THE PARAMETER BLOCK.
THIS METHOD OF EXPANSION SHOULD ENABLE ONE TO EXAMINE
CLEARLY WHAT IS BEING DONE.
ALL STRUCTURES ARE IDENTICAL WITH EXCEPTION TO THE
FIRST 3 TESTS WHICH CHECK. IF CORRECT INFO IS PUSHED
ON STACK DURING ERROR.
OTHER TEST STRUCTURES LOAD A PARAMETER BLOCK AND
EXECUTES THE INSTRUCTION.
NOTE: THIS TEST ONLY CHECKS FOR ONE STATUS CONDITION AT
A TIME.

```

10004 APK.T

```

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

```

SWITCH COMMANDS

```

18.2 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.

```

```

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

```

```

0005 APK.T
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
;8.2.1 OTHER COMMANDS
;
"CR" A "RETURN" CAN BE TYPED TO CONTINUE THE PROGRAM
AFTER ITS LOCKED. IN A SWITCH MODIFICATION MODE
;
"0 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.
;
"0 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.
;

10006 APK.T
01
02
03
04
05
06
07
08
09
10
11
12
13
14
;9.0
;
9.1 OPERATING PROCEDURE IN THE COMPUTER
1. LOAD THE PROGRAM IN THE COMPUTER
USING THE BINARY LOADER OR DTOS TAPE.
2. SET SWITCHES TO ONE OF THE FOLLOWING
STARTING ADDRESSES:
500,200 - RUN ALL TESTS
;
3. PRESS START.
4. THE PROGRAM PRINTS "PASS" FOLLOWING
THE SUCCESSFUL COMPLETION OF THE TEST
;

```

10007 APK.T

```

01 110.
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 ;

```

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 RCCMP, 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,I,J,K 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
 ACO 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)

10008 APK.T

```

01 ;
02 ;
03 ;
04 ;
05 ;
06 ;
07 ;
08 ;
09 ;
10 ;
11 ;
12 ;
13 ;
14 ;
15 ;
16 ;
17 ;

```

: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.

10009 APK.T

```

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

```

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.

```

10010 APK.T

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19

```

```

C) SLICING ROUTINES ~
THESE TYPES OF ROUTINES EXIST IN SEVERAL DIFFERENT
THEY ALL HAVE VERY SIMILAR NAMES, FOR EXAMPLE - SLICE,
SLICC, SLICA, SLICB. THESE ARE ALL JSR'S TO SOME SLICING
ROUTINE.
THE SLICING ROUTINES WILL SET UP VARIOUS RANDOM
SLICES IN AP RAM FOR THE VARIOUS INPUT AND OUTPUT
ARRAYS USED IN THE AP INSTRUCTIONS.
TO CREATE THESE SLICES, RANDOM N AND STARTING ADDRESS
ARE CREATED. THE COMMENTS ON THE CALL WILL INDICATE
THE DESTINATION OF THESE RANDOM PARAMETERS WITHIN THE
TEST CODE. THE ROUTINES ALL SLICE DIFFERENTLY.
ONE WILL DIVIDE THE RAM IN HALF AND THEN FIND
A RANDOM SLICE WITHIN EACH HALF. ANOTHER MIGHT
FIND A RANDOM SLICE FOR EACH THIRD OF RAM.

```

10011 APK.T

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
- APP EXER
- APH EXER
- API EXER
- APJ EXER
- APK EXER

10012 APK.T

070TD 12

OCTAL DEBUG TOOL (ODT)

:12.

THE DIAGNOSTIC IS EQUIPPED WITH A BUILT IN ODT WHICH CAN BE ACCESSED BY HITTING CONTROL 0 (-0) AT ANY TIME DURING THE EXECUTION OF THE PROGRAM (AFTER SETTING THE PARAMETERS).

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:
- ! POUND WITH A "!"
- @ 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. LEADING 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.

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

FOR ACCUMULATORS 0-3

FOR PC OF THE NEXT INSTRUCTION TO BE EXECUTED IN THE EVENT OF A "P" COMMAND.

CPU AND I/O STATUS

INTERPRETATION

STATUS OF I/O DONE FLAG

STATUS OF INTERRUPTS (ION FLAG)

STATUS OF CARRY BIT

ADDRESS OF THE LOCATION HAVING THE BREAK POINT (IF ANY)

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"/ AND "ADR" TO THE POINTER, OPEN THE CELL AND PRINT ITS CONTENTS.
- "ADR"/ AND "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.

```

0013 APK.1
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

"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 + "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 "R" OR "A" 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 OOT 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 OOT 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 "A".
"ADR"R START EXECUTING THE PROGRAM AT "ADR" AFTER AN
TO-RESET.
K KILL THE STRING TYPED SO FAR. THE OOT 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
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.

0014 APK.1
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

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 (16M) SAME AS STANDARD
ADDRESSING
2) TWO WORD MODE (20M) - EACH 32 BITS IS
NOW ONE ADDRESS SPACE. THIS IS USED
IN THE AP TO SIMPLIFY REAL NUMBER
ADDRESSING.
3) FOUR WORD MODE (40M) - 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 2MM (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.

```


10015 APK.T

01 ;14.0 RUN TIME
02 ;
03 ; 14.1 PASS 1
04 ;
05 ; 14.2 SUBSEQUENT PASSES
06 ;
08 ;TITL APK.TX
;END

**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS

0016 APK.T

070TD 001520 MC 12/01
S?WPD 001044 MC 4/01