

TEXT LISTING

068-001076-00

PROGRAM

AP FLOATING POINT DIAGNOSTIC

TEXT TAPE

097-001076-00

ABSTRACT

THIS PROGRAM IS A HARDWARE DIAGNOSTIC FOR THE ARRAY PROCESSOR. THE PROGRAM IS EXECUTED BY THE HOST COMPUTER CONTROLLING THE AP AND IS A GATE BY GATE TEST OF THE LOGIC IN THE ARRAY PROCESSOR. THE PROGRAM ASSUMES ONLY THE THE HOST COMPUTER IS FUNCTIONING PROPERLY.

```

0001 .MAIN          MACRO REV 06.30          14:04:22 08/11/78
01
02
03
04
05
06
07 *****
08 ? NAME: APPP.TX          PART NUMBER: 097-001076
09 ?
10 ?
11 ? DESCRIPTION: AP FLOATING POINT DIAGNOSTIC
12 ?
13 ?
14 ? REVISION HISTORY:
15 ?
16 ? REV.          DATE
17 ?
18 ?          00          08/11/78
19 ?
20 ?
21 ? COPYRIGHT (C) DATA GENERAL CORPORATION, 1978
22 ? ALL RIGHTS RESERVED.
23 ? LICENSED MATERIAL-PROPERTY OF DATA GENERAL CORPORATION.
24 ? *****
25
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
29
30
31
32
33
34
35
36
37
38
39

? 1. PROGRAM NAME: APPP_SR -ARRAY PROCESSOR MICRO DIAGNOSTIC -FP
?
? 2. REVISION HISTORY: N/A
?
? 3. MACHINE REQUIREMENTS:
? 1. ECLIPSE S130 CENTRAL PROCESSOR
? 2. MINIMUM OF 20K MEMORY
? 3. API,AP2,AP3 PROCESSOR BOARDS
? 4. BASIC I/O TELETYPE/DISPLAY AND CONTROL
?
? 4. TEST REQUIREMENTS: N/A
?
? 5. SUMMARY
? THIS PROGRAM IS A HARDWARE DIAGNOSTIC FOR THE
? ARRAY PROCESSOR. THE PROGRAM IS EXECUTED BY THE
? HOST COMPUTER CONTROLLING THE AP AND IS
? A GATE BY GATE TEST OF THE LOGIC IN THE ARRAY
? PROCESSOR. THE PROGRAM ASSUMES ONLY THAT THE
? HOST COMPUTER IS FUNCTIONING PROPERLY.
? THE PROGRAM UTILIZES ONLY THE STANDARD ECLIPSE
? INSTRUCTION SET AND THE AP-MAINTENANCE INSTRUCTION
? SET DESCRIBED IN SECTION 11C. NO CHECKING OF THE USER AP
? INSTRUCTION SET(SIS) AND ASSOCIATED CONTROL STORE
? IS CHECKED IN THIS PROGRAM.
?
? THE AP FLOATING POINT DIAGNOSTIC CHECKS ONLY THE
? AP2 AND AP3 FLOATING POINT LOGIC ALONG WITH
? ASSOCIATED I/O REGISTERS AND DATA PATHS. THE AP
? STORAGE AND INTEGER ARITHMETIC LOGIC IS CHECKED
? WITH THE AP INTEGER DIAGNOSTIC, PART NUMBER NNNNNN.
?
? DUE TO THE LIMITATIONS OF THE MAINTENANCE INSTRUCTION
? SET, VARIOUS AREAS OF THE LOGIC (ESSENTIALLY IN THE
? AREA OF THE CONTROL STORE AND DECODE) WILL REMAIN
? UNCHECKED. THUS RUNNING THE AP-EXERCISER PROGRAMS
? IS STILL REQUIRED. HOWEVER, THE BULK OF THE AP STORAGE
? AND ARITHMETIC LOGIC IS CHECKED AND IT IS HOPED THAT A
? GREAT DEAL MORE DIAGNOSTIC INFORMATION CAN BE DERIVED
? USING THIS PROGRAM.

```

10003 .MAIN

```

01 :
02 :
03 :
04 :
05 :
06 :
07 :
08 :
09 :
10 :
11 :
12 :
13 :
14 :
15 :
16 :
17 :
18 :
19 :
20 :
21 :
22 :

```

6. RESTRICTIONS:  
THIS PROGRAM CAN NOT BE RUN IN AN ICP-AP CONFIGURATION  
AS THE IOP DOES NOT SUPPORT THE MIS MICROCODE.

7. PROGRAM DESCRIPTION/THEORY OF OPERATION

THIS PROGRAM UTILIZES THE MAINTENANCE INSTRUCTION  
SET) OF THE ARRAY PROCESSOR TO EXERCISE AND PROVIDE  
DIAGNOSTIC INFORMATION ABOUT THE FOLLOWING SECTIONS  
OF THE ARRAY PROCESSOR:

- AP3 FLOATING PT. ADDER AND REGISTERS
- AP2 FLOATING PT. MULTIPLIER AND REGISTERS
- AP3 FLOATING PT. LATCH LOGIC
- AP2 FLOATING PT. LATCH LOGIC

THE MIS PROVIDES A MEANS FOR EXERCISING WELL DEFINED  
HARDWARE AREAS OF THE ARRAY PROCESSOR. A COMPLETE  
SET OF INSTRUCTIONS DEFINITIONS AND EXPLANATIONS  
IS FOUND IN SECTION 11C OF THIS TEXT FILE AS WELL AS  
IN THE ACTUAL PROGRAM LISTING.

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 :

```

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 MCNITOR.  
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 OPTICS 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	INTERPRETATION
VALUE	VALUE	VALUE	
1	40000	1	LOOP ON ERROR
2	20000	1	SKIP LOOPING ON ERROR
3	10000	1	PRINT TO CONSOLE
4	04000	1	ABORT PRINT CLT TO CONSOLE
5	02000	1	DO NOT PRINT _% FAILURE
6	01000	1	PRINT _% FAILURE
10	00040	0	ALLOW END OF PASS PRINT OUT
		1	SUPPRESS END CF PASS PRINT OUT
		0	DO NOT PRINT CN THE LINE PRINTER
		1	PRINT ON THE LINE PRINTER
		0	DO NOT HALT ON ERROR
		1	HALT ON ERROR
		0	PRINT ERROR WORDS IN OCTAL FORMAT
		1	PRINT ERROR WORDS IN HEX FORMAT

10005 .MAIN

01 SWITCH COMMANDS  
02 ONCE THE PROGRAM STARTS EXECUTING THE STATE OF ANY OF  
03 THE BITS CAN BE CHANGED BY HITTING KEYS 1-9, A-F. THE  
04 PROGRAM WILL CONTINUE RUNNING AFTER UPDATING THE OPTIONS  
05 EACH KEY WILL COMPLEMENT THE STATE OF THE BIT AFFILIAT-  
06 ED WITH IT, THUS BIT 4 CAN BE ALTERED BY HITTING KEY 4.  
07 SETTING OF ANY BIT OF LOCATION "SMREG" WILL SET BIT 0.  
08 (DEFAULT MODE IS DEFINED AS ALL BITS OF SMREG SET TO 0)  
09 THE PROGRAM CAN BE LOCKED INTO SWITCH MODIFICATION MODE  
10 BY TYPING A 0, IN WHICH CASE MORE THAN ONE BIT CAN BE  
11 CHANGED BEFORE CONTROL IS ALLOWED TO RETURN TO THE  
12 MAIN PROGRAM.  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35

10006 .MAIN

01 OPERATING PROCEDURE  
02 1. LOAD THE PROGRAM IN THE HOST COMPTLER  
03 2. SET SWITCHES EQUAL TO ONE OF THE FOLLOWING  
04 STARTING ADDRESSES:  
05 176 ENTER OCTAL DEBUGGER (OOB)  
06 177 ENTER MEMORY DUMP  
07 200,500  
08 RUNALL TESTS  
09 AP3 FLOATING POINT ADDER  
10 AP2 FLOATING POINT MULTPLIER  
11 AND ALL FLOATING POINT LATCH TESTS  
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

PROGRAM OUTPUT/ERROR DESCRIPTION

WHEN AN ERROR IS DETECTED A PRINTOUT OF THE FOLLOWING GENERAL FORMAT IS OUTPUT TO EITHER THE TTY AND OR THE LINE PRINTER:

OPTION PRINTOUT \*\*EXPLAINED BELOW

PC NNNNN  
AC0= XXXXX AC1= YYYYY AC2= ZZZZZ

THE PC VALUE MAY POINT TO ONE OF THE FOLLOWING:

1. A CALL OF EHALT  
NO OPTION PRINTOUT; SIGNIFICANCE OF AC'S 0-2 (IF ANY) IS EXPLAINED IN THE PROGRAM LISTING AT OR NEAR THE EHALT CALL.

2. A CALL OF EITHER RNCOMP OR CNCOMP  
IN THE CASE OF RNCOMP, WHICH COMPARES 2 REAL (32 BITS) NUMBERS, THE OPTICA PRINTOUT IS THE ENTIRE GOOD AND BAD 32 BIT NUMBERS IN EITHER A HEX OR OCTAL FORMAT. (SEE SW10 OPTION)

IN THE CASE OF CNCOMP, WHICH COMPARES 2 COMPLEX (64 BITS) NUMBERS, THE OPTICA PRINTOUT IS THE EN GOOD AND BAD 64 BIT NUMBERS IN EITHER A HEX OR OCTAL FORMAT.

FOR EITHER OF THE ABOVE, ACC CONTAINS THE GOOD 16 BITS, AC1 THE BAD 16 BITS CAUSING THE ERROR, AND AC2 POINTS TO THE START OF THE BAD 2 OR 4 WORD DATA BLOCK.

```

10007 .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 ?

3. A CALL TO ONE OF THE FOLLGKING:
A1PB1,MA1PB1,A1MB2,MA2MB1,M0XM2,M1XM3,A1PB3
A3PB1,A1B0VM,A0B1VM

THESE SUBROUTINES ALL HAVE THE FOLLOWING
CALL FORMAT. USING A1PB1 AS AN EXAMPLE:
A1PB1      LOAD A1,B1/EXECUTE P1P1
N          ARG1(32 BIT REAL # FCRMAT)
N          ARG2(32 BIT REAL # FCRMAT)
N          ARG1 OP ARG2 RESULT/OP =+,-,*, OR
           f(32 BIT REAL FCRMAT)

IN GENERAL ARG1 AND ARG2 ARE REAL 32 BIT
ARGUMENTS THAT ARE LOADED INTO SPECIFIC AP
REGISTERS AS EXPLAINED IN THE LISTING AND
OPERATED ON BY EITHER THE AP3 F.P. ADDER OR
THE AP2 F.P. MULTIPLIER. THE RESULT VIA WREG
IS LOADED BACK INTO MAIN MEMORY AND CHECKED
AGAINST THE ARG1 OP ARG2 RESULT.

IF AN ERROR IS DETECTED VIA THE ABOVE CHECK,
THE OPTION PRINTOUT IS THE ARG1,ARG2 VALUES
AS WELL AS THE 32 BIT GOOD/BAD VALUES. ALL
ARE PRINTED IN EITHER A HEX OR OCTAL FORMAT AS
PER THE SW10 CONTROL.

FOR THE ABOVE,AC0,AC1 CONTAIN THE GOOD/BAD
16 BIT VALUES,CAUSING THE ERROR AND AC2 POINTS
TO THE START OF THE 2 WORD BAD DATA BLOCK.

10008 .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 ?

11.  DEBUG HELP
11.A  SUBROUTINE DEFINITIONS

STORES LOOP ADDRESS IN LOC ERR1,
WHICH THE ERROR HANDLER (.EHALT) USES
AS A RETURN ADDRESS TO LOOP
FOLLOWING AN ERROR.

FLOATING PT ADD/SUBTRACT/MULT ROUTINES
SFADD/SFSUB/SFMUL
ADDRESS OF ARG1
ADDRESS OF ARG2
ADDRESS OF RESULT

RNFSP      GENERATE A RANDOM REAL #
ADDRESS    ADDRESS OF STORAGE LOC
***NUMBER GENERATED IS A NORMALIZED
REAL 32 BIT # WITH EXPONENT BOUNDED
AT 65. +-7.

*****
THE FOLLOWING SUBROUTINES HAVE A COMMON CALL FORMAT
AS DESCRIBED IN SECTION 10.3. THE INDIVIDUAL FUNCTIONS
FOLLOW:

A1PB1      LOAD A1,B1 AND EXECUTE P1P1(A1+B1)
MA1PB1     LOAD A1,B1 AND EXECUTE M1P1(-A1*B1)
MA2MB1     LOAD A2,B1 AND EXECUTE M2M1(-A2-B1)
A1MB2     LOAD A1,B2 AND EXECUTE P1M2 (A1-B2)
M0XM2     LOAD M0,M2 AND EXECUTE M02LL(M0XM2)
M1XM3     LOAD M1,M3 AND EXECUTE M13(M1 X M3)
A1PB3     LOAD A1,B3, ZERO B1 AND EXECUTE
           P1P1(A1+0 >B3) AND P1P3 (A1+B3)
A3PB1     LOAD A1,B1, AND ZERO B2 AND EXECUTE
           A1MB2(A1-0 >A3) AND P3P1 (A3*B1)
A1B0VM    LOAD M3,A1, AND 1E0 TO M0 AND EXECUTE
M03A1 (1 X M3 TO B0/A1*B0 TO WREG)
A0B1VM    LOAD M1,B1, AND 1E0 TO M2 AND EXECUTE
M12B1 (M1 X 1E0 TO A0/ A0*B1 TO WREG)

*****
M.02LL    EXECUTE M02LL WITH THE FOLLOWING CONDITI
N         TEST CONDITION BITS TO AC1
ARG1     REAL 32 BIT ARG1 TO M0,B1
ARG2     REAL 32 BIT ARG2 TO M1,A1
NORMAL RETURN **LATCH SET OR NOT SET ON BASIS OF
TEST CONDITION AND ARG1,ARG2

```

```

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

; 118 TEST SEQUENCE
;
; THE HARDWARE TESTING VIA RUNALL FOLLCS:
;
; EO-E2
;
; E10-E95
;
; E100-E128
;
; FO-F3
;
; F10-F71
;
; F80-F91
;
; F100-F115
;
; F116-F121
;
; 11C.
; LRO
; LRX
; RFX
; RRO
; RMO
; RND01
; ROR01
; RND01
; ROR01
; ADD01
; SUB01
; INCO
; DECO
; ADI05
; SBD07
; DBL0
; DBI0
; LDW
; STW
; LSOT
; LOZYX

; AP1, AP3 A1,B1 DATA REGISTERS AND DATA
; PATHS. ***DATA PATH IS VIA F.P. ADDER
; AP3 FLOATING POINT ADDER VIA A1,B1 DATA
; REGISTERS.
; AP3- A2,A3,B2,B3 REGISTERS AND SELECTION
; AS WELL AS SUBTRACT LOGIC OF THE F.P.
; ADDER.
;
; AP2 M0-M3 DATA REGISTERS, AP3 MD REGISTER
; A 1.0*N MULTIPLY FOR THE AP2 F.P. MULTIP
; AP2 FLOATING POINT MULTIPLY
; AP3 A0-B0 REGISTERS (LOADED WITH THE
; MULTIPLIER OUTPUT) AND SELECTION.
; AP3 ADDER (A1,B1) LATCH LOGIC
; AP2 MULTIPLIER (M0,M2) LATCH LOGIC
;
; 1) AP2 INSTRUCTIONS
;
; AC0<4-15>=>AP REG.& AC1<13-15>
; AC0<4-15>=>AP REG.& AC1<13-15>
; AC0<4-15>=>AP REG.& AC1<13-15>
; R0<15-4>=>AC0<4-15>
; COMPLEMENT OF R0=>AC0<4-15>
; (R0 AND R1)>AC0<4-15>
; (R0 OR R1)>AC0<4-15>
; (R0 XOR R1)>AC0<4-15>
; COMPLEMENT OF (R0 AND R1)>AC0<4-15>
; COMPLEMENT OF (R0 OR R1)>AC0<4-15>
; ((R2 MINUS R1)>R0 AND AC0<4-15>, CARRY=>L
; ((R2 MINUS R1)>R2 AND AC0<4-15>
; I=>LATCH IF RESULT IS ZERO
; O=>LATCH IF RESULT IS NOT
; (R0 PLUS 1)>AC0<4-15>
; (R0 MINUS 1)>AC0<4-15>
; (R4 PLUS R5 PLUS 1)>R4 AND AC0<4-15>
; (R6 MINUS R7 MINUS 1)>R6 AND AC0<4-15>
; I=>LATCH IF RESULT
; O=>LATCH IF RESULT
; (R0 PLUS R0)>AC0<4-15>
; (R0 PLUS R0 PLUS 1)>AC0<4-15>
;
; (2) AP1 INSTRUCTIONS
;
; AC0<0-14,15>=>SCT REGISTER IN API
; TAR CONTENTS ARE DESTROYED. MICRO CODE
; USES MICRO-CRDS LK1,LMB, LWC,
; CONTENTS OF TAR AND AC2 REMAIN
; WREG<0-63>=>4 CONSECUTIVE WCRDS IN MEMO
; TAR CONTENTS ARE DESTROYED. MICRO CODE
; USES MICR0-CRDER RN1,RNB,RMC, A
; CONTENTS OF TAR AND AC2 REMAIN
;
; (3) AP1 AND AP2 INSTRUCTIONS
;
; AC0<0-14,15>=>SCT REGISTER IN API
; NOTE: SCT REGISTER BITS 14 AND 15 EFF
; TABLE THEREFORE, SCT SHOULD B
; LOADED PRIOR TO EXECUTING RT1,
; OR STW,
; AC0<4-15>=>R0=>ZREG=>YREG=>XREG
; NOTE: THE PREVIOUS CONTENTS OF REGIST
; SIDE OF THE ARROW ARE WRITTEN I
; ON THE RIGHT SIDE OF THE ARROW.
;
; 10010 -MAIN
;
; 11C.
;
; LRO
; LRX
; RFX
; RRO
; RMO
; RND01
; ROR01
; RND01
; ROR01
; ADD01
; SUB01
; INCO
; DECO
; ADI05
; SBD07
; DBL0
; DBI0
; LDW
; STW
; LSOT
; LOZYX
;
; =101470
; =125470
; =131470
; =101570
; =105570
; =111570
; =115570
; =121570
; =125570
; =131570
; =135570
; =141570
; =145570
; =151570
; =155570
; =161570
; =165570
; =171570
; =101670
; =105670
; =175570
; =135670

```



```

10013 .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

0070D 110
OCTAL DEBUG TOOL (ODT)

:110.
THE DIAGNOSTIC IS EQUIPPED WITH A BUILT IN ODT WHICH CAN BE
ACCESS BY HITTING CONTROL 0 (C) AT ANY TIME DURING THE
EXECUTION OF THE PROGRAM (AFTER SETTING THE PARAME-
METERS).
ON ENTERING ODT THE ADDRESS OF THE LOCATION HAVING THE
NEXT INSTRUCTION TO BE EXECUTED WILL BE TYPED-OUT.

:110.1
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.

:110.2
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

:110.3
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.

:110.3.1
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
FOR ACCUMULATORS 0-3
THE EVENT OF A "R" COMMAND.
CPU AND I/O STATUS
BIT INTERPRETATION
15 STATUS OF I/O DONE FLAG
14 STATUS OF INTERRUPTS (ION FLAG)
13 STATUS OF CARRY BIT
6 ADDRESS OF THE LOCATION HAVING THE BREAK POINT (
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 POINTE
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 CLSE THE OPEN CELL
WITH OR WITHOUT MODIFICATION.

10014 .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

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

:110.3.2
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 "R" 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.

:110.3.3
OTHER ODT COMMANDS
THIS KEY IS USED TO DELETE ERRONEOUSLY TYPED
DIGITS. EACH TIME THE KEY IS PRESSED THE RIGHT M
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
CONTENTS. THIS ALLOWS THE MODIFICATION OF THE CE
AS IF ITS CONTENTS WERE TYPED IN JUST BEFORE THE
KEY WAS PRESSED.
"ADR" 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 "A".
"ADR" 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
ORIGINAL PROGRAM AREA. IF A BREAK POINT IS
PLACED OUTSIDE THIS AREA THE RESULTS WILL
BE UNPREDICTABLE.

```



```

10015 .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

M?DWD 11E

;11E.0 MEMORY DUMP UTILITY
;
;11E.0.1 THIS UTILITY AFFORDS THE USER THE CAPABILITY
; OF DISPLAYING IN OCTAL FORMAT; THE CONTENTS
; OF CONTIGUOUS MEMORY LOCATIONS OF VARIABLE BLOCK
; LENGTHS.
;
;11E.0.2 BLOCKS OF CONTIGUOUS MEMORY THAT ARE IDENTICAL
; AND GREATER THAN 64 ENTRIES IN LENGTH ARE OUT-
; PUT IN AN ABBREVIATED FORMAT. (SEE PAR. 11E.3.2).
; THIS FEATURE WILL CONSERVE HARD COPY AND EXECUTION
; TIME.
;
;11E.0.3 THE USER MAY ALSO SPECIFY A SEARCH WORD. THE
; TOTAL NUMBER OF ENTRIES FOUND; MATCHING THIS
; WORD; WILL BE DISPLAYED AT THE END OF THE PRINTOUT
; AND WILL ALSO BE SAVED IN LOCATION #D370.
;
;11E.0.4 THIS PROGRAM MAY BE MANUALLY STARTED AT LOCATION
; "MDWPP" SYMBOLIC (I.E. THE FIRST ADDRESS OF
; THE UTILITY); OR DYNAMICALLY CALLED BY A USER
; PROGRAM VIA THE D?MP MACRO.
;
;11E.1 DIALOGUE
;11E.1.1 PROGRAM DIALOGUE TERMINATED BY A "2" REQUIRES A
; USER RESPONSE BEFORE PROGRAM EXECUTION CAN CON-
; TINUE. IN THE FOLLOWING DIALOGUE USER RESPONSE IS
; INDICATED BY " ":
;
; FST ADR? "AAAAAA"
; LST ADR? "BBBBBB"
; WD? "CCCCC"
;
; WHERE:
; "AAAAAA" IS ANY OCTAL NUMBER IN THE RANGE OF 00000
; THRU 077776; AND EQUAL TO, OR LESS THAN "BBBBBB".
;
; "BBBBBB" IS ANY OCTAL NUMBER IN THE RANGE OF 00001
; THRU 077777; AND EQUAL TO, OR GREATER THAN "AAAAAA".
;
; "CCCCC" IS ANY OCTAL NUMBER IN THE RANGE OF 00000
; THRU 177777.
;
;11E.1.2 A RESPONSE OF "0", "CR", "LF", "TAB", OR "SPACE"
; TO ANY REQUEST WILL BE INTERPRETED AS A "0" RE-
; SPONSE.
;
; .EJEC

```

```
00016 .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

;11E.2 ERRORS
;11E.2.1 AN ILLEGAL RESPONSE TO A REQUEST, (I.E. A NON-
; OCTAL CHARACTER), WILL RESULT IN A REPEAT OF THAT
; REQUEST.
;11E.2.2 A RANGE ERROR RESPONSE, (I.E. FIRST ADDRESS
; GREATER THAN LAST ADDRESS), WILL RESULT IN THE RE-
; START OF THE PROGRAM IF ENTERED MANUALLY; OR A
; RETURN TO PC +3 IF ENTERED DYNAMICALLY.
;
;11E.3 TYPICAL PROGRAM RESPONSE
;
;ADR:> 0 1 2 3 4 5 6 7
;0 NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN
;10 NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN
;SAME NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN
;100 NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN
;110 NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN
;WD'S FOUNDE= MM-MM
;
;11E.3.1 WHERE:
; THE "NN-NN" ENTRIES ABOVE CORRESPOND TO THE CON-
; TENTS OF THE ASSOCIATED ADDRESSES.
;
; THE "MM-MM" ENTRY ABOVE REPRESENTS THE TOTAL NUMBER OF
; WORDS (OCTAL) FOUND MATCHING THE SEARCH WORD.
;
;11E.3.2 IN THE EXAMPLE ABOVE IT IS ASSUMED THAT THE CON
; OF LOCATIONS 10 THRU 107 INCLUSIVE ARE IDENTICAL. THERE-
; FORE THE ABBREVIATED OUTPUT. (I.E. LOCATIONS 20 THRU 107
; INCLUSIVE ARE REPLACED BY THE TEXT MESSAGE "SAME".
;
; NOTE:
; FOR MANUAL MODE OF OPERATION SWITCH "2" IN LOCATION
; "SWREG" MUST BE "0" OR THE PROGRAM WILL HANG IN A
; LOOP.

```

10017 .MAIN

01 ? 12. SPECIAL NOTES/FEATURES

02 ?  
03 ? 1. AGAIN, IT IS NOTED THAT NO CHECKING OF THE MICRO-  
04 ? CODE/CONTROL STORE USED FOR EXECUTED THE USER  
05 ? AP INSTRUCTIONS(SIS) IS DONE IN THIS PROGRAM.

06 ?  
07 ? 2. THE CAT/KITTEN DATA CHANNEL EXERCISERS MAY BE RUN  
08 ? WITH THIS PROGRAM VIA A CLOAD COMMAND TO EITHER A  
09 ? DTOS TAPE OR A DDOS DISK. IF THE CAT PROGRAM  
10 ? HAS BEEN LOADED MANUALLY VIA A PAPER TAPE, A START  
11 ? MAY BE OBTAINED VIA LOADING LOC 62 TC A -1 VIA THE  
12 ? ODT PROGRAM.

13 ? 13. PROGRAM RUNTIME  
14 ? RUNTIME FOR ALL PASSES IS APPROXIMATELY 30 SEC  
15 ?

10018 .MAIN

\*\*00000 TOTAL ERRORS, 00000 PASS 1 ERRORS

0019 .MAIN

MFDMO 001526 MC 15701  
DFDOD 000454 MC 13701