

**DataGeneral**

---

---

**TECHNICAL  
STATEMENT**

---

---

TEXT LISTING

068-000239-05

PROGRAM

EXERCISER FOR ECLIPSE  
PART 1

TEXT TAPE

097-000239-05

ABSTRACT

'ECLIPSF10' IS AN EXERCISER PROGRAM USED TO TEST THE RELIABILITY OF THE CENTRAL PROCESSOR INSTRUCTIONS OF THE ECLIPSE COMPUTER. 'ECLIPSF10' EXERCISES THE EXTENDED INSTRUCTIONS OF THE ECLIPSE EXTENSIVELY AND ASSURES OF ITS RELIABLE OPERATION.



0003 FCL10

```

01 *M?FSS: THIS MACRO IS USED TO BUILD A CALL TO THE MESS?S ROUTINE OF THE
02 * TTYTO PACKAGE. IT REQUIRES ONE ARGUMENT, THE LOCATION
03 * OF THE MESSAGE.
04
05 *?T?TTO: THIS MACRO IS USED TO BUILD A CHECK FOR TTI INPUT AND TO
06 * TRANSFER CONTROL TO THE APPROPRIATE ROUTINE IF A TTI KEY
07 * HAS BEEN HIT.
08
09 *?T?TII: THIS MACRO IS USED TO BUILD A CHECK FOR TTI DONE WITH ODTPK
10 * PRESENT. ON FINDING AN INPUT KEY HAVING BEEN STRUCK IT WILL
11 * CALL SWPAK AND THEN ODTPK IF A CONTROL 0 WAS INPUTED.
12
13 *?T?TTP: THIS MACRO IS USED TO BUILD A WAIT FOR TTI INPUT AND TO
14 * TRANSFER CONTROL TO THE APPROPRIATE ROUTINE IF A TTI KEY
15 * HAS BEEN HIT.
16
17 *?T?TTS: THIS MACRO IS USED TO BUILD A WAIT FOR TTI DONE WITH ODTPK
18 * PRESENT. ON FINDING AN INPUT KEY HAVING BEEN STRUCK IT WILL
19 * CALL SWPAK AND THEN ODTPK IF A CONTROL 0 WAS INPUTED.
20
21 *R?AN: THIS MACRO IS USED TO CALL THE RANDOM NUMBER GENERATOR.
22
23 *R?ANA: THIS MACRO IS USED TO GENERATE A RANDOM ADDRESS BY CALLING THE
24 * RAN? AND RAN?O ROUTINES. THE USER MUST DEFINE TWO PARAMETERS
25 * IN THE MACRO CALL. THEY ARE:
26
27 1) THE ADDRESS CONTAINING THE LOWER
28 * ADDRESS LIMIT.
29 2) THE ADDRESS CONTAINING THE UPPER
30 * ADDRESS LIMIT.
31
32 *R?AND: THIS MACRO IS USED TO GENERATE A RANDOM DATA WORD BY CALLING
33 * THE RAN? AND RAN?O ROUTINES. THE USER MUST DEFINE TWO PARAMTE
34 * IN THE MACRO CALL. THEY ARE:
35
36 1) THE LOWER DATA LIMIT.
37 2) THE UPPER DATA LIMIT.
38
39 *D?VCC: THIS MACRO IS USED TO CALL THE DVC?H ROUTINE. THE USER MUST
40 * DEFINE FOUR PARAMETERS IN THE MACRO CALL. THEY ARE:
41
42 1) THE DEVICE CODE TO BE CHANGED.
43 2) THE ADDRESS OF THE NEW DEVICE CODE
44 3) THE STARTING LOCATION OF THE SCAN
45 4) THE ENDING LOCATION OF THE SCAN
46
47 *D?VCD: THIS MACRO IS USED TO BUILD A DEVICE CODE CHANGE ROUTINE
48
49 *D?VCF: THIS MACRO IS USED TO BUILD A DEVICE CODE CHANGE ROUTINE
50 * FOR ECLIPSE TYPE CPU'S.
51
52 *SETUP: THIS MACRO IS USED TO CALL EITHER S?TUP, M?NST, OR S?EUP,
53 * DEPENDING ON THE VALUE OF COM?P. IF COM?P=0 IT WILL CALL S?TUP
54 * COM?P=1 IT WILL CALL M?NST. IF COM?P=2 IT WILL CALL S?EUP.
55 * NOTE! COM?P MUST BE DEFINED BEFORE SETUP
56 * IS CALLED. THIS MACRO REQUIRES ONE USER DEFINED PARAMETER WHIC
57 * IS THE ITERATION COUNT FOR THE SURTEST.
58
59 *FROR: THIS MACRO IS USED TO CALL EITHER F?ROR, M?NER, OR F?ERR,
60 * DEPENDING ON THE VALUE OF COM?P. IF COM?P=0 IT WILL CALL F?OR

```

0004 FCL10

```

01 *M?P=1 IT WILL CALL M?NER. IF COM?P=4 IT WILL CALL E?ERR.
02 * NOTE: COM?P MUST BE DEFINED BEFORE ERROR IS CALLED.
03
04 *!00P: THIS MACRO IS USED TO CALL L?OOP, OP L?OP AND IS USED IN UPDAT
05 * OLD PROGRAMS. IF COM?P=4 L?OOP. IF COM?P=4 IT WILL CALL L?OP.
06 * WILL CALL L?OP.
07
08 *M?MP: THIS MACRO IS USED TO BUILD A CALL TO THE MEMORY DUMP
09 * UTILITY. IT REQUIRES THREE ARGUMENTS IN THE CALL. THEY ARE
10
11 1) THE SEARCH WORD
12 2) THE STARTING ADDRESS OF THE DUMP
13 3) THE ENDING ADDRESS OF THE DUMP
14
15 *M?DMP: THIS MACRO IS USED TO BUILD A MEMORY DUMP UTILITY. THIS
16 * UTILITY PROVIDES A PROGRAM WITH A MEANS TO DISPLAY THE
17 * CONTENTS OF CONTIGUOUS MEMORY LOCATIONS IN AN OCTAL FORMAT.
18
19 *M?NPTS: THIS MACRO IS USED BY THE N?EWT MACRO
20 * TO START A SURTEST. THE MACRO WILL
21 * NUMBER THE SURTEST, GENERATE A SURTEST
22 * HEADING AND OPTIONALLY CALL THE S?UENCE
23 * CHECKING ROUTINE AND THE SETUP MACRO.
24 * THE MACRO HAS 1? POSSIBLE ARGUMENTS IN THE
25 * CALL. THEY ARE DEFINED AS FOLLOWS:
26
27 1) THE ITERATION COUNT FOR THE SURTEST
28 * (IF THIS IS 0 THE CALL TO THE SETUP MACRO
29 * IS DELETED)
30
31 2) THIS OPTIONAL ARGUMENT IS
32 * A COMMENT TO BE INSERTED IN THE SURTEST
33 * HEADER (ALL EMBEDDED SPACES MUST BE PRECEDED
34 * BY 3 BACK ARROW CHARACTERS (^))
35
36 3) THIS OPTIONAL ARGUMENT IF
37 * DEFINED AS AN "S" WILL DEIFTE THE CALL
38 * TO THE SEQUENCE CHECKING ROUTINE
39 * (THIS SAVES TWO LOCATIONS PER SURTEST)
40
41 4-11) MUST BE THE DIGITS 0-8 RESPECTIVELY
42
43 1?) THIS IS AN OPTIONAL ARGUMENT WHICH IF DEFINED
44 * AS A "F" WILL DELETE THE _FJECT FROM THE MACRO.
45
46 *N?EWT: THIS MACRO IS USED TO CALL THE N?NPTS MACRO; WHICH
47 * IS USED TO START A SURTEST. IT SETS UP PARAMETERS
48 * FOR THE N?NPTS MACRO AND HAS FOUR POSSIBLE
49 * ARGUMENTS. THEY ARE:
50
51 1) THE ITERATION COUNT FOR THE SUBTEST
52 * (IF THIS IS 0 THE CALL TO THE SETUP MACRO
53 * IS DELETED)
54
55 2) THIS OPTIONAL ARGUMENT WHICH IS A
56 * COMMENT TO BE INSERTED IN THE SUB TEST
57 * HEADING (ALL EMBEDDED SPACES MUST BE PRECEDED
58 * BY THREE BACK ARROW (^) CHARACTERS)
59
60 3) THIS OPTIONAL ARGUMENT

```

0005 FCL10

```

01      IF DEFINED AS A "S WILL DELETE
02      THE CALL TO THE SEQUENCE CHECKING
03      ROUTINE (THIS SAVES TWO LOCATIONS PER SUBTEST)
04
05      4) THIS IS AN OPTIONAL ARGUMENT WHICH IF DEFINED
06      AS A "E WILL DELETE THE .FJEC FROM THE MACRO.
07
08      THIS MACRO BUILDS A SEQUENCE CHECKING
09      ROUTINE
10
11      PROVIDES THE PRE DIAGNOSTIC COMMUNICATIONS WITH
12      DTOS. SIZES MEMORY, MAP TYPE AND CPU TYPE.
13
14      CATP: USED BY P2REM. SETS START OF CAT/KITTEN SUCH THAT TTI
15      IS NOT MASKED OUT.
16
17      CATPF: USED BY P2REM. SETS START OF CAT/KITTEN SUCH THAT TTI
18      IS MASKED OUT.
19
20      P2STM: PROVIDES POST DIAGNOSTIC OPTIONAL FUNCTIONS. I.E. PRINT
21      "PASS", RETURN TO DTOS, RESTART PROGRAM.
22
23      CAT: USED BY P2STM. CHECKS IF CAT/KITTEN WAS LOADED, AND
24      IF YES, STARTS CAT/KITTEN.
25
26      P2PAS: USED BY P2STM. PRINTS THE END OF DIAGNOSTIC
27      MESSAGE "PASS".
28
29      P2MFR: PROVIDES A TIMING ROUTINE, A CALTRRATE ROUTINE, AND
30      A DELAY ROUTINE.
31
32      P2LAY: PROVIDES A CALL TO THE DELAY ROUTINE IN THE T2MER
33      MACRO.
34
35      P2MTT: PROVIDES A CALL TO THE TIMING ROUTINE IN THE T2MER
36      MACRO.
37
38      NOTE: FOR FURTHER INFORMATION ON HOW TO USE THESE
39      MACRO'S SEE THE COMMENTS INCLUDED IN EACH MACRO.

```

10006 FCL10

```

01      .MACRO S?MPD
02
03      SWITCH SETTINGS
04
05      LOCATION "SWREG" IS USED TO SELECT THE PROGRAM OPTIONS
06      (NOT SYSTEM CONFIGURATION). WHILE RUNNING UNDER DTOS,
07      THIS LOCATION WILL BE LOADED BY THE MONITOR.
08      HOWEVER UNDER STAND ALONE AND PROGRAM LOAD MODES THIS
09      LOCATION WILL BE SET ACCORDING TO THE ANSWERS SUPPLIED
10      BY THE OPERATOR. IN ANY CASE THE OPTIONS CAN BE CHANGED
11      OR VERIFIED BY USING ONE OF THE COMMANDS GIVEN IN SEC.
12      1.2
13
14      SWITCH OPTIONS
15      DIFFERENT RITS AND THEIR INTERPRETATION AT LOCATION
16      "SWREG" IS AS FOLLOWS:
17
18      RIT      OCTAL      BINARY      INTERPRETATION
19      VALUE     VALUE
20
21      1          40000     1          LOOP ON ERROR
22
23      2          20000     0          PRINT TO CONSOLE
24
25      3          10000     0          ABORT PRINT OUT TO CONSOLE
26
27      4          04000     1          DO NOT PRINT "%X FAILURE
28
29      5          02000     0          PRINT "%X FAILURE
30
31      6          01000     0          ALLOW END OF PASS PRINT OUT
32
33      7          00400     1          SUPPRESS END OF PASS PRINT OUT
34
35      8          00200     0          DO NOT PRINT ON THE LINE PRINTER
36
37
38      9          00200     1          PRINT ON THE LINE PRINTER
39
40
41
42
43
44
45
46
47

```

10007 FCL10

01  
02 SWITCH COMMANDS  
03 ONCE THE PROGRAM STARTS EXECUTING THE STATE OF ANY OF  
04 THE BITS CAN BE CHANGED BY HITTING KEYS 1-9, 4-F. THE  
05 PROGRAM WILL CONTINUE RUNNING AFTER UPDATING THE OPTIONS.  
06 EACH KEY WILL COMPLETE THE STATE OF THE BIT AFFILIAT-  
07 ED WITH IT, THUS BIT 4 CAN BE ALTERED BY HITTING KEY 4.  
08 SHIFTING OF ANY BIT OF LOCATION "SWREG" WILL SET BIT 0.  
09 (DEFAULT MODE IS DEFINED AS ALL BITS OF SWREG SET TO 0)  
10 THE PROGRAM CAN BE LOCKED INTO SWITCH MODIFICATION MODE  
11 BY TYPING A 0. IN WHICH CASE MORE THAN ONE RTI CAN BE  
12 CHANGED BEFORE CONTROL IS ALLOWED TO RETURN TO THE  
13 MAIN PROGRAM.  
14  
15 1-1.2 OTHER COMMANDS  
16  
17 "CR" A "RETURN" CAN BE TYPED TO CONTINUE THE PROGRAM  
18 AFTER ITS LOCKED IN A SWITCH MODIFICATION MODE  
19  
20 "END" THIS COMMAND GIVEN AT ANY TIME WILL RESET "SWREG"  
21 TO DEFAULT MODE AND RESTART THE PROGRAM.  
22  
23 "R" THIS COMMAND GIVEN AT ANY TIME WILL RESTART THE  
24 PROGRAM. SWITCHES ARE LEFT WITH THE VALUES THEY  
25 HAD BEFORE THE COMMAND WAS ISSUED.  
26  
27 "R0" THIS COMMAND GIVEN AT ANY TIME WILL CAUSE THE  
28 PROGRAM CONTROL TO GO TO ODT (NOTE: THIS IS AN  
29 OPTIONAL COMMAND AND IS AVAILBLE ONLY IF  
30 ODTPK IS PRESENT)  
31  
32 "M" THIS COMMAND GIVEN AT ANY TIME WILL PRINT THE  
33 CURRENT OPERATING MODES.  
34  
35  
36  
37

10008 FCL10

01  
02 -MACRO 0?D?D  
03  
04 OCTAL DERUG TOOL (ODT)  
05  
06  
07 THE DIAGNOSTIC IS EQUIPPED WITH A BUILT IN ODT WHICH CAN  
08 BE ACCESSED BY HITTING CONTROL 0 (R0) AT ANY TIME DURING  
09 THE EXECUTION OF THE PROGRAM (AFTER SETTING THE PARA-  
10 METERS).  
11 ON ENTERING ODT THE ADDRESS OF THE LOCATION HAVING THE  
12 NEXT INSTRUCTION TO BE EXECUTED WILL BE TYPED-OUT.  
13  
14 CONVENTIONS AND SYMBOLS  
15 THE FOLLOWING CONVENTIONS ARE USED BY THE ODT:  
16 ? PRESSING ANY ILLEGAL KEY CAUSES THE ODT TO RES-  
17 POND WITH A "?".  
18 @ ODT IS READY AND AT YOUR SERVICE.  
19  
20 COMMAND STRUCTURE  
21 AN ODT COMMAND HAS THE FOLLOWING FORMAT:  
22 [ARGUMENT] [COMMAND]  
23 AN ARGUMENT MAY BE ONE OF THE FOLLOWING:  
24 "EXP" AN OCTAL EXPRESSION CONSISTING OF OCTAL NUMBERS  
25 SEPARATED BY PLUS (+) OR MINUS (-) SIGNS. LEAD-  
26 ING ZEROS NEED NOT BE TYPED.  
27 "ADR" AN ADDRESS IS THE SAME AS AN EXPRESSION EXCEPT  
28 THAT RTI 0 IS NEGLECTED.  
29 A COMMAND IS A SINGLE TELETYPE CHARACTER  
30  
31 ODT COMMANDS  
32 THE LOCATIONS THAT CAN BE EXAMINED AND MODIFIED BY THE  
33 USFR ARE CALLED CELLS. THESE CELLS ARE OF TWO TYPES:  
34 INTERNAL CPU CELLS AND MEMORY LOCATIONS.  
35  
36 1-1.3 OPENING INTERNAL CELLS  
37 THE COMMAND TO OPEN ONE OF THE INTERNAL REGISTERS IS OF  
38 THE FORM "NA" WHERE N IS ANY OCTAL EXPRESSION BETWEEN  
39 0 AND 7  
40 FOR ACCUMULATORS 0-3  
41 FOR PC OF THE NEXT INSTRUCTION TO BE EXECUTED IN  
42 THE EVENT OF A "P" COMMAND.  
43 CPU AND I/O STATUS  
44 BIT INTERPRETATION  
45 STATUS OF I/O DONE FLAG  
46 STATUS OF INTERRUPTS (I/O FLAG)  
47 STATUS OF CARRY BIT  
48 ADDRESS OF THE LOCATION HAVING THE BREAK POINT (IF  
49 ANY)  
50 INSTRUCTION AT THE BREAK POINT LOCATION  
51  
52 OTHER COMMANDS TO OPEN CELLS ARE:  
53 "ADR"/ OPEN THE CELL AND PRINT ITS CONTENTS  
54 -/ OPEN THE CELL CURRENTLY POINTED TO BY THE POINTER  
55 AND PRINT ITS CONTENTS.  
56 -+ "ADR"/ ADD "ADR" TO THE POINTER, OPEN THE CELL  
57 AND PRINT ITS CONTENTS.  
58 -"ADR"/ SUBTRACT "ADR" FROM THE POINTER, OPEN  
59 THE CELL AND PRINT ITS CONTENTS.  
60 "CR" THE RETURN KEY IS USED TO CLOSE THE OPEN CELL

01 WITH OR WITHOUT MODIFICATION.  
02 LINE FEED IS USED TO CLOSE THE OPEN CELL WITH OR  
03 WITHOUT MODIFICATION AND TO OPEN THE SUCCEEDING  
04 CELL.  
05 CLOSE THE OPEN CELL WITH OR WITHOUT MODIFICATION  
06 AND OPEN THE PRECEDING CELL  
07 CLOSE THE OPEN CELL WITHOUT MODIFICATION, AND  
08 OPEN THE CELL POINTED TO BY ITS CONTENTS.  
09 \*"ADR"/ CLOSE THE OPEN CELL WITHOUT MODIFICATION, AND  
10 OPEN THE CELL POINTED TO BY ITS CONTENTS + "ADR".  
11 \*"ADR"/ CLOSE THE OPEN CELL WITHOUT MODIFICATION, AND  
12 OPEN THE CELL POINTED TO BY ITS CONTENTS - "ADR".  
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

1.3.2 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 ". " OR "-/ -OCTAL EXPRESS-  
ION". A RIBROUT 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.

1.3.3 OTHER ODT COMMANDS  
RIBROUT 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 RIBROUT KEYS 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.

10011 FCL10

10012 FCL10

```

01 .MACRO M2DMD
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

```

MEMORY DUMP UTILITY

THIS UTILITY AFFORDS THE USER THE CAPABILITY OF DISPLAYING IN OCTAL FORMAT; THE CONTENTS OF CONTIGUOUS MEMORY LOCATIONS OF VARIABLE BLOCK LENGTHS.

BLOCKS OF CONTIGUOUS MEMORY THAT ARE IDENTICAL AND GREATER THAN 64 ENTRIES IN LENGTH ARE OUTPUT IN AN ABBREVIATED FORMAT. (SFF PAR. 1.3.2). THIS FEATURE WILL CONSERVE HARD COPY AND EXECUTION TIME.

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 WD390.

THIS PROGRAM MAY BE MANUALLY STARTED AT LOCATION "MDM?P" SYMBOLIC (I.E. THE FIRST ADDRESS OF THE UTILITY); OR DYNAMICALLY CALLED BY A USER PROGRAM VIA THE D?MP MACRO.

DIALOGUE

PROGRAM DIALOGUE TERMINATED BY A "?" REQUIRES A USER RESPONSE BEFORE PROGRAM EXECUTION CAN CONTINUE. IN THE FOLLOWING DIALOGUE USER RESPONSE IS INDICATED BY " ":

FST ADR? "AAAAA"  
LST ADR? "BBBBBB"  
WD? "CCCCC"

WHERE:  
"AAAAA" IS ANY OCTAL NUMBER IN THE RANGE OF 000000 THRU 077776; AND EQUAL TO, OR LESS THAN "BBBBBB".  
"BBBBBB" IS ANY OCTAL NUMBER IN THE RANGE OF 000001 THRU 077777; AND EQUAL TO, OR GREATER THAN "AAAAAA".  
"CCCCC" IS ANY OCTAL NUMBER IN THE RANGE OF 000000 THRU 17777.

A RESPONSE OF "0", "CR", "LF", "TAB", OR "SPACE" TO ANY REQUEST WILL BE INTERPRETED AS A "0" RESPONSE.

.FJEC

```

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

```

ERRORS

1.2.1 AN ILLEGAL RESPONSE TO A REQUEST, (I.E. A NON-OCTAL CHARACTER), WILL RESULT IN A REPEAT OF THAT REQUEST.

1.2.2 A RANGE ERROR RESPONSE, (I.F. FIRST ADDRESS GREATER THAN LAST ADDRESS), WILL RESULT IN THE RESTART OF THE PROGRAM IF ENTERED MANUALLY; OR A RETURN TO PC +3 IF ENTERED DYNAMICALLY.

TYPICAL PROGRAM RESPONSE

```

0 1 2 3 4 5 6 7
NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN
NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN
NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN
NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN NN-NN
MM-MM

```

1.3.1 WHERE:  
THE "NN-NN" ENTRIES ABOVE CORRESPOND TO THE CONTENTS OF THE ASSOCIATED ADDRESSES.

THE "MM-MM" ENTRY ABOVE REPRESENTS THE TOTAL NUMBER OF WORDS (OCTAL) FOUND MATCHING THE SEARCH WORD.

1.3.2 IN THE EXAMPLE ABOVE IT IS ASSUMED THAT THE CONTENTS OF LOCATIONS 10 THRU 107 INCLUSIVE ARE IDENTICAL. THEREFORE THE ABBREVIATED OUTPUT. (I.F. 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.

.MACRO P?PFD  
PRE MACRO  
THIS MACRO PROVIDES ALL THE NECESSARY PRE-DIAGNOSTIC PROGRAM COMMUNICATIONS WITH DTOS. IT SIZES MEMORY, MONITORS THE EGGS BLOCK, AND PROVIDES TO THE USER THE FOLLOWING:

1) THE LOGICAL TOP OF MEMORY  
2) THE TOPMOST MEMORY LOCATION AVAILABLE FOR THE DIAGNOSTIC PROGRAM  
3) THE HIGHEST 1K BLOCK OF MEM (IF MAPPED)

```

0013 FCL10
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

0014 FCL10
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

41) A STATUS WORD CONTAINING THE STATE OF
    THE CATSWITCH, THE MODE OF OPERATION,
    AND THE TEST PROGRAM PASS COUNT

51) A STATUS WORD CONTAINING: CPU TYPE,
    MAP STATUS, (MAP-NO MAP), AND THE
    MAP TYPE, IF APPLICABLE.

6) A POINTER TO THE I/O MODULE

7) A POINTER TO THE CAT/KITTEN PROGRAM
    IF DESIRED.

*1.1 THE USER MUST ALLOCATE PAGE ZERO STORAGE FOR
    THESE PARAMETERS, AND DEFINE THEM IN THE MACRO
    CALL.

*1.2 CALLING SEQUENCE
    C?ATF
    P?REM MTOP, I?TOP, STAT1, STAT2, BEGIN, C?ATF, 5
    BLANK

*1.2.1 ARGUMENTS
    ARG 1 IS THE LOCATION WHERE THE PHYSI-
    CAL MEMORY SIZE WILL BE STORED.
    (OR THE HIGHEST 1K BLOCK IF MAPPED)
    ARG 2 IS THE LOCATION WHERE THE TOP-
    MOST MEMORY LOCATION FOR THE
    DIAGNOSTIC PROGRAM WILL BE
    STORED.
    ARG 3 IS THE LOCATION WHERE THE FIRST
    STATUS WORD WILL BE STORED.
    ARG 4 IS THE LOCATION WHERE THE SECOND
    STATUS WORD WILL BE STORED
    ARG 5 IS THE STARTING ADDRESS OF THE
    PRE?M MACRO. (SEE NOTE).
    ARG 6
    C?ATF
    C?ATP
    (OPTIONAL) THESE CALLS MUST BE
    USED IF THE USER WANTS THE
    CAT/KITTEN PROGRAM TO RUN.
    C?ATF MASKS OUT THE I?I, I?O,
    I?P, AND MTA INTERRUPTS.
    C?ATP MASKS OUT ALL OF THE
    ABOVE, EXCEPT THE I?I.
    ARG 7 CPU NUMBER TO INHIBIT EXPANSION
    OF UNNECESSARY CODE WHEN THE CPU
    TYPE IS PRE-DEFINED. BY DEFAULT, (NO
    ARGUMENT PASSED), ALL CODE IS
    EXPANDED. THE NUMBERS ARE
    AS FOLLOWS:
    0 = 1200
    1 = 800
    2 = NOVA/2
    3 = NOVA/3
    4 = MICRONOVA
    5 = ECLIPSE
    6 = RESERVED

*1.2.2 EXAMPLE OF A MACRO CALL FOR ECLIPSE
    P?REM MTOP, I?TOP, STAT1, STAT2, BEGIN, C?ATF, 5

* NOTE: IF THE USER INTENDS TO USE THE PAGE ZERO DEFINITION
    MACROS P?G00 OR M?P00, AND HE ALSO INTENDS TO START
    HIS DIAGNOSTIC PROGRAM BY CALLING THE PRE?M MACRO,
    THEN HE SHOULD PASS THE LABEL "BEGIN" AS THE FIFTH
    ARGUMENT OF THE CALL.
    IF THE USER INTENDS TO USE THE P?G00 MACRO AND
    TO START HIS PROGRAM BY CALLING THE PRE?M MACRO,
    HE SHOULD PASS THE SAME STARTING ADDRESS TO THE
    PRE?M MACRO AS TO THE P?G00 MACRO.
    IF THE USER DOES NOT INTEND TO USE ANY OF THE PAGE
    ZERO DEFINITIONS MACROS, HE CAN PASS ANY STARTING
    ADDRESS.

* NOTE: IF SIZING A SYSTEM WITH AN "MMPUI". THE BANK JUMPERS
    MUST BE PROPERLY INSERTED OR INCORRECT SIZING WILL
    RESULT.

*1.3 SIZING
    SIZES LOGICAL AND PHYSICAL MEMORY
    CHECKS IF CAT/KITTEN PROGRAM WAS LOADED
    IF CAT/KITTEN WAS LOADED IT PROVIDES ROOM AT MEMORY
    TOP FOR MINI-MONITOR, I/O MODULE, CAT/KITTEN PROGRAM
    AND THE BUFFERS FOR THE DISK/IORD OPERATIONS
    IF CAT/KITTEN WAS NOT LOADED IT PROVIDES ROOM ONLY
    FOR MINI-MONITOR AND I/O MODULE
    CHECKS FOR THE PRESENCE OF A MAP AND THE MAP TYPE
    WHERE APPLICABLE. PROVIDES THE HIGHEST 1K PHYSICAL
    BLOCK OF MEMORY (IN OCTAL)

*1.4 THE MACRO ALSO SETS TWO STATUS WORDS WHICH CONTAIN
    THE FOLLOWING:
    (STATUS WORD 1)
    BIT NUMBER SIGNIFICANCE
    0 - 7 SPARE
    RESERVED FOR I/O MODULE OP
    CODES (NOT IMPLEMENTED)
    8 - 10
    0 = NO CAT/KITTEN
    1 = CAT/KITTEN PROG HAS BEEN LOADED
    12
    0 = MANUAL OR DERIG MODE
    1 = AUTO OR SEMI-AUTO MODE
    13 - 15 # OF PASSES OF PROG TO RUN
    (STATUS WORD 2)
    BIT NUMBER SIGNIFICANCE
    0 0 = UNMAPPED
    1 1 = MAPPED

```



0015 FCL10

```

01 0 = MPMU (FCLIPSF ONLY)
02 1 = MPMU1 (ECLIPSF ONLY)
03 SPARE
04
05 2 = 12
06 13 = 15
07 1 = RESERVED
08 2 = RESERVED
09 3 = NOVA/3
10 4 = MICRONOVA
11 5 = ECLIPSE
12 6 = RESERVED
13 7 = RESERVED
14
15
16 THE MACRO ALSO PROVIDES A POINTER TO THE
17 T/O MODULE. THE CALLING SEQUENCE IS:
18
19 JSR @IOM?00
20 Op CODE
21
22 THE MACRO ALSO PROVIDES A POINTER TO THE START
23 OF THE CAT/KITTEN PROGRAM (IF THE USER PASSES
24 THE SIXTH ARGUMENT OF THE MACRO CALL)
25 THE CALLING SEQUENCE IS:
26 JSR @ICAT?
27
28 PAGE ZERO DEFINITION REQUIREMENTS
29
30 1) THE FIRST FOUR ARGUMENTS USED IN THE MACRO
31 CALL MUST BE DEFINED IN PAGE ZERO
32
33 2) THE POINTER TO THE I/O MODULE MUST BE
34 DEFINED IN PAGE ZERO AS FOLLOWS:
35 IOM?00: 0
36
37 3) IF THE USER WANTS THE CAT/KITTEN PROGRAM TO RUN
38 HE MUST DEFINE IN PAGE ZERO THE FOLLOWING:
39 ICA??: 0
40
41 IF THE USER HAS NOT USED EITHER THE DEFAULT
42 PAGE ZERO DEFINITIONS MACRO (P?G00) OR THE
43 USER DEFINED PAGE ZERO DEFINITIONS MACRO
44 (P?G001) HE MUST DEFINE A POINTER TO THE EGGS
45 BLOCK AS FOLLOWS:
46 IEGG??: F666
47
48 IF THE USER DOES NOT INTEND TO USE ANY OF THE
49 PAGE ZERO DEFINITIONS MACROS, HE MUST DEFINE
50 THE FOLLOWING PAGE ZERO COUNTER:
51 PAS??: 0
52
53 .MACRO P?ST0
54 P?STM
55
56 THIS MACRO IS USED TO HANDLE ALL POST-DIAGNOSTIC
57 PROGRAM COMMUNICATIONS WITH DTOS
58 IT'S PURPOSE IS TO MONITOR THE EGGS BLOCK AND THE DTOS
59 PASS COUNTERS (PA?SIN AND PAS?S)
60

```

0016 FCL10

```

01 IT WILL DECIDE WHETHER TO RETURN TO THE START OF THE
02 DIAGNOSTIC PROGRAM, JUMP TO A USER DEFINED PRINT MACRO
03 (OR PRINT ROUTINE) OR TO START THE CAT/KITTEN PROGRAM
04
05 THE USER MAY DEFINE FOUR ARGUMENTS IN THE MACRO
06 CALL. THE FIRST ARGUMENT MUST BE DEFINED, THE
07 NEXT TWO ARE OPTIONAL. THESE ARE:
08 ARG 1 THE "START OF A NEW PASS" ADDRESS
09 ARG 2 (OPTIONAL) THE USER CAN EITHER PASS THE MACRO
10 CALL "P?PAS" AS THE SECOND ARGUMENT, IN WHICH
11 CASE, THE FOLLOWING MESSAGE WILL BE PRINTED:
12 " PASS # "
13 WHERE # IS THE PRESENT PASS COUNT IN DECIMAL
14 OR HE COULD PASS A CALL TO A USER DEFINED
15 PRINT MACRO OR A CALL WHICH EXPANDS TO A JSR
16 TO A PRINT ROUTINE. (IF THE USER DOES NOT
17 INTEND TO USE THE SECOND ARGUMENT (NO PRINT=
18 OUT DEFSTED) HOWEVER HE DOES INTEND TO USE
19 THE THIRD ARGUMENT (L?CAT) HE SHOULD PLACE TWO
20 CONSECUTIVE COMMAS DIRECTLY AFTER ARGUMENT 1).
21 ARG 3 (OPTIONAL) IF THE USER WANTS THE CAT/KITTEN
22 PROGRAM TO RUN HE MUST PASS "L?CAT" AS THE
23 THIRD ARGUMENT OF THE MACRO CALL.
24 ARG 4 (OPTIONAL) THIS ARGUMENT MAY BE USED TO INSERT
25 USER DEFINED CODE.
26 THE CALLING SEQUENCE IS:
27 P?STM ARG 1,ARG 2,L?CAT,ARG 4
28 P?PAS
29 WHERE ARG 1 IS THE STARTING ADDRESS OF A NEW
30 PASS
31 (OPTIONAL) THE USER CAN EITHER
32 PASS THE MACRO CALL "P?PAS" AS
33 SECOND ARGUMENT, IN WHICH CASE
34 THE FOLLOWING MESSAGE WILL BE
35 PRINTED:
36 " PASS # "
37 WHERE # IS THE PRESENT PASS COUNT
38 IN DECIMAL
39 OR THE USER COULD PASS A CALL TO
40 A USER DEFINED PRINT MACRO OR A
41 CALL WHICH EXPANDS TO A JSR TO A
42 USER DEFINED PRINT ROUTINE.
43 (IF THE USER DOES NOT INTEND TO
44 USE THE SECOND ARGUMENT (NO PRINT=
45 OUT DEFSTED) HOWEVER HE DOES INTEND
46 TO USE THE THIRD ARGUMENT (L?CAT)
47 HE SHOULD PLACE TWO CONSECUTIVE
48 COMMAS DIRECTLY AFTER ARGUMENT 1)
49 L?CAT (OPTIONAL) IS THE CALL WHICH THE
50

```



0019 FCL10

```

01 ** S?MPK
02 -TFF D?IAG
03 ** COM?P
04 ** 1,2,3,4,5,6
05 -FNDC -TFF
06 ** M?NDG
07 ** 1,2,3,4,5,6
08 -FNDC
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 **

```

0020 FCL10

```

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 **

```

10021 FCL10

```

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

```

10022 FCL10

```

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

```

0023 FCL10

01 01 :THE CALL  
02 02 :  
03 03 :  
04 04 :  
05 05 :  
06 06 :  
07 07 :  
08 08 :  
09 09 :  
10 10 :  
11 11 :

10024 FCL10

```

01 STA 2,0,3
02 JMP 4,3
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

```

```

:MACRO D?VCE
:SUBROUTINE DVC?H (FOR ECLIPSE CPU'S)
:
: THIS SUBROUTINE CHANGES THE DEVICE CODE OF I/O INSTRUCTIONS
: WITH DEV CODE IN (CALL+1) TO THE DEV CODE DEFINED IN
: LOCATION IN (CALL*P). THE BEGIN AND END OF SCAN ADDRESSES
: ARE DEFINED IN (CALL+3) AND (CALL+4) RESPECTIVELY.
: RETURN IS AT (CALL+5). ON RETURN THE SUBROUTINE WILL HAVE CHANGED
: THE ARGUMENT IN CALL+1 TO THE NEW DEV CODE. THIS ALLOWS
: THE SAME CALL TO THE DEV CODE CHANGE ROUTINE TO BE RE-USED
: TO CHANGE THE I/O INSTRUCTIONS TO SOME OTHER DEV CODE WITHOUT
: HAVING THE USER CHANGE THE CALL.
:
:
: NOTF: THIS ROUTINE WILL CHECK EACH INSTRUCTION BEFORE CHANGING
: IT, AND SKIP THE APPROPRIATE NUMBER OF INSTRUCTIONS IN
: CASE OF MULTIPLE-WORD INSTRUCTIONS.
:
: CALLING SEQUENCE IS:
:
: JSR @DVC?H
: NEW DEV CODE LOC--*1
: START OF SCAN--*2
: END OF SCAN--*3
: RETURN
:
: WHERE DVC?H IS A PAGE ZERO POINTER TO DVC?H

```

```

DVC?H: STA 3,DVC?X ;SAVE EXIT PNTR
LDA 0,0,3 ;C(AC0)= DEVICE CODE TO BE
ADD 1,0 ; CHANGED + 60000
LDA 2,1,3 ;
ADD 3,2 ;
LDA 2,0,2 ;
ADD 2,1 ;
STA 1,DVT?1 ;
LDA 2,3,3 ;
ADD 3,2 ;
STA 2,DVT?3 ;
LDA 2,2,3 ;
ADD 2,3 ;
STA 0,DCC?0 ;SV AC0
STA 3,DCC?3 ;SV AC3
JSR DCH?T ;GET TABLE ADDR
STA 3,DTA?8 ;SAVE TABLE STRT ADDR
LDA 3,DCC?3 ;RESTORE AC3
ONE?H: LDA 2,DTA?8 ;GET TABLE START
DCC?H: LDA 0,DVT?3 ;GET END PNTR
;SKP IF AC0 => AC3
JMP DXT?T ;ALL DONE
LDA 0,0,2 ;GET # TO INC RY
STA 0,DTN?C ;SAVE IT
DCH?H: LDA 1,0,3 ;AC1 HAS INSTRUCTION
INC 2,2 ;NEXT TABLE LOC.
LDA 0,0,2 ;AC0 HAS MASK

```



0027 FCL10

```
01 177777 ;
02
03 103777 ;MASK
04 103770 ;TORT=XORI-ANDI-ADDI
05 177777 ;
06
07 166377 ;MASK
08 102070 ;EJMP=ETISZ
09 106070 ;EJSR=EDSZ
10 177777 ;
11
12 176377 ;MASK
13 102370 ;PSHJ
14 177777 ;
15
16 162377 ;MASK
17 142170 ;DSPA
18 177777 ;
19
20 103477 ;MASK
21 102050 ;FLDS=FLDD-FSTS-FSTD
22 177777 ;
23
24 103777 ;MASK
25 102550 ;FLMD
26 102750 ;FFMD
27 177777 ;
28
29 103077 ;MASK
30 101050 ;FAMS=FAMD-FSMS-FSMD-FMMS-FMMD-FDMS-FDMD
31 177777 ;
32
33 143777 ;MASK
34 103350 ;FSST=FLST
35 177777 ;
36
37 140377 ;MASK
38 100170 ;ELDR=ESTB
39
40 0 ;END OF 2-WORD INSTRUCTIONS
41 0 ;*****END OF TABLE*****
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
```

MACRO SETUP  
-TFE COM?P  
S?TUP  
-ENDC  
-TFN (COM?P&1)  
M?NST  
-ENDC  
-TFN (COM?P&4)  
S?EUP  
-FENDC

MACRO LOOP  
-TFE (COM?P&4)  
L?OOP  
-ENDC  
-TFN (COM?P&4)