POINT 4 COMPUTER
DIAGNOSTICS MANUAL

POINT 4
DATA CORPORATION

POINT 4 COMPUTER
DIAGNOSTICS MANUAL

POINT 4
DATA CORPORATION

TABLE OF CONTENTS

POINT 4 SELF-TEST PROGRAM

The POINT 4 CPU has a comprehensive built-in diagnostic program, contained in a PROM (Programmable Read-Only Memory). This SELF-TEST program fully tests all the CPU logic and performs a comprehensive memory test of all the memory on the board - either 32K or 64K words. The only logic not tested is the DMA Data Channel, and the Interrupt and I/O logic are tested only to the extent possible without requiring a separate controller to exercise them. After the CPU and memory tests are completed the program relocates itself and repeats. This process continues until an error is found or the STOP switch is pressed.

How to run SELF-TEST:

1. Remove the front cover of the POINT 4 CPU.

2. While holding down the SELF-TEST pushbutton on the front edge of the CPU board, press APL on the mini-panel.

3. Then press CONT. SELF-TEST is now running.

What to expect:

After pressing APL (Step 2 above), the RUN light should be out. After pressing CONT (Step 3), the RUN light should come on and the CARRY light should flash on and off in a somewhat irregular pattern repeating about every 3 seconds.

If a terminal is connected to the CPU through a master terminal interface (device code 10/11), the program will give some informative output, as follows.

a. After sufficient preliminary tests have been run, it types:

"EDS POINT 4 SELF-TEST".

b. After first completion of CPU test, it types "32K CPU OK" or "64K CPU OK", whichever applies.

c. After first completion of memory test, it types "MEMORY OK".

d. Thereafter each time a complete test sequence is completed (roughly once every 3 seconds), it types "V".

Note 1. After running SELF-TEST 64K times (about 59 hours), type-outs b. and c. will be repeated.

Note 2. If no master terminal controller is plugged in no harm results. The program simply skips the type-out.

To test the Power-Fail Auto-Restart capability:

SELF-TEST contains the necessary code to test this option, if it is installed. Simply turn the key switch on the mini-panel to AUTO while SELF-TEST is running, and then unplug the a-c line cord. When it is plugged back in, SELF-TEST should resume running where it left off.

If SELF-TEST stops:

Any HALT indicates an error. Press APL to read in MANIP (the Virtual Front Panel program), assuming a master terminal is connected, and type A. This will indicate where SELF-TEST has halted, and also display the contents of the accumulators and carry at the time of the HALT. Refer to the Point 4 User's Manual, Section 5.5, for instructions on how to use MANIP. Dump the first few locations of memory. Normally,

Location 0 indicates the starting location to which SELF-TEST has relocated itself; i.e., 0 real = 20000 virtual.

Location 1 is the interrupt vector for the Power-Fail Auto-Restart test.

Location 2 indicates the (real) address of the last test started.

Two exceptions to this pattern exist:

1. If an interrupt other than Power-Fail has occurred, word 0 contains the value of the program counter at the time of the interrupt.

2. If a Power-Fail interrupt has occurred, the first four words will be:

```
0: 102520  SUBZL  0,0
1:  60377  NIOP   CPU    ;set 64K mode, if available
2:   2401  JMP    @.+1
3:    xxx  ;auto-restart pointer into SELF-TEST
```

If it is desired to use MANIP's "F" Offset (virtual) capability, type F(content of 0),20000. This will interpret all addresses as they are in the listing, regardless of their actual location. If A is typed after the offset has been entered, the HALT location is displayed in virtual form. By careful analysis of the program listing preceding the HALT, and of the tell-tale evidence left in the accumulators and temporary storage locations, it is usually possible to deduce the failure mechanism giving rise to the error.

Detailed description of SELF-TEST:

The SELF-TEST diagnostic contains the following tests (see program listing in back of this manual).

1. First, it tests the HALT instruction, because a HALT is used to indicate any subsequent error. This means that when SELF-TEST is initiated, the CPU should HALT (i.e. RUN light should go out), and CONT must be pressed to actually run the program.

2. SELF-TEST then performs a few preliminary tests to detect certain specific failures. These include in particular the compare instructions that will be used in subsequent tests, and the instructions used in the type-out subroutine.

3. Next SELF-TEST sets up the interrupt service vector at location 1 and enables interrupts, after masking out TTO (master terminal output, device code 11) interrupts. If an interrupt subsequently occurs, the CPU Power-Fail flag is tested by means of a SKPDN CPU instruction. If Power-Fail has been sensed, an auto-restart routine is put at location 0 and the program shuts down. Otherwise the CPU HALTs with the device code of the interrupting device in A1.

4. It then types "EDS POINT 4 SELF-TEST". This type-out is included only on the first pass; that is, after SELF-TEST relocates itself and restarts itself this type-out is suppressed. If no master terminal controller (Device Code 10/11) is included and operational, the program simply continues with no ill effects. At this time the program writes into each word its own address (except where SELF-TEST is located) to clear memory of parity errors which may exist after a power up.

5. ALU and Data Bus test: Increments a counter, using an ISZ instruction, 64K times starting from 0. To test that it takes 64K counts before the counter overflows (resulting in a skip), it uses the four accumulators in four nested loops doing sixteen 1-bit shifts each. This test uses all possible 16-bit numbers as the "Destination" input to the ALU (Arithmetic-Logic Unit), and checks that carry propagation can occur from the least significant position up to any other position. Also tests left and right shift capability for each bit.

6. ALU Source Operand test: Sums all numbers from 0 through 64K, and checks that the total is correct (to 16 bits). Uses all possible 16-bit numbers as the "Source" input to the ALU.

7. Exhaustive test of all ALU instructions. Executes all arithmetic and logical instructions from 100000 = COM 0,0 through 177777 = ANDCS# 3,3,SBN and checks that the final result is correct. This test exercises all operations the ALU is capable of, using a pseudo-random sequence of operands, and also uses all possible bit combinations in the instruction register (except msb = 0).

8. Page 0 and Base 3 addressing: Writes into each word of page zero (except when SELF-TEST is currently in page zero) that words own address, using Page Zero addressing mode. Then reads the value back using Base 3 addressing mode and confirms that it is correct.

9. Relative, Base 2, and indirect addressing modes: Reads each word in the 256-word region addressable by relative addressing three different ways and checks that the same value is being read each way. Exercises all possible address displacements in the memory reference type of instruction. Each of these displacements is used with the same value in A2 and the program counter -- different values in A2 and PC will be tested when SELF-TEST relocates itself and repeats this test.

10. SELF-TEST then determines if it is operating in a 64K system or a 32K system. If in a 64K system, the following three tests are done in both 64K mode and in 32K mode. Note: If SELF-TEST has relocated itself into upper 32K, then these tests can of course not be performed in 32K mode since SELF-TEST must remain in 64K mode.

11. Auto-Index: Tests the auto-increment capability using all possible values in location 20, and the auto-decrement capability using all possible values in location 37. Then tests that all other locations differing in only one bit from the range 20-37 do not auto-index.

12. I/O tests: Since no external I/O devices are required to run SELF-TEST, it only tests CPU I/O instructions. The I/O Skip instructions are tested by use of INTEN and INTDS: when interrupts are enabled SKPBN CPU should skip and SKPBZ CPU should not, and vice versa when interrupts are disabled. DIA -,CPU is a READS instruction and is tested by reading into two different accumulators and checking that they have picked up up the same value. DIB -,CPU is an INTA instruction and should pick up a zero value. DIC -,CPU is an IORST instruction and should not change the content of the accumulator specified. (Note: The IORST test can not be run when executing in upper 32K because it automatically sets the CPU into 32K mode.)

The program then tests the DIC instruction with all other device codes differing from 77 in only one bit, using two different accumulators and checking that they pick up the same value. Note that in the highly unusual case of a peripheral controller with device code 37, 57, 67, 73, 75, or 76 and responding to successive DIC instructions with different values, SELF-TEST would HALT at this test.

13. Multi-level indirect addressing test: If executing in 32K mode, a 3-level indirect addressing chain is tested, checking that the correct value is obtained. If executing in 64K mode, the 16-bit indirect addressing capability is tested instead.

14. In a 64K system, if currently in 32K mode, a small amount of code is moved into upper 32K and tested for proper execution. This test is included mainly so that any faults in execution in upper 32K can be detected by SELF-TEST on the first pass, without having to wait until it has relocated itself into upper RAM.

15. When all these tests have been passed correctly, it types out either "64K CPU OK" or "32K CPU OK", as the case may be. This type-out is suppressed after the first pass of SELF-TEST, though it will occur again after 64K passes (about 59 hours).

16. Memory test: If SELF-TEST is currently located below the midpoint of available memory, it tests all memory above itself; otherwise it tests all memory below itself (except locations 0 and 1 which are always reserved for the current location of SELF-TEST and for the interrupt vector). The memory test algorithm consists of four tests as follows:

1. Write a 1 into each bit of the first word, then change it to a 0, then change it back to 1. Do the same for each successive word until all words contain 177777. Now test the first word, check that it contains 177777. Change it to 0, retest, then change it back to 177777. Repeat for each successive word. This algorithm insures that between the time any word was set to 177777 and the time it is tested, all other words have been toggled back and forth between 0 and 177777.

2. Repeat the algorithm of test 1 except with 1's and 0's interchanged.

3. Write into each word its own address.

4. Write into each word a 73077 (HALT). This serves two purposes:

   a. 73077 has odd parity, thus insuring that parity RAM will toggle between 1 and 0.

   b. If SELF-TEST ever jumps out of itself it will halt, saving the accumulators for failure analysis.

17. When Memory test is passed, type "MEMORY OK". Suppressed after first pass.

18. SELF-TEST now copies itself to a location slightly more than 20000 words below itself, wrapping around to the top of memory if necessary. Since its initial location is 20000, the first move will bring it into upper RAM (straddling location 0 is not allowed).

```
ASM ,@$LPT1,SELF.SN
JAN 25, 1980  14:41:02

; SELF-TEST PROGRAM FOR POINT 4 CPU
; WRITTEN BY RENNY BOSCH
; 12-10-79

;                    All Rights Reserved
;      Copyright (C) 1979, Educational Data Systems


          20000 SELF= 20000

              1 .TXTM 1
          20000 .LOC  SELF
   20000  20000       SELF              ;TELLS APL LOGIC WHERE TO LOAD SELF

   20001  63077       HALT              ;TEST HALT (MUST PRESS CONT.)

; A FEW BASIC ALU TESTS

   20002 102001       ADC    0,0,SKP    ;A0 = 177777
   20003  63077       HALT              ;UNCONDITIONAL "SKP" FAILED TO SKIP
   20004 126424       SUBZ   1,1,SZR    ;A1=0
   20005  63077       HALT
   20006 152000       ADC    2,2
   20007 151404       INC    2,2,SZR    ;A2=0
   20010  63077       HALT
   20011 176000       ADC    3,3
   20012 162415       SNE    3,0        ;A3,A0 SHOULD = 177777
   20013 132414       SEQ    1,2        ;A1,A2 SHOULD = 0
   20014  63077       HALT

; A FEW BASIC JMP, LDA, STA, ISZ TESTS USING RELATIVE ADDRESSING

   20015  20405       LDA    0,.+5
   20016 116414       SEQ    0,3
   20017  63077       HALT              ;A0 & A3 SHOULD = 177777
   20020  30457       LDA    2,C100K
   20021 102621       SUBZR  0,0,SKP
   20022 177777       177777
   20023 112414       SEQ    0,2
   20024  63077       HALT              ;A0 & A2 SHOULD = 100000
   20025    402       JMP    .+2        ;TEST JMP REL.
   20026  63077       HALT              ;SHOULD JUMP OVER THIS
   20027  40401       STA    0,.+1
   20030  63077 COM00:HALT              ;PGM CHANGES TO 100000=COM 0,0
   20031 112415       SNE    0,2
   20032  63077       HALT              ;A0 SHOULD = 77777, A2 = 100000
   20033 100000       COM    0,0
   20034  24774       LDA    1,COM00
   20035 112415       SNE    0,2
   20036 132414       SEQ    1,2
   20037  63077       HALT              ;A0, A1, A2 SHOULD = 100000
```

```
  20040   4404        JSR    JMP4       ;TEST INSTRS. USED IN "TYPE" S\R
  20041  63277 REF1: HALT!INTDS
  20042    416 REF2: JMP    INTSU
  20043  63077        HALT
  20044  54002 JMP4: STA    3,2        ;LOC. 2 --> LAST TEST BEGUN
  20045  25400        LDA    1,0,3
  20046  20773        LDA    0,REF1
  20047 106414        SEQ    0,1
  20050  63077        HALT              ;A0 & A1 SHOULD = 63277 = (REF1)
  20051 175420        INCZ   3,3
  20052  25400        LDA    1,0,3
  20053  20767        LDA    0,REF2
  20054 106414        SEQ    0,1
  20055  63077        HALT              ;A0 & A1 SHOULD = 416 = (REF2)
  20056   1400        JMP    0,3        ;SHOULD GET TO "INTSU" VIA "REF2"
  20057  63077        HALT

  20060   4426 INTSU:JSR    SETUP      ;PICK UP POINTER TO INTERRUPT SERVICE
                      ;
; INTERRUPT SERVICE ROUTINE -- ONLY USED IF AN INTERRUPT OCCURS
                      ;
  20061  61477 INTSV:INTA   0          ;GET INTERRUPTING DEVICE CODE
  20062  63677        SKPDN  CPU        ;IS IT A POWER FAILURE ?
  20063  63077        HALT              ;  NO, OTHER INTRPT (A0 = DEV. CODE)
  20064   4403        JSR    PWRF       ;  YES, PICK UP AUTO RESTART POINTER
                      ;
  20065    773 AUTOS:JMP    INTSU      ;AUTO RESTART WILL ENTER HERE

  20066      1 PCNT: 1      ;POWER FAIL COUNTER
  20067  20420 PWRF: LDA    0,.SUBZ   ;POWER FAIL; SET UP RESTART ROUTINE:
  20070  40000        STA    0,0        ;     0: 102520  SUBZL  0,0
  20071  20417        LDA    0,.NIOP
  20072  40001        STA    0,1        ;     1:  60377  NIOP   CPU
  20073  20406        LDA    0,C2401
  20074  40002        STA    0,2        ;     2:   2401  JMP    @.+1
  20075  54003        STA    3,3        ;     3: POINTER TO "AUTOS"
  20076  10770        ISZ    PCNT
  20077 100000 C100K:100000
  20100  63077        HALT

  20101   2401 C2401:2401
  20102 177777 INIFL:-1                 ;INITIAL FLAG FOR TYPE-OUT
  20103  20000 .SELF:SELF
  20104      1 CNTR: 1
  20105  20767 END.: END-1
```

```
20106  54001 SETUP:STA    3,1        ;SET UP INTERRUPT SERVICE VECTOR
20107 102520 .SUBZ:SUBZL  0,0
20110  60377 .NIOP:NIOP   CPU        ;SET 64K MODE, IF AVAIL.
20111  62077       MSKO   0          ;MASK OUT TTO INTERRUPTS
20112  60177       INTEN
20113  20767       LDA    0,INIFL
20114 101015       SNZ    0,0        ;INITIAL ENTRY TO SELF-TEST ?
20115    444       JMP    TISZ       ;  NO, SKIP TYPE-OUT
20116 176420       SUBZ   3,3        ;A3 = 0
20117  54763       STA    3,INIFL    ;INIFL = 0
20120  60211       NIOC   TTO
20121  63511       SKPBZ  TTO        ;MAKE SURE TTY WON'T HANG UP
20122  63077       HALT
20123   4556       JSR    TYPE       ;"EDS POINT 4 SELF-TEST"
20124   6412 .TXT "<15><12>
20125  42504 ED
20126  51440 S
20127  50117 PO
20130  44516 IN
20131  52040 T
20132  32040 4
20133  51505 SE
20134  46106 LF
20135  26524 -T
20136  42523 ES
20137  52015 T<15>
20140   5000 <12>"

20141 176000       ADC    3,3        ;CHECK IF 64K MEMORY AVAILABLE
20142 152220       ADCZR  2,2
20143  55001       STA    3,1,2      ;STORE 177777 AT 100000 (IF 64K)
20144  50000       STA    2,0        ;STORE 77777 AT 0
20145  31001       LDA    2,1,2      ;PICK UP FROM 100000 (IF 64K)
20146 156554       SUBOL# 2,3,SZR
20147  63077       HALT              ;PICKED UP NEITHER 77777 NOR 177777
20150  50545       STA    2,TOPWD    ;SAVE TOP WORD (77777 OR 177777)

20151  20732       LDA    0,.SELF
20152 176520       SUBZL  3,3        ;A3 = 1
20153 175400 FILL:INC     3,3        ;POINTER TO NEXT ADDRESS
20154  41400       STA    0,0,3      ;CLEAR PARITY ERROR FROM MEMORY
20155 162415       SNE    3,0        ;ARE WE AT THE BEGINING OF SELF TEST?
20156  34727       LDA    3,END.     ;YES, SET A3 = FIRST WORD AFTER SELF TEST
20157 172414       SEQ    3,2        ;HAVE WE FINISHED CLEARING MEMORY?
20160    773       JMP    FILL       ;NO, DO NEXT ADDRESS
20161 102400 TISZ: SUB    0,0
20162  40722       STA    0,CNTR
20163  10721       ISZ    CNTR       ;TEST ISZ AND DSZ INSTR'S
20164  14720       DSZ    CNTR
20165  63077       HALT
20166  14716       DSZ    CNTR
20167  10715       ISZ    CNTR
20170  63077       HALT
```

```
; TEST ALU: INCREMENT A COUNTER (VIA ISZ INSTRUCTION) 64K TIMES,
; CHECK THAT NO COUNTS ARE SKIPPED BY USING 4 NESTED LOOPS OF 16 BIT SHIFTS

   20171   4534 TALU: JSR    PIKUP     ;(SKIPS NEXT WORD)
                      ;
   20172  42263 X:    42263  ;CHECKSUM FOR "EXHAUSTIVE ALU TEST"
                      ;
   20173 176120       ADCZL  3,3       ;A3 = 177776
   20174 152620 LP3:  SUBZR  2,2       ;A2 = 100000
   20175 126220 LP2:  ADCZR  1,1       ;A1 = 77777
   20176 101015 LP1:  SNZ    0,0       ;VERY FIRST PASS ?
   20177 102521       SUBZL  0,0,SKP   ;  YES, SET A0 = 1 & SKIP
   20200  10704 LP0:  ISZ    CNTR      ;  NO, COUNT UP CNTR
   20201 101015       SNZ    0,0
   20202  63077       HALT
   20203 101123       MOVZL  0,0,SNC   ;A0 SHIFTED 16 BITS ?
   20204    774       JMP    LP0       ;  NO, LOOP
   20205 101015       SNZ    0,0
   20206 124015       COM#   1,1,SNR
   20207  63077       HALT
   20210 102520       SUBZL  0,0
   20211 125242       MOVOR  1,1,SZC   ;HAS A1 SHIFTED 16 TIMES ?
   20212    764       JMP    LP1       ;  NO, REPEAT
   20213 124015       COM#   1,1,SNR
   20214 151015       SNZ    2,2
   20215  63077       HALT
   20216 151223       MOVZR  2,2,SNC   ;A2 SHIFTED 16 TIMES ?
   20217    756       JMP    LP2       ;  NO
   20220 151015       SNZ    2,2
   20221 174015       COM#   3,3,SNR
   20222  63077       HALT
   20223 175142       MOVOL  3,3,SZC   ;A3 SHIFTED 16 TIMES ?
   20224    750       JMP    LP3       ;  NO
   20225  20657       LDA    0,CNTR   ;CNTR HAS NOW BEEN INC'D 16*16*16*16-1
   20226 116415       SNE    0,3       ;  = 177777(8) TIMES
   20227  10655       ISZ    CNTR      ;ONE MORE ISZ SHOULD PRODUCE SKIP
   20230  63077       HALT

   20231 102400       SUB    0,0       ;TEST EVERY POSS. NO. AS SOURCE OPND.
   20232 126400       SUB    1,1
   20233 107000 ADDLP:ADD    0,1       ;ADD ALL NUMBERS 0 THROUGH 64K-1
   20234 101404       INC    0,0,SZR
   20235    776       JMP    ADDLP
   20236 102620       SUBZR  0,0       ;SUM SHOULD = 100000 MOD 64K
   20237 106414       SEQ    0,1
   20240  63077       HALT
```

```
; EXHAUSTIVE TEST OF ALL ALU INSTRUCTIONS

   20241 176220        ADCZR  3,3        ;A3 = 77777 (ARBITARY INITIAL COND. )
   20242 171300        MOVS   3,2        ;A2 = 177577
   20243 145520        INCZL  2,1        ;A2 = 177401
   20244 102620        SUBZR  0,0        ;A0 = 100000
   20245  40401        STA    0,.+1

   20246  63077 ALUI:  HALT              ;CYCLES THROUGH ALL ALU INSTR.
   20247 147100        ADDL   2,1        ; \
   20250 123100        ADDL   1,0        ;  } FOLD RESULT INTO A3
   20251 117100        ADDL   0,3        ; /
   20252  10774        ISZ    ALUI       ;MODIFY INSTRUCTION; ALL DONE ?
   20253    773        JMP    ALUI       ;  NO, CONTINUE
   20254  20716        LDA    0,X        ;  YES
   20255 162414        SEQ    3,0        ;IS FINAL RESULT CORRECT ?
   20256  63077        HALT              ;  NO, ALU ERROR

; BASE 3 ADDRESSING VS. PAGE ZERO

   20257   4446        JSR    PIKUP
   20260    730        R-SELF+400
   20261 172032        SGE    3,2        ;IS SELF ABOVE PAGE ZERO ?
   20262    446        JMP    TJSR       ;  NO, SKIP PZ TEST
   20263 176520        SUBZL  3,3        ;SET UP FOR PAGE ZERO TEST
   20264  20002        LDA    0,2
   20265  20777        LDA    0,.-1
   20266  40403        STA    0,LDA0
   20267 175400 B3LP:  INC    3,3        ;  INCREMENT BY 1 WORD
   20270  55400        STA    3,0,3      ;INTO EACH WORD WRITE ITS OWN ADDRESS
   20271  20002 LDA0:  LDA    0,2        ;*******GETS MODIFIED BY PROGRAM******
   20272 116414        SEQ    0,3        ;DID WE GET BACK WHAT WE WROTE?
   20273  63077        HALT              ;  NO
   20274  10775        ISZ    LDA0       ;MODIFY THE LOAD INSTRUCTION
   20275  20423        LDA    0,C377
   20276 162032        SGE    3,0        ;IS A3 < 377
   20277    770        JMP    B3LP       ;  NO, REPEAT LOOP
   20300    430        JMP    TJSR       ;YES, GO ON TO NEXT TEST

   20301  25400 TYPE:  LDA    1,0,3      ;TYPE-OUT SUBROUTINE
   20302 175420        INCZ   3,3
   20303  20411 TYPE2:LDA     0,C377L    ;OUTPUT LEFT BYTE FIRST
   20304 123705        ANDS   1,0,SNR    ;ZERO BYTE (TERMINATOR) ?
   20305   1400        JMP    0,3        ;  YES, RETURN TO CALLER
   20306  61111        DOAS   0,TTO      ;OUTPUT BYTE TO TTY
   20307  63511        SKPBZ  TTO        ;WAIT FOR TTY NOT BUSY
   20310    777        JMP    .-1
   20311 125362        MOVCS  1,1,SZC    ;HAVE WE OUTPUT BOTH BYTES YET ?
   20312    771        JMP    TYPE2      ;  NO, DO THE RIGHT BYTE NOW
   20313    766        JMP    TYPE       ;  YES, GET NEXT 2 BYTES

   20314 177400 C377L:177400
```

```
 20315 177777 TOPWD:177777
 20316 177777 CURMO:177777 ;CURRENT ADDRESSING MODE (77777 OR 177777)
 20317 177777 UPPER:177777 ;177777 IF SELF IS IN UPPER 32K, ELSE 77777
 20320    377 C377: 377
 20321 177600 CM200:-200
 20322     20 C20:  20
 20323 177737 C40C: -1-40
 20324  20152 ADR:  LDREL-200         ;USED IN RELATIVE ADDRESSING TEST

; SUB-ROUTINE TO PICK UP POINTER TO CENTRAL REFERENCE POINT

 20325  54002 PIKUP:STA    3,2        ;LOC. 2 --> LAST TEST STARTED
 20326  31400       LDA    2,0,3      ;LOAD PARAMETER WORD
 20327   5401       JSR    1,3        ;SKIP-RETURN WITH POINTER TO "R"
        20330 R=     .          ;REFERENCE POINT USED FOR ADDRESSING EXTENSION

; BASE 2, RELATIVE, AND INDIRECT ADDRESSING - ALL WITHIN +-200 OF HERE

 20330   4775 TJSR: JSR    PIKUP
 20331     22       LDREL-R
 20332 173000       ADD    3,2        ;CALC. LOC. OF "LDREL"
 20333  20766       LDA    0,CM200
 20334 143040       ADDO   2,0
 20335  40767       STA    0,ADR      ;SET UP "ADR" = LDREL - 200
 20336  35200       LDA    3,-200,2
 20337  20777       LDA    0,.-1      ;PICK UP BASE 2 INSTR.
 20340  34600       LDA    3,.-200
 20341  34777       LDA    3,.-1      ;PICK UP REL. ADDR. INSTR.
 20342  40404 SETAD:STA    0,LDAB2    ;SET UP BASE 2 INSTRUCTION
 20343  54407       STA    3,LDREL    ;SET UP REL. ADDR. INSTR.
 20344  24755       LDA    1,CM200

 20345   4401 B2LP: JSR    .+1        ;FOR WORST CASE ADDRESS CALC. TIME
 20346  35200 LDAB2:LDA    3,-200,2   ;*** GETS MODIFIED BY PROGRAM ***
 20347  22755       LDA    0,@ADR
 20350 116414       SEQ    0,3        ;A0 = INDIR., A3 = BASE 2 ADDRESSING
 20351  63077       HALT              ;THEY DON'T MATCH !?
 20352  34600 LDREL:LDA    3,.-200    ;*** GETS MODIFIED BY PROGRAM ***
 20353 116414       SEQ    0,3
 20354  63077       HALT
 20355  10747       ISZ    ADR        ;INCREMENT INDIRECT ADDRESS
 20356  10770       ISZ    LDAB2      ;AND BASE 2 LOAD INSTRUCTION,
 20357  10773       ISZ    LDREL      ;AND RELATIVE LOAD INSTRUCTION
 20360 125404       INC    1,1,SZR    ;HAVE WE TESTED 200 LOCATIONS ?
 20361    764       JMP    B2LP       ;  NOT YET, REPEAT LOOP
 20362  35000       LDA    3,0,2      ;PREPARE FOR 2ND 200 LOCATIONS
 20363  20777       LDA    0,.-1      ;PICK UP BASE 2 INSTR.
 20364  34400       LDA    3,.        ;PICK UP REL. ADDR. INSTR.
 20365 101002       MOV    0,0,SZC    ;HAVE WE DONE 2ND PASS ALREADY ?
 20366    754       JMP    SETAD      ;  NO, DO IT NOW

 20367   4736       JSR    PIKUP      ;PICK UP REFERENCE POINTER
 20370  77340       R-END+100000
 20371 172432       SGR    3,2        ;ARE WE EXECUTING IN UPPER 32K ?
 20372 102221       ADCZR  0,0,SKP    ;  NO, THEN USE 32K MODE FIRST
 20373 102000       ADC    0,0        ;  YES, THEN NEED 64K MODE
 20374  40723       STA    0,UPPER
 20375  40721       STA    0,CURMO
```

; TEST AUTO INDEX CAPABILITY

```
   20376  20720 AUTOX:LDA    0,CURMO
   20377 101520       INCZL  0,0
   20400  60377       NIOP   CPU      ;SET CPU IN CURRENT MODE (32K OR 64K)
   20401   4724       JSR    PIKUP
   20402    370       R-SELF+40
   20403 172432       SGR    3,2      ;DOES SELF OVERLAP LOC. 0-40 ?
   20404    451       JMP    IOTST    ;  YES, SKIP AUTO INDEX TESTS
   20405 152000       ADC    2,2
   20406  50020       STA    2,20     ;-1 IN AUTO INCREMENT CELL
   20407  34707       LDA    3,CURMO
   20410 175400       INC    3,3
   20411  54037       STA    3,37     ;0 (OR 100000) IN AUTO DECR. CELL
   20412 151400 AXLP: INC    2,2
   20413  22020       LDA    0,@20    ;CAUSES (20) TO INCREMENT
   20414  24020       LDA    1,20
   20415 132414       SEQ    1,2      ;DID THE VALUE IN 20 INCREMENT ?
   20416  63077       HALT            ;  NO - ERROR
   20417  25000       LDA    1,0,2
   20420 106414       SEQ    0,1      ;DID WE LOAD @20 CORRECTLY ?
   20421  63077       HALT            ;  NO ?!
   20422 174400       NEG    3,3      ;DECREMENT A3
   20423 174000       COM    3,3
   20424  22037       LDA    0,@37    ;CAUSES (37) TO DECREMENT
   20425  20037       LDA    0,37
   20426 116414       SEQ    0,3      ;DID CONTENT OF 37 DECREMENT ?
   20427  63077       HALT            ;  NO, ERROR
   20430 175014       SKZ    3,3
   20431    761       JMP    AXLP
```

; NOW TEST THAT <20 AND >37 DO NOT AUTO INDEX

```
   20432  20670       LDA    0,C20
   20433  40005       STA    0,5
   20434  26005       LDA    1,@5     ;LOC. 5 SHOULD NOT AUTO-INDEX
   20435  24005       LDA    1,5
   20436 106414       SEQ    0,1      ;DID WE GET BACK WHAT WE WROTE ?
   20437  63077       HALT            ;  NO !?
   20440 111000       MOV    0,2      ;NOW A0 = A2 = 20
   20441 113000 NAXLP:ADD    0,2      ;TEST 40, THEN 60, 120, 220, 420, ETC.
   20442  25000       LDA    1,0,2
   20443 125113       SSN    1,1      ;PREVENT POSS. INF. INDIRECT CHAIN
   20444  37000       LDA    3,@0,2   ;SHOULD NOT AUTO-INDEX
   20445  35000       LDA    3,0,2
   20446 136414       SEQ    1,3      ;DID 2 LDA INSTR'S GIVE SAME RESULT ?
   20447  63077       HALT            ;  NO, ERROR
   20450 112400       SUB    0,2
   20451 101120       MOVZL  0,0      ;PREPARE FOR NEXT LOCATION
   20452  24644       LDA    1,CURMO  ;IF 32K MODE,
   20453 107414       AND#   0,1,SZR  ;DON'T TEST LOC. 100020 (= 20)
   20454    765       JMP    NAXLP
```

```
; I\O TESTS

   20455   4650 IOTST:JSR    PIKUP     ;(SKIPS NEXT WORD)
                      ;
   20456    623 JTYPE:JMP    TYPE      ;ELEVATOR TO "TYPE"
                      ;
   20457  60277       INTDS
   20460  63477       SKPBN  CPU
   20461  63577       SKPBZ  CPU
   20462  63077       HALT             ;ION SHOULD BE OFF
   20463  60177       INTEN
   20464  63577       SKPBZ  CPU
   20465  63477       SKPBN  CPU
   20466  63077       HALT             ;ION SHOULD BE ON
   20467  20627       LDA    0,CURMO
   20470 101112       SSP    0,0       ;ARE WE CURRENTLY IN 64K MODE ?
   20471    415       JMP    IO64K     ;  YES, CAN'T TEST IORST
   20472  62677       IORST
   20473  63477       SKPBN  CPU
   20474  63577       SKPBZ  CPU
   20475  63077       HALT             ;ION SHOULD BE OFF AGAIN

   20476 102521       SUBZL  0,0,SKP
                      ;
   20477    626 JPIK: JMP    PIKUP     ;ELEVATOR TO "PIKUP"
                      ;
   20500 126220       ADCZR  1,1

   20501  62477 DIC07:DIC    0,77      ; = IORST
   20502  66477 DIC17:DIC    1,77      ;(SHOULD NOT CHANGE ACCUMULATOR)
   20503 106415       SNE    0,1
   20504  63077       HALT             ;A0 S/B 1, A1 S/B 77777
   20505  62077       MSKO   0         ;MASK OUT TTO INTERRUPTS

   20506  61477 IO64K:DIB    0,CPU     ; = INTA
   20507 101014       SKZ    0,0
   20510  63077       HALT             ;NO DEVICE SHD. ACK. INTERRUPT
   20511  60677       DIAC   0,CPU     ; = READS
   20512  64577       DIAS   1,CPU     ;ALSO ENABLES ION
   20513 106414       SEQ    0,1
   20514  63077       HALT             ;A0 AND A1 SHD = MINI-SWITCHES

; NOW CHECK THAT DEVICE CODES OTHER THAN 77 DO A DIC NORMALLY

   20515  30606       LDA    2,C40C

   20516  20763 IOLP: LDA    0,DIC07
   20517  24763       LDA    1,DIC17
   20520 143400       AND    2,0
   20521 147400       AND    2,1
   20522  40401       STA    0,.+1
   20523  62437       DIC    0,37      ;PGM CHANGES TO DEV. 57,67,73,75,76
   20524  44401       STA    1,.+1
   20525  66437       DIC    1,37      ;DITTO
   20526 106414       SEQ    0,1
   20527  63077       HALT             ;DIC INSTR'S SHD PICK UP SAME VALUE
   20530 151242       MOVOR  2,2,SZC   ;HAVE WE DONE DEVICE 76 ?
   20531    765       JMP    IOLP      ;  NO, GO BACK
```

```
; MULTI-LEVEL @ IF IN 32K MODE -- OR 16-BIT @ IF IN 64K MODE

   20532   4745 MULTI:JSR    JPIK      ;PICK UP POINTER TO R
                      ;
   20533 120541 A:     @B
                      ;
   20534  54404        STA    3,C       ;SET UP C TO POINT TO R
   20535  21400        LDA    0,0,3     ;SAVE THE VALUE AT R
   20536  31766        LDA    2,CURMO-R,3
   20537   4403        JSR    .+3
                      ;
   20540  20330 C:     R
   20541 120540 B:     @C
                      ;
   20542 177245        ADDOR  3,3,SNR   ;COMLEMENT MSB OF A3 (GIVES C')
   20543    446        JMP    CPUDN     ;PROTECT WORD 1 (INTSV)
   20544  45401        STA    1,1,3     ;PUT A KNOWN VALUE IN B'
   20545 151112        SSP    2,2       ;ARE WE IN 64K MODE ?
   20546 121000        MOV    1,0       ;  YES, REMEMBER THE VALUE IN B'
   20547  54772        STA    3,B       ;SET UP B TO POINT @C (IF 32K)
   20550 175400        INC    3,3
   20551  54762        STA    3,A       ;A POINTS @B (32K) OR AT B' (64K)
   20552  26761        LDA    1,@A      ;THIS IS THE LOAD @ INSTRUCTION
                                        ;32K: @A = A --> B --> C --> R
                                        ;64K: @A = A --> B' (= B W.OPP.MSB)
   20553 106414        SEQ    0,1
   20554  63077        HALT             ;MULTI @ FAILED (IF CURMO = 77777)

   20555   4722        JSR    JPIK      ;PICK UP POINTER TO R
                      ;
   20556    700 JJTYP:JMP    JTYPE     ;ELEVATOR TO "TYPE" (PGM SKIPS)
                      ;
   20557  21766        LDA    0,CURMO-R,3
   20560  25765        LDA    1,TOPWD-R,3
   20561 106033        SLS    0,1       ;CURR. IN 32K BUT 64K AVAIL. ?
   20562    427        JMP    CPUDN     ;  NO, THEN CPU TEST IS DONE
   20563  45766        STA    1,CURMO-R,3;  YES, THEN SET 64K MODE
   20564 102520        SUBZL  0,0
   20565  60377        NIOP   CPU
```

```
  20566   4403        JSR    .+3        ;MOVE SOME CODE TO UPPER 32K
                      ;
  20567  63077 HALT.:HALT               ;THESE TWO WORDS WILL BE MOVED
  20570   5401        JSR    1,3        ;TO UPPER 32K
                      ;
  20571  20776        LDA    0,.-2
  20572  24776        LDA    1,.-2
  20573 177240        ADDOR  3,3        ;SET MSB OF A3
  20574  45400        STA    1,0,3      ;MOVE THE 2 WORDS,
  20575  41401        STA    0,1,3      ;  REVERSING THEIR ORDER
  20576  24771        LDA    1,HALT.
  20577 122414        SEQ    1,0        ;DID WE CLOBBER LOWER CORE ?
  20600  63077        HALT              ;  YES - 64K MODE NOT WORKING
  20601 171400        INC    3,2        ;SAVE A3 VALUE
  20602   5400        JSR    0,3        ;JMP TO THE JSR 1,3 IN UPPER 32K
  20603  63077        HALT              ;IT SHD. RETURN TO THE NEXT LOCATION:
  20604 172414        SEQ    3,2        ;DID IT PICK UP THE RIGHT A3 ?
  20605  63077        HALT              ;  NO
  20606   4671        JSR    JPIK       ;PICK UP POINTER TO R
                      ;
  20607    670 JJPIK:JMP     JPIK       ;ELEVATOR TO "PIKUP" (PGM SKIPS)
                      ;
  20610   1446        JMP    AUTOX-R,3;REDO TESTS IN 64K MODE

; CPU LOGIC TEST IS NOW COMPLETED

  20611  10546 CPUDN:ISZ    INIFC      ;IS THIS FIRST PASS OF SELF-TEST ?
  20612    420        JMP    TMEM       ;  YES, SKIP TYPE-OUT
  20613 125113        SSN    1,1        ;TYPE "64K" OR "32K"
  20614    405        JMP    TP32K
  20615   4741        JSR    JJTYP
  20616  33064 .TXT "64
  20617  45400 K"

  20620    404        JMP    TPCPU

  20621   4735 TP32K:JSR    JJTYP
  20622  31462 .TXT "32
  20623  45400 K"

  20624   4732 TPCPU:JSR    JJTYP      ;TYPE " CPU OK,"
  20625  20103 .TXT " C
  20626  50125 PU
  20627  20117  O
  20630  45454 K,
  20631  20000  "



; MEMORY TEST:  FIRST PASS: SET A BIT TO 1, SET IT TO 0, THEN SET
; IT BACK TO 1, THEN DO THE SAME TO NEXT BIT, ETC.
; SECOND PASS: TEST THAT THE BIT = 1, TOGGLE IT TO 0, RETEST,
; AND BACK TO 1, THEN DO SAME FOR NEXT BIT --
; THUS EACH BIT IS TESTED AFTER ALL OTHER BITS HAVE BEEN TOGGLED.
; THEN REPEAT THE WHOLE TEST WITH 0'S AND 1'S INTERCHANGED
; THIRD TEST: USE EACH WORD'S ADDRESS IN PLACE OF 0'S OR 1'S
; FOURTH TEST : USE 73077 HALT (HAS ODD PARITY) IN PLACE OF ADDRESS
```

```
20632   4755 TMEM: JSR    JJPIK
20633    440       END-R
20634 173000       ADD    3,2       ;A2 = FIRST LOC. ABOVE SELF-TEST
20635  21765       LDA    0,TOPWD-R,3
20636  34530       LDA    3,NWDS    ;A3 = -(LENGTH OF SELF-TEST)
20637 101220       MOVZR  0,0       ;MIDPOINT OF AVAILABLE RAM
20640 142033       SLS    2,0       ;ARE WE CURRENTLY ABOVE MIDPOINT ?
20641 157001       ADD    2,3,SKP   ;  YES, TEST LOWER MEMORY
20642 115141       MOVOL  0,3,SKP   ;  NO, TEST UPPER PORTION
20643  30513       LDA    2,C2      ;PROTECT LOC. 0 (INTRPT VECTOR)
20644  50523       STA    2,FIRST
20645 102040       ADCO   0,0       ;SET A0 = 177777, C = 1

20646  30521 MTEST:LDA    2,FIRST   ;FIRST PASS - SET UP MEMORY
20647 101003 LOOP1:MOV    0,0,SNC   ;IS THIS THE THIRD TEST ?
20650 141000       MOV    2,0       ;  YES: USE ADDRESS
20651  41000       STA    0,0,2
20652 104000       COM    0,1
20653  45000       STA    1,0,2     ;TOGGLE MEMORY WORD
20654  41000       STA    0,0,2     ;TOGGLE BACK AGAIN
20655 151400       INC    2,2
20656 156032       SGE    2,3       ;ALL SET UP ?
20657    770       JMP    LOOP1     ;  NOT YET
20660  30507       LDA    2,FIRST   ;SECOND PASS - TEST MEMORY
20661 101003 LOOP2:MOV    0,0,SNC   ;ARE WE ON THE THIRD TEST ?
20662 141000       MOV    2,0       ;  YES, USE ADDRESS
20663  25000       LDA    1,0,2
20664 106414       SEQ    0,1
20665  63077       HALT             ;A0 = S/B, A1 = IS, A2 = ADR.
20666 104000       COM    0,1
20667  45000       STA    1,0,2     ;TOGGLE MEMORY
20670  25000       LDA    1,0,2     ;RETEST
20671 124000       COM    1,1
20672 106414       SEQ    0,1
20673  73077 HALTI:73077
20674  41000       STA    0,0,2     ;TOGGLE MEMORY WORD BACK AGAIN
20675 151400       INC    2,2
20676 156032       SGE    2,3       ;TESTED ALL LOCATIONS ?
20677    762       JMP    LOOP2     ;  NO
20700 101466       INCC   0,0,SEZ   ;NOW PREPARE FOR NEXT TEST
20701  20772       LDA    0,HALTI   ;GET THE HALT INSTRUCTION
20702  24771       LDA    1,HALTI   ;GET 73077 INSTRUCTION
20703 122014       ADC#   1,0,SZR   ;HAVE WE DONE FOUR TESTS?
20704    742       JMP    MTEST     ;NO, DO NEXT TEST

20705  10453 TFP:  ISZ    INIFM     ;IS THIS FIRST PASS OF SELF-TEST ?
20706    410       JMP    TP.VS     ;  YES, SKIP TYPE-OUT
20707   4647       JSR    JJTYP     ;TYPE "MEMORY OK."
20710  46505 .TXT "ME
20711  46517 MO
20712  51131 RY
20713  20117  O
20714  45456 K.
20715      0 "

       20715 .LOC  .-1     ;TO AVOID A ZERO WORD (ENDS APL LOADING)
20715      1       1
```

```
   20716  14444 TP.VS:DSZ     CCNT        ;TYPE A "V"
   20717    406       JMP     TYPV
   20720  20441       LDA     0,LCNT
   20721  40441       STA     0,CCNT
   20722   4634       JSR     JJTYP
   20723   6412       15*400+12           ;CR, LF
   20724      1       1
   20725   4631 TYPV: JSR     JJTYP
   20726  53000       "V*400


; MOVE TEST PGM THRU CORE + REPEAT

   20727   4660 MOVE: JSR     JJPIK
   20730 130406       -DIST*2             ;TENT. ASSUME DOUBLE MOVE REQUIRED
   20731  20432       LDA     0,R.MIN
   20732  24432       LDA     1,R.MAX
   20733 162433       SLE     3,0         ;IS SELF WHERE SINGLE MOVE WOULD
   20734 166033       SLS     3,1         ;    CAUSE STRADDLING WORDS 0-3 ?
   20735 151240       MOVOR   2,2         ;  NO, THEN DO SINGLE MOVE
   20736  21765       LDA     0,TOPWD-R,3
   20737  24426       LDA     1,R.OFS
   20740 136400       SUB     1,3         ;A3 = CURRENT LOC. OF SELF
   20741 173000       ADD     3,2
   20742 113400       AND     0,2         ;A2 = NEW LOCATION OF SELF
   20743  24423       LDA     1,NWDS
   20744  21400 MOVLP:LDA     0,0,3       ;NOW DO THE MOVE LOOP
   20745  41000       STA     0,0,2
   20746 175400       INC     3,3
   20747 151400       INC     2,2
   20750 125404       INC     1,1,SZR     ;MOVE DONE ?
   20751    773       JMP     MOVLP       ;  NO
   20752  24414       LDA     1,NWDS
   20753 133000       ADD     1,2
   20754  50000       STA     2,0         ;FOR EASILY FINDING SELF WHEN MOVED
   20755   1002       JMP     2,2

   20756      2 C2:   2
   20757 177777 INIFC:177777 ;INITIAL FLAG FOR "CPU OK." MESSAGE
   20760 177777 INIFM:177777 ;INITIAL FLAG FOR "MEMORY OK." MESSAGE
   20761    110 LCNT: 110    ;LINE COUNT = NO. OF V'S BEFORE A CR\LF
   20762      1 CCNT: 1      ;COLUMN COUNT OF V'S TYPED
   20763  23134 R.MIN:DIST-END+R-1
   20764  24131 R.MAX:DIST+R-SELF+4
   20765    330 R.OFS:R-SELF
   20766 177010 NWDS: SELF-END ;NO. OF WORDS IN SELF PROGRAM
   20767      0 FIRST:0      ;TELLS APL LOGIC: END OF PROGRAM

          23575 DIST= 23575  ;DISTANCE SELF IS MOVED EACH TIME
          20770 END=  .
             10 .LOC  SELF+1000-.       ;OVERFLOW TEST


                .END  ;POINT 4 SELF-TEST
```

```
---

A       20533   ADDLP   20233   ADR     20324   ALUI    20246   AUTOS   20065
AUTOX   20376   AXLP    20412   B       20541   B2LP    20345   B3LP    20267
C       20540   C100K   20077   C2      20756   C20     20322   C2401   20101
C377    20320   C377L   20314   C40C    20323   CCNT    20762   CM200   20321
CNTR    20104   COM00   20030   CPUDN   20611   CURMO   20316   DIC07   20501
DIC17   20502   DIST    23575   END     20770   END.    20105   FILL    20153
FIRST   20767   HALTI   20673   HALT.   20567   INIFC   20757   INIFL   20102
INIFM   20760   INTSU   20060   INTSV   20061   IO64K   20506   IOLP    20516
IOTST   20455   JJPIK   20607   JJTYP   20556   JMP4    20044   JPIK    20477
JTYPE   20456   LCNT    20761   LDA0    20271   LDAB2   20346   LDREL   20352
LOOP1   20647   LOOP2   20661   LP0     20200   LP1     20176   LP2     20175
LP3     20174   MOVE    20727   MOVLP   20744   MTEST   20646   MULTI   20532
NAXLP   20441   NWDS    20766   PCNT    20066   PIKUP   20325   PWRF    20067
R       20330   REF1    20041   REF2    20042   R.MAX   20764   R.MIN   20763
R.OFS   20765   SELF    20000   SETAD   20342   SETUP   20106   TALU    20171
TFP     20705   TISZ    20161   TJSR    20330   TMEM    20632   TOPWD   20315
TP32K   20621   TPCPU   20624   TP.VS   20716   TYPE    20301   TYPE2   20303
TYPV    20725   UPPER   20317   X       20172   .NIOP   20110   .SELF   20103
.SUBZ   20107
```

PROGRAM LISTING

MANIP

```
ASM ,@$LPT,MANIP.R
JUL  6, 1979   7:05:26

, MANIP  --  RELOCATABLE RAM MANIPULATOR AND DEBUGGER
; WRITTEN BY RENNY BOSCH
; 6-30-79

;                All Rights Reserved
;     Copyright (C) 1975, Educational Data Systems
;     Copyright (C) 1979, Educational Data Systems
;    This document contains secret and confidential
;   information of Educational Data Systems, and may
;  not be reproduced, used,  or disclosed without the
; prior written permission of Educational Data Systems


          77000 .LOC  77000           ;ACTUAL LOC. = 177000 IF 64K AVAIL.

   77000 177000 PC:   177000          ;INITIAL PROGRAM COUNTER SAVED HERE


; ON ENTRY TO EACH OF THE "COMMAND LETTER" PROCEDURES,
;      A0 = FIRST OPERAND
;      A1 = SECOND OPERAND
;      A2 = FIRST OPERAND AS AN ADDRESS (INCL. OFFSET IF ANY)
;      A3 = B = CENTRAL REFERENCE POINT


   77001  40436 MANIP:STA   0,A        ;START HERE
   77002  44436       STA   1,A+1      ;SAVE ACCUMULATORS AND CARRY
   77003  50436       STA   2,A+2
   77004  54436       STA   3,A+3
   77005 102560       SUBCL 0,0
   77006  40435       STA   0,A+4
   77007  60477       READS 0          ;READ FRONT EDGE MINI-SWITCHES
   77010  24515       LDA   1,C200
   77011 122423       SUBZ  1,0,SNC    ;ARE SWITCHES = 200 + DEV.CODE ?
   77012    566       JMP   INCMD      ;  NO, AWAIT COMMAND INPUT
   77013  30422 .P:   LDA   2,C77      ;"P" = AUTO PROGRAM LOAD FROM DISC
   77014 112033       SLS   0,2        ;OPERAND >= 77 ?
   77015    560       JMP   ABORT      ;  YES, ERROR
   77016 101015       SNZ   0,0        ;WAS AN OPERAND ENTERED ?
   77017  60477       READS 0          ;  NO, READ SWITCHES
   77020 143405       AND   2,0,SNR    ;MASK DEVICE CODE: IS IT 0 ?
   77021    554       JMP   ABORT      ;  YES, ERROR
   77022   4401       JSR   .+1        ;PICK UP INITIAL VALUE OF A3
   77023 126000       ADC   1,1        ;FORM -1
   77024 130700       NEGS  1,2        ;FORM 400
   77025  51777       STA   2,-1,3     ;WRITE IN ALL WORDS (TO
   77026 137004       ADD   1,3,SZR    ;  REMOVE VIRGIN PARITY ERRORS)
   77027    776       JMP   .-2
   77030  24406       LDA   1,NIOSX
   77031  62677       IORST            ;TURN OFF 64K MODE
   77032 107000       ADD   0,1        ;CALCULATE THE NIOS DISC INSTR.
   77033  44376       STA   1,376      ;STORE THE INSTRUCTION
   77034    376       JMP   376        ;AND JUMP TO IT

   77035     77 C77:  77
   77036  60100 NIOSX:NIOS   0
```

```
77037       5 A:     .BLK    5          ;CPU STATUS IS SAVED HERE

77044   30460 .C:    LDA     2,C4       ;"C" = CHANGE ACCUMULATOR, C
77045 112432         SGR     0,2        ;IS FIRST OPND <= 4 ?
77046    4406        JSR     .C1        ;  YES
77047    5467 .CREF: JSR     TYPE-B,3 ;TYPE "=F" AND VIRTUAL EQUIVALENT
77050   43075        "F*L+"=
77051   25400        LDA     1,OP1-B,3
77052    5443        JSR     TPADR-B,3
77053     525        JMP     INCMD

77054 117000 .C1:    ADD     0,3
77055   45770        STA     1,A-.CREF,3;SAVE 2D OPND AS NEW CPU STATUS
77056     522        JMP     INCMD

77057   24721 .A:    LDA     1,PC       ;"A" = DUMP PC AND ACCUMULATORS
77060    5443        JSR     TPADR-B,3
77061    5471        JSR     TPCLN-B,3
77062   24755        LDA     1,A
77063    5445        JSR     TPOCT-B,3
77064   24754        LDA     1,A+1
77065    5445        JSR     TPOCT-B,3
77066   24753        LDA     1,A+2
77067    5445        JSR     TPOCT-B,3
77070   24752        LDA     1,A+3
77071    5445        JSR     TPOCT-B,3
77072   24751        LDA     1,A+4      ;  AND CARRY
77073    5445 TOCTI: JSR     TPOCT-B,3
77074     504        JMP     INCMD

77075   20746 .J:    LDA     0,A+4      ;"J" = JUMP; LOAD ACCUMULATORS
77076 100440         NEGO    0,0        ;  AND CARRY
77077   20740        LDA     0,A
77100   24740        LDA     1,A+1
77101   30740        LDA     2,A+2
77102   34740        LDA     3,A+3
77103   60211        NIOC    TTO
77104    2564        JMP     @ADR1      ;JUMP TO USER PROGRAM

77105 106400 .M:     SUB     0,1        ;"M" = MOVE
77106   35406        LDA     3,ADR3-B,3
77107 102520         SUBZL   0,0        ;TENTATIVELY DELTA = 1
77110 156033         SLS     2,3        ;IS DIRECTION OF MOVE DOWNWARD ?
77111     404        JMP     .M1        ;  YES
77112 102000         ADC     0,0        ;  NO - MAKE DELTA = -1 AND
77113 133000         ADD     1,2        ;  START AT OTHER END
77114 137000         ADD     1,3
77115   25000 .M1:   LDA     1,0,2      ;FINALLY DO THE MOVE ITSELF
77116   45400        STA     1,0,3
77117 113000         ADD     0,2        ;ADD DELTA TO SOURCE & DEST. PNTRS.
77120 117000         ADD     0,3
77121   10564        ISZ     COUNT      ;DONE ?
77122     773        JMP     .M1        ;  NO
77123     455        JMP     INCMD      ;  YES
```

```
; * * * * * START OF MANIP'S "PAGE 0" (ACCESSIBLE IF A3 = B) * * * * *

   77124      4 C4:    4
   77125    200 C200:  200
   77126 177755 N.TS:  B-TSEND;NO.  TS CELLS TO BE CLEARED FOR NEW CMD
   77127 177777 DELAY: -1      ;- DELAY AFTER CR

; BRANCH.  BRANCHES TO THE DESTINATION INDICATED IN TABLE ENTRY IF THE
; RIGHT-MOST 7 (OR 5) BITS THEREOF AGREE WITH A0.  CALLING SEQUENCE:
;     JSR    BRNC7    (OR BRNC5 FOR 5-BIT, WITH A1 = 37)
;     DEST1-.-1*K+CHAR1(OR F INSTEAD OF K FOR 5-BIT)
;     DEST2-.-1*K+CHAR2
;       ...
;     END OF LIST IS INDICATED BY 7 (OR 5) LSB'S = 0

; A -1 IN THE TABLE IS USED TO DETERMINE MAX ALLOWABLE NO. OF OPERANDS

   77130  24527 BRNC7: LDA    1,C177
   77131 123400 BRNC5: AND    1,0
   77132  31400        LDA    2,0,3
   77133 175400        INC    3,3
   77134 147415        AND#   2,1,SNR   ;END OF LIST ?
   77135    526 RTN1:  JMP    RTNA3     ;  YES
   77136 150015        COM#   2,2,SNR   ;IS LIST ENTRY = -1 ?
   77137  10544        ISZ    N.OP      ;  YES: MAX. NO. OPNDS. EXCEEDED ?
   77140 112421        SUBZ   0,2,SKP   ;  NO OR YES,NO
   77141    434        JMP    ABORT     ;  YES,YES
   77142 133414        AND#   1,2,SZR   ;MATCH ?
   77143    767        JMP    BRNC5+1   ;  NO
   77144 151113        SSN    2,2       ;IS DISPLACEMENT NEGATIVE ?
   77145 125620        INCZR  1,1       ;  NO - CHANGE A1 TO 100 (OR 20)
   77146 151200        MOVR   2,2
   77147 125224        MOVZR  1,1,SZR   ;SHIFTED 7 (OR 5) PLACES ?
   77150    776        JMP    .-2       ;  NO
   77151  20513        LDA    0,OP1
   77152  24513        LDA    1,OP2
   77153 157000        ADD    2,3       ;YES - ADD TO "." IN LIST & GO THERE
   77154  30514        LDA    2,ADR1
   77155    506        JMP    RTNA3


   77156 126421 RDWD:  SUBZ   1,1,SKP   ;READ A WORD FROM TTI OR PTR
   77157 126440 RDCHA: SUBO   1,1       ;READ A CHARACTER
   77160  14504        DSZ    OP1       ;WHICH READER ?
   77161  60510        DIAS   0,TTI     ;  TTI
   77162  10502        ISZ    OP1
   77163  60512        DIAS   0,PTR     ;  PTR
   77164  63410        SKPBN  TTI
   77165  63512        SKPBZ  PTR
   77166    776        JMP    .-2
   77167 107363        ADDCS  0,1,SNC   ;FOLD IN NEW CHAR: = A WORD ?
   77170    770        JMP    RDCHA+1   ;  NOT YET, READ ANOTHER
   77171  20517        LDA    0,CKSUM   ;  YES, UPDATE THE CHECKSUM
   77172 123000        ADD    1,0
   77173  40515        STA    0,CKSUM
   77174    467        JMP    RTNA3
   77175  60210 ABORT: NIOC    TTI        ; ABORT, TYPE "\"
   77176   4555         JSR     TYPE
   77177    134         "\
   77200  20461 INCMD: LDA     0,F.CUR    ; INPUT A COMMAND
   77201  40507         STA     0,OFSET
   77202   4512         JSR     TCRLF      ; TYPE CR, LF
   77203  54504         STA     3,.TS      ; INITIALIZE OPERAND STORAGE POINTER
   77204  14503         DSZ     .TS
   77205  24721         LDA     1,N.TS
   77206 102400         SUB     0,0
   77207  41400         STA     0,0,3      ; CLEAR TEMP STORE AREA
   77210 175400         INC     3,3
   77211 125404         INC     1,1,SZR
   77212    775         JMP     .-3
   77213 101400 INCHA: INC     0,0        ; WAIT FOR INPUT
   77214  63610         SKPDN   TTI
   77215    776         JMP     .-2
   77216  60610         DIAC    0,TTI
   77217   4711         JSR     BRNC7      ; SEE IF IT'S AN ACTIVE CHARACTER

   77220  37015         .CR-.-1*K+15       ; CARRIAGE RETURN
   77221  50336         .UP-.-1*K+"^       ; UP ARROW (EXAMINE PREVIOUS)

   77222  20000 C20K:  20000              ; SERVES AS LIST TERMINATOR

   77223  30466         LDA     2,C40
   77224 112032         SGE     0,2        ; IS IT A CONTROL CHARACTER ?
   77225    750         JMP     ABORT      ;   YES, ABORT
   77226  61111         DOAS    0,TTO      ; NOT ACTIVE, ECHO IT
   77227  30463         LDA     2,C60
   77230 142400         SUB     2,0
   77231  34462         LDA     3,C7
   77232 116432         SGR     0,3        ; IS IT AN OCTAL DIGIT ?
   77233    404         JMP     OCTAL      ;   YES
   77234 143023         ADDZ    2,0,SNC    ; RECONSTITUTE CHAR.; IS IT COMMA ?
   77235  40444 INCH2: STA     0,T        ;   NO, SAVE IT
   77236    417         JMP     SOCTF

   77237  10443 OCTAL: ISZ     OCTFL      ; FIRST OCTAL DIGIT OF A NUMBER ?
   77240  10447         ISZ     .TS        ;   YES, ADVANCE PARAMETER POINTER
   77241  26446         LDA     1,@.TS     ; PROCESS OCTAL CHARACTER
   77242 125120         MOVZL   1,1        ; SHIFT PREV. NO. LEFT 3 BITS
   77243 125120         MOVZL   1,1
   77244 125120         MOVZL   1,1
   77245 107000         ADD     0,1        ; ADD NEW DIGIT TO PREV. NO.
   77246  46441         STA     1,@.TS
   77247  20441         LDA     0,OFSET
   77250 101400         INC     0,0
   77251 107000         ADD     0,1        ; CONVERT TO ADDRESS: ADD OFFSET IF ANY
   77252  30435         LDA     2,.TS
   77253  45004         STA     1,4,2      ; SAVE AS AN ADDRESS
   77254 102000         ADC     0,0
   77255  40425 SOCTF: STA     0,OCTFL    ; SET OCTAL FLAG
   77256    735         JMP     INCHA

   77257    177 C177:  177
```

```
77260 177777 F.PRE:-1       ;PREVIOUS VALUE OF "F" OFFSET-1
77261 177777 F.CUR:-1       ;CURRENT VALUE OF "F" OFFSET-1
                    ; (WE USE OFFSET-1 TO AVOID AN INITIAL VALUE OF 0
                    ;  WHICH WOULD END APL LOADING ON POINT 4 CPU)


77262  34422 RTNTS:LDA     3,TS      ;RETURN VIA TS
77263   5400 RTNA3:JSR     0,3       ;RETURN VIA A3

       77264 B=       .     ;USED AS THE CENTRAL LOCATION REFERENCE ***

77264  77377 OP1:   VAR.    ;FIRST OPERAND TYPED IN (OCTAL)

77265  77377 OP2:   VAR.    ;2D OPND. (VALUE IN C, CONTROL IN D, O)

77266  77377 OP3:   VAR.    ;THIRD OPERAND (CONSTANT IN K, N, S)

77267  77377 OP4:   VAR.    ;FOURTH OPERAND (MASK IN N, S)

77270  77377 ADR1:  VAR.    ;FIRST OPERAND, CONVERTED TO AN ADDRESS

77271  77377 ADR2:  VAR.    ;2D OPND, CONVERTED TO AN ADDRESS

77272  77377 ADR3:  VAR.    ;THIRD ADDRESS (DESTINATION IN M)

77273      6 .BLK   6       ;ADD'L BUFFER, USED IN R, V

77301  77377 T:     VAR.    ;COMMAND LETTER

77302  77377 OCTFL:VAR.     ;OCTAL FLAG, CONTROLS OPERAND COUNTING

77303  77377 N.OP:  VAR.    ;COUNTS NO. OF OPERANDS ENTERED

77304  77377 TS:    VAR.    ;GENERAL SUBROUTINE RETURN ADDRESS

77305  77377 COUNT:VAR.     ;NO. WORDS TO BE MOVED, SEARCHED, OR READ
                            ;WORD/BYTE INDICATOR IN D

77306  77377 FLAG:  VAR.    ;FLAG USED IN S/N AND R/V AND E/:

       77307 TSEND=.        ;END OF VARIABLES INITIALIZED TO 0

77307  77377 .TS:   VAR.    ;POINTER TO ABOVE TEMP. STORE (INCMD, R, V)

       77310 OFSET:         ;WORKING VALUE OF ADDRESS OFFSET
77310  77377 CKSUM:VAR.     ;CHECKSUM USED IN R, V

       77377 VAR. = 77377   ;(PREVENTS UNNECESSARY PUNCHING)



77311     40 C40:   40
77312     60 C60:   60
77313      7 C7:    7
```

; TYPE-OUT ROUTINES

```
  77314  54770 TCRLF:STA     3,TS       ;TYPE CARRIAGE RETURN, LINE FEED
  77315   4436        JSR     TYPE
  77316   5015        12*L+15
  77317  35643        LDA     3,DELAY-B,3;WAIT SPECIFIED DELAY AFTER CR, LF
  77320  20770        LDA     0,OFSET    ;(INSIDE LOOP FOR LONGER MAX. DELAY)
  77321 175404        INC     3,3,SZR
  77322    776        JMP     .-2
  77323 101404        INC     0,0,SZR    ;IS THERE AN OFFSET ?
  77324  20451        LDA     0,C.F      ;  YES, TYPE AN F, SPACE
  77325   4431        JSR     TP2CH
  77326    734        JMP     RTNTS

  77327  20761 TPADR:LDA     0,OFSET    ;TYPE A1 AS AN ADDRESS
  77330 106000        ADC     0,1
  77331 152421 TPOCT:SUBZ    2,2,SKP    ;SUPPRESS LEADING ZEROES
  77332  30757 TPOCL:LDA     2,C40      ;TYPE SPACES FOR LEADING ZEROES
  77333  20756        LDA     0,C40      ;TYPE ONE INITIAL SPACE
  77334  54750        STA     3,TS       ;TYPE THE OCTAL NO. IN A1, AFTER
  77335   4422        JSR     TPCHA      ;  TYPING THE CHARACTER IN A0
  77336 102620 TPA01:SUBZR   0,0        ;PREPARE TO MOVE MSB OF A1 INTO A0
  77337 101041        MOVO    0,0,SKP    ;SET CARRY TO FORM "PUSHER" BIT
  77340  20662 TPNXT:LDA     0,C20K     ;LEFT-SHIFT ONE DIGIT FROM A1 INTO A0
  77341 125105        MOVL    1,1,SNR    ;INITIALLY INSERTS "PUSHER" BIT
  77342    720        JMP     RTNTS      ;EXIT WHEN "PUSHER" BIT IS GONE
  77343 101103        MOVL    0,0,SNC
  77344    775        JMP     .-3
  77345 101015        SNZ     0,0        ;NON-ZERO DIGIT ...
  77346 125135        MOVZL#  1,1,SNR    ;  ... OR LAST DIGIT ?
  77347  30743        LDA     2,C60      ;  YES: ADDEND FOR ASCII DIGIT
  77350 143040        ADDO    2,0
  77351   4406        JSR     TPCHA
  77352    766        JMP     TPNXT

  77353  21400 TYPE: LDA     0,0,3      ;TYPE THE CHAR.(S) FOLL. THE JSR
  77354 175401        INC     3,3,SKP
  77355  20421 TPCLN:LDA     0,COLON    ;TYPE COLON
  77356 101020 TP2CH:MOVZ    0,0        ;TYPE 2 CHARACTERS IN A0
  77357  54533 TPCHA:STA     3,RTNTP    ;TYPE 1 CHAR. IN A0 IF C=1, 2 IF C=0
  77360  63710        SKPDZ   TTI        ;(PRESERVES A1,A2)
  77361    614        JMP     ABORT
  77362  63511        SKPBZ   TTO
  77363    775        JMP     .-3
  77364  34530        LDA     3,C377
  77365 117414        AND#    0,3,SZR    ;NULL CHARACTER ?
  77366  61111        DOAS    0,TTO
  77367 101362 TPCH2:MOVCS   0,0,SZC    ;SECOND CHAR. TO BE TYPED ?
  77370    770        JMP     TPCHA+1    ;  YES
  77371  63511        SKPBZ   TTO        ;NO, WAIT FOR OUTPUT TO FINISH
  77372    777        JMP     .-1
  77373  34517        LDA     3,RTNTP
  77374    667        JMP     RTNA3

  77375  20106 C.F:   " *L+"F
  77376     72 COLON:":
```

```
; CONVERT ADDRESSING MODE, INCL. VIRTUAL (USING "F" OFFSET) AND/OR
; BYTE (LOWER OR UPPER CORE OR VIRTUAL) - USED IN D,I,J,O

  77377 125015 CNVRT:SNZ     1,1        ;BYTE ADDRESSING MODE ?
  77400    405        JMP     CNV2       ;  NO
  77401 131225        MOVZR   1,2,SNR    ;PUT LSB IN C; WAS OP2 = 1 ?
  77402 124020        COMZ    1,1        ;  YES, SET TO -1 & SET C = 0
  77403 101200        MOVR    0,0        ;CONVERT BYTE TO WORD ADDR., USING C
  77404 125103        MOVL    1,1,SNC    ;SAVE C IN LSB OF A1
  77405  30676 CNV2:  LDA     2,N.OP
  77406 151414        INC#    2,2,SZR    ;MAX. NO. OPERANDS ?
  77407  30701        LDA     2,OFSET    ;  NO, USE OFFSET
  77410  50700        STA     2,OFSET
  77411 151400        INC     2,2
  77412 113000        ADD     0,2        ;NOW CALC. DESIRED ADDRESS
  77413  50655        STA     2,ADR1
  77414    647        JMP     RTNA3


  77415 122000 .CR:   ADC     1,0        ;PROCESS CARRIAGE RETURN
  77416  41421        STA     0,COUNT-B,3
  77417  20670        LDA     0,.TS
  77420  25640        LDA     1,C4-B,3
  77421 122400        SUB     1,0
  77422 162422        SUBZ    3,0,SZC    ;> 4 OPERANDS ENTERED ?
  77423   1711        JMP     ABORT-B,3;  YES, ERROR
  77424  40657        STA     0,N.OP     ;NO. OF OPERANDS - 5
  77425  20654        LDA     0,T        ;BRANCH ON INITIAL LETTER
  77426  24465        LDA     1,C37
  77427   5645        JSR     BRNC5-B,3;COMMAND LETTER BRANCH TABLE

  77430   3256        .N-.-1*F+"N-H
  77431   3163        .S-.-1*F+"S-H
  77432 177777            -1         ;MAX 3 OPERANDS HEREAFTER
  77433  11713        .K-.-1*F+"K-H
  77434 162415        .M-.-1*F+"M-H
  77435 177777            -1         ;MAX 2 OPERANDS HEREAFTER
  77436 160243        .C-.-1*F+"C-H
  77437   4204        .D-.-1*F+"D-H
  77440   1405        .E-.-1*F+"E-H
  77441   6146        .F-.-1*F+"F-H
  77442   6751        .I-.-1*F+"I-H
  77443  10417        .O-.-1*F+"O-H
  77444  15130        .X-.-1*F+"X-H
  77445    472        .CLN-.-1*F+":-F
  77446 177777            -1         ;MAX 1 OPERAND HEREAFTER
  77447 160341        .A-.-1*F+"A-H
  77450 161212        .J-.-1*F+"J-H
  77451 156060        .P-.-1*F+"P-H
  77452  11262        .R-.-1*F+"R-H
  77453  11266        .V-.-1*F+"V-H
  77454   5331        .Y-.-1*F+"Y-H

  77455 177400 C377L:177400             ;SERVES AS LIST TERMINATOR
  77456   1711        JMP     ABORT-B,3

; * * * * * * * * * * END OF MANIP'S "PAGE ZERO" * * * * * * * * * * *
```

```
77457  11417 .CLN:  ISZ    N.OP-B,3 ;INPUT = COLON:  TWO OPERANDS ?
77460    423        JMP    .CLN1    ;  NO, DISPLAY CONTENT
77461  45000        STA    1,0,2    ;  YES, STORE OP2 AT ADR1
77462 102521 .CLN2: SUBZL  0,0,SKP  ;<< FROM .CLN1
77463 102000 .UP:   ADC    0,0      ;"^" = EXAMINE PREVIOUS ADDRESS
77464  25400        LDA    1,OP1-B,3
77465 107000        ADD    0,1
77466  45400        STA    1,OP1-B,3
77467  31404        LDA    2,ADR1-B,3
77470 113001        ADD    0,2,SKP
77471   5513 .E:    JSR    CNVRT-B,3;"E" = PREPARE TO ENTER DATA
77472  51404        STA    2,ADR1-B,3
77473   5430 NXTL:  JSR    TCRLF-B,3
77474  25404        LDA    1,ADR1-B,3
77475   5443        JSR    TPADR-B,3
77476  45422        STA    1,FLAG-B,3;SET "EXAMINE" FLAG = 0
77477  45401 .CLN3: STA    1,OP2-B,3;PREPARE FOR OCTAL INPUT --> OP2
77500  55423        STA    3,.TS-B,3;(PRETEND ONE OPERAND HAS COME IN)
77501   5471        JSR    TPCLN-B,3;TYPE A COLON
77502   1751        JMP    INCH2-B,3;COUNT AS ONE OPERAND, SET T = ":"

77503  21422 .CLN1: LDA    0,FLAG-B,3;2D OPERAND NOT TYPED IN
77504 101014        SKZ    0,0      ;HAVE WE ALREADY EXAMINED IT ?
77505    755        JMP    .CLN2    ;  YES, GO TO NEXT LINE
77506  11422        ISZ    FLAG-B,3 ;  NO
77507  25000        LDA    1,0,2
77510   5446        JSR    TPOCL-B,3;TYPE THE VALUE AT ADR1
77511    766        JMP    .CLN3    ;TYPE A COLON & WAIT FOR INPUT


77512  77377 RTNTP: VAR.            ;RETURN ADDRESS USED BY TPCHA
77513     37 C37:   37
77514    377 C377:  377


77515  15422 .S:    DSZ    FLAG-B,3 ;"S" = SEARCH FOR GIVEN VALUE
77516  31404 .N:    LDA    2,ADR1-B,3;"N" = SEARCH FOR NOT-EQUAL
77517  21000        LDA    0,0,2
77520  25403        LDA    1,OP4-B,3;4TH OPERAND (IF ANY) IS THE MASK
77521 125014        SKZ    1,1
77522 123400        AND    1,0
77523  25402        LDA    1,OP3-B,3;THIRD OPERAND IS THE COMPAREE
77524 122404        SUB    1,0,SZR  ;A0 = 0 MEANS EQUAL
77525 102520        SUBZL  0,0      ;A0 = 1 MEANS NOT EQUAL
77526  31422        LDA    2,FLAG-B,3
77527 113213        ADDR#  0,2,SNC  ;HIT (IE. = IF S OR <> IF N) ?
77530    407        JMP    NXTSN    ;  NO
77531   5430        JSR    TCRLF-B,3;NEXT LINE - TYPE CR, LF
77532  25404        LDA    1,ADR1-B,3
77533   5443        JSR    TPADR-B,3;TYPE THE ADDRESS
77534   5471        JSR    TPCLN-B,3;TYPE A COLON
77535  27404        LDA    1,@ADR1-B,3;   NO
77536   5446        JSR    TPOCL-B,3;   NO, TYPE VALUE IN OCTAL FORM
77537  11404 NXTSN: ISZ    ADR1-B,3
77540    401        JMP    .+1
77541  11421        ISZ    COUNT-B,3
77542    754        JMP    .N
77543   1714        JMP    INCMD-B,3
```

```
77544   5513 .D:    JSR    CNVRT-B,3;"D" = DUMP OCTAL - WORDS OR BYTES
77545 125224        MOVZR  1,1,SZR  ;WORD MODE ?
77546 126100        ADCL   1,1      ;NO, FORM BYTE COUNT = -1 OR -2
77547  45421        STA    1,COUNT-B,3
77550   5430 DLINE: JSR    TCRLF-B,3;TYPE CR
77551  25400        LDA    1,.OP1-B,3
77552   5445        JSR    TPOCT-B,3;TYPE THE (WORD OR BYTE) ADDRESS
77553   5471        JSR    TPCLN-B,3;TYPE A COLON
77554  11400 DLIN2: ISZ    OP1-B,3
77555    401        JMP    .+1
77556  27404        LDA    1,@ADR1-B,3;FETCH NEXT WORD OR BYTE
77557  21421        LDA    0,COUNT-B,3
77560 101015        SNZ    0,0      ;WORD MODE ?
77561    411        JMP    DWORD    ;  YES
77562 101414        INC#   0,0,SZR  ;LEFT BYTE ?
77563 125300        MOVS   1,1      ;  YES
77564  20730        LDA    0,C377
77565 107400        AND    0,1
77566   5446        JSR    TPOCL-B,3;TYPE THE BYTE IN OCTAL FORM
77567  11421        ISZ    COUNT-B,3;WAS IT THE SECOND BYTE ?
77570    764        JMP    DLIN2    ;  NO
77571 126121        ADCZL  1,1,SKP  ;PREPARE BYTE COUNT = -2
77572   5446 DWORD: JSR    TPOCL-B,3;TYPE THE WORD IN OCTAL
77573  45421        STA    1,COUNT-B,3
77574  11404 DNEXT: ISZ    ADR1-B,3
77575    401        JMP    .+1
77576  25400        LDA    1,OP1-B,3
77577  21427        LDA    0,C7-B,3
77600 107414        AND#   0,1,SZR  ;IS LSD OF ADDRESS = 7 ?
77601    753        JMP    DLIN2    ;  NO, TYPE ANOTHER NO.
77602    746        JMP    DLINE    ;  YES, TYPE NEW LINE


77603  41643 .Y:    STA    0,DELAY-B,3;"Y" = SET RETURN DELAY
77604   1714        JMP    INCMD-B,3


77605 122000 .F:    ADC    1,0      ;"F" = ESTABLISH ADDRESS OFFSET
77606 105000        MOV    0,1
77607  21775        LDA    0,F.CUR-B,3
77610  31417        LDA    2,N.OP-B,3
77611 150234        COMZR# 2,2,SZR  ;ANY OPERANDS ENTERED ?
77612  25774        LDA    1,F.PRE-B,3;  NO, SWAP WITH PREVIOUS OFFSET
77613 106414        SEQ    0,1      ;IS NEW VALUE DIFF. FROM CURRENT ?
77614  41774        STA    0,F.PRE-B,3;  YES, SAVE CURRENT VALUE
77615  45775        STA    1,F.CUR-B,3
77616   5471        JSR    TPCLN-B,3
77617 125400        INC    1,1
77620   1607        JMP    TOCTI-B,3;TYPE THE NEW OFFSET VALUE
```

```
77621     233 C.ESC: 233      ;ESCAPE CODE RECOGNIZED BY I

77622    5513 .I:     JSR     CNVRT-B,3;"I" = INPUT ASCII
77623    5471         JSR     TPCLN-B,3
77624 125200          MOVR    1,1        ;PUT BYTE INDICATOR IN C
77625   21000         LDA     0,0,2
77626   25571         LDA     1,C377L-B,3
77627 123703          ANDS    1,0,SNC    ;START WITH RIGHT BYTE ?
77630 102440 .I2:     SUBO    0,0        ;   NO, GET READY FOR 2 CHARACTERS
77631 105360          MOVCS   0,1        ;   YES, SAVE FIRST CHARACTER
77632   63610         SKPDN   TTI
77633     777         JMP     .-1
77634   60610         DIAC    0,TTI      ;INPUT A CHARACTER
77635   45000         STA     1,0,2      ;TENTATIVELY STORE TERMINATOR
77636   25641         LDA     1,C200-B,3
77637 107415          AND#    0,1,SNR    ;IS MSB ALREADY SET ?
77640 123000          ADD     1,0        ;   NO, THEN SET IT
77641   24760         LDA     1,C.ESC
77642 106415          SNE     0,1        ;WAS INPUT ESCAPE ?
77643    1714         JMP     INCMD-B,3;   YES - END INPUT
77644   61111         DOAS    0,TTO      ;ECHO IT
77645 101012          MOV#    0,0,SZC    ;IS THIS THE 1ST CHAR. OF A PAIR ?
77646     763         JMP     .I2+1      ;   YES
77647   25000         LDA     1,0,2
77650 107000          ADD     0,1
77651   45000         STA     1,0,2      ;   NO - STORE TWO CHAR.S
77652 151400          INC     2,2
77653     755         JMP     .I2


77654    5513 .O:     JSR     CNVRT-B,3;"O" = OUTPUT ASCII
77655    5471         JSR     TPCLN-B,3
77656 125200          MOVR    1,1        ;PUT INITIAL BYTE INDICATOR IN C
77657   24635 .O2:    LDA     1,C377
77660   21000         LDA     0,0,2
77661 101003          MOV     0,0,SNC    ;INITIAL PASS TYPE 1 BYTE ?
77662 101300          MOVS    0,0        ;   NO
77663 107405          AND     0,1,SNR    ;IS FIRST BYTE = 0 ?
77664    1714         JMP     INCMD-B,3;   YES
77665    5473         JSR     TPCHA-B,3;   NO, TYPE 1 OR 2 BYTES
77666 106415          SNE     0,1        ;WAS SECOND BYTE = 0 ?
77667    1714         JMP     INCMD-B,3;   YES
77670 151400          INC     2,2        ;   NO
77671     766         JMP     .O2


77672   21402 .K:     LDA     0,OP3-B,3;"K" = ENTER A CONSTANT IN CORE
77673   41000         STA     0,0,2
77674 151400          INC     2,2
77675   11421         ISZ     COUNT-B,3
77676     775         JMP     .-3
77677    1714         JMP     INCMD-B,3
```

```
; READ OR VERIFY PAPER TAPE

   77700  11422 .R:    ISZ    FLAG-B,3 ;"R" = READ
   77701   5673 .V:    JSR    RDCHA-B,3;"V" = VERIFY
   77702   5673        JSR    RDCHA-B,3; << FROM RPEAT
   77703 125025        MOVZ   1,1,SNR  ;IN LEADER ?
   77704    776        JMP    .-2      ;  YES
   77705   5674        JSR    RDCHA+1-B,3;READ 2D HALF OF FIRST WORD
   77706  45421        STA    1,COUNT-B,3;FIRST WORD = - COUNT
   77707  45424        STA    1,CKSUM-B,3;INITIALIZE THE CHECKSUM
   77710   5672        JSR    RDWD-B,3
   77711  44466        STA    1,ADDRS  ;SECOND WORD = ADDRESS
   77712   5672        JSR    RDWD-B,3
   77713  31421        LDA    2,COUNT-B,3
   77714 151113        SSN    2,2      ;SPECIAL BLOCK ?
   77715    436        JMP    SPBLK    ;  YES
   77716  55423        STA    3,.TS-B,3
   77717  11423 RDTMP: ISZ    .TS-B,3  ;PROTECT OP1
   77720   5672        JSR    RDWD-B,3 ;READ A BLOCK INTO TEMP. STORAGE
   77721  47423        STA    1,@.TS-B,3
   77722  24444        LDA    1,CM20
   77723 146032        SGE    2,1      ;REPEAT BLOCK (COUNT > 20) ?
   77724    421        JMP    RPEAT    ;  YES
   77725 151404        INC    2,2,SZR
   77726    771        JMP    RDTMP
   77727  55423        STA    3,.TS-B,3
   77730 151015 STORE: SNZ    2,2      ;IS THIS A REPEAT BLOCK ?
   77731  11423        ISZ    .TS-B,3  ;  NO
   77732  21424        LDA    0,CKSUM-B,3;STORE THE BLOCK IF AOK
   77733 101014        SKZ    0,0      ;CHECKSUM ERROR ?
   77734    426        JMP    ABORV    ;  YES
   77735  27423        LDA    1,@.TS-B,3;  NO
   77736  11422        ISZ    FLAG-B,3 ;VERIFY OR READ ?
   77737  15422        DSZ    FLAG-B,3
   77740  46437        STA    1,@ADDRS ;  READ
   77741  22436        LDA    0,@ADDRS
   77742 106414        SEQ    0,1      ;VERIFY ERROR ?
   77743    417        JMP    ABORV    ;  YES
   77744  10433        ISZ    ADDRS
   77745  11421 RPEAT: ISZ    COUNT-B,3;ALL DONE ?
   77746    762        JMP    STORE    ;  NO
   77747  21400        LDA    0,OP1-B,3;  YES
   77750 101015        SNZ    0,0      ;ARE WE USING TTY ?
   77751  61111        DOAS   0,TTO    ;  YES - GIVE A HICKUP
   77752    730        JMP    .V+1     ;READ NEXT BLOCK

   77753 151235 SPBLK: MOVZR# 2,2,SNR  ;SPECIAL BLOCK: WORD COUNT > 1 ?
   77754 101014        SKZ    0,0      ;OR CHECKSUM ERROR ?
   77755    405        JMP    ABORV    ;  YES
   77756  30421        LDA    2,ADDRS
   77757 151112        SSP    2,2      ;STARTING ADDRESS ?
   77760   1714        JMP    INCMD-B,3;  NO, RETURN TO MANIP
   77761   1000        JMP    0,2      ;  YES, JUMP THERE
```

```
77762   5467 ABORV: JSR     TYPE-B,3 ;ABORT IN READ OR VERIFY
77763  56007        "\*L+7           ;BELL, BACKSLASH
77764  24413        LDA     1,ADDRS
77765   1607        JMP     TOCTI-B,3;TYPE THE OFFENDING ADDRESS

77766 177760 CM20:  -20


77767 126440 .X:    SUBO    1,1       ;"X" = CALCULATE CHECKSUM
77770  21000        LDA     0,0,2
77771 106500        SUBL    0,1       ;FORM NEGATIVE ROTATING CHECKSUM
77772 151400        INC     2,2
77773  11421        ISZ     COUNT-B,3;DONE ?
77774    774        JMP     .-4       ;   NO
77775 125200        MOVR    1,1
77776   1607        JMP     TOCTI-B,3;   YES, TYPE IT OUT


77777      0 .BLK   77777-.

77777      0 ADDRS: 0        ;BLOCK ADDRESS, AND APL TERMINATOR



            40 F=     40
           100 H=     100
           200 K=     200
           400 L=     400




                    .END     ;MANIP
```

```
---

A        77037  ABORT    77175  ABORV    77762  ADDRS    77777  ADR1     77270
ADR2     77271  ADR3     77272  B        77264  BRNC5    77131  BRNC7    77130
C177     77257  C200     77125  C20K     77222  C37      77513  C377     77514
C377L    77455  C4       77124  C40      77311  C60      77312  C7       77313
C77      77035  CKSUM    77310  CM20     77766  CNV2     77405  CNVRT    77377
COLON    77376  COUNT    77305  C.ESC    77621  C.F      77375  DELAY    77127
DLIN2    77554  DLINE    77550  DNEXT    77574  DWORD    77572  F           40
FLAG     77306  F.CUR    77261  F.PRE    77260  H          100  INCH2    77235
INCHA    77213  INCMD    77200  K          200  L          400  MANIP    77001
NIOSX    77036  NXTL     77473  NXTSN    77537  N.OP     77303  N.TS     77126
OCTAL    77237  OCTFL    77302  OFSET    77310  OP1      77264  OP2      77265
OP3      77266  OP4      77267  PC       77000  RDCHA    77157  RDTMP    77717
RDWD     77156  RPEAT    77745  RTN1     77135  RTNA3    77263  RTNTP    77512
RTNTS    77262  SOCTF    77255  SPBLK    77753  STORE    77730  T        77301
TCRLF    77314  TOCTI    77073  TP2CH    77356  TPA01    77336  TPADR    77327
TPCH2    77367  TPCHA    77357  TPCLN    77355  TPNXT    77340  TPOCL    77332
TPOCT    77331  TS       77304  TSEND    77307  TYPE     77353  VAR.     77377
.A       77057  .C       77044  .C1      77054  .CLN     77457  .CLN1    77503
.CLN2    77462  .CLN3    77477  .CR      77415  .CREF    77047  .D       77544
.E       77471  .F       77605  .I       77622  .I2      77630  .J       77075
.K       77672  .M       77105  .M1      77115  .N       77516  .O       77654
.O2      77657  .P       77013  .R       77700  .S       77515  .TS      77307
.UP      77463  .V       77701  .X       77767  .Y       77603
```