

DIGITAL RESEARCH

Post Office Box 579, Pacific Grove, California 93950, (408) 373-3403

DYNAMIC DEBUGGING TOOL (DDT)

CP/M VERSION _____

COPYRIGHT © 1976

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA. 93950

SER. # _____

DEMMOV

CP/M VERSION
(CONTROL DISKETTE)

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

1>
2>
3> 000E = VERSION EQU 14
4> ; DDT RELOCATOR PROGRAM, INCLUDED WITH THE MODULE TO PERFORM
5> ; THE MOVE FROM 200H TO THE DESTINATION ADDRESS
6> 0703 = BIAS EQU 703H ;DISTANCE UP TO DEMON
7> 0100 ORG 100H
8> 0200 = STACK EQU 200H
9> 0005 = BDOS EQU 0005H
10> 0009 = PRHT EQU 9 ;BDOS PRINT FUNCTION
11> 0200 = MODULE EQU 200H ;MODULE ADDRESS
12>
13> 0100 010000 LXI B, 0 ;ADDRESS FIELD FILLED-IN WHEN MODULE BUILT
14> 0103 C33D01 JMP START
15> 0106 434F505952 DB 'COPYRIGHT (C) 1976, DIGITAL RESEARCH'
16> 0130 4444542056SIGNOH, DB 'DDT VERS '
17> 0139 312E DB VERSION/10+'0', ''
18> 013B 3424 DB VERSION MOD 10 + '0', ''\$'
19> 013D 310002 START, LXI SP,STACK
20> 0140 C5 PUSH B
21> 0141 C5 PUSH B
22> 0142 113001 LXI D.SIGHON
23> 0145 0E09 MVI C,PRNT
24> 0147 CD0500 CALL BDOS
25> 014A C1 POP B ;RECOVER LENGTH OF MOVE
26> 014B 210700 LXI H,BDOS+2;ADDRESS FIELD OF JUMP TO BDOS (TOP MEMORY)
27> 014E 7E MOV A,M ;A HAS HIGH ORDER ADDRESS OF MEMORY TOP
28> 014F 3D DCR A ;PAGE DIRECTLY BELOW BDOS
29> 0150 98 SUB B ;B HAS HIGH ORDER ADDRESS OF RELOC AREA
30> 0151 57 MOV D,A
31> 0152 1E00 MVI E,0 ;D,E ADDRESSES BASE OF RELOC AREA
32> 0154 D5 PUSH D ;SAVE FOR RELOCATION BELOW
33>
34> 0155 210002 MOVE, LXI H,MODULE;READY FOR THE MOVE
35> 0158 78 MOV A,B ;BC=0?
36> 0159 B1 ORA C
37> 015A CA6501 JZ RELOC
38> 015D 0B DCX B ;COUNT MODULE SIZE DOWN TO ZERO
39> 015E 7E MOV A,M ;GET NEXT ABSOLUTE LOCATION
40> 015F 12 STAK D ;PLACE IT INTO THE RELOC AREA
41> 0160 13 INX D
42> 0161 23 INX H
43> 0162 C35801 JMP MOVE
44>
45> RELOC, ;STORAGE MOVED, READY FOR RELOCATION
46> ;HL ADDRESSES BEGINNING OF THE BIT MAP FOR RELOCATION
47> 0165 D1 POP D ;RECALL BASE OF RELOCATION AREA
48> 0166 C1 POP B ;RECALL MODULE LENGTH
49> 0167 E5 PUSH H ;SAVE BIT MAP BASE IN STACK
50> 0168 62 MOV H,D ;RELOCATION BIAS IS IN D
51>
52> 0169 78 RELW, MUV H,B ;BL=DY
53> 016A B1 ORA C
54> 016B CA8701 JZ ENDREL
55>
56>
57> 016E 0B NOT END OF THE RELOCATION, MAY BE INTO NEXT BYTE OF BIT MAP
58> 016F 7B DCX B ;COUNT LENGTH DOWN
59> 0170 E607 MOV A,E
60> 0172 C27A81 ANI 111B ;0 CAUSES FETCH OF NEXT BYTE
JNZ REL1
61> ;FETCH BIT MAP FROM STACKED ADDRESS
62> 0175 E3 XTHL
63> 0176 7E MOV A,M ;NEXT 8 BITS OF MAP
64> 0177 23 INX H
65> 0178 E3 XTHL ;BASE ADDRESS GOES BACK TO STACK
66> 0179 6F MOV L,A ;L HOLDS THE MAP AS WE PROCESS 8 LOCATION
67> 017A 7D REL1: MOV A,L ;JCY SET TO 1 IF RELOCATION NECESSARY
68> 017B 17 RAL ;BACK TO L FOR NEXT TIME AROUND
69> 017C 6F MOV L,A ;BACK TO L FOR NEXT TIME AROUND
JNC REL2 ;SKIP RELOCATION IF CY=0
70> 017D D28301
71>
72>
73> 0180 1A CURRENT ADDRESS REQUIRES RELOCATION
74> 0181 84 LDAX D
ADD H ;APPLY BIAS IN H
75> 0182 12 STAX D
76> 0183 13 REL2: INX D ;TO NEXT ADDRESS
77> 0184 C36901 JMP RELB ;FOR ANOTHER BYTE TO RELOCATE
78>
79>
80> 0187 D1 ENDREL: ;END OF RELOCATION
POP D ;CLEAR STACKED ADDRESS
81> 0188 2E00 MVI L,B
82> 018A 110307 LXI D,BIAS
83> 018D 19 DAD D
84> 018E E9 PCHL ;GO TO RELOCATED PROGRAM
85> 018F END

CP/M VERSION
COPYRIGHT © 1976
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

```

2>
3> 0100      ORG    100H
4> 1206 =    BDOS    EQU    $+1106H
5>          ; CP/M DEBUGGER DISASSEMBLER/ASSEMBLER MODULE
6>          ; RELOCATABLE VERSION
7>
8>          ; COPYRIGHT (C) 1976
9>          ; DIGITAL RESEARCH
10>         ; BOX 579 PACIFIC GROVE
11>         ; CALIFORNIA 93950
12>
13>          ; CP/M VERSION _____
14> 0000 =    FALSE   EQU    0
15> FFFF =    TRUE    EQU    NOT FALSE
16>
17> 0000 =    DEMON   EQU    $+700H
18>
19> 0005 =    JLOC1   EQU    5H
20>
21> 0009 =    GETBUFF EQU    DEMON+9H
22> 000C =    GHC     EQU    DEMON+0CH
23>
24> 000C =    CI      EQU    GHC      ;SYNONYM FOR GHC
25> 0002 =    CONO    EQU    2      ;CONSOLE OUTPUT FUNCTION
26> 000B =    CHKIO   EQU    11     ;CHECK FOR CHARACTER INPUT DURING LIST
27>
28> 000D =    CR      EQU    0DH
29> 000A =    LF      EQU    0AH
30> 0009 =    TAB     EQU    09H
31>
32> MODLOC: ;MODULE LOCATION
33> 0100 C30008 JMP    DEMON   ;BRANCH AROUND DISASSEMBLER IF USED
34> 0103 C38D01 JMP    INIT    ;INITIALIZE JUMPS
35> 0106 C38604 JMP    DISENT
36> 0109 C34006 JMP    ASMEN   ;ENTRY POINT FOR ASSEMBLER
37> 010C PC:    DS     2      ;CURRENT FADED PC DURING DISASSEMBLY
38> 010E MPC:   DS     2      ;MAX VALUE FOR PC (STOP ADDRESS)
39> 0110 PAGM:  DS     1      ;PAGE MODE IF NON ZERO
40> 0111 TPC:   DS     2      ;TEMPORARY PC FOR ASSEMBLER RESTORE ON ERROR
41> 0113 OLDSP: DS     2      ;ENTRY SP VALUE
42>
43>
44>
45> CO:    ;PRINT CHARACTER IN REGISTER C
46> 0115 F5 PUSH   PSW
47> 0116 C5 PUSH   B
48> 0117 D5 PUSH   D
49> 0118 E5 PUSH   H
50> 0119 59 MOV    E,C      ;CHARACTER MOVES TO E
51> 011A 0E02 MVI    C,CONO   ;OUTPUT FUNCTION
52> 011C CD0612 CALL   BDOS
53> 011F E1 POP    H
54> 0120 D1 POP    D
55> 0121 C1 POP    B
56> 0122 F1 POP    PSW
57> 0123 C9 RET
58>
59>
60> DELIM: ;CHECK FOR DELIMITER

```

CP/M VERSION _____
 COPYRIGHT © 1976
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # **DISASSEMBLER/ASSEM**

```

61> 0124 FE20 CPI    '
62> 0126 CB RZ
63> 0127 FE09 CPI    TAB
64> 0129 CB RZ
65> 012A FE2C CPI    '
66> 012C CB RZ
67> 012D FE0D CPI    CR
68> 012F CB RZ
69> 0130 FE7F CPI    7FH
70> 0132 CA4006 JZ    ASHEN   ;RESTART CURRENT LINE
71> 0135 C9 RET
72>
73>          ;CRLF: ;RETURN AND LINE FEED
74> 0136 0E0D MVN   C,CR
75> 0138 CD1501 CALL   CO
76> 0138 0E0A MVN   C,LF
77> 013D CD1501 CALL   CO
78> 0140 C9 RET
79>
80>          ;SCAN: ;FILL OPCODE WITH CHARACTERS
81>          ;
82> 0141 CD0C08 SC1,  ;ENTER HERE IF CHARACTER SCANNED
83>          ;SCAN#:
84> 0144 FE0D CPI
85> 0146 CA2E06 JZ    ERR
86> 0149 CD2401 CALL   DELIM
87> 014C CA4101 JZ    SC1
88>
89>          ;CLEAR BUFFER
90> 014F 0E04 MVN   C,4
91> 0151 218F07 LXI   H,OPCODE
92> 0154 3620 SC0,  ;MVI   M,' '
93> 0156 23 INX   H
94> 0157 0D DCR   C
95> 0158 C25401 JNZ   SC0
96>
97>          ;GARBAGE REMOVED AT BEGINNING OF SCAN
98> 015B 0E05 MVN   C,5
99> 015D 218F07 LXI   H,OPCODE
100> 0160 77 SC2,  ;MOV   M,A ;STORE CHARACTER
101> 0161 CD0C08 CALL   CI
102> 0164 CD2401 CALL   DELIM
103> 0167 CA7201 JZ    SC3
104> 016A 23 INX   H
105> 016B 0D DCR   C
106> 016C CA2E06 JZ    ERR   ;TOO LONG
107> 016F C36001 JMP   SC2
108>
109>          ;SC3: ;END OF CURRENT SCAN, COMPARE FOR EMPTY
110> 0172 3A8F07 LDA   OPCODE
111> 0175 FE20 CPI
112> 0177 C9 RET
113>
114>          ;HEX: ;CONVERT ACCUMULATOR TO HEXADECIMAL
115> 0178 D630 SUI   '0'
116> 017A FE0A CPI   10
117> 017C D8 RC    ;'0' - '9'
118> 017D C6F9 ADI   ('0'-'A'+10) AND 0FFH
119> 017F FE10 CPI   16
120> 0181 D8 RC

```

CP/M VERSION _____
 COPYRIGHT © 1976
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. #

```

121> 0182 C32E86      JMP    ERR
122> ;GADDR, ;GET ADDRESS VALUE TO B (HIGH ORDER) AND C (LOW) WITH COPY OF
123> 0185 CD4101      CALL   SCAN
124> 0188 CA2E06      JZ    ERR
125> 0188 110000      LXI   D, 0
126> 018E 010000      LXI   B, 0 ;OPCODE INDEX
127> 0191 218F87      CAB,  LXI   H, OPCODE
128> 0194 09          DAD   B
129> 0195 7E          MOV   A, M ;NEXT CHARACTER
130> 0196 FE20          CPI   '
131> 0199 CAB001      JZ    GHI
132> 0198 CD7801      CALL   HEX ;CONVERT ACCUMULATOR
133> 019E 6B          MOV   L, E
134> 01A0 62          MOV   H, D ;COPY D,E TO H,L
135> 01A0 29          DAD   H ;+2
136> 01A1 29          DAD   H ;+4
137> 01A2 29          DAD   H ;+8
138> 01A3 29          DAD   H ;+16
139> 01A4 5F          MOV   E, A
140> 01A5 1600        MVI   D, 0
141> 01A7 19          DAD   D ;+CHAR
142> 01A8 EB          XCHG
143> 01A9 03          INX   B ;BACK TO D,E
144> 01AA 79          MOV   A, C ;NEXT POSITION
145> 01AB FE04          CPI   4
146> 01AD C29101      JNZ   GAB ;FOR ANOTHER
147>
148>
149> GAI, ;D,E CONTAINS RESULT
150> 01B0 42          MOV   B, D
151> 01B1 4B          MOV   C, E
152> 01B2 78          MOV   A, E ;COPY OF LOW BYTE TO ACCUMULATOR
153> 01B3 05          DCR   B
154> 01B4 04          IHR   B ;SETS ZERO FLAG IF B IS ZERO
155> 01B5 C9          RET
156>
157> GBYTE, ;GET BYTE VALUE TO ACCUMULATOR AND C, CHECK FOR HIGH ORDER ZE
158> 01B6 CD6501      CALL   GADDR
159> 01B9 C22E86      JNZ   ERR
160> 01BC C9          RET
161>
162> INIT, ;INITIALIZE THE JUMPS AROUND DISASSEMBLER
163> 01BD 210600      LXI   H, JLOC1+1 ;ADDRESS FIELD OF JUMP AT SH
164> 01C0 3E20          MVI   A, MODLOC AND 0FFH ;LOW ORDER ADDRESS OF THIS MO
165> 01C2 56          SUB   H ;ALREADY ADDRESSING THIS MODULE
166> 01C3 23          INX   H ;FOR A MODULE BELOW THIS POINT?
167> 01C4 3E01          MVI   A, MODLOC SHR 8 ;IF SO, SKIP THIS OPERATION
168> 01C6 9E          SBB   M
169> 01C7 D8          RNC   ;NO CARRY IF <5H> <= MODLOC
170> 01C8 210001      LXI   H, MODLOC ;CHANGE ADDRESS
171> 01CB 220600      SHLD  JLOC1+1 ;FIELD OF JUMP AT SH
172> 01CE C9          RET
173>
174> **** ASSEMBLER MODULE STARTS HERE ****
175>
176>
177>
178> ADJ, ;MOVE REGISTER INDICATOR TO MIDDLE FIELD OF CODE
179> 01CF 17          RAL
180> 01D0 17          RAL

```

CP/M VERSION
 COPYRIGHT © 1976
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA. 93950
 SER. #

<pre> 01D1 17 RAL 01D2 E638 ANI 111000B 01D4 C9 RET 01D5 17 JADJ4, ;MOVE TO LEFT BY 4 AND MASK 01D6 17 RAL 01D7 17 RAL 01D8 17 RAL 01D9 E630 RAL 01DB C9 AH1 110000B 01DB C9 RET 01DC EB SEAR2, ;SAME AS SEAR, EXCEPT 2 CHARACTER MATCH 01DD 2AF07 XCHG 01E0 EB LHLD OPCODE ;2ND BYTE IN D, 1ST BYTE IN E 01E1 7B XCHG 01E2 BE SEAO, ;H,L ADDRESS TABLE TO MATCH ON 01E3 C2E801 MOV A, E ;GET 1ST BYTE 01E6 23 CMP M ;MATCH? 01E7 7A JNZ SEA1 ;TO ADDRESS NEXT ELT 01E8 BE INX H ;NEXT TO MATCH 01E9 C8 MOV A, D ;2ND CHAR 01EA 28 CMP M 01EB 28 RZ 01EC 28 DCX 01ED 0D DCX 01EE C2E101 DCR C ;MATCH AT CURRENT ENTRY 01F1 0D SEA1, ;ADDRESSES NEXT ELEMENT 01F2 C9 JNZ SEA0 ;FOR ANOTHER COMPARE 01F3 0604 NO MATCH IN TABLE, RETURN WITH NON-ZERO VALUE 01F5 D5 DCR C 01F6 118F07 RET 01F9 1A SE1, ;SEARCH FOR MATCH IN OPCODE TABLE, LENGTH OF TABLE IN REG-D 01FA BE D, E CONTAINS ADDRESS OF BINARY EQUIVALENT OF OPCODE 01FB C20602 H,L ADDRESS FOUR CHARACTER OPCODE TO MATCH 01FE 23 OPCODE CONTAINS FOUR BYTE OPCODE TYPED AT CONSOLE 01FF 13 RETURNS WITH ZERO VALUE IF OPCODE FOUND, WITH D,E 0200 05 ADDRESSING PROPER BYTE, NON-ZERO IF NOT FOUND. 0201 C2F901 MVI B, 4 ;4 CHARACTER MATCH 0204 D1 PUSH D ;SAVE THE CURRENT BYTE VALUE LOCATION 0205 C9 LXI D, OPCODE ;ADDRESS CHARACTERS TYPED 0206 23 LDAX 0207 05 SE2, ;POINT TO FIRST BYTE TO MATCH 0208 23 CMP M ;SAME CHARACTER AS TABLE? 0209 13 JNZ SE2 ;NO, SKIP TO NXT TABLE ENTRY 020A 05 INX H ;YES, LOOK AT NEXT CHARACTER 020B 05 INX D ;MOVE TO NEXT CHARACTER TYPED 020C 05 DCR B ;DECREMENT CHARACTER COUNT 020D 05 JNZ SE1 ;MORE TO MATCH? 020E 05 POP D ;COMPLETE MATCH, RETURN WITH D,E ADDRESSING BYTE VALUE 020F 05 RET 0210 05 SE2, ;MISMATCH, FINISH COUNT 0211 05 INX H 0212 05 DCR B </pre>	CP/M VERSION COPYRGHT © 1976 DIGITAL RESEARCH P. O. BOX 579 PACIFIC GROVE, CA. 93950 SER. #
--	---

```

241> 0208 C28602      JNZ    SE2
242>
243> ; H,L AT END OF FOUR BYTE AREA, MOVE BACK 8
244> 020B 11F6FF      LXI    D,-8
245> 020E 19          DAD    D      ;H,L READY FOR NXT. MATCH
246>
247> 020F D1          POP    D      ;RESTORE BYTE POINTER
248> 0210 13          INX    D      ;MOVE TO NEXT IN CASE MATCH OK
249> 0211 00          DCR    C      ;MORE OPCODES TO MATCH?
250> 0212 C2F301      JNZ    SEAR   ;LOOK FOR MORE
251>
252> ; NO MATCH FOUND IN TABLE, SET NON-ZERO VALUE AND RETURN
253> 0215 00          DCR    C
254> 0216 C9          RET
255>
256>
257> GETREG: ;SCAN FOR SIMPLE REGISTER REFERENCE
258> 0217 C5          PUSH   B
259> 0210 CD4101      CALL   SCAN
260> 021B CA2E06      JZ     ERR
261> 021E BE08      MVI    C,8   ;8 REGISTERS
262> 0220 217907      LXI    H,SREG ;SIMPLE REGISTERS
263> 0223 CDC081      CALL   SEAR2 ;LOOK FOR 2 CHAR MATCH
264> 0226 C22E06      JNZ    ERR
265> 0229 00          DCR    C
266> 023A 79          MOV    A,C
267> 022B C1          POP    B
268> 022C C9          RET
269>
270> GETD: ;GET DOUBLE PRECISION REGISTER
271> 022D C5          PUSH   B
272> 022E CD4101      CALL   SCAN
273> 0231 CA2E06      JZ     ERR
274> 0234 BE05      MVI    C,5
275> 0235 218807      LXI    H,DREG
276> 0239 CDF301      CALL   SEAR
277> 023C C22E06      JNZ    ERR
278> 023F 00          DCR    C
279> 0240 79          MOV    A,C
280> 0241 C1          POP    B
281> 0242 C9          RET
282>
283> GETDR: ;GET DOUBLE REGISTER (BDHSP)
284> 0243 CD2D02      CALL   GETD
285> 0246 FE04      CPI    4      ;PSW?
286> 0248 CA2E06      JZ     ERR
287> 0248 C9          RET
288>
289> GETPR: ;GET PUSH/POP REGISTER (BDH OR PSW)
290> 024C CD2D02      CALL   GETD
291> 024F FE03      CPI    3
292> 0251 CA2E06      JZ     ERR
293> 0254 FE04      CPI    4
294> 0256 C0          PHZ
295> 0257 3D          DCR    A      ;PSW MUST BE ADJUSTED
296> 0258 C9          RET
297>
298> GCON: ;GET CONDITION CODE
299> ; BUFFER IS SCANNED, MOVE LEFT BEFORE COMPARE
300> 0259 218F07      LXI    H,OPCODE

```

CPI..VERSION
 COPYRIGHT © 1976
 DIGITAL RESEARCH
 P.O. BOX 579
 PACIFIC GROVE, CA 93050
 SER. #

```

301> 025C 119007      LXI    D,OPCODE+1
302> 025F 0E02      MVI    C,2   ;MOVE TWO CHARACTERS
303> 0261 1A          LDAX   D      ;LOAD CHARACTER TO MOVE
304> 0262 77          MOV    M,A   ;MOVE LEFT
305> 0263 23          INX    H      ;NEXT DESTINATION
306> 0264 13          INX    D      ;NEXT SOURCE
307> 0265 0D          DCR    C
308> 0266 C26102      JNZ    MOP
309>
310>
311> 0269 1A          ;MUST BE BLANK AT END
312> 026A FE20      CPI
313> 026C C22E06      JHZ    ERR
314> 026F 77          MOV    M,A
315>
316> ;NOW READY TO DO THE COMPARE
317> 0270 216907      LXI    H,CREG
318> 0273 0E08      MVI    C,8
319> 0275 CDC081      CALL   SEAR2
320> 0278 C22E06      JNZ    ERR
321> 027B 0D          DCR    C
322> 027C 79          MOV    A,C
323> 027D CDCF01      CALL   ADJ   ;MOVE TO BITS 3,4,5 OF BYTE (LSB = 0)
324> 0280 C9          RET
325>
326>
327> 0281 CD5902      GCONA: ;GET CONDITION CODE TO REGISTER A. DOUBLE ADDRESS TO B,C
328> 0284 F5          PUSH   PSW
329> 0285 CD8501      CALL   GADDR ;VALUE TO B,C
330> 0288 F1          POP    PSW
331>
332> 0289 F6C0      INCLUDE HIGH ORDER 11'S FOR J AND C OPCODES
333> 028B C9          RET
334>
335> 028C 1A          SETMD: ;SET MEMORY AT LOCATION PC TO VALUE ADDRESSED BY B
336> 028C 1A          LDAX   D      ;VALUE TO ACCUM
337>
338>
339> 028D 218C01      SETM: ;SET MEMORY AT LOCATION PC TO VALUE IN ACCUM. INC PC
340> 0290 77          LHLD   PC
341> 0291 23          MOV    M,A   ;STORE AT PC
342> 0292 228C01      INX    H      ;PC=PC+1
343> 0295 C9          SHLD   PC
344>
345>
346>
347>
348> 0296 CDC088      GETOP: ;PROCESS NEXT OPCODE
349> 0299 FE0D      CALL   CI
350> 029B CA5906      CPI    CR
351> 029E CD4401      JZ     GOBACK ;RETURN IF SIMPLE INPUT
352> 02A1 CA2E06      CALL   SCAN0
353>
354>
355> 02A4 0E11      JZ     ERR
356> 02A6 21BF06      CALL   SEAR ;LOOK FOR MATCH
357> 02A9 115E06      JNZ    CHK1 ;NO MATCH. CHECK FOR GROUP-1
358> 02AC CDF301
359> 02AF C2B502
360>

```

CPI/VERSION
 COPYRIGHT © 1976
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93050
 SER. #

P. O. BOX 579
 PACIFIC GROVE, CA 93050

```

361>      ; MATCHED OPCODE, D/E ADDRESS BYTE VALUE
362> 02B2 C38C02  JMP  SETMD ;SET MEMORY AT PC AND INC PC
363>      ;
364>      ; CHECK GROUP-1 VALUES
365> 02B5 0E0A  CHK1,  MVI  C,10 ;LENGTH OF GROUP-1
366> 02B7 21E706  LXI  H,ETAB2
367> 02BA CDF301  CALL  SEAR ;D,E REMAIN SET
368> 02BD C2C902  JHZ  CHK2 ;NO MATCH, CHECK NEXT GROUP
369>
370>      ;
371> 02C0 CD8C02  CALL  SETMD
372> 02C3 C18601  CALL  GBYTE ;GETS BYTE VALUE TO ACCUMULATOR
373> 02C6 C38D02  JMP  SETM ;PUTS BYTE VALUE TO MEMORY AT PC
374>
375>      ;
376> 02C9 0E06  CHK2,  MVI  C,6
377> 02CB 21FF06  LXI  H,ETAB3
378> 02CE CDF301  CALL  SEAR
379> 02D1 C2E102  JHZ  CHK3 ;NO MATCH
380>
381>      ;
382> 02D4 CD8C02  CALL  SETMD
383> OP2,   ENTER HERE FOR DOUBLE BYTE OPERANDS
384> 02D7 CD8501  CALL  GADDR ;VALUE IN B,A
385> 02DA C08D02  CALL  SETM
386> 02DD 78      MOV   A,B
387> 02DE C38D02  JMP  SETM
388>
389>      ;
390> 02E1 0E01  CHK3,  MVI  C,1
391> 02E3 210307  LXI  H,PMOV
392> 02E6 CDF301  CALL  SEAR
393> 02E9 C2FD02  JHZ  CHK4
394>
395>      ;
396> 02EC CD1702  MOV   INSTRUCTION GET DESTINATION OPERAND
397> 02EF CDCF01  CALL  GETREG ;VALUE TO ACCUMULATOR
398> 02F2 47      CALL  ADJ
399> 02F3 0E40      MOV   B,A ;SAVE IN B
400>      ;
401> OP1,   GET NEXT OPERAND FOR MOV, FIRST OPERAND FOR ACCUM/REG OPERAT
402> 02F5 CD1702  CALL  GETREG
403> 02F8 B1      ORA   C ;SETS HIGH ORDER TWO BITS
404> 02F9 B0      ORA   B ;SETS DESTINATION/OPERATOR
405> 02FA C38D02  JMP  SETM
406>
407>      ;
408> 02FD 0E08  CHK4,  MVI  C,8
409> 02FF 212307  LXI  H,ETAB5
410> 0302 CDF301  CALL  SEAR
411> 0305 C21303  JHZ  CHK5
412>
413>      ;
414> 0308 0D      ACCUM/REG INSTRUCTION, C COUNTS OPERATORS AS SEARCH PROCEEDS
415> 0309 79      DCR   C
416> 030A CDCF01  MOV   A,C
417> 030D 47      CALL  ADJ
418>      ;
419> 030E 0E00  OPERATOR NUMBER (SHIFTED) SAVED FOR LATER MASK
420> 0310 C3F502  MVI  C,1000000000 ;ACCUM/REG OPERATOR INDICATOR
                JMP  OP1  ;GETS OPERAND AND SAVES BYTE IN MEMORY

```

```

421>      ;
422> 0313 0E02  CHK5,  ;MAY BE INR/DCR
423> 0315 212B07  MVI  C,2
424> 0318 CDF301  LXI  H,PDCR
425> 031B C22B03  CALL  SEAR
426>      ;JNZ  CHK6
427>      ;
428> 031E 0C      ;C=2 IF DCR, =1 IF INR
429> 031F 0C      IHR  C ;+1
430> 0320 0C      IHR  C ;+2
431> 0321 CD1702  IHR  C ;+3
432> 0324 CDCF01  CALL  GETREG ;VALUE TO ACCUM
433> 0327 B1      CALL  ADJ
434> 0328 C38D02  ORA   C ;FILL PROPER INSTRUCTION INDICATOR
435>      ;JMP  SETM
436>      ;
437> 032B 0E01  CHK6,  ;MAY BE A MVI INSTRUCTION
438> 032D 212F07  MVI  C,1
439> 032E CDF301  LXI  H,PMVI
440> 0330 C08D02  CALL  SEAR
441> 0333 C24703  JNZ  CHK7
442>      ;
443> 0336 CD1702  MVI INSTRUCTION, GET REGISTER
444> 0339 CDCF01  CALL  GETREG ;VALUE GOES TO ACCUMULATOR
445> 033C F606  CALL  ADJ
446> 033E C08D02  ORI   110B
447> 0341 C08D02  CALL  SETM
448> 0344 C38D02  CALL  GBYTE
449>      ;JMP  SETM
450>      ;
451> 0347 0E06  CHK7,  ;CHECK FOR GROUP-7
452> 0349 214707  MVI  C,6
453> 034C CDF301  LXI  H,ETAB7
454> 034F C26D03  CALL  SEAR
455>      ;JNZ  CHK8
456>      ;
457> 0352 79      LXI, STAX, INX, DAD, LDA, OR DCX
458> 0353 FE04  MOV   A,C ;A=1...6
459> 0355 DA5A03  CPI   4
460> 0355 DA5A03  JC    IN0
461>      ;
462> 0358 C605  MUST BE DAD, LDA, OR DCX
463> 035B C085  ADI   5 ;CHANGES ACCUM TO 9.10, OR 11
464>      ;IN0,  ;ACCUMULATOR CONTAINS CODE, SAVE IT
465> 035A 47      MOV   B,A
466> 035B CD4302  CALL  GETDR ;DOUBLE REGISTER VALUE TO ACCUM
467> 035E C0D501  CALL  ADJ4 ;ADJUST VALUE TO MIDDLE FIELD
468> 0361 B0      ORA   B ;FILLS REMAINING BITS
469> 0362 C08D02  CALL  SETM
470>      ;MAY BE LXI
471> 0365 E6CF  ANI   11001111B
472> 0367 FE01  CPI   1
473> 0369 C0      RHZ  ;NOT LXI
474> 036A C3D702  JMP  OP2  ;PICK UP OPERAND
475>      ;
476>      ;
477>      ;
478>      ;
479> 036D 0E01  CHK8,  ;RST?
480> 036F 214B07  MVI  C,1
                LXI  H,PRST

```

CP/M VERSION _____
COPYRGHT © 1976
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____

```

481> 0372 CDF301      CALL    SEAR
482> 0375 C28803      JHZ    CHK9
483> ; RST. GET OPERAND
484> ; CALL GBYTE
485> 0376 CD6601      CPI    8
486> 037B FE08      JHC    ERR
487> 037D D22E06      CALL    ADJ
488> 0380 CDCF01      ORI    11000111B
489> 0383 F6C7      JMP    SETM
490> 0385 C38D02
491>
492> ; CHK9. ;POP/PUSH?
493> 0388 0E02      MVI    C, 2
494> 038A 215707      LXI    H, PPOP+4
495> 038D CDF301      CALL    SEAR
496> 0390 C2A803      JHZ    CHK10
497>
498> ; C=2 IF PUSH, 1 IF POP
499> 0393 0D      DCR    C
500> 0394 C29C03      JHZ    PP0
501>
502> ; POP, SET BIT PATTERN
503> 0397 0EC1      MVI    C, 1100001B
504> 0399 C39E03      JMP    PPI
505>
506> ; PP0. ;PUSH
507> 039C 0EC5      MVI    C, 11000101B
508> 039E CD4C02      PP1.   CALL    GETPR  ;DOUBLE PUSH/POP REGISTER TO PROPER FIELD
509> 03A1 CDD501      CALL    ADJ4  ;MOVE TO FIELD
510> 03A4 B1      ORA    C
511> 03A5 C38D02      JMP    SETM
512>
513> ; CHK10. ;J/C/R?
514> 03A8 3A8F07      LDA    OPCODE
515> 03AB FE4A      CPI    'J'
516> 03AD C28803      JHZ    CHK11
517> 03B0 CDB102      CALL    GCOHA
518> ; CONDITION CODE TO FIELD IN ACCUM. ADDRESS TO B,C
519> 03B3 F602      ORI    010B
520> 03B5 C3C203      JMP    FADDR ;FILL ADDRESS
521>
522> 03B8 FE43      CPI    'C'
523> 03B9 C2C003      JHZ    CHK12
524> 03B0 CDB102      CALL    GCOHA
525> 03C0 F604      ORI    100B
526>
527> 03C2 CDB102      FADDR. CALL    SETM
528> 03C5 79      MOV    A, C
529> 03C6 CDB102      CALL    SETM
530> 03C9 78      MOV    A, B
531> 03CA C38D02      JMP    SETM
532>
533> 03CD FE52      CPI    'R'
534> 03CF C22E06      JHZ    ERR
535> 03D2 C05902      CALL    GCON
536> 03D5 F6C0      ORI    1100000B
537> 03D7 C38D02      JMP    SETM
538>
539> ; **** END OF ASSEMBLER MODULE, START DISASSEMBLER ****

```

CP/M VERSION
 COPYRIGHT © 1976
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA. 93950
 SER. #

```

541> ; **** END OF ASSEMBLER MODULE, START DISASSEMBLER ****
542> ; **** END OF ASSEMBLER MODULE, START DISASSEMBLER ****
543> 03DA 2A0E01      RDBYTE, LHLD MPC
544> 03DD D5      PUSH D ;SAVE DE
545> 03DE EB      XCHG ;MAX PC TO D,E
546> 03DF 2A0C01      LHLD PC ;CURRENT PC
547> ; SUBTRACT PC FROM MPC, STOP IF CARRY GENERATED
548> 03E2 7B      MOV A, E
549> 03E3 95      SUB L
550> 03E4 7A      MOV A, D
551> 03E5 9C      SBB H
552> 03E6 D2EE03      JHC RD0
553>
554> ; PC EXCEEDS MPC, RETURN
555> 03E9 2A1301      LHLD OLDSP
556> 03EC F9      SPHL ;RESTORE ORIGINAL STACK POINTER
557> 03ED C9      RET
558>
559> 03EE D1      RD0, POP D ;RESTORE D,E
560> 03EF 7E      MOV A, M
561> 03F0 23      INX H
562> 03F1 220C01      SHLD PC
563> 03F4 C9      RET
564>
565> 03F5 3C      RGPRHT, INR A
566> 03F6 E607      ANI 87
567> 03F8 FE06      CPI 06
568> 03FA DAFF03      JC RGP1
569> 03FD C603      ADI 03
570> 03FF FE05      RGP1, CPI 05
571> 0401 DA0604      JC RGP2
572> 0404 C602      ADI 02
573> 0406 C641      RGP2, ADI 41H
574> 0408 4F      MOV C, A
575> 0409 C31501      JMP CO
576>
577> 040C 47      DECODE, MOV B, A
578> 040D E6F0      ANI 0F0H
579> 040F 0F      RRC
580> 0410 0F      RRC
581> 0411 0F      RRC
582> 0412 0F      RRC
583> 0413 C690      ADI 90H
584> 0415 27      DAA
585> 0416 CE40      ACI 40H
586> 0418 27      DAA
587> 0419 4F      MOV C, A
588> 041A CD1501      CALL CO
589> 041D 78      MOV A, B
590> 041E E60F      ANI 0FH
591> 0420 C690      ADI 90H
592> 0422 27      DAA
593> 0423 CE40      ACI 40H
594> 0425 27      DAA
595> 0426 4F      MOV C, A
596> 0427 C31501      JMP CO
597>
598> 042A 0604      PRINT, MVI B, 4
599> 042C 4E      P1, MOV C, M
600> 042D CD1501      CALL CO

```

CP/M VERSION
 COPYRGHT © 1976
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA. 93950
 SER. #

```

601> 0430 23      INX    H
602> 0431 05      DCR    B
603> 0432 C22C04   JHZ    P1
604> 0435 0E20     MVI    C.20H
605> 0437 C31501   JMP    CO
606> 043A 7A      XTRACT: MOV    A,D
607> 043B E638     AHI    38H
608> 043D 0F      RRC
609> 043E 0F      RRC
610> 043F 0F      RRC
611> 0440 C9      RET
612>
613> 0441 CD3A04   CCPRHT: CALL  XTRACT
614> 0444 87      ADD    A
615> 0445 4F      MOV    C,A
616> 0446 215807   LXI    H,CCODE
617> 0449 09      DAD    B
618> 044A 4E      MOV    C,M
619> 044B CD1501   CALL   CO
620> 044E 23      INX    H
621> 044F 4E      MOV    C,M
622> 0450 CD1501   CALL   CO
623> 0453 0E20     MVI    C.20H
624> 0455 CD1501   CALL   CO
625> 0458 C31501   JMP    CO
626>
627> 045B CD3A04   RPPRHT: CALL  XTRACT
628> 045E E606     AHI    B6
629> 045F FE06     CPI    B6
630> 0462 C2F503   JHZ    RCPRT
631> 0465 0E53     MVI    C.53H
632> 0467 CD1501   CALL   CO
633> 046A 0E50     MVI    C.50H
634> 046C C31501   JMP    CO
635>
636> PRPC: ;PRINT CRLF FOLLOWED BY PC VALUE
637> 046F CD3601   CALL   CRLF
638> 0472 2A0C01   LHLD  PC
639> 0475 7C      MOV    A,H
640> 0476 CD8C04   CALL   DECODE
641> 0479 7D      MOV    A,L
642> 047A CD8C04   CALL   DECODE
643> 047D 0E20     MVI    C,' '
644> 047F CD1501   CALL   CO
645> 0482 CD1501   CALL   CO
646> 0485 C9      RET
647>
648> DISENT: ;ENTER HERE FROM DEBUGGER - SET UP Jmps
649> 0486 210000   LXI    H,0
650> 0489 39      DAD    SP
651> 048A 221301   SHLD  OLDSP  ;SP SAVED FOR LATER RETURN
652> 048D CDBD01   CALL   INIT
653>
654> ;CHECK FOR PAGE MODE DISPLAY
655> 0490 3A1001   LDA    PAGM  ;GET PAGE MODE (NUMBER OF LINES TO PRINT)
656> 0493 87      ORA    A      ;SET FLAGS
657> 0494 CAAB04   JZ    DISASM ;NOT PAGE MODE
658>
659> ;SET MPC TO 0FFFFH
660> 0497 21FFFF   LXI    H,0FFFFH

```

CP/M VERSION

COPYRIGHT © 1976
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA. 93950

SER. # _____

```

661> 049A 220E01
662>
663> 049D 3C
664> 049E C2AB04
665>
666> 04A1 3C
667> 04A2 321001
668> 04A5 2A0C01
669> 04A8 C3C404
670>
671>
672>
673> ;DISASM:
674> 04AB 0E0B
675> 04AD D0612
676> 04B0 E601
677> 04B2 C25906
678>
679> ;CHECK TO SEE IF ENOUGH LINES PRINTED IN PAGE MODE
680> 04B5 211001
681> 04B8 7E
682> 04B9 B7
683> 04BA CAC104
684>
685> ;PAGE MODE. DECREMENT AND CHECK FOR ZERO
686> 04BD 35
687> 04BE CA5906
688>
689> 04C1 CD6F04
690> 04C4 CDDA03
691> 04C7 57
692> 04C8 215E06
693> 04C9 011100
694> 04CE BE
695> 04CF CA2006
696> 04D2 23
697> 04D3 0D
698> 04D4 C2CE04
699> 04D7 0E0A
700> 04D9 BE
701> 04DA CABC06
702> 04DD 23
703> 04DE 0D
704> 04DF C2D904
705> 04E2 0E06
706> 04E4 BE
707> 04E5 CAF005
708> 04E8 23
709> 04E9 0D
710> 04EA C2E404
711> 04ED E6C0
712> 04EF FE40
713> 04F1 CAD605
714> 04F4 FE80
715> 04F6 CAC705
716> 04F9 7A
717> 04FA E6C7
718> 04FC D684
719> 04FE CAB805
720> 0501 3D

```

SHLD MPC
255 IMPLIES TRACE MODE
INR A
JHZ DISASM ;NOT TRACE MODE IF SR
TRACE MODE, SET TO 1 AND IGNORE ADDRESS FIELD
INR A ;1 IN ACC
STA PAGM
LHD PC ;RECOVER PC
JMP DIS1

CP/... VERSION

COPYRIGHT © 1976
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE CA. 93950

SER. # _____

721> 0502 CAB205	JZ	MG3	781> 0590 CD2A04	MG51.	CALL	PRINT
722> 0505 3D	DCR	A	782> 0593 CD5B04		CALL	RPRNT
723> 0506 CA9E05	JZ	MG4	783> 0596 0E2C		MVI	C,2CH
724> 0509 7A	MOV	A,D	784> 0598 CD1501		CALL	CO
725> 050A E6C8	ANI	0C0H	785> 0598 C3FB05		JMP	D7
726> 050C CA7205	JZ	MG5	786> 059E 212F07	MG4.	LXI	H,PMVI
727> 050F 7A	MOV	A,D	787> 05A1 CD2A04		CALL	PRINT
728> 0510 E6C7	ANI	0C7H	788> 05A4 CD3A04		CALL	XTRACT
729> 0512 D6C0	SUI	0C0H	789> 05A7 CDF503		CALL	RGRNRT
730> 0514 CA6705	JZ	MG6	790> 05AA 0E2C		MVI	C,2CH
731> 0517 D602	SUI	B2	791> 05AC CD1501		CALL	CO
732> 0519 CASC05	JZ	MG7	792> 05AF C31706		JMP	D8
733> 051C D602	SUI	B2	793> 05B2 212B07	MG3.	LXI	H,PDCR
734> 051E CAS105	JZ	MG8	794> 05B5 C3BB05		JMP	D5
735> 0521 D603	SUI	B3	795> 05B8 212B07	MG2.	LXI	H,PINR
736> 0523 CA4205	JZ	MG9	796> 05B8 CD2A04	D5.	CALL	PRINT
737> 0526 7A	MOV	A,D	797> 05B8 CD3A04		CALL	XTRACT
738> 0527 E607	ANI	B7	798> 05C1 CDF503	D6.	CALL	RGRNRT
739> 0529 4F	MOV	C,A	799> 05C4 C3AR04		JMP	DISASM
740> 052A 215207	LXI	H,PPOP-1	800> 05C7 7A	MG1.	MOV	A,D
741> 052D 69	DAD	B	801> 05C8 E638		ANI	38H
742> 052E CD2A04	CALL	PRINT	802> 05CA 0F		RR	
743> 0531 CD3A04	CALL	XTRACT	803> 05CB 4F		MOV	C,A
744> 0534 FE06	CPI	B6	804> 05CC 210707		LXI	H,PADD
745> 0536 C2C105	JNZ	D6	805> 05CF 09		DAD	B
746> 0539 214F07	LXI	H,PPSW	806> 05D0 CD2A04		CALL	PRINT
747> 053C CD2A04	CALL	PRINT	807> 05D3 C3E705		JMP	D9
748> 053F C3AB04	JMP	DISASM	808> 05D6 210307	MG8.	LXI	H,PMOV
749> 0542 214B07	LXI	H,PRST	809> 05D9 CD2A04		CALL	PRINT
750> 0545 CD2A04	CALL	PRINT	810> 05DC CD3A04		CALL	XTRACT
751> 0548 CD3A04	CALL	XTRACT	811> 05DF CDF503		CALL	RGRNRT
752> 0548 CD0C04	CALL	DECODE	812> 05E2 0E2C		MVI	C,2CH
753> 054E C3AB04	JMP	DISASM	813> 05E4 CD1501		CALL	CO
754> 0551 0E43	MGB.	MVI	814> 05E7 7A	D9.	MOV	A,D
755> 0553 CD1501	CALL	CO	815> 05E8 E607		ANI	B7
756> 0556 CD4104	CALL	CCPRT	816> 05EA CDF503		CALL	RGRNRT
757> 0559 C3FB05	JMP	D7	817> 05ED C3AB04		JMP	DISASM
758> 055C 0E4A	MGB.	MVI	818> 05F0 79	TG3.	MOV	A,C
759> 055E CD1501	CALL	CO	819> 05F1 87		ADD	A
760> 0561 CD4104	CALL	CCPRT	820> 05F2 87		ADD	A
761> 0564 C3FB05	JMP	D7	821> 05F3 4F		MOV	C,A
762> 0567 0E52	MGB.	MVI	822> 05F4 21E706		LXI	H,TAB3-4
763> 0569 CD1501	CALL	CO	823> 05F7 09		DAD	B
764> 056C CD4104	CALL	CCPRT	824> 05F8 CD2A04		CALL	PRIHT
765> 056F C3AB04	JMP	DISASM	825> 05FB CDDA03	D7.	CALL	RDBYTE
766> 0572 213307	MGS.	LXI	826> 05FE 57		MOV	D,A
767> 0573 7A	MOV	A,D	827> 05FF CDDA03		CALL	RDBYTE
768> 0576 E60F	ANI	BFH	828> 0602 CD0C04		CALL	DECODE
769> 0578 3D	DCR	A	829> 0605 7A		MOV	A,D
770> 0579 CA9005	JZ	MG51	830> 0606 CD0C04		CALL	DECODE
771> 057C FE04	CPI	B4	831> 0609 C3AB04		JMP	DISASM
772> 057E DA8305	JC	D4	832> 060C 79	TG2.	MOV	A,C
773> 0581 D605	SUI	B5	833> 060D 87		ADD	A
774> 0583 87	D4.	ADD	834> 060E 87		ADD	A
775> 0584 87	ADD	A	835> 060F 4F		MOV	C,A
776> 0585 4F	MOV	C,A	836> 0610 21BF06		LXI	H,TAB2-4
777> 0586 09	DAD	B	837> 0613 09		DAD	B
778> 0587 CD2A04	CALL	PRINT	838> 0614 CD2A04		CALL	PRINT
779> 058A CD5B04	CALL	RPRNT	839> 0617 CDDA03	D8.	CALL	RDBYTE
780> 058D C3AB04	JMP	DISASM	840> 061A CD0C04		CALL	DECODE

CP/M VERSION

COPYRIGHT © 1976

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA. 93950

SER. #

CP/M VERSION

COPYRIGHT © 1976

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA. 93950

SER. #

841> 061D C3AB04	JMP	DISASM	901>				
842> 0620 79	TG1,	MOV A,C	902>				
843> 0621 87		ADD A	903>	06D3 5342492049	DB	'SBI ','IN ','SUI ','OUT '	
844> 0622 87		ADD A	904>				
845> 0623 4F		MOV C,A	905>				
846> 0624 217B06	LXI	H,TAB1-4	906>	06E3 41434920	DB	'ACI '	
847> 0627 09	DAD	B	907>	06E7 41444920	ETAB2,	DB	'ADI '
848> 0628 CD2A04	CALL	PRINT	908>				
849> 062B C3AB04	JMP	DISASM	909>	06EB 43414C4C4ATAB3,	DB	'CALL','JMP ','LDA ','STA '	
850>	ERR,	JENTER HERE FOR ERROR REPORTING	910>				
851> 062E CD3601	CALL	CRLF	911>				
852> 0631 0E3F	MVI C,'?'		912>	06FB 4C484C44	DB	'LHLD'	
853> 0633 CD1501	CALL	CO	913>	06FF 53484C44	ETAB3,	DB	'SHLD'
854>			914>				
855> 0636 2A1301	LHLD	OLDSP	915>				
856> 0639 F9	SPHL		916>	0703 4D4F5620	PMOV,	DB	'MOV '
857> 063A 2A1101	LHLD	TPC ; RESTORE PC	917>	0707 4144442041PADD,	DB	'ADD ','ADC ','SUB ','SBB '	
858> 063D 220C01	SHLD	PC	918>				
859>			919>				
860>			920>	0717 414E412058	DB	'ANA ','XRA ','ORA '	
861> ASMEH,	JENTER HERE FROM DEBUGGER		921>	0723 434D5020	ETAB5,	DB	'CMP '
862> 0640 210000	LXI	H,0	922>				
863> 0643 39	DAD	SP	923>	0727 494E5220	PINR,	DB	'INR '
864> 0644 221301	SHLD	OLDSP	924>	0728 44435220	PDCR,	DB	'DCR '
865> 0647 CDBD01	CALL	INIT	925>	072F 4D564920	PMVI,	DB	'MVI '
866>			926>	0733 4C58492053PLXI,	DB	'LXI ','STAX','INX ','DAD '	
867> 064A CD6F04	ASM0:	CALL PRPC ;PRINT PC VALUE	927>				
868> 064D 221101	SHLD	TPC ;SAVE PC VALUE	928>				
869> 0650 CD8508	CALL	GETBUFF ;FILL INPUT BUFFER	929>	0743 4C444158	DB	'LDAX'	
870> 0653 CD9602	CALL	GETOP ;GET OPERATION	930>	0747 44435820	ETAB7,	DB	'DCX '
971> 0656 C34A06	JMP	ASM0	931>				
872>			932>				
873> 0659 2A1301	G0BACK,	LHLD OLDSP	933>	074B 52535420	PRST,	DB	'RST '
874> 065C F9	SPHL		934>	074F 50535720	PPSW,	DB	'PSW '
875> 065D C9	RET		935>	0753 504F502050PPOP,	DB	'POP ','PUSH'	
876>			936>				
877> 065E 00070F17	TABLE,	DB 000H,007H,00FH,017H	937>	075B 4E5A5A204ECCODE,	DB	'NZ ','Z ','NC ','C '	
878> 0662 1F272F37	DB	01FH,027H,02FH,037H	938>				
879> 0666 3F76C9E3	DB	03FH,076H,0C9H,0E3H	939>	0763 504F504558	DB	'PO ','PE ','P '	
880> 066A E9EF3F9	DB	0E3H,0E6H,0F3H,0F9H	940>	0769 4D20 CREG,	DB	'M '	
891> 065E FBC6CED3	DB	0FBH,0C6H,0CEH,0D3H	941>				
892> 0672 D6DBDEE6	DB	0D6H,0DBH,0DEH,0E6H	942>				
893> 0676 EEF6FE22	DB	0EEH,0F6H,0FEH,B22H	943>	0768 4220432044	DB	'B ','C ','D ','E '	
894> 067A 2A323AC3	DB	02AH,032H,03AH,0C3H	944>	0773 48204C204D	DB	'H ','L ','M '	
895> 067E CD	DB	0CDH	945>	0779 4120 SREG,	DB	'A '	
896> 067F 4549202053TAB1,	DB	'EI ','SPHL','DI ','XCHG'	946>				
897>			947>	077B 4220202044	DB	'B ','D ','H ','SP '	
898>			948>	078B 50535720 DREG,	DB	'PSW '	
899>			949>				
900> 068F 5043484C58	DB	'PCHL','XTHL','RET ','HLT '	950>	078F OPCODE,	DS	4	
901>			951>	0793 END			
902>							
903>							
904>							
905>							
906> 06AF 5241522052	DB	'RAR ','RAL ','RRD ','RLC '					
907>							
908>							
909> 06BF 4E4F5020 ETAB1,	DB	'NOP '					
908> 06C3 435049204FTAB2,	DB	'CPI ','ORI ','XRI ','ANI '					

CPI/M VERSION
 COPYRIGHT © 1976
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. #

1>
 2>
 3> 0100
 4> ;
 5> ORG 100H
 6> ; CP/M DEBUGGER VERSION 1.0
 7> ; RELOCATABLE VERSION
 8>
 9> ;
 10> COPYRIGHT (C) 1976
 11> DIGITAL RESEARCH
 12> BOX 579 PACIFIC GROVE
 13> CALIFORNIA 93950
 14>
 15> 0000 = FALSE EQU 0
 16> 0FFF = TRUE EQU NOT FALSE
 17> 0103 = DISIN EQU \$+3
 18> 0000 = DEMON EQU \$+700H
 19> 1206 = BDOS EQU \$+1106H
 20> 0005 = BDOSE EQU 5H ; ENTRY POINT TO DOS FROM USER PROGRAMS
 21> 0100 = PCB BASE EQU 100H ; DEFAULT PC
 22> 0100 = SPBASE EQU 100H ; DEFAULT SP
 23> 0106 = DISEN EQU DISEN+3 ; DISASSEMBLER ENTRY POINT
 24> 0109 = ASSEM EQU DISEN+3 ; ASSEMBLER ENTRY POINT
 25> 010C = DISPC EQU ASSEM+3 ; DISASSEMBLER PC VALUE
 26> 010E = DISPM EQU DISPC+2 ; DISASSEMBLER PC MAX VALUE
 27> 0110 = DISPG EQU DISPM+2 ; DISASSEMBLER PAGE MODE IF NON ZERO
 28> 000C = PSIZE EQU 12 ; NUMBER OF ASSEMBLY LINES TO LIST WITH
 29> 0020 = CSIZE EQU 32 ; COMMAND BUFFER SIZE
 30> 0032 = SSIZE EQU 50 ; LOCAL STACK SIZE
 31>
 32> ; BASIC DISK OPERATING SYSTEM CONSTANTS
 33> 0001 = CIF EQU 1
 34> 0002 = COF EQU 2
 35> 0003 = RIF EQU 3
 36> 0004 = POF EQU 4
 37> 0005 = LOF EQU 5
 38> 0007 = IDS EQU 7
 39> 000A = GETF EQU 10 ; FILL BUFFER FROM CONSOLE
 40> 000B = CHKIO EQU 11 ; CHECK IO STATUS
 41> 000C = LIFT EQU 12 ; LIFT HEAD ON DISK
 42> 000F = OPF EQU 15 ; DISK FILE OPEN
 43> 0014 = RDF EQU 20 ; READ DISK FILE
 44> 001A = DMAF EQU 26 ; SET DMA ADDRESS
 45> 0058 = DBP EQU 5BH ; DISK BUFFER POINTER
 46> 0050 = DBF EQU 80H ; DISK BUFFER ADDRESS
 47> 005C = DFCB EQU 5CH ; DISK FILE CONTROL BLOCK
 48> 005C = FC8 EQU DFCB
 49> 0000 = FDN EQU 0 ; DISK NAME
 50> 0001 = FFH EQU 1 ; FILE NAME
 51> 0009 = FFT EQU 9 ; FILE TYPE
 52> 000C = FRL EQU 12 ; REEL NUMBER
 53> 000F = FRC EQU 15 ; RECORD COUNT
 54> 0020 = FCR EQU 32 ; CURRENT RECORD
 55> 0021 = FLN EQU 33 ; FCB LENGTH
 56> 001A = DEOF EQU 1AH ; CONTROL-Z (EOF)
 57> 003D = CR EQU 0DH
 58> 008A = LF EQU BAH
 59>
 60> 0007 = RSTNUM EQU 7 ; RESTART NUMBER

DEM
 CP/M VERSION _____
 COPYRIGHT © 1976
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # DEBUGGER

61>	0030	RSTLOC	EQU	RSTNUM+S	RESTART LOCATION
62>	00FF	RSTIN	EQU	BC7H OR (RSTNUM SHL 3)	RESTART INSTRUCTION
63>		;		-----	CP/M VERSION
64>		;		TEMPLATE FOR PROGRAMMED BREAK	COPYRIGHT © 1976
65>		;		-----	DIGITAL RESEARCH
66>		;		PCH : PCL	P. O. BOX 579
67>		;		HLH : HLL	PACIFIC GROVE, CA 93950
68>		;		SPH : SPL	SER. #
69>		;		RA : FLG	MZ010E1C (MINUS, ZERO, INC, EVEN, CARRY)
70>		;		B : C	
71>		;		D : E	
72>		;		-----	
73>		;		FLG FIELD:	
74>		;		-----	
75>	0005	AVAL	EQU	5	JA REGISTER COUNT IN HEADER
76>	0006	BVAL	EQU	6	
77>	0007	DVAL	EQU	7	
78>	0008	HVAL	EQU	8	
79>	0009	SVAL	EQU	9	
80>	000A	PVAL	EQU	10	
81>		;			
82>		;			
83>	0000	ORG		DEMON	/START OF DEBUGGER
84>		DEM0N		ENTRY POINTS	
85>		JMP		TRAPAD	/TRAP ADDRESS FOR RETURN IN CASE INTERRUPT
86>		JMP		BEGIN	
87>		BREAKA:			
88>	0006	JMP		BREAKP	
89>				USEFUL ENTRY POINTS FOR PROGRAMS RUNNING WITH DDT	
90>	0009	JMP		GETBUFF	/GET ANOTHER BUFFER FULL
91>	000C	JMP		GNC	/GET NEXT CHARACTER
92>	000F	JMP		PCCHAR	/PRINT A CHARACTER FROM A
93>	0012	JMP		PBYTE	/PRINT BYTE IN REGISTER A
94>	0015	JMP		PADDX	/PRINT ADDRESS IN REGISTERS D,E
95>	0018	JMP		SCAHEXP	/SCAN 0,1,2, OR 3 EXPRESSIONS
96>	001B	JMP		GETVAL	/GET VALUE TO H,L
97>		;			
98>		;			
99>		;		TRAPAD:	/GET THE RETURN ADDRESS FOR THIS JUMP TO BDOS IN CASE OF
100>					A SOFT INTERRUPT DURING BDOS PROCESSING.
101>	001E	XTHL		E3	/PC TO HL
102>	001F	SHLD		222711	RETLOC /MAY NOT NEED IT
103>	0022	XTHL		E3	
104>	0023	JMP		C30612	BDOS
105>					
106>		BEGIN:			
107>					
108>	0026	LXI		210008	H, DEMON
109>	0029	SHLD		220600	BDOS+E1 /CHANGE ENTRY POINT ADDRESS FOR DOS
110>	002C	XRA		AF	A /ZERO ACC
111>	002D	STA		322C11	BREAKS /CLEAR BREAK POINT COUNT
112>					
113>	0030	LXI		210001	H, PCB BASE
114>	0033	SHLD		220C01	DISPC /INITIAL VALUE FOR DISASSEMBLER
115>	0036	SHLD		223A11	DISLOC /INITIAL VALUE FOR DISPLAY
116>	0039	SHLD		226411	MLOAD /MAX LOAD LOCATION
117>					
118>		;		SETUP RESTART TEMPLATE	
119>	003C	SHLD		229611	PLOC
120>	003F	LXI		210001	H, SPBASE

121> 0842 319411
 122> 0845 E5
 123> 0846 210200
 124> 0849 E5
 125> 084A 2B
 126> 084B 2B
 127> 084C 229411
 128> 084F E5
 129> 0850 E5
 130> 0851 222A11
 131> 0854 222511
 132>
 133> 0857 3EC3
 134> 0859 323000
 135> 085C 210600
 136> 085F 223900
 137>
 138>
 139> 0862 3A5D00
 140> 0865 FE20
 141> 0867 CA7108
 142>
 143>
 144> 086A 210000
 145> 086D E5
 146> 086E C34100
 147>
 148>
 149>
 150>
 151>
 START:
 152> 0871 318C11
 153> 0874 CDE0D0
 154> 0877 3E2D
 155> 0879 CDEF0D
 156>
 157>
 158> 087C CD5EAD
 159>
 160> 087F CD356D
 161> 0882 FE0D
 162> 0884 CA7108
 163> 0887 D641
 164> 0892 DA5300
 165> 089C FE1A
 166> 089E D25300
 167>
 168> 0891 SF
 169> 0892 1600
 170> 0894 219E00
 171> 0897 19
 172> 0898 19
 173> 0899 SE
 174> 089A 23
 175> 089B 56
 176> 089C EB
 177> 089D E9
 178>
 179> JMPTAB: ;JUMP TABLE TO SUBROUTINES
 180> 089E E500

LXI SP,STACK-4 ;INITIAL SP
 PUSH H ;INITIAL PSW
 LXI H,10B ;INITIAL PSW
 PUSH H
 PUSH H
 DCX H
 SHLD HLOC ;H,L CLEARED
 PUSH H ;B,C CLEARED
 PUSH H ;D,E CLEARED
 SHLD TRACER ;CLEAR TRACE FLAG
 SHLD USERBRK ;CLEAR USER BREAK DURING TRACE/UNTRACE
 MVN A,0C3H ;(JMP RESTART)
 STA RSTLOC
 LXI H,BREAKA ;BREAK POINT SUBROUTINE
 SHLD RSTLOC+1 ;RESTART LOCATION ADDRESS FIELD
 CHECK FOR FILE NAME PASSED TO DEMON, AND LOAD IF PRESENT
 LDA FCB+FFN ;BLANK IF NO NAME PASSED
 CPI
 JZ START
 PUSH A ZERO, AND READ
 LXI H,0
 PUSH H
 JMP RINIT
 MAIN COMMAND LOOP
 SER. # _____
 LXI SP,STACK-12 ;INITIALIZE SP IN CASE OF ERROR
 CALL CRLF ;INITIAL CRLF
 MVN A,'-'
 CALL PCHAR ;OUTPUT PROMPT
 GET INPUT BUFFER
 CALL GETBUFF ;FILL COMMAND BUFFER
 CALL GNC ;GET CHARACTER
 CPI CR
 JZ START
 SUI 'A' ;LEGAL CHARACTER?
 JC CERROR ;COMMAND ERROR
 CPI 'Z'-'A'+1
 JNC CERROR
 CHARACTER IN REGISTER A IS COMMAND, MUST BE IN THE RANGE A-Z
 MOV E,A ;INDEX TO E
 MVN D,0 ;DOUBLE PRECISION INDEX
 LXI H,JMPTAB;BASE OF TABLE
 DAD D
 DAD D ;INDEXED
 MOV E,M ;LD BYTE
 IHX H
 MOV D,M ;HD BYTE
 XCHG ;TO H,L
 PCHL ;GONE...
 DW ASSM ;A ENTER ASSEMBLER LANGUAGE

181> 08A0 5300
 182> 08A2 3009
 183> 08A4 6009
 184> 08A6 5300
 185> 08A8 F609
 186> 08AA 000A
 187> 08AC 740A
 188> 08AE 300A
 189> 08B0 5300
 190> 08B2 5300
 191> 08B4 0109
 192> 08B6 F40A
 193> 08B8 5000
 194> 08BA 5300
 195> 08BC 5300
 196> 08BE 5300
 197> 08C0 3000
 198> 08C2 0E0C
 199> 08C4 570C
 200> 08C6 5300
 201> 08C8 5300
 202> 08CA 5300
 203> 08CC 6F0C
 204> 08CE 5300
 205> 08D0 5300
 206>
 207>
 208> OPN: ;FILE OPEN ROUTINE. THIS SUBROUTINE OPENS THE DISK INPUT
 209> 08D2 E5
 210> 08D3 D5
 211> 08D4 C5
 212> 08D5 AF
 213> 08D6 325000
 214> 08D9 0E0F
 215> 08DB 115C00
 216> 08DE CD1E00
 217> 08E1 C1
 218> 08E2 D1
 219> 08E3 E1
 220> 08E4 C9
 221>
 222> ASSM: ;ASSEMBLER LANGUAGE INPUT
 223>
 224> 08E5 210901
 225> 08E8 CD150B
 226> 08EB D25300
 227>
 228> 08EE CD380E
 229> 08F1 3D
 230> 08F2 C2530D
 231> 08F5 CD6E0E
 232> 08F8 220C01
 233> 08FB CD0901
 234> 08FE C37100
 235>
 236> LASSM: ;ASSEMBLER LANGUAGE OUTPUT LISTING
 237>
 238>
 239> 240> 0901 CD270B

DW CERROR ;B
 DW CALLPR ;C CALL PROGRAM
 DW DISPLAY ;D DISPLAY RAM MEMORY
 DW CERROR ;E
 DW FILL ;F FILL MEMORY
 DW GOTO ;G GO TO MEMORY ADDRESS
 DW HEXARI ;H HEXADECIMAL SUM AND DIFFERENCE
 DW INFCB ;I FILL INPUT FILE CONTROL BLOCK
 DW CERROR ;J
 DW CERROR ;K
 DW LASSM ;L LIST ASSEMBLY LANGUAGE
 DW MOVE ;M MOVE MEMORY
 DW CERROR ;N
 DW CERROR ;O
 DW CERROR ;P
 DW CERROR ;Q
 DW READ ;R READ HEXADECIMAL FILE
 DW SETHEM ;S SET MEMORY COMMAND
 DW TRACE ;T
 DW UNTRACE ;U
 DW CERROR ;V
 DW CERROR ;W
 DW EXAMINE ;X EXAMINE AND MODIFY REGISTERS
 DW CERROR ;Y
 DW CERROR ;Z

STA DBP ;CLEAR BUFFER POINTER
 CPI... VERSION
 COPYRGHT © 1976
 DIGITAL RESEARCH
 P. O. BOX 379
 PACIFIC GRO. E. CA 93950

LXI D,DFCB
 CALL TRAPAD ;TO BDS
 POP B
 POP D
 POP H
 RET

LXI H,ASSEM ;BASE OF ASSEMBLER
 CALL COMLOAD ;COMPARE AGAINST MLOAD
 JNC CERROR ;NOT THERE

CALL SCANEXP ;SCAN THE EXPRESSIONS WHICH FOLLOW
 DCR A ;ONE EXPRESSION EXPECTED
 JNZ CERROR
 CALL GETVAL ;GET EXPRESSION TO H,L
 SHLD DISPC
 CALL ASSEM
 JMP START

LXI H,ASSEM ;BASE OF ASSEMBLER
 CALL COMLOAD ;COMPARE AGAINST MLOAD
 JNC CERROR ;NOT THERE

LCR > LISTS FROM CURRENT DISASSM PC FOR SEVERAL LINES
 L<NUMBER><CR> LISTS FROM <NUMBER> FOR SEVERAL LINES
 L<NUMBER>,<NUMBER> LISTS BETWEEN LOCATIONS
 CALL CHKDIS ;DISASSM PRESENT?

```

241> 0904 D2538D      JNC  CERROR
242>
243> 0907 CD398E      CALL  SCANEKP ;SCAN EXPRESSIONS WHICH FOLLOW
244> 090A CH2309      JZ   SPAGE ;DRANCH IF NOT EXPRESSIONS
245> 090D CD8E0E      CALL  GETVAL ;EXP1 TO H,L
246> 0910 220C01      SHLD DISPC ;SETS BASE PC FOR LIST
247> 0913 3D          DCR  A    ;ONLY EXPRESSION?
248> 0914 CA2509      JZ   SPAGE ;SETS SINGLE PAGE MODE
249>
250>           ;ANOTHER EXPRESSION FOLLOWS
251> 0917 CD8E0E      CALL  GETVAL
252> 091A 220E01      SHLD DISPM ;SETS MAX VALUE
253> 091D 3D          DCR  A    ;CLEAR PAGE MODE
254> 091E C2538D      JHZ  CERROR ;ERROR IF MORE EXPN'S
255> 0921 AF          XRA  A    ;CLEAR PAGE MODE
256> 0922 C32709      JMP  SPAGE
257>
258> 0925 3E0C      SPACE, MVI A,PSIZE ;SCREEN SIZE FOR LIST
259> 0927 321001      SPAGE0 STA  DISPG
260> 092A CD0601      CALL  DISEN ;CALL DISASSEMBLER
261> 092D C37108      JMP  START ;FOR ANOTHER COMMAND
262>
263>           ;DISPLAY MEMORY, FORMS ARE
264>           ;D      DISPLAY FROM CURRENT DISPLAY LINE
265>           ;DHNN      SET DISPLAY LINE AND ASSUME D
266>           ;DHNN,MMM      DISPLAY HNN TO MMM
267>           ;NEW DISPLAY LINE IS SET TO NEXT TO DISPLAY
268>
269> CALLPR:          ;CALL USER PROGRAM FROM DDT
270>
271> 0930 CD380E      CALL  SCANEKP
272> 0933 IA538D      JC   CERROR ;CANNOT BE ,XXX
273> 0935 CA538D      JZ   CERROR ;CANNOT BE C ALONE
274> 0939 CD8E0E      CALL  GETVAL ;ADDRESS TO CALL IN H,L
275> 093C E5          PUSH H    ;READY FOR CALL
276>
277>           ;GET REMAINING PARAMETERS
278>           ;REG-A CONTAINS 1,2,OR 3 CORRESPONDING TO NUMBER OF VALUES
279> 093D 010000      LXI  B,0
280> 0940 3D          DCR  A
281> 0941 C24909      JNZ  CALL0
282> 0944 C5          PUSH B
283> 0945 C5          PUSH B
284> 0946 C35909      JMP  CALL2
285> CALL0:          ;AT LEAST ONE PARAMETER
286>           ;CALL GETVAL
287> 094C E5          PUSH H
288> 094D 3D          DCR  A
289> 094E C25509      JNZ  CALL1
290>
291> 0951 C5          PUSH B
292> 0952 C35909      JMP  CALL2
293> CALL1:          ;MUST BE TWO PARAMETERS FOR THE CALL
294>           ;CALL GETVAL
295> 0958 E5          PUSH H
296> CALL2:          ;SET UP PARAMETERS IN B,C AND D,E
297> 0959 D1          POP  D    ;RECALL SECOND PARAMETER
298> 095A C1          POP  B    ;RECALL FIRST PARAMETER
299>
300>           ;READY FOR THE USER PROGRAM CALL
301>           ;LXI  H,START ;RETURN ADDRESS
301> 095E E3          XTHL
302> 095F E9          PCHL
303>
304>           ;DISPLAY:
305> 0960 CD380E      CALL  SCANEKP ;GET 0,1,OR 2 EXPNS
306> 0963 CA7F09      JZ   DISP1 ;ASSUME CURRENT DISLOC
307> 0966 CD8E0E      CALL  GETVAL ;GET VALUE TO H,L
308> 0969 DA6F09      JC   DISFO ;CARRY SET IF ,B FORM
309> 096C 223A11      SHLD DISLOC ;OTHERWISE DISPC ALREADY SET
310>           ;DISP0: ;GET NEXT VALUE
311> 096F E67F      ANI  7FH ;IN CASE ,B
312> 0971 3D          DCR  A
313> 0972 CA7F09      JZ   DISP1 ;SET HALF PAGE MODE
314> 0975 CD8E0E      CALL  GETVAL
315> 0978 3D          DCR  A    ;A,B,C NOT ALLOWED
316> 0979 C2538D      JNZ  CERROR
317> 097C C38A09      JMP  DISP2
318>
319>           ;DISP1: ;0 DR 1 EXPN, DISPLAY HALF SCREEN
320> 097F 2A3A11      LHLD DISLOC
321> 0982 7D          MOV  A,L
322> 0983 E6F0      AHI  8F0H ;NORMALIZE TO LINE START
323> 0985 6F          MOV  L,A
324> 0986 11BF00      LXI  D,PSIZE*16-1
325> 0989 19          DAD  D
326> 098A 223C11      SHLD DISMAX
327>           ;DISP2: ;DISPLAY MEMORY FROM DISLOC TO DISMAX
328> 098D CDDB0D      CALL  CRLF
329> 0990 CDC70D      CALL  BREAK ;BREAK KEY?
330> 0993 C27108      JNZ  START ;STOP CURRENT EXPANSION
331> 0996 2A3A11      LHLD DISLOC
332> 0999 223E11      SHLD TDISP
333> 099C CD600D      CALL  PADD ;PRINT LINE ADDRESS
334> 099F CD600D      CALL  BLANK
335> 09A2 7E          MOV  A,M ;GET NEXT DATA BYTE
336> 09A3 CDAD0D      CALL  POYTE ;PRINT BYTE
337> 09A6 23          INX  H
338> 09A7 CDDE0D      CALL  DISCOM ;COMPARE H,L WITH DISMAX
339> 09AA DAB509      JC   DISCH ;CARRY SET IF H,L > DISMAX
340> 09AD 7D          MOV  A,L ;CHECK FOR LINE OVERFLOW
341> 09AE E60F      ANI  8FH
342> 09B0 C29F09      JNZ  DISP4 ;JUMP FOR ANOTHER BYTE
343>
344>           ;DISCH: ;DISPLAY AREA IN CHARACTER FORM
345> 09B3 223A11      SHLD DISLOC ;UPDATE FOR NEXT WRITE
346> 09B6 2A3E11      LHLD TDISP
347> 09B9 EB          XCHG
348> 09BA CD6D0D      CALL  BLANK
349>
350> 09BD 1A          DISCH0: LDAX D ;GET BYTE
351> 09BE CDDE0D      CALL PGGRAPH ;PRINT IF GRAPHIC CHARACTER
352> 09C1 13          INX  D
353> 09C2 2A3A11      LHLD DISLOC ;COMPARE FOR END OF LINE
354> 09C5 7D          MOV  A,L
355> 09C6 93          SUB  E
356> 09C7 C2BD09      JNZ  DISCH0
357> 09CA 7C          MOV  A,H
358> 09CB 92          SUB  D
359> 09CC C2BD09      JNZ  DISCH0
360>

```

CPM VERSION
 COPYRIGHT © 1976
 DIGITAL RESEARCH
 P.O. BOX 573
 PACIFIC GROVE, CA 93950
 SER. #

```

361> ; DROP THRU AT END OF CHARACTERS
362> 09CF 2A3A11 ;LHLD DISLOC
363> 09D2 CDE0D0 ;CALL DISCOM ;END OF DISPLAY?
364> 69D5 DA7100 ;JC START
365>
366> ; NO, CONTINUE WITH NEXT LINE
367> 09D8 C38D09 ;JMP DISP3
368>
369>
370> ; FILL MEMORY AREA WITH FIXED DATA ELEMENT
371>
372> SCAN3: ;SCAN THREE EXPN'S FOR FILL AND MOVE
373> 09D8 CD380E ;CALL SCANEXP
374> 09DE FE03 ;CPI 3
375> 09E0 C2530D ;JHZ CERROR
376> 09E3 CD0E0E ;CALL GETVAL
377> 09E6 E5 ;PUSH H
378> 09E7 CD0E0E ;CALL GETVAL
379> 09EA E5 ;PUSH H
380> 09EB CD0E0E ;CALL GETVAL
381> 09EE D1 ;POP D
382> 09EF C1 ;POP B ;BC, DE, HL
383> 09F0 C9 ;RET
384>
385> BCDE: ;COMPARE BC > DE (CARRY GEN'D IF TRUE)
386> 09F1 7B ;MOV A, E
387> 09F2 91 ;SUB C
388> 09F3 7A ;MOV A, D
389> 09F4 98 ;SSB B
390> 09F5 C9 ;RET
391>
392> ; FILL:
393> 09F6 CDD809 ;CALL SCAN3 ;EXPRESSIONS SCANNED BC , DE , HL
394> 09F9 7C ;MOV A, H ;MUST BE ZERO
395> 09FA B7 ;ORA A
396> 09FB C2530D ;JHZ CERROR
397> 09FE CDF109 ;CALL BCDE ;END OF FILL?
398> 0A01 DA7100 ;JC START
399> 0A04 7D ;MOV A, L ;DATA
400> 0A05 02 ;STAX B ;TO MEMORY
401> 0A06 03 ;IHX B ;NEXT TO FILL
402> 0A07 C3FE09 ;JMP FILLO
403>
404> ; GO COMMAND WITH OPTIONAL BREAKPOINTS
405>
406> GOTO: ;GOTO
407> 0A0A C05D0D ;CALL CRLF ;READY FOR GO.
408> 0A0D CD380E ;CALL SCANEXP ;0,1, OR 2 EXPs
409> 0A10 CD0E0E ;CALL GETVAL
410> 0A13 E5 ;PUSH H ;START ADDRESS
411> 0A14 CD0E0E ;CALL GETVAL
412> 0A17 E5 ;PUSH H ;BKPT1
413> 0A18 CD0E0E ;CALL GETVAL
414> 0A1B 44 ;MOV B, H ;BKPT2
415> 0A1C 4D ;MOV C, L
416> 0A1D D1 ;POP D ;BKPT1
417> 0A1E E1 ;POP H ;GOTO ADDRESS
418>
419> COPR: ;DI
420> 0A1F F3

```

CP/M VERSION
 COPYRIGHT © 1976
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA. 93950
 SER. # _____

CP/M VERSION
 COPYRIGHT © 1976
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA. 93950
 SER. # _____

```

421> 0A20 CA3B0A ;JZ GOP1 ;NO BREAK POINTS
422> 0A23 DA290A ;JC GOP0
423> ;SET PC
424> 0A26 229611 ;SHLD PLOC ;INTO MACHINE STATE
425> ;SET BREAKS
426> 0A29 E67F ;ANI 7FH ;CLEAR . BIT
427> 0A2B 3D ;DCR A ;IF 1 THEN SKIP (2,3 IF BREAKPOINTS)
428> 0A2C CA3B0A ;JZ GOP1
429> 0A2F CD4C0A ;CALL SETBK ;BREAK POINT FROM D.E
430> 0A32 3D ;DCR A
431> 0A33 CA3B0A ;JZ GOP1
432> ;SECOND BREAK POINT
433> 0A36 59 ;MOV E, C
434> 0A37 50 ;MOV D, B ;TO D.E
435> 0A38 CD4C0A ;CALL SETBK ;SECOND BREAK POINT SET
436> ;GOP1: ;RESTORE MACHINE STATE AND START IT
437> ;LXI SP, STACK-12
438> 0A3B 318C11 ;POP D
439> 0A3E D1 ;POP B
440> 0A3F C1 ;POP PSW
441> 0A40 F1 ;POP H ;JSP IN HL
442> 0A41 E1 ;SPHL
443> 0A42 F9 ;LHLD PLOC ;JPC IN HL
444> 0A43 2A9611 ;PUSH H ;INTO USER'S STACK
445> 0A46 E5 ;LHLD HLOC ;JHL RESTORED
446> 0A47 2A9411 ;EI
447> 0A48 FB ;RET
448> 0A4B C9
449>
450> ;SETBK: ;SET BREAK POINT AT LOCATION D.E
451> 0A4C F5 ;PUSH PSW
452> 0A4D C5 ;PUSH B ;NUMBER OF BREAKS SET SO FAR
453> 0A4E 212C11 ;LXI H, BREAKS
454> 0A51 7E ;MOV A, M
455> 0A52 34 ;INR M ;COUNT BREAKS UP
456> 0A53 B7 ;ORA A ;ONE SET ALREADY?
457> 0A54 CA670A ;JZ SETBK0
458> ;ALREADY SET, MOVE PAST ADDR, DATA FIELDS
459> 0A57 23 ;INX H
460> 0A58 7E ;MOV A, M ;CHECK = ADDRESSES
461> 0A59 23 ;INX H
462> 0A5A 46 ;MOV B, M ;CHECK NO ADDRESS
463> 0A5B 23 ;INX H
464> ;DON'T SET TWO BREAKPOINTS IF EQUAL
465> 0A5C BB ;CMP E ;LDW =?
466> 0A5D C2670A ;JHZ SETBK0
467> 0A60 78 ;MOV A, B
468> 0A61 BA ;CMP D ;HIGH =?
469> 0A62 C2670A ;JHZ SETBK0
470> ;EQUAL ADDRESSES, REPLACE REAL DATA
471> 0A65 7E ;MOV A, M ;GET DATA BYTE
472> 0A66 12 ;STAX D ;PUT BACK INTO CODE
473> 0A67 23 ;SETBK0: ;ADDRESS FIELD
474> 0A68 73 ;INX H
475> 0A69 23 ;MOV H, E ;LSB
476> 0A6A 72 ;INX H
477> 0A6B 23 ;MOV H, D ;MSB
478> 0A6C 1A ;INX H ;DATA FIELD
479> 0A6D 77 ;LDAX D ;GET BYTE FROM PROGRAM
480> 0A6E 3EFF ;MOV H, A ;TO BREAKS VECTOR
        ;MVI A, RSTIN ;RESTART INSTRUCTION

```

```

481> 0A70 12      STAX D      ;TO CODE
482> 0A71 C1      POP B
483> 0A72 F1      POP PSW
484> 0A73 C9      RET
485>
486>
487> ;HEXADECIMAL ARITHMETIC
488> HEXARI: CALL SCANEXP
489> CPI 2
490> JHZ CERROR
491> CALL GETVAL ;FIRST VALUE TO H,L
492> PUSH H
493> CALL GETVAL ;SECOND VALUE TO H,L
494> POP D ;FIRST VALUE TO D,E
495> PUSH H ;SAVE A COPY OF SECOND VAALUE
496> 0A83 D1
497> 0A84 E5
498> 0A85 CDBD0D
499> 0A88 19
500> 0A89 CDD60D
501> 0A8C CD6D0D
502> 0A8F E1
503> 0A90 AF
504> 0A91 95
505> 0A92 6F
506> 0A93 3E00
507> 0A95 9C
508> 0A96 67
509> 0A97 19
510> 0A98 CDD60D
511> 0A9B C37100
512>
513> ;SET INPUT FILE CONTROL BLOCK (AT 5CH) TO SIMULATE CONSOLE COMMAND
INFCB:
514> ;FILL FCB AT 5CH
515> XRA A
516> STA FCB+FCR ;CLEAR CURRENT RECORD
517> 0A9F 327C00
518> 0AA2 325C00
519> 0AA5 CD350D
520> 0AA8 0E09
521> 0AAA 215D00
522>
523> FLP: ;FILL NAME
524> MOV H,A
525> INX H
526> DCR C
527> JZ CERROR ;FILE NAME TOO LONG.
528>
529> 0A63 CD350D
530> 0A86 FE2E
531> 0A88 CAC00A
532> ;NOT .. MAY BE CR
533> 0A8B FE0D
534> 0A8D C2A00A
535>
536> ;NAME FILLED, EXTEND WITH BLANKS
537> 0AC0 0D
538> 0AC1 CACABA
539> 0AC4 3620
540> 0AC6 23

```

CP/M VERSION
 COPYRIGHT © 1976
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. #

<pre> 541> 0AC7 C3C00A 542> ;JMP FLB 543> ;BLANKS FILLED, SCAN FILE TYPE IF '.' FOUND 544> 0ACA 0E04 TFT 545> 0ACC FE2E 546> 0ACE C2E50A 547> 548> 549> 0AD1 216500 550> ;SCAN FILE TYPE 551> 0AD4 CD850D FLP1: CALL GNC 552> 0AD7 FE0D CPI CR 553> 0AD9 CAE50A JZ FLB1 554> 0ADC 77 MOV M,A 555> 0ADD 23 INX H 556> 0ADE 0D DCR C 557> 0ADF CA530D JZ CERROR ;TOO LONG 558> 0AE2 C3D40A JMP FLP1 559> 560> ;FILL WITH BLANKS 561> 0AE5 0D FLP1: DCR C 562> 0AE6 CAEF0A JZ FLZ 563> 0AE9 3620 MOV M,' ' 564> 0AEB 23 INX H 565> 0AEC C3E50A JMP FLB1 566> 567> ;ZERO THE EXTENT 568> 0AEF 3600 FLZ: MOVI M,0 569> 0AF1 C37100 JMP START 570> 571> ;MOVE MEMORY 572> MOVE: CALL SCAN3 ;BC,DE,HL 573> 0AF4 CDDB09 MOVEB: ;HAS B,C PASSED D,E? 574> ;CALL BCDE 575> 0AF7 CDF109 JC START ;END OF MOVE 576> 0AF8 DA7108 LDAX B ;CHAR TO ACCUM 577> 0AFD 0A INX B ;NEXT TO GET 578> 0AFE 03 MOV M,A ;MOVE IT TO MEMORY 579> 0AFF 77 INX H 580> 0B00 23 JMP MOVE0 ;FOR ANOTHER 581> 0B01 C3F70A 582> 583> ;READ FILES (HEX OR COM) 584> 585> QCOM: ;COM FILE IF ZERO AT END 586> 0B04 216500 LXI H,FCB+FFT 587> 0B07 7E MOVI A,M 588> 0B08 FE43 CPI 'C' 589> 0B0A C0 RHZ 590> 0B0B 23 INX H 591> 0B0C 7E MOV A,M 592> 0B0D FE4F CPI '0' 593> 0B0F C0 RHZ 594> 0B10 23 INX H 595> 0B11 7E MOVI A,M 596> 0B12 FE4D CPI 'M' 597> 0B14 C9 RET 598> 599> ;COMLOAD: ;COMPARE HL > MLOAD 600> 0B15 EB XCHG ;H,L TO D,E </pre>	CP/M VERSION COPYRIGHT © 1976 DIGITAL RESEARCH P. O. BOX 579 PACIFIC GROVE, CA 93950 SER. #
---	---

```

601> 0B16 2A6411      LHLD   MLOAD ;MLOAD TO H,L
602> 0B19 7D          MOV    A,L  ;MLOAD LSB
603> 0B1A 93          SUB    E
604> 0B1B 7C          MOV    A,H
605> 0B1C 9A          SBB    D
606> 0B1D EB          XCHG
607> 0B1E C9          RET
608>
; CKMLOAD,           ;CHECK FOR HL > MLOAD AND SET MLOAD IF SO
609> 0B1F CD1508      CALL   COMLOAD ;CARRY IF HL>MLOAD
610> 0B22 30          RNC
611> 0B23 226411      SHLD   MLOAD ;CHANGE IT
612> 0B26 C9          RET
613>
; CHKDIS,           ;CHECK FOR DISASSM PRESENT
614> 0B27 E5          PUSH   H
615> 0B28 210601      LXI    H,DISEN ;ENTRY POINT
616> 0B29 CD1508      CALL   COMLOAD
617> 0B2E E1          POP    H
618> 0B2F C9          RET
619>
; READ,              CALL   SCAHEXP
620> 0B30 CD380E      LXI    H,B
621> 0B33 210000      JZ    READN
622> 0B36 CA400B      DCR    A ;ONE EXPRESSION?
623> 0B39 3D          JNZ    CERROR
624> 0B3A C2530D      CALL   GETVAL ;EXPRESSION TO H,L
625> 0B3D CD6E0E      PUSH   H ;SAVE IT FOR BELOW
626> 0B40 E5          RINIT, CALL   OPH  ;OPEN INPUT FILE
627> 0B41 CD1208      CPI    255
628> 0B44 FEFF      JZ    CERROR
629> 0B46 CA530D      JCONT  IF FILE OPEN WENT OK
630> 0B47 C2750B      DISK   FILE OPENED AND INITIALIZED
631>
; CHECK FOR 'COM' FILE AND LOAD DIRECT TIL EOF
632> 0B49 CD0408      CALL   QCOM ;LOOK FOR 'COM'
633> 0B4C C2750B      JHZ    HREAD
634>
; COM FILE, LOAD WITH OFFSET GIVEN BY PUSHED REGISTER H
635> 0B4F E1          POP    H
636> 0B50 110601      LXI    D,100H ;BASE OF TRANSIENT AREA
637> 0B53 19          DAD    D
638>
; LCOM8,             REG H HOLDS LOAD ADDRESS
639> 0B54 E5          PUSH   H ;SAVE DMA ADDRESS
640> 0B55 115C00      LXI    D,DFCB
641> 0B58 0E14          MVF    C,RDF ;READ SECTOR
642> 0B5A CD1E08      CALL   TRAPAD
643> 0B5D E1          POP    H
644> 0B5E B7          ORA    A ;SET FLAGS TO CHECK RETURN CODE
645> 0B5F C2DA0B      JNZ    RLIFT
646>
; MOVE FROM 0BH TO LOAD ADDRESS IN H,L
647> 0B62 110000      LXI    D,DBF
648> 0B65 0E60          MVF    C,00H ;BUFFER SIZE
649> 0B67 14          LDAX   D ;LOAD NEXT BYTE
650> 0B68 13          IHX
651> 0B69 77          MOV    M,A ;STORE NEXT BYTE
652> 0B6A 23          IHX
653> 0B6B 00          DCR    C
654>
; LCOM1,             ;CHECK ADDRESS AGAINST MLOAD
655> 0B6C C2670B      JNZ    LCOM1
656> 0B6D CD1F0B      CALL   CKMLOAD
657> 0B6E C3540B      JMP    LCOM8
658>
; OTHERWISE ASSUME HEX FILE IS BEING LOADED
659> 0B6F CD1C0D      HREAD, CALL   DISKR ;NEXT CHAR TO ACCUM
660> 0B70 FE1A          CPI    DEOF ;PAST END OF TAPE?
661> 0B71 DE3A          JZ    CERROR ;FOR ANOTHER COMMAND
662> 0B72 C2750B      SBI    '
663> 0B73 HREAD,        JNZ    HREAD ;LOOKING FOR START OF RECORD
664>
; START FOUND, CLEAR CHECKSUM
665> 0B74 0882 57          MOV    D,A
666> 0B75 0883 E1          POP    H
667> 0B76 0884 E5          PUSH   H
668> 0B77 0885 CDBA0B      CALL   RBYTE
669> 0B78 0883 5F          MOV    E,A ;SAVE LENGTH
670> 0B79 0889 CDBA0B      CALL   RBYTE ;HIGH ORDER ADDR
671> 0B7A 088C F5          PUSH   PSW
672> 0B7B 0881 CDBA0B      CALL   RBYTE ;LOW ORDER ADDR
673> 0B7C 0880 C1          POP    B
674> 0B7D 0881 4F          MOV    C,A
675> 0B7E 0892 09          DAD    B ;BIASED ADDR IN H
676> 0B7F 0893 7B          MOV    A,E ;CHECK FOR LAST RECORD
677> 0B7G 0894 B7          ORA    A
678> 0B7H 0895 C2A00B      JNZ    RDTYPE
679> 0B7I 0898 60          END OF TAPE, SET LOAD ADDRESS
680> 0B7J 0899 69          MOV    H,B
681> 0B7K 089A 229611      MOV    L,C
682> 0B7L 089D C3DA0B      SHLD   PLOC ;SET PC VALUE
683> 0B7M 08A0 CDBA0B      JMP    RLIFT ;FOR ANOTHER COMMAND
684>
; RDTYPE,            CALL   RBYTE ;RECORD TYPE = 0
685> 0B7N 08A3 CDBA0B      ;LOAD RECORD
686> 0B7O 08A6 77          RED1,  CALL   RBYTE
687> 0B7P 08A7 23          MOV    M,A
688> 0B7Q 08A9 1D          INX    H
689> 0B7R 08A9 C2A30B      DCR    E
690> 0B7S 08B1 40          JNZ    RED1 ;FOR ANOTHER BYTE
691> 0B7T 08B2 40          OTHERWISE AT END OF RECORD - CHECKSUM
692> 0B7U 08B3 F1          CALL   RBYTE
693> 0B7V 08B4 C2530D      PUSH   PSW ;FOR CHECKSUM CHECK
694> 0B7W 08B7 C3750B      CALL   CKMLOAD ;CHECK AGAINST MLOAD
695> 0B7X 08B8 C5          POP    PSW
696> 0B7Y 08B9 E5          JZ    CERROR ;CHECKSUM ERROR
697> 0B7Z 08B0 C5          JMP    HREAD ;FOR ANOTHER RECORD
698>
; RBYTE,              ;READ ONE BYTE FROM BUFF AT WBP TO REG-A
699> 0B80 08B0 C5          COMPUTE CHECKSUM IN REG-D
700> 0B81 08B1 E5          PUSH   B
701> 0B82 08B2 D5          PUSH   H
702> 0B83 08B3 40          PUSH   D
703> 0B84 08B4 C2530D      ;READ ONE BYTE FROM BUFF AT WBP TO REG-A
704> 0B85 08B5 C5          CALL   DISKR ;GET ONE MORE CHARACTER
705> 0B86 08B6 E5          CALL   HEXCON ;CONVERT TO HEX (OR ERROR)
706> 0B87 08B7 C5          ;COMPUTE CHECKSUM IN REG-D
707> 0B88 08B8 E5          PUSH   B
708> 0B89 08B9 D5          PUSH   H
709> 0B8A 08B0 C5          PUSH   D
710> 0B8B 08B1 E5          ;READ ONE BYTE FROM BUFF AT WBP TO REG-A
711> 0B8C 08B2 D5          CALL   DISKR ;GET ONE MORE CHARACTER
712> 0B8D 08B3 40          CALL   HEXCON ;CONVERT TO HEX (OR ERROR)
713> 0B8E 08B4 C2530D      ;COMPUTE CHECKSUM IN REG-D
714> 0B8F 08B5 C5          PUSH   B
715> 0B90 08B6 E5          PUSH   H
716> 0B91 08B7 D5          PUSH   D
717> 0B92 08B8 C5          ;READ ONE BYTE FROM BUFF AT WBP TO REG-A
718> 0B93 08B9 E5          CALL   DISKR ;GET ONE MORE CHARACTER
719> 0B94 08B0 C5          CALL   HEXCON ;CONVERT TO HEX (OR ERROR)
720>

```

CPI. VERSION
COPYRIGHT © 1976
DIGITAL RESEARCH
P. O. BOX 519
PACIFIC GROVE, CA 93950

SER. #

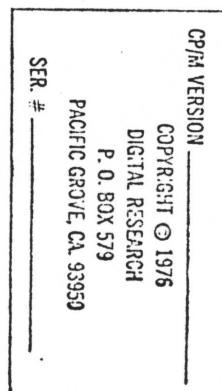
721> SHIFT LEFT AND MASK
 722> 0BC3 07 RLC
 723> 0BC4 07 RLC
 724> 0BC5 07 RLC
 725> 0BC6 07 RLC
 726> 0BC7 E6F0 ANI 0F0H
 727> 0BC9 F5 PUSH PSW ;SAVE FOR A FEW STEPS
 728> 0BCA CD1C0D CALL DISKR
 729> 0BCD CD010E CALL HEXCON
 730>
 731> OTHERWISE SECOND NIBBLE OK, SO MERGE
 732> 0BD0 C1 POP B ;PREVIOUS NIBBLE TO REG-B
 733> 0BD1 B0 ORA B
 734> 0BD2 47 MOV B,A ;VALUE IS NOW IN B TEMPORARILY
 735> 0BD3 D1 POP D ;CHECKSUM
 736> 0BD4 82 ADD D ;ACCUMULATING
 737> 0BD5 57 MOV D,A ;BACK TO CS
 738> ZERO FLAG REMAINS SET
 739> 0BD6 78 MOV A,B ;BRING BYTE BACK TO ACCUMULATOR
 740> 0BD7 E1 POP H
 741> 0BD8 C1 POP B ;BACK TO INITIAL STATE WITH ACCUM SET
 742> 0BD9 C9 RET
 743> RLIFT: ;LIFT HEAD ON DISK BEFORE RETURNING
 744> 0BD4 0EBC MVI C,LIFT
 745> 0BD5 CD1E0B CALL TRAPAD
 746>
 747> 0BD7 21030C LXI H,LMSG ;LOAD MESSAGE
 748> 0BE2 7E RL10: MOV A,M ;LAST CHAR
 749> 0BE3 B7 ORA A ;LAST CHAR?
 750> 0BE4 CAEE0B JZ RLI1
 751> 0BE7 CD6F0D CALL PCHAR
 752> 0BEA 23 INH H ;NEXT CHAR
 753> 0BE8 C3E20B JMP RLIB
 754> 0BEE CDB10D RL11: CALL CRLF
 755> 0EF1 2A6411 LHLD MLOAD
 756> 0EF4 CDD60D CALL PADDR
 757> 0EF7 CD610D CALL BLANK
 758> 0BFA 2A9611 LHLD PLOC
 759> 0BFD CDD60D CALL PADDR
 760> 0C00 C37108 JMP START
 761> 0C03 0D0A4E4558LMSG: DB CR,LF,'NEXT PC',0
 762>
 763> ;SET MEMORY COMMAND
 764>
 765> SETMEM: ;ONE EXPRESSION EXPECTED
 766> 0C0E CD380E CALL SCANEWP ;SETS FLAGS
 767> 0C11 3D DCR A ;ONE EXPRESSION ONLY
 768> 0C12 C2530D JNZ CERROR
 769> 0C15 CD0E0E CALL GETVAL ;START ADDRESS IS IN H,L
 770> 0C18 CDB10D SETM0: CALL CRLF ;NEW LINE
 771> 0C1B E5 PUSH H ;SAVE CURRENT ADDRESS
 772> 0C1C CDD60D CALL PADDR ;PRINTED
 773> 0C1F C0610D CALL BLANK ;SEPARATOR
 774> 0C22 E1 POP H ;GET DATA
 775> 0C23 7E MOV A,M
 776> 0C24 E5 PUSH H ;SAVE ADDRESS TO FILL
 777> 0C25 CDA00D CALL PBYTE ;PRINT BYTE
 778> 0C28 CDG10D CALL BLANK ;ANOTHER SEPARATOR
 779> 0C2B CD5E0D CALL GETBUFF ;FILL INPUT BUFFER
 780> 0C2E CD850D CALL GNC ;MAY BE EMPTY (NO CHANGE)
 781> 0C31 E1 POP H ;RESTORE ADDRESS TO FILL
 782> 0C32 FE0D CPI CR
 783> 0C34 CA4F0C JZ SETM1
 784> 0C37 FE2E CPI ''
 785> 0C39 CA7100 JZ START
 786> DATA IS BEING CHANGED
 787> 0C3C E5 PUSH H ;SAVE ADDR TO FILL
 788> 0C3D CD380E CALL SCANEWP ;FIRST CHARACTER ALREADY SCANNED
 789> 0C40 3D DCR A ;ONE ITEM?
 790> 0C41 C2530D JNZ CERROR ;MORE THAN ONE
 791> 0C44 CD0E0E CALL GETVAL ;VALUE TO H,L
 792> 0C47 7C MOV A,H
 793> 0C48 B7 ORA A ; ;HO ZERO?
 794> 0C49 C2530D JNZ CERROR ;DATA IS IN L
 795> 0C4C 7D MOV A,L
 796> 0C4D E1 PDP H ;RESTORE DATA VALUE
 797> 0C4E 77 MOV M,A
 798> 0C4F 23 SETM1: INX H ;NEXT ADDRESS READY
 799> 0C50 C3180C JMP SETM0
 800>
 801> ;UNTRACE MODE
 802> UNTRACE:
 803> 0C53 AF XRA A ;CLEAR TRACE MODE FLAG
 804> 0C54 C3590C JMP ETRACE
 805>
 806> ;START TRACE
 807> 0C57 3EFF TRACE: MVI A,0FFH ;SET TRACE MODE FLAG
 808> ETRACE:
 809> 0C59 322911 STA TMODE
 810> 0C5C CD300E CALL SCANEWP
 811> 0C5F 210000 LXI H,0
 812> 0C62 222511 SHLD USERBRK ;CLEAR USERBRK
 813> 0C65 CA860C JZ TRAC0
 814> ;EXPRESSIONS WERE GIVEN, FORMS ARE
 815> 0C67 3000 TX TRACE FOR X STEPS ACC = 1
 816> 0C68 222511 TX,BRK TRACE FOR X STEPS. CALL "BRK" AT EACH STOP ACC=2
 817> 0C69 3000 T,BRK CALL "BRK" ACC = 1, CY = 1
 818>
 819> 0C6B DA760C JC SETTR0
 820> 0C6B CD0E0E CALL GETVAL ;TO H,L
 821> 0C6E F5 PUSH PSW
 822> 0C6F 7D MOV A,L ;CHECK FOR ZERO
 823> 0C70 B4 ORA H
 824> 0C71 C2530D JZ CERROR
 825> 0C74 F1 POP PSW ;RECALL NUMBER OF PARAMETERS
 826> 0C75 2B DCX H ;TRACE VALUE - 1
 827> 0C76 E5 SETTR0: ;H,L CONTAINS TRACE COUNT, SAVE IT FOR LATER
 828> 0C76 E5 PUSH H
 829>
 830> 0C77 3D LOOK FOR BREAK ADDRESS
 831> 0C78 CA850C DCR A ;IF ONLY ONE SPECIFIED, THEN SKIP USERBRK
 832> 0C78 3D JZ SETTR1
 833> 0C7C C2530D DCR A ;MUST BE TWO VALUES
 834> 0C7F CD0E0E JNZ CERROR ;MORE THAN TWO SPECIFIED
 835> 0C82 222511 CALL GETVAL ;VALUE TO H,L
 836>
 837> 0C85 E1 SHLD USERBRK
 838> 0C86 222A11 SETTR1: ;RECALL TRACE COUNT
 839> 0C89 CDEC0E POP H
 840> 0C8C C31F0A TRAC0: SHLD TRACER
 841> 0C8D 3000 CALL DSTATE ;STARTING STATE IS DISPLAYED
 842> 0C8E C31F0A JMP GOPR ;SETS BREAKPOINTS AND STARTS EXECUTION

CP/M VERSION
 COPYRIGHT © 1976
 DIGITAL RESEARCH
 P.O. BOX 579
 PACIFIC GROVE, CA. 93950
 SER. #

```

841> ; EXAMINE AND MODIFY CPU REGISTERS.
842> EXAMINE:
843>     CALL  GNC    ;CR?
844>     CPI   CR
845>     JHZ   EXAM0
846>     CALL  DSTATE ;DISPLAY CPU STATE
847>     JMP   START
848>     ; EXAM0: ;REGISTER CHANGE OPERATION
849>     LXI   B, PVAL+1 ;B=0, C=PVAL (MAX REGISTER NUMBER)
850>     ; LOOK FOR REGISTER MATCH IN RVECT
851>     LXI   H, RVECT
852>     ; EXAM1: ;MATCH IN RVECT?
853>     CMP   M
854>     JZ    EXAM2
855>     INX   H      ;NEXT RVECT
856>     IHR   B      ;INCREMENT COUNT
857>     DCR   C      ;END OF RVECT?
858>     JHZ   EXAM1
859>     ; NO MATCH
860>     JMP   CERROR
861>     ; EXAM2: ;MATCH IN RVECT, B HAS REGISTER NUMBER
862>     CALL  GNC
863>     CPI   CR      ;ONLY CHARACTER?
864>     JHZ   CERROR
865>     ; WRITE CONTENTS, AND GET ANOTHER BUFFER
866>     PUSH  B      ;SAVE COUNT
867>     CALL  CRLF   ;NEW LINE FOR ELEMENT
868>     CALL  DELT   ;ELEMENT WRITTEN
869>     CALL  BLANK
870>     CALL  GETBUFF ;FILL COMMAND BUFFER
871>     CALL  SCANEXP ;GET INPUT EXPRESSION
872>     ORA   A      ;NDNE?
873>     JZ    START
874>     DCR   A      ;MUST BE ONLY ONE
875>     JHZ   CERROR
876>     CALL  GETVAL   ;VALUE IS IN H,L
877>     POP   B      ;RECALL REGISTER NUMBER
878>     ; CHECK CASES FOR FLAGS, REG-A, OR DOUBLE REGISTER
879>     MOV   A,B
880>     CPI   AVAL
881>     JHC   EXAH4
882>     ; SETTING FLAGS, MUST BE ZERO OR ONE
883>     MOY   A,H
884>     ORA   A
885>     JNZ   CERROR
886>     MOY   A,L
887>     CPI   2
888>     JHC   CERROR
889>     ; 0 OR 1 IN H,L REGISTERS - GET CURRENT FLAGS AND MASK POSITION
890>     CALL  FLGSHF
891>     ; SHIFT COUNT IN C, D,E ADDRESS FLAG POSITION
892>     MOY   H,A      ;FLAGS TO H
893>     MOV   B,C      ;SHIFT COUNT TO B
894>     MVI   A,0FEH  ;11111110 IN ACCUM TO ROTATE
895>     CALL  LRotate ;ROTATE REG-A LEFT
896>     ANA   H      ;MASK ALL BUT ALTERED BIT
897>     MOV   B,C      ;RESTORE SHIFT COUNT TO B
898>     ; 901> 0CF1 67      ;SAVE MASKED FLAGS
899>     902> 0CF2 7D      ;A,L ;0/1 TO LSB OF ACCUM
900>     903> 0CF3 CDFB0C  ;CALL LROTATE ;ROTATED TO CHANGED POSITION
901>     904> 0CF6 B4      ;ORA H      ;RESTORE ALL OTHER FLAGS
902>     905> 0CF7 12      ;STAK D      ;BACK TO MACHINE STATE
903>     906> 0CF8 C37108  ;JMP START ;FOR ANOTHER COMMAND
904>     ; LROTATE: ;LEFT ROTATE FOR FLAG SETTING
905>     907>          ;PATTERN IS IN REGISTER A, COUNT IN REGISTER B
906>     908>          ;DCR B      ;ROTATE COMPLETE
907>     909>          ;RZ      ;END-AROUND ROTATE
908>     910>          ;RCF C3FB0C ;JMP LROTATE
909>     ; EXAM4: ;MAY BE ACCUMULATOR CHANGE
910>     911> 0D01 C2110D ;JNZ EXAMS
911>     912>          ;MUST BE BYTE VALUE
912>     913> 0D04 7C      ;MOV A,H
913>     914> 0D05 B7      ;ORA A
914>     915> 0D06 C2530D  ;JNZ CERROR
915>     916> 0D09 7D      ;MOV A,L ;GET BYTE TO STORE
916>     917> 0D0A 219111  ;LXI H, ALOC ;A REG LOCATION IN MACHINE STATE
917>     918> 0D0D 77      ;MOV M,A ;STORE IT AWAY
918>     919> 0D0E C37108  ;JMP START
919>     ; EXAM5: ;MUST BE DOUBLE REGISTER PAIR
920>     921> 0D11 E5      ;PUSH H      ;SAVE VALUE
921>     922> 0D12 CDA90E  ;CALL GETDBA ;DOUBLE ADDRESS TO HL
922>     923> 0D15 D1      ;POP D      ;VALUE TO D,E
923>     924> 0D16 73      ;MOV M,E
924>     925> 0D17 23      ;INX H
925>     926> 0D18 72      ;MOV M,D ;ALTERED MACHINE STATE
926>     927> 0D19 C37108  ;JMP START
927>     ; DISKR: ;DISK READ
928>     929> 0D1C E5      ;PUSH H
929>     930> 0D1D D5      ;PUSH D
930>     931> 0D1E C5      ;PUSH B
931>     ; RDI: ;READ DISK INPUT
932>     933>          ;940> 0D1F 3A5B00
933>     934> 0D22 F67F
934>     935> 0D24 CA3C0D  ;JZ HDI ;GET NEXT DISK INPUT RECORD
935>     ; RDC: ;READ CHARACTER
936>     937> 0D27 1600  ;MVI D,0
937>     938> 0D29 5F      ;MOV E,A
938>     939> 0D2A 218000  ;LXI H,DBF
939>     940> 0D2D 19      ;DAD D
940>     941> 0D2E 7E      ;MOV A,M
941>     942> 0D2F FE1A  ;CPI DEOF
942>     943> 0D31 CA4E0D  ;JZ DEF ;END OF FILE
943>     944> 0D34 215B00  ;LXI H,DBP
944>     945> 0D38 B7      ;INR M
945>     946> 0D39 C34F0D  ;ORA A
946>     947> 0D3C 0E14  ;JMP RRET
947>     ; NDI: ;NEXT BUFFER IN
948>     949> 0D3D 0E14  ;MVI C,RDF

```



SER. # _____
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

```

961> 0D3E 115C00      LXI   D, DFCB
962> 0D41 CD1E08      CALL  TRAPAD
963> 0D44 B7          ORA   A
964> 0D45 C24E0D      JHZ   DEF
965> ;                 ;
966> ;                 BUFFER READ OK
967> 0D48 325B00      STA   DBP   ;STORE 00H
968> 0D4B C3270D      JMP   RDC
969> ;                 ;
970> DEF.  ;SET CARRY AND RETURN (END FILE)
971> 0D4E 37          STC
972> RRET.             ;
973> 0D4F C1          POP   B
974> 0D50 D1          POP   D
975> 0D51 E1          POP   H
976> 0D52 C9          RET
977> ;                 ;
978> CERROR. ;ERROR IN COMMAND
979> 0D53 CDBD0D      CALL  CRLF
980> 0D56 3E3F          MVI  A, '?'
981> 0D58 CD6F0D      CALL  PCHAR
982> 0D5B C37100      JMP   START
983> ;                 ;
984> ; SUBROUTINES
985> GETBUFF. ;FILL COMMAND BUFFER AND SET POINTERS
986> 0D5E 0E0A          MVI   C, GETF ;GET BUFFER FUNCTION
987> 0D60 114211      LXI   D, COMLEN;START OF COMMAND BUFFER
988> 0D63 CD1E08      CALL  TRAPAD ;FILL BUFFER
989> 0D65 214411      LXI   H, COMBUF;NEXT TO GET
990> 0D69 224011      SHLD  NEXTCOM
991> 0D6C C9          RET
992> ;                 ;
993> BLANK.             MVI   A, ' '
994> 0D6D 3E20          ;
995> ;                 ;
996> PCHAR. ;PRINT CHARACTER TO CONSOLE
997> 0D6F E5          PUSH  H
998> 0D70 D5          PUSH  D
999> 0D71 C5          PUSH  B
000> 0D72 5F          MOV   E, A
001> 0D73 0E02          MVI   C, COF
002> 0D75 CD1E08      CALL  TRAPAD
003> 0D78 C1          POP   B
004> 0D79 D1          POP   D
005> 0D7A E1          POP   H
006> 0D7B C9          RET
007> ;                 ;
008> TRANS.             ;
009> ;                 TRANSLATE TO UPPER CASE
010> 0D7C FE7F          CPI   7FH   ;RUBOUT?
011> 0D7E C8          RZ
012> 0D7F FE61          CPI   ('A' OR 0100000B) ;UPPER CASE A
013> 0D81 D8          RC
014> 0D82 E65F          AH1   1011111B ;CLEAR UPPER CASE BIT
015> 0D84 C9          RET
016> ;                 ;
017> GNC.               ;
018> ;                 GET NEXT BUFFER CHARACTER FROM CONSOLE
019> 0D85 E5          PUSH  H   ;SAVE FOR REUSE LOCALLY
020> 0D86 214311      LXI   H, CURLEN

```

CP/M VERSION
 COPYRIGHT © 1976
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA. 93950
 SER. # _____

```

.021> 0D99 7E          MOV   A, M
.022> 0D9A B7          ORA   A   ;ZERO?
.023> 0D9B 3E0D          MVI   A, CR
.024> 0D9D CA9C0D      JZ    GNCRET ;RETURN WITH CR IF EXHAUSTED
.025> 0D90 35          DCR   M   ;CURLEN=CURLEN-1
.026> 0D91 2A4011      LHLD  NEXTCOM
.027> 0D94 7E          MOV   A, M   ;GET NEXT CHARACTER
.028> 0D95 23          INX   H   ;NEXTCOM=NEXTCOM+1
.029> 0D96 224011      SHLD  NEXTCOM ;UPDATED
.030> 0D99 CD7C0D      CALL  TRANS
.031> 0D9C E1          GNCRET. POP  H   ;RESTORE ENVIRONMENT
.032> 0D9D C9          RET
.033> ;                 ;
.034> PHIB. ;PRINT NIBBLE IN LO ACCUM
.035> 0D9E FE0A          CPI   10
.036> 0D98 D2A80D      JNC   PHIBH ;JUMP IF A-1
.037> 0DA3 C630          ADI   '0'
.038> 0DA5 C36F0D      JMP   PCHAR ;JRET THRU PCHAR SER.#
.039> 0D9A C637          PHIBH. ADI   'A'-10
.040> 0DAA C36F0D      JMP   PCHAR
.041> ;                 ;
.042> 0DAD F5          PBYTE. PUSH PSW ;SAVE A COPY FOR LO NIBBLE
.043> 0DAE 1F          RAR
.044> 0DAF 1F          RAR
.045> 0D80 1F          RAR
.046> 0D81 1F          RAR
.047> 0D82 E60F          ANI   0FH ;MASK HD NIBBLE TO LO NIBBLE
.048> 0D84 CD9E0D      CALL  PHTB
.049> 0D87 F1          POP   PSW ;RECALL BYTE
.050> 0D86 E60F          ANI   0FH
.051> 0D8A C39E0D      JMP   PNIB
.052> ;                 ;
.053> CRLF. ;CARRIAGE RETURN LINE FEED
.054> 0D8D 3E0D          MVI   A, CR
.055> 0DBF CD6F0D      CALL  PCHAR
.056> 0DC2 3E0A          MVI   A, LF
.057> 0DC4 C36F0D      JMP   PCHAR
.058> ;                 ;
.059> BREAK. ;CHECK FOR BREAK KEY
.060> 0DC7 C5          PUSH  B
.061> 0DC8 D5          PUSH  D
.062> 0DC9 E5          PUSH  H
.063> 0DCA 070B          MVI   C, CHKIO
.064> 0DCC CD1E08      CALL  TRAPAD
.065> 0DCF E601          ANI   1B
.066> 0DD1 E1          POP   H
.067> 0DD2 D1          POP   D
.068> 0DD3 C1          POP   B
.069> 0DD4 C9          RET
.070> ;                 ;
.071> PADDX. ;SAME AS PADDR, EXCEPT PRINT VALUE IN D.E
.072> 0DD5 EB          XCHG
.073> ;                 ;
.074> PADDR. ;PRINT THE ADDRESS VALUE IN H.L
.075> 0DD6 7C          MOV   A, H
.076> 0DD7 CDAD0D      CALL  PBYTE
.077> 0DDA 7D          MOV   A, L
.078> 0DDB C3AD0D      JMP   PBYTE
.079> ;                 ;
.080> PGRAFH. ;PRINT GRAPHIC CHARACTER IN REG-A OR '.' IF NOT

```

081> 0DDE FE7F CPI 7FH
 082> 0DE0 D2E80D JNC PPERIOD
 083> 0DE3 FE28 CPI ''
 084> 0DE5 D26F0D JNC PCHAR
 085>
 PPERIOD:
 086> MVI A, ','
 087> JMP PCHAR
 ;
 DISCOM: ;COMPARE H,L AGAINST DISMAX. CARRY SET IF HL > DISMAX AND
 XCHG
 LHLD DISMAX
 MOV A,L
 SUB E
 MOV L,A ;REPLACE FOR ZERO TESTS LATER
 MOV A,H
 SBB D
 XCHG
 RET
 ;
 DELIM: ;CHECK FOR DELIMITER CHARACTER
 CPI CR
 RZ
 CPI ''
 RZ
 CPI ''
 RET
 ;
 HEXCON: ;CONVERT ACCUMULATOR TO PURE BINARY FROM EXTERNAL HEX
 SUI '0'
 CPI 10
 RC ;MUST BE 0-9
 ADI ('0'-'A'+10) AND 0FFH
 CPI 16
 RC ;MUST BE 0-15
 JMP CERROR ;BAD HEX DIGIT
 ;
 GETVAL: ;GET NEXT EXPRESSION VALUE TO H,L (POINTER IN D,E ASSUMED)
 XCHG
 MOV E,M
 INX H
 MOV D,M
 INX H
 XCHG
 RET
 ;
 GETEXP: ;GET HEX VALUE TO D,E
 XCHG
 LXI H,B
 GETEXP:
 CALL HEXCON
 DAD H ;*2
 DAD H ;*4
 DAD H ;*8
 DAD H ;*16
 ORA L ;HL=HL+HEX
 MOV L,A
 CALL GNC
 CALL DELIM ;DELIMITER?
 JNZ GETEXP
 XCHG

141> 0E2C C9 RET
 142>
 143>
 144> 0E2D 73 ;STORE D,E TO H,L AND INCREMENT ADDRESS
 145> 0E2E 23
 146> 0E2F 72
 147> 0E30 23
 148> 0E31 E5
 149> 0E32 213311
 150> 0E35 34
 151> 0E36 E1
 152> 0E37 C9
 153>
 154>
 155>
 156>
 157> 0E38 CD850D
 158>
 159> 0F3B 213311
 160> 0E3E 3600
 161> 0E40 23
 162> 0E41 FE0D
 163> 0E43 CA7D8E
 164>
 165>
 166> 0E46 FE2C
 167> 0E48 C2560E
 168>
 169> 0E4B 3E80
 170> 0E4D 323311
 171> 0E50 110000
 172> 0E53 C3590E
 173>
 174>
 175> 0E56 CD150E
 176> 0E59 CD2D0E
 177> 0E5C FE0D
 178> 0E5E CA7D8E
 179> 0E61 CD850D
 180> 0E64 CD150E
 181> 0E67 C1D0E
 182>
 183> 0E6A FE0D
 184> 0E6C CA7D8E
 185> 0E6F CD850D
 186> 0E72 CD150E
 187> 0E75 CD2D0E
 188> 0E78 FE0D
 189> 0E7A C2530D
 190>
 191> 0E7D 113311
 192> 0E80 1A
 193> 0E81 FE81
 194> 0E83 CA530D
 195> 0E86 13
 196> 0E87 B7
 197> 0E88 07
 198> 0E89 0F
 199> 0E8A C9
 200>

;
 SCSTORE: ;STORE D,E TO H,L AND INCREMENT ADDRESS
 MOV H,E
 IHX H
 MOV M,D
 INX H
 PUSH H
 LXI H,EXPLIST
 IHR M ;COUNT NUMBER OF EXPN'S
 POP H
 RET
 ;
 SCANEXP: ;SCAN EXPRESSIONS - CARRY SET IF .B
 ;ZERO SET IF NO EXPRESSIONS, A SET TO NUMBER OF EXPRESSIONS
 ;HI ORDER BIT SET IF .B ALSO
 CALL GNC
 SCANEX: ;ENTER HERE IF CHARACTER ALREADY SCANNED
 LXI H,EXPLIST
 MVI M,0 ;ZERO EXPRESSIONS
 INX H ;READY TO FILL EXPRESSION LIST
 CPI CR ;END OF LINE
 JZ SCANRET
 ;
 NOT CR, MUST BE DIGIT OR COMMA
 CPI ''
 JNZ SCANEQ
 MARK AS COMMA
 MVI A,80H
 STA EXPLIST
 LXI D,0
 JMP SCANE1
 ;
 SCANEQ: ;NOT CR OR COMMA
 CALL GETEXP ;EXPRESSION TO D,E
 SCANE1: CALL SCSTORE ;STORE THE EXPRESSION AND INCREMENT H,L
 CPI CR
 JZ SCANRET
 CALL GNC
 CALL GETEXP
 CALL SCSTORE
 SECOND DIGIT SCANNED
 CPI CR
 JZ SCANRET
 CALL GNC
 CALL GETEXP
 CALL SCSTORE
 CPI CR
 JNZ CERROR
 SCANRET: ;LOOK AT COUNT
 LXI D,EXPLIST
 LDAX D ;LOAD COUNT TO ACC
 CPI B1H ;WITHOUT B?
 JZ CERROR
 INX D ;READY TO EXTRACT EXPN'S
 ORA A ;ZERO FLAG MAY BE SET
 RLC
 RET ;SET CARRY IF NO BIT SET (.B)
 ;WITH FLAGS SET

```

201> ; SUBROUTINES FOR CPU STATE DISPLAY
202> FLCSHF: ;SHIFT COMPUTATION FOR FLAG GIVEN BY REG-B
203> ; REG A CONTAINS FLAG UPDH EXIT (UNSHIFTED)
204> ; REG C CONTAINS NUMBER OF SHIFTS REQUIRED+1
205> ; REGS D,E CONTAIN ADDRESS OF FLAGS IN TEMPLATE
206>
207> 0E3B E5
208> 0E3C 21680F
209> 0EFF 58
210> 0E90 1600
211> 0E92 19
212> 0E93 4E
213> 0E34 219011
214> 0E97 7E
215> 0E38 EB
216> 0E99 E1
217> 0E9A C9
218>
219> ; GETFLG: ;GET FLAG GIVEN BY REG-B TO REG-A AND MASK
220> 0E9B CD880E
221> 0E9E 0D
222> 0E9F CAA60E
223> 0EA2 1F
224> 0EA3 C39E0E
225> 0EA6 E601
226> 0EA8 C9
227>
228> ; GETDBA: ;GET DOUBLE BYTE ADDRESS CORRESPONDING TO REG-A TO HL
229> 0EA9 D606
230> 0EAB 21660F
231> 0EAE 5F
232> 0EAF 1600
233> 0EB1 19
234> 0EB2 5E
235> 0E93 16FF
236> 0E85 219811
237> 0E88 19
238> 0EB9 C9
239>
240> ; GETDBL: ;GET DOUBLE BYTE CORRESPONDING TO REG-A TO HL
241> 0EBA CD980E
242> 0EBD 5E
243> 0EBE 23
244> 0EBF 56
245> 0EC0 EB
246> 0EC1 C9
247>
248> ; DELT: ;DISPLAY CPU ELEMENT GIVEN BY COUNT IN REG-B, ADDRESS IN H,L
249> 0EC2 7E
250> 0EC3 CD6F0D
251> 0EC6 78
252> 0EC7 FE05
253> 0EC9 D2D30E
254>
255> ; DISPLAY FLAG
256> 0ECC CD960E
257> 0ECF CD9E0D
258> 0ED2 C9
259>
260> ; DELTB: ;NOT FLAG, DISPLAY = AND DATA

```

SER. #	CPI.. VERSION COPYRIGHT © 1976 DIGITAL RESEARCH P. O. BOX 579 PACIFIC GROVE, CA 93950
--------	---

```

0ED3 F5
0ED4 3E3D
0ED6 CD6F0D
0ED9 F1
0EDA C2E50E
;
PUSH PSW
MVI A, '='
CALL PCHAR
POP PSW
JNZ DELT1 ;JUMP IF NOT REG-A
;
REGISTER A, DISPLAY BYTE VALUE
LXI H, ALOC
MOV A, M
CALL PBYTE
RET
;
DELT1, ;DOUBLE BYTE DISPLAY
CALL GETDBL ;TO H,L
CALL PADDR ;PRINTED
RET
;
DSTATE, ;DISPLAY CPU STATE
LXI H, RVECT ;REGISTER VECTOR
MVI B, B ;REGISTER COUNT
CALL CRLF
DSTA0, ;ELEMENT DISPLAYED
PUSH B
PUSH H
CALL DELT ;RVECT ADDRESS RESTORED
POP H
POP B ;COUNT RESTORED
INR B ;NEXT COUNT
INX H ;NEXT REGISTER
MOV A, B ;LAST COUNT?
CPI PVAL+1
JNC DSTAI ;JMP IF PAST END
CPI AVAL ;BLANK AFTER?
JC DSTAB
YES, BLANK AND GO AGAIN
CALL BLANK
JMP DSTAB
;
READY TO SEND DECODED INSTRUCTION
DSTA1, ;COMPUTE BREAKPOINTS IN CASE OF TRACE
CALL BLANK
CALL HERK ;SAVE EXPRESSION COUNT - B,C AND D,E HAVE
PUSH PSW ;SAVE BP ADDRESS
PUSH D ;SAVE AUX BREAKPOINT
CALL CHKDIS ;CHECK TO SEE IF DISASSEMBLER IS HERE
JNC DCHEX ;DISPLAY HEX IF NOT
DISASSEMBLE CODE
LHLD PLOC ;GET CURRENT PC
SHLD DISPC ;SET DISASSM PC
LXI H, DISPG ;PAGE MODE = BFFH TO TRACE
MVI M, BFFH
CALL DISEN
JMP DSTRET
;
DCHEX, ;DISPLAY HEX
DCX H ;POINT TO LAST TO WRITE
SHLD DISMAK ;SAVE FOR COMPARE BELOW
LHLD PLOC ;START ADDRESS OF TRACE
MOV A, M ;GET OPCODE
CALL PBYTE

```

CP.. VERSION
COPYRIGHT © 1976
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

```

21> 0F39 23      INX H      ;READY FOR NEXT BYTE
22> 0F3A C0E0D0  CALL DISCOM ;ZERO SET IF ONE BYTE TO PRINT, CARRY IF NO MO
23> 0F3D DAS70F  JC DSTRET
24> 0F40 F5      PUSH PSW   ;SAVE RESULT OF ZERO TEST
25> 0F41 C0D10D  CALL BLANK  ;SEPARATOR
26> 0F44 F1      POP PSW   ;RECALL ZERO TEST
27> 0F45 B3      ORA E     ;ZERO TEST
28> 0F46 CA530F  JZ DSTA2
29>          ;DISPLAY DOUBLE BYTE
30> 0F49 5E      MOV E,M
31> 0F4A 23      INX H
32> 0F4B 56      MOV D,M
33> 0F4C E8      XCHG
34> 0F4D CDD60D  CALL PADDR ;PRINT ADDRESS
35> 0F50 C3570F  JMP DSTRET
36>          ;DSTA2. ;PRINT BYTE VALUE
37> 0F53 7E      MOV A,M
38> 0F54 CDA0D0  CALL PBYTE
39>          ;DSTRET:
40> 0F57 C1      POP B     ;AUX BREAKPOINT
41> 0F58 D1      POP D     ;RESTORE BREAKPOINT
42> 0F59 F1      POP PSW   ;RESTORE COUNT
43> 0F5A C9      RET
44>          ;DATA VECTORS FOR CPU DISPLAY
45> 0F5B 435A4D4549RVECT. DB 'CZMEIABDHSP'
46> 0F66 F6      RIHX, DB '(BLDC-STACK) AND 0FFH' ;LOCATION OF BC
47> 0F67 F4      DB '(DLDC-STACK) AND 0FFH' ;LOCATION OF DE
48> 0F68 FC      DB '(HLDC-STACK) AND 0FFH' ;LOCATION OF HL
49> 0F69 FA      DB '(SLDC-STACK) AND 0FFH' ;LOCATION OF SP
50> 0F6A FE      DB '(PLDC-STACK) AND 0FFH' ;LOCATION OF PC
51>          ;FLGTAB ELEMENTS DETERMINE SHIFT COUNT TO SET/EXTRACT FLAGS
52> 0F6B 0107088305FLGTAB: DB . 1,7,8,3,5 ;CY, ZER, SIGN, PAR, IDCY
53>          ;CLRTRACE: ;CLEAR THE TRACE FLAG
54> 0F70 210000  LXI H,0
55> 0F73 222A11  SHLD TRACER
56> 0F76 C9      RET
57>          ;BREAKP: ;ARRIVE HERE WHEN PROGRAMMED BREAK OCCURS
58> 0F77 F3      DI
59> 0F78 229411  SHLD HLLOC ;JHL SAVED
60> 0F78 E1      POP H     ;RECALL RETURN ADDRESS
61> 0F7C 2B      DCX H     ;DECREMENT FOR RESTART
62> 0F7D 229611  SHLD PLOC
63>          ;DAD SP BELOW DESTROYS CY, SO SAVE AND RECALL
64> 0F80 F5      PUSH PSW   ;INTO USER'S STACK
65> 0F81 210200  LXI H,2   ;BIAS SP BY 2 BECAUSE OF PUSH
66> 0F84 39      DAD SP    ;SP IN HL
67> 0F85 F1      POP PSW   ;RESTORE CY AND FLAGS
68> 0F86 319411  LXI SP,STACK-4;LOCAL STACK
69> 0F89 E5      PUSH H     ;SP SAVED
70> 0F8A F5      PUSH PSW
71> 0F8B C5      PUSH B
72> 0F8C D5      PUSH D
73>          ;MACHINE STATE SAVED, CLEAR BREAK POINTS
74> 0F8D 2A9611  LHLD PLOC ;CHECK FOR RST INSTRUCTION
75> 0F90 7E      MOV A,M   ;OPCODE TO A
76> 0F91 FEFF  CPI RSTIN

```

CP1. VERSION
 Copyright © 1976
 DIGITAL RESEARCH
 P. O. Box 579
 PACIFIC GROVE, CA. 93950
 SER. #

```

381> 0F93 F5      ;SAVE CONDITION CODES FOR LATER TEST
382>          ;PUSH PSW
383>          ;SAVE PLOC FOR LATER INCREMENT OR DECREMENT
384> 0F94 E5      ;PUSH H
385>          ;CLEAR BREAKPOINTS WHICH ARE PENDING
386>          ;LXI H,BREAKS
387> 0F95 212C11  ;MOV A,H
388> 0F98 7E      ;MVI H,0   ;SET TO ZERO BREAKS
389> 0F99 3600
390> 0F98 87      ;CLER0: ORA A     ;ANY MORE?
391> 0F9C CAA0BF  ;JZ CLER1
392> 0F9F 3D      ;DCR A
393> 0FA0 47      ;MOV B,A   ;SAVE COUNT
394> 0FA1 23      ;INX H     ;ADDRESS OF BREAK
395> 0FA2 5E      ;MOV E,M   ;LOW ADDR
396> 0FA3 23      ;INX H
397> 0FA4 75      ;MOV D,M   ;HIGH ADDR
398> 0FA5 23      ;INX H
399> 0FA6 7E      ;MOV A,M   ;INSTRUCTION
400> 0FA7 12      ;STAK D    ;BACK TO PROGRAM
401> 0FA8 78      ;MOV A,B   ;RESTORE COUNT
402> 0FA9 C39B0F  ;JMP CLER0
403>          ;CLER1: ;CLEARED, CONTINUE TRACING, OR STOP EXECUTION
404>          ;POP H     ;RESTORE PLOC
405> 0FAC E1      ;POP PSW   ;RESTORE CONDITION FLAGS
406> 0FAD F1      ;JZ BREAK0 ;BRANCH IF PROGRAMMED INTERRUPT
407> 0FAE CACD0F
408>          ;MUST BE FRONT PANEL INTERRUPT, CHECK IF IN BDOS
409>          ;INX H     ;DON'T DECREMENT ON PANEL INTERRUPT
410> 0FB1 23      ;SHLD PLOC ;RESTORE TO NEXT LOGICAL INSTRUCTION
411> 0FB2 229611  ;XCHG    ;TO D,E FOR COMPARE
412> 0FB5 EB      ;LXI B,BDOS ;BASE OF DOS
413> 0FB6 010612  ;CALL BCDE ;CY IF BDOS>^PLOC
414> 0FB9 CDC109  ;JC BREAK0 ;BRANCH IF PLOC <= BDOS
415> 0FBC DACD0F
416>          ;IN THE BDOS, DON'T BREAK UNTIL THE RETURN OCCURS
417>          ;CALL CLRTRACE;CLEAR TRACE FLAGS
418> 0FBF CD700F  ;LHLD RETLOC ;TRAPPED RETLOC ON ENTRY TO DOS
419> 0FC2 242711  ;XCHG    ;TO D,E READY FOR BREAKPOINT
420> 0FC5 EB      ;MVI A,82H ;LOOKS LIKE C,BBBB
421> 0FC6 3E82  ;ORA A     ;SETS FLAGS
422> 0FC8 87      ;STC
423> 0FC9 37      ;JMP GOPR  ;SUBSEQUENT TEST FOR CY
424> 0FC4 C31F0A  ;START PROGRAM EXECUTION, WITH BREAKPOINT
425>          ;BREAK0: ;NORMAL BREAKPOINT
426>          ;EI
427> 0FCD FB      ;LHLD TRACER
428> 0FCE 2A2A11  ;MOV A,H
429> 0FD1 7C      ;ORA L
430> 0FD2 B5      ;JZ STOPEX
431> 0FD3 CAF30F
432>          ;TRACE IS ON
433>          ;DCX H
434> 0FD6 2B      ;SHLD TRACER
435> 0FD7 222A11  ;CALL BREAK ;BREAK KEY DEPRESSED?
436> 0FDA CDC70D  ;JNZ STOPEX
437> 0FD8 C2F30F  ;LDA TMODE ;TRACE MODE T IF BFFF
438> 0FE0 3A2911  ;ORA A
439> 0FE3 B7      ;JNZ BREAK1
440> 0FE4 C2ED0F

```

SER. #

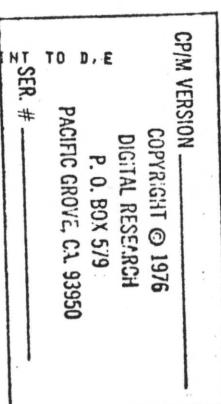
COPYRIGHT © 1976
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA. 93950

441> ; NOT TRACING, BUT MONITORING, SO SET BREAKPOINTS
 442> 0FE7 CD3010 CALL HBRK
 443> 0FEA C31F8A JMP GOPR
 444>
 445> ;BREAK1: ;TRACING AND MONITORING
 446> 0FED CDEC0E CALL DSTATE ;STATE DISPLAYED, CHECK FOR BREAKPOINTS
 447> 0FF0 C31F8A JMP GOPR ;STARTS EXECUTION
 448>
 449> ;STOPEX:
 450> 0FF3 210000 LXI H, 0
 451> 0FF6 222511 SHLD USERBRK ;CLEAR USER BREAK ADDRESS
 452> 0FF9 CD700F CALL CLRTRACE ;TRACE FLAGS GO TO ZERO
 453> 0FFC 3E2A MVI A, '*'
 454> 0FFE CD6F8D CALL PCHAR
 455> 1001 2A9611 LHLD PLOC
 456> ;CHECK TO ENSURE DISASSEMBLER IS PRESENT
 457> 1004 CD270B CALL CHKDIS
 458> 1007 D20D10 JHC STOP0
 459> 100A 220C01 SHLD DISPC
 460> 100D CD160D STOP0: CALL PADDR
 461> 1010 2A9411 LHLD HL0C
 462> 1013 223A11 SHLD DISLOC
 463> 1016 C37108 JMP START
 464>
 465> ;CAT: ;DETERMINE OPCODE CATEGORY - CODE IN REGISTER B
 466> ;D, E CONTAIN DOUBLE PRECISION CATEGORY NUMBER ON RETURN
 467> 1019 110D00 LXI D, OPMAX ;D=0, E=OPMAX
 468> 101C 210A11 LXI H, OPLIST
 469> 101F 7E CAT0: MOV A, M ;MASK TO A
 470> 1020 A8 AHA B ;MASK OPCODE FROM B
 471> 1021 23 IHX H ;READY FOR COMPARE
 472> 1022 BE CMP H ;SAME AFTER MASK?
 473> 1023 23 IHX H ;READY FOR NEXT COMPARE
 474> 1024 CA2C10 JZ CAT1 ;EXIT IF COMPARED OK
 475> 1027 14 IHR D ;UP COUNT IF NOT MATCHED
 476> 1028 1D DCR E ;FINISHED?
 477> 1029 C21F10 JHZ CAT0
 478> 102C 5A CAT1: MOV E, D ;E IS CATEGORY NUMBER
 479> 102D 1600 MVI D, 0 ;DOUBLE PRECISION
 480> 102F C9 RET
 481>
 482> ;HBRK: ;FIND NEXT BREAK POINT ADDRESS
 483> ;UPON RETURN, REGISTER A IS SETUP AS IF USER TYPED G,B1,B2 OR
 484> ;G,B1 DEPENDING UPON OPERATOR CATEGORY. B,C CONTAINS SECOND BP
 485> ;D,E CONTAINS PRIMARY BP. HL ADDRESS NEXT OPCODE BYTE
 486> 1030 2A9611 LHLD PLOC
 487> 1033 46 MOV B, M ;GET OPERATOR
 488> 1034 23 IHX H ;HL ADDRESS BYTE FOLLOWING OPCODE
 489> 1035 E5 PUSH H ;SAVE IT FOR LATER
 490> 1036 CD1910 CALL CAT ;DETERMINE OPERATOR CATEGORY
 491> 1039 212411 LXI H, CATNO ;SAVE CATEGORY NUMBER
 492> 103C 73 MOV H, E
 493> 103D 214710 LXI H, CATTAB;CATEGORY TABLE BASE
 494> 1040 19 DAD D ;INHXED
 495> 1041 19 DAD D ;INHXED+2
 496> 1042 5E MOV E, M ;LOW BYTE TO E
 497> 1043 23 IHX H
 498> 1044 56 MOV D, M ;HIGH BYTE TO D
 499> 1045 EB XCHG PCHL ;JUMP INTO TABLE
 500> 1046 E9
 ;NOT TRACING, BUT MONITORING, SO SET BREAKPOINTS
 501> 1847 6310 CATTAB: DW JMPOP ;JUMP OPERATOR
 502> 1849 8910 DW CCOP ;JUMP CONDITIONAL
 503> 184B 6310 DW JMPOP ;CALL OPERATOR (TREATED AS JMP)
 504> 184D 8910 DW CCOP ;CALL CONDITIONAL
 505> 184F 6910 DW RETOP ;RETURN FROM SUBROUTINE
 506> 1851 9810 DW RSTOP ;RESTART
 507> 1853 AD10 DW PCOP ;PCHL
 508> 1855 CF10 DW IMOP ;SINGLE PRECISION IMMEDIATE (2 BYTES)
 509> 1857 CF10 DW IMOP ;JADI... CPT
 510> 1859 CC10 DW DIMOP ;DOUBLE PRECISION IMMEDIATE (3 BYTES)
 511> 185B CC10 DW DIMOP ;JLHLD... STA
 512> 185D C210 DW RCND ;RETURN CONDITIONAL
 513> 185F CF10 DW IMOP ;IN/OUT
 514> ;NEXT DW MUST BE THE LAST IN THE SEQUENCE
 515> 1061 BD10 DW SIMOP ;SIMPLE OPERATOR (1 BYTES)
 516>
 517> ;JMPOP: ;GET OPERAND FIELD, CHECK FOR BDOS
 518> 1863 CD7710 CALL GETOPA ;GET OPERAND ADDRESS TO D,E AND COMPARE WITH
 519> 1866 C2D210 JHZ ENDOP ;TREAT AS SIMPLE OPERATOR IF NOT BDOS
 520>
 521> ;OTHERWISE, TREAT AS A RETURN INSTRUCTION
 522> 1869 CD8210 RETOP: CALL GETSP ;ADDRESS AT STACKTOP TO D,E
 523> 186C C3D210 JMP ENDOP ;TREAT AS SIMPLE OPERATOR
 524>
 525> ;CBDOS: ;COMPARE D,E WITH BDOS ADDRESS, RETURN ZERO IF EQUAL
 526> 186F 3E86 MVI A, BDOS AND OFFH
 527> 1071 BB CMP E
 528> 1072 C8 RHZ
 529> 1073 3E12 MVI A, BDOS SHR 8
 530> 1075 BA CMP D
 531> 1076 C9 RET
 532>
 533> ;GETOPA: ;GET OPERAND ADDRESS AND COMPARE WITH BDOS
 534> 1077 C1 POP B ;GET RETURN ADDRESS
 535> 1078 E1 POP H ;GET OPERAND ADDRESS
 536> 1079 5E MOV E, M
 537> 107A 23 INX H
 538> 107B 56 MOV D, M
 539> 107C 23 INX H
 540> 107D E5 PUSH H ;UPDATED PC INTO STACK
 541> 107E C5 PUSH B ;RETURN ADDRESS TO STACK
 542> 107F C36F10 JMP CBDOS ;RETURN THROUGH CBDOS WITH ZERO FLAG SET
 543>
 544> ;GETSP: ;GET RETURN ADDRESS FROM USER'S STACK TO D,E
 545> 1085 5E LHLD SLOC
 546> 1086 23 MOV E, M
 547> 1087 56 INX H
 548> 1088 C9 MOV D, M
 549> RET
 550>
 551> ;CCOP: ;CALL CONDITIONAL OPERATOR
 552> 1089 CD7710 CALL GETOPA ;GET OPERAND ADDRESS TO D,E / COMPARE WITH
 553> 108C CA9610 JZ CCOP1
 554>
 555> ;NOT THE BDOS, BREAK AT OPERAND ADDRESS AND NEXT ADDRESS
 556> 108F C1 POP B ;NEXT ADDRESS TO B,C
 557> 1090 C5 PUSH B ;BACK TO STACK
 558> 1091 3E02 MVI A, 2 ;TWO BREAKPOINTS
 559> 1093 C3D410 JMP RETCAT ;RETURN FROM HBRK
 560>
 561> ;CCOP1: ;BREAK ADDRESS AT NEXT LOCATION ONLY. WAIT FOR RETURN FROM
 562> 1096 D1 POP B

COPYRIGHT © 1976
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

561> 1097 D5
 562> 1098 C3D210 PUSH D ;BACK TO STACK
 563> JMP ENDP ;ONE BREAKPOINT ADDRESS
 564> ,
 565> 109B 78 RSTOP, ;RESTART INSTRUCTION - CHECK FOR RST 7
 566> 109C FFFF MOV A,B
 567> 109E C2A510 CPI RSTIN ;RESTART INSTRUCTION USED FOR SOFT INT
 568> JHZ RST0
 569> ,
 570> 10A1 AF SOFT RST, NO BREAK POINT SINCE IT WILL OCCUR IMMEDIATELY
 571> 10A2 C3D610 XRA A
 572> 10A5 E638 JMP RETCAT1 ;ZERO ACCUMULATOR
 573> RST0, AH1 11100BB ;GET RESTART NUMBER
 574> 10A7 5F MOY E,A
 575> 10A8 1600 MVI D,0 ;DOUBLE PRECISION BREAKPOINT TO D,E
 576> 10AA C3D210 JMP ENDP
 577> ,
 578> 10AD 2A9411 PCOP, ;PCHL
 579> 10B0 EB LHLD HLOC
 580> 10B1 CD6F10 XCHG ;HL VALUE TO D,E FOR BREAKPOINT
 581> CALL CBDS ;JBDS VALUE?
 582> 10B4 C2D210 JNZ ENDP
 583> ,
 584> 10B7 C36910 PCHL TO BDOS, USE RETURN ADDRESS
 585> 10B8 C3D210 JMP RETOP
 586> ,
 587> 10BD D1 SIMOP, ;SIMPLE OPERATOR, USE STACKED PC
 588> POP D
 589> PUSH D
 590> 10BF C3D210 JMP ENDP
 591> ,
 592> RCOND, ;RETURN CONDITIONAL
 593> 10C2 CD8210 CALL GETSP ;GET RETURN ADDRESS FROM STACK
 594> 10C5 C1 POP B ;B,C ALTERNATE LOCATION
 595> 10C6 C5 PUSH B ;REPLACE IT
 596> 10C7 3E02 MVI A,2
 597> 10C9 C3D410 JMP RETCAT ;TO SET FLAGS AND RETURN
 598> ,
 599> DIMOP, ;DOUBLE PRECISION IMMEDIATE OPERATOR
 600> 10CC D1 POP D
 601> 10CD 13 INX D ;INCREMENTED ONCE, DROP THRU FOR ANOTHER
 602> 10CE D5 PUSH D ;COPY BACK
 603> ,
 604> IMOP, ;SINGLE PRECISION IMMEDIATE
 605> 10CF D1 POP D
 606> 10D0 13 INX D
 607> 10D1 D5 PUSH D
 608> ,
 609> ENDOP, ;END OPERATOR SCAN
 610> 10D2 3E01 MVI A,1 ;SINGLE BREAKPOINT
 611> RETCAT, ;RETURN FROM HBRK
 612> 10D4 3C INR A ;COUNT UP FOR G,...
 613> 10D5 37 STC
 614> RETCAT1, PUSH PSW ;SAVE REGISTER STATE IN CASE USERBRK
 615> 10D6 F5 LHLD USERBRK
 616> 10D7 2A2511 MOV A,H
 617> 10D8 7C ORA L
 618> 10D8 B5 JZ RETCAT2 ;NO USERBRK IF ZERO
 619> 10DC CA0711 ,
 620>

10E1 10DF D5 PUSH D ;SAVE BREAK POINT
 10E0 C5 PUSH B ;SAVE AUX BREAK POINT
 10E1 E5 PUSH H ;SAVE USERBRK ADDRESS FOR PCHL BELOW
 10E2 212411 USER BREAK OCCURS HERE, CALL USER ROUTINE AND CHECK RETURN
 10E3 4E LXI H,CATHO
 10E6 2A9611 MOV C,M ;OPCODE CATEGORY IS IN C
 10E9 EB LHLD PLOC
 10EA 21EF10 XCHG ;LOCATION OF INSTRUCTION IN D,E
 10ED E3 LXI H,RETUSER
 10EE E9 XTHL ;RETURN ADDRESS TO STACK, USERBRK TO H,L
 10EF B7 PCHL
 10F0 C1 ORA A
 10F1 D1 POP B ;RESTORE BREAKPOINTS
 10F2 CA0711 PDP D
 10F3 F5 JZ RETCAT2
 10F6 3E23 ABORT THE OPERATION WITH A CONDITION
 10F8 CD6F0D PUSH PSW
 10FB F1 MVN A,'B'
 10FC CDAD0D CALL PCHAR
 10FF 3E20 POP PSW
 1101 CD6F0D CALL PBYTE
 1104 C3F30F MVN A,' '
 1107 F1 CALL PCHAR
 1108 E1 JMP STOPEX
 1109 C9 POP H ;STOP EXECUTION
 1110 0000 = RET
 1111 0000 =
 1112 FFC9
 1113 C7C7
 1114 C7C7
 1115 FFE9
 1116 C786
 1117 C7C6
 1118 C7C0
 1119 C7C0
 1120 F7D3
 1121 0000 = OPLIST, DB 1111\$1111B, 1100\$0011B, 10 JMP
 1122 F7D3 OPMAX EQU (\$-OPLIST)/2
 1123 1124 CATNO, DS 1 ;CATEGORY NUMBER SAVED IN HBRK
 1124 1125 USERBRK, DS 2 ;USER BREAK ADDRESS IF NON-ZERO
 1125 1126 RETLOC, DS 2 ;RETURN ADDRESS TO USER FROM BDOS
 1126 1127 TMODE, DS 1 ;TRACE MODE
 1127 1128 TRACER, DS 2 ;TRACE COUNT
 1128 1129 BREAKS, DS 7 ;#BREAKS/BKPT1/DAT1/BKPT2/DAT2
 1129 1130 EXPLIST, DS 7 ;COUNT+(EXP1)(EXP2)(EXP3)
 1130 1131 DISLOC, DS 2 ;DISPLAY LOCATION
 1131 1132 DISMAX, DS 2 ;MAX VALUE FOR CURRENT DISPLAY
 1132 1133 TDISP, DS 2 ;TEMP 16 BIT LOCATION
 1133 1134 NEXTCOM, DS 2 ;NEXT LOCATION FROM COMMAND BUFFER
 1134 1140 COMLEN, DB 2 ;CSIZE ;MAX COMMAND LENGTH



CP/M VERSION
 COPYRIGHT © 1976
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. #

```
681> 1143      CURLEN: DS      1      /CURRENT COMMAND LENGTH  
682> 1144      COMBUF: DS     CSIZE  /COMMAND BUFFER  
683> 1164      MLOAD:  DS      2      /MAX LOAD ADDRESS  
684> 1166      DS      SSIZE  /STACK AREA  
685>  
686> 1196 =     PLOC   EQU    STACK-2 /PC IN TEMPLATE  
687> 1194 =     HLOC   EQU    STACK-4 /HL  
688> 1192 =     SLOC   EQU    STACK-6 /SP  
689> 1191 =     ALOC   EQU    STACK-7 /A  
690> 1190 =     FLOC   EQU    STACK-8 /FLAGS  
691> 118E =     BLOC   EQU    STACK-10 /BC  
692> 118C =     DLOC   EQU    STACK-12 /D,E  
693>  
694> 1198 00      NOP  
695> 1199      END          /FOR RELOCATION BOUNDARY
```

