

TITLE 'CP/M DEBUGGER (DEMON) 1/80'
CP/M DEBUGGER VERSION 2.2

COPYRIGHT (C) 1980
DIGITAL RESEARCH
BOX 579 PACIFIC GROVE
CALIFORNIA 93950

0000 = FALSE EQU 0
FFFF = TRUE EQU NOT FALSE
0000 = DEBUG EQU FALSE ;TRUE IF DEBUGGING
FFFF = RELOC EQU TRUE ;TRUE IF RELOCATING

IF DEBUG
ORG 1000H
ELSE
IF RELOC
ORG 0000H
ELSE
ORG 0D000H ;TESTING IN 64K
ENDIF
ENDIF

COPYRIGHT © 1980
DIGITAL RESEARCH, INC.
P. O. BOX 579
PACIFIC GROVE, CA 93950

SERIAL # _____

0000 = MODBAS EQU \$;BASE OF ASSEM/DISASSEM MODULE
0000 DS 680H ;SIZE OF ASSEM/DISASSEM
0680 = DEMON EQU \$;BASE OF DEMON MODULE
0003 = DISIN EQU MODBAS+3
1686 = BDOS EQU \$+1006H
0005 = BDOSE EQU 5H ;ENTRY POINT TO DOS FROM USER PROGRAMS
0100 = PCBASE EQU 100H ;DEFAULT PC
0100 = SPBASE EQU 100H ;DEFAULT SP
0006 = DISEN EQU DISIN+3 ;DISASSEMBLER ENTRY POINT
0009 = ASSEM EQU DISEN+3 ;ASSEMBLER ENTRY POINT
000C = DISPC EQU ASSEM+3 ;DISASSEMBLER PC VALUE
000E = DISPM EQU DISPC+2 ;DISASSEMBLER PC MAX VALUE
0010 = DISPG EQU DISPM+2 ;DISASSEMBLER PAGE MODE IF NON ZERO
000C = PSIZE EQU 12 ;NUMBER OF ASSEMBLY LINES TO LIST WITH
0020 = CSIZE EQU 32 ;COMMAND BUFFER SIZE
0032 = SSIZE EQU 50 ;LOCAL STACK SIZE

BASIC DISK OPERATING SYSTEM CONSTANTS

0001 = CIF EQU 1
0002 = COF EQU 2
0003 = RIF EQU 3
0004 = POF EQU 4
0005 = LOF EQU 5
0007 = IDS EQU 7
000A = GETF EQU 10 ;FILL BUFFER FROM CONSOLE
000B = CHKIO EQU 11 ;CHECK IO STATUS
000C = LIFT EQU 12 ;LIFT HEAD ON DISK
000F = OPF EQU 15 ;DISK FILE OPEN
0014 = RDF EQU 20 ;READ DISK FILE
001A = DMAF EQU 26 ;SET DMA ADDRESS
005B = DBP EQU 5BH ;DISK BUFFER POINTER
0080 = DBF EQU 80H ;DISK BUFFER ADDRESS


```

;
;
TRAPAD: ;GET THE RETURN ADDRESS FOR THIS JUMP TO BDOS IN CASE OF
; A SOFT INTERRUPT DURING BDOS PROCESSING.
06A2 E3 XTHL ;PC TO HL
06A3 224A0F SHLD RETLOC ;MAY NOT NEED IT
06A6 E3 XTHL
TRAPJMP: ;ADDRESS FILLED AT "BEGIN"
06A7 C30000 JMP 0000H
;
BEGIN:
;
06AA 2A0600 LHLD BDOSE+1
06AD 22A806 SHLD TRAPJMP+1 ;FILL JUMP TO BDOS
06B0 21A206 LXI H,TRAPAD
06B3 220100 SHLD MODBAS+1 ;ADDRESS FIELD CHANGED
06B6 210000 LXI H,MODBAS
06B9 220600 SHLD BDOSE+1 ;NOW INCLUDES ASSEM/DISASSEM
;
06BC AF XRA A ;ZERO TO ACCUM
06BD 324F0F STA BREAKS ;CLEARS BREAK POINT COUNT
;
06C0 210001 LXI H,PCBASE
06C3 220C00 SHLD DISPC ;INITIAL VALUE FOR DISASSEMBLER PC
06C6 225D0F SHLD DISLOC ;INITIAL VALUE FOR DISPLAY
06C9 22870F SHLD MLOAD ;MAX LOAD LOCATION
;
;
SETUP RESTART TEMPLATE
06CC 22B90F SHLD PLOC
06CF 210001 LXI H,SPBASE
06D2 31B70F LXI SP,STACK-4
06D5 E5 PUSH H ;INITIAL SP
06D6 210200 LXI H,10B ;INITIAL PSW
06D9 E5 PUSH H
06DA 2B DCX H
06DB 2B DCX H ;CLEARED
06DC 22B70F SHLD HLOC ;H,L CLEARED
06DF E5 PUSH H ;B,C CLEARED
06E0 E5 PUSH H ;D,E CLEARED
06E1 224D0F SHLD TRACER ;CLEAR TRACE FLAG
;
06E4 3EC3 MVI A,0C3H ;(JMP RESTART)
06E6 323800 STA RSTLOC
06E9 218606 LXI H,BREAKA ;BREAK POINT SUBROUTINE
06EC 223900 SHLD RSTLOC+1 ;RESTART LOCATION ADDRESS FIELD
;
;
CHECK FOR FILE NAME PASSED TO DEMON, AND LOAD IF PRESENT
06EF 3A5D00 LDA FCB+FFN ;BLANK IF NO NAME PASSED
06F2 FE20 CPI
06F4 CAFE06 JZ START
;
;
PUSH A ZERO, AND READ
06F7 210000 LXI H,0
06FA E5 PUSH H
06FB C3AD09 JMP RINIT
;

```

COPYRIGHT © 1980
 DIGITAL RESEARCH, INC.
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SERIAL # _____

```

;
;      MAIN COMMAND LOOP
;
START:
06FE 31AFOF      LXI      SP,STACK-12      ;INITIALIZE SP IN CASE OF ERROR
;      CHECK FOR DISASSEMBLER OVERLOAD
0701 CD9309      CALL     CHKDIS
0704 DA0D07      JC       DISASMOK

;
;      DISASSEMBLER NOT PRESENT, SET BDOS JMP
0707 218006      LXI      H,DEMON
070A 220600      SHLD     BDOSE+1 ;(RE)SET JMP ADDRESS
DISASMOK:
070D CD150C      CALL     CRLF      ;INITIAL CRLF
;      IF DEBUG
;      MVI      A,':'
;      ELSE
0710 3E2D        MVI      A,'-'
;      ENDIF
0712 CDC70B      CALL     PCHAR      ;OUTPUT PROMPT

;
;      GET INPUT BUFFER
0715 CDB60B      CALL     GETBUFF ;FILL COMMAND BUFFER

;
0718 CDDDOB      CALL     GNC        ;GET CHARACTER
071B FE0D        CPI      CR
071D CAFE06      JZ       START
0720 D641        SUI      'A'      ;LEGAL CHARACTER?
0722 DAAB0B      JC       CERROR    ;COMMAND ERROR
0725 FE1A        CPI      'Z'-'A'+1
0727 D2AB0B      JNC      CERROR

;      CHARACTER IN REGISTER A IS COMMAND, MUST BE IN THE RANGE A-Z
072A 5F          MOV      E,A      ;INDEX TO E
072B 1600        MVI      D,0      ;DOUBLE PRECISION INDEX
072D 213707      LXI      H,JMPTAB;BASE OF TABLE
0730 19          DAD      D
0731 19          DAD      D      ;INDEXED
0732 5E          MOV      E,M      ;LO BYTE
0733 23          INX      H
0734 56          MOV      D,M      ;HO BYTE
0735 EB          XCHG      ;TO H,L
0736 E9          PCHL      ;GONE...

;
JMPTAB: ;JUMP TABLE TO SUBROUTINES
0737 7E07      DW      ASSM      ;A ENTER ASSEMBLER LANGUAGE
0739 AB0B      DW      CERROR    ;B
073B AB0B      DW      CERROR    ;C
073D C607      DW      DISPLAY   ;D DISPLAY RAM MEMORY
073F AB0B      DW      CERROR    ;E
0741 5C08      DW      FILL      ;F FILL MEMORY
0743 7008      DW      GOTO      ;G GO TO MEMORY ADDRESS
0745 DA08      DW      HEXARI    ;H HEXADECIMAL SUM AND DIFFERENCE
0747 0409      DW      INFCB     ;I FILL INPUT FILE CONTROL BLOCK
0749 AB0B      DW      CERROR    ;J
074B AB0B      DW      CERROR    ;K
074D 9707      DW      LASSM     ;L LIST ASSEMBLY LANGUAGE

```

COPYRIGHT © 1980
 DIGITAL RESEARCH, INC.
 P. O. BOX 573
 PACIFIC GROVE, CA 93950
 SERIAL # _____

```

074F 5A09      DW      MOVE      ;M MOVE MEMORY
0751 AB0B      DW      CERROR    ;N
0753 AB0B      DW      CERROR    ;O
0755 AB0B      DW      CERROR    ;P
0757 AB0B      DW      CERROR    ;Q
0759 9C09      DW      READ      ;R READ HEXADECIMAL FILE
075B 7A0A      DW      SETMEM     ;S SET MEMORY COMMAND
075D C30A      DW      TRACE      ;T
075F BFOA      DW      UNTRACE    ;U
0761 AB0B      DW      CERROR    ;V
0763 AB0B      DW      CERROR    ;W
0765 E70A      DW      EXAMINE    ;X EXAMINE AND MODIFY REGISTERS
0767 AB0B      DW      CERROR    ;Y
0769 AB0B      DW      CERROR    ;Z

```

```

;
;

```

```

OPN:           ;FILE OPEN ROUTINE.  THIS SUBROUTINE OPENS THE DISK INPUT

```

```

076B E5        PUSH     H
076C D5        PUSH     D
076D C5        PUSH     B
076E AF        XRA      A
076F 325B00     STA      DBP      ;CLEAR BUFFER POINTER
0772 0E0F      MVI      C,OPF
0774 115C00     LXI      D,DFCB
0777 CDA206     CALL     TRAPAD    ;TO BDS
077A C1        POP      B
077B D1        POP      D
077C E1        POP      H
077D C9        RET

```

```

;

```

```

ASSM:          ;ASSEMBLER LANGUAGE INPUT

```

```

;             CHECK FOR ASSM PRESENT

```

```

077E CD9309     CALL     CHKDIS    ;ASSM/DISASSM PRESENT
0781 D2AB0B     JNC      CERROR    ;NOT THERE

```

```

;

```

```

0784 CD900C     CALL     SCANEXP   ;SCAN THE EXPRESSIONS WHICH FOLLOW
0787 3D        DCR      A          ;ONE EXPRESSION EXPECTED
0788 C2AB0B     JNZ      CERROR
078B CD660C     CALL     GETVAL     ;GET EXPRESSION TO H,L
078E 220C00     SHLD     DISPC
0791 CD0900     CALL     ASSEM
0794 C3FE06     JMP      START

```

```

;

```

```

LASSM:         ;ASSEMBLER LANGUAGE OUTPUT LISTING

```

```

;             L<CR> LISTS FROM CURRENT DISASSM PC FOR SEVERAL LINES

```

```

;             L<NUMBER><CR> LISTS FROM <NUMBER> FOR SEVERAL LINES

```

```

;             L<NUMBER>,<NUMBER> LISTS BETWEEN LOCATIONS

```

```

0797 CD9309     CALL     CHKDIS    ;DISASSM PRESENT?
079A D2AB0B     JNC      CERROR

```

```

;

```

```

079D CD900C     CALL     SCANEXP   ;SCAN EXPRESSIONS WHICH FOLLOW
07A0 CABB07     JZ       SPAGE      ;BRANCH IF NOT EXPRESSIONS
07A3 CD660C     CALL     GETVAL     ;EXP1 TO H,L
07A6 220C00     SHLD     DISPC      ;SETS BASE PC FOR LIST
07A9 3D        DCR      A          ;ONLY EXPRESSION?
07AA CABB07     JZ       SPAGE      ;SETS SINGLE PAGE MODE

```

COPYRIGHT © 1980
 DIGITAL RESEARCH, INC.
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SERIAL # _____

```

;
; ANOTHER EXPRESSION FOLLOWS
07AD CD660C CALL GETVAL
07B0 220E00 SHLD DISPM ;SETS MAX VALUE
07B3 3D DCR A
07B4 C2AB0B JNZ CERROR ;ERROR IF MORE EXPN'S
07B7 AF XRA A ;CLEAR PAGE MODE
07B8 C3BD07 JMP SPAGO

;
07BB 3E0C SPAGE: MVI A,PSIZE ;SCREEN SIZE FOR LIST
07BD 321000 SPAGO: STA DISPG
07C0 CD0600 CALL DISEN ;CALL DISASSEMBLER
07C3 C3FE06 JMP START ;FOR ANOTHER COMMAND

;
; DISPLAY MEMORY, FORMS ARE
; D DISPLAY FROM CUPRENT DISPLAY LINE
; DNNN SET DISPLAY LINE AND ASSUME D
; DNNN,MMM DISPLAY NNN TO MMM
; NEW DISPLAY LINE IS SET TO NEXT TO DISPLAY
;
; DISPLAY:
07C6 CD900C CALL SCANEXP ;GET 0,1,OR 2 EXPNS
07C9 CAE507 JZ DISP1 ;ASSUME CURRENT DISLOC
07CC CD660C CALL GETVAL ;GET VALUE TO H,L
07CF DAD507 JC DISPO ;CARRY SET IF ,B FORM
07D2 225D0F SHLD DISLOC ;OTHERWISE DISPC ALREADY SET
DISPO: ;GET NEXT VALUE
07D5 E67F ANI 7FH ;IN CASE ,B
07D7 3D DCR A
07D8 CAE507 JZ DISP1 ;SET HALF PAGE MODE
07DB CD660C CALL GETVAL
07DE 3D DCR A ;A,B,C NOT ALLOWED
07DF C2AB0B JNZ CERROR
07E2 C3F007 JMP DISP2

;
DISP1: ;0 OR 1 EXPN, DISPLAY HALF SCREEN
07E5 2A5D0F LHLD DISLOC
07E8 7D MOV A,L
07E9 E6F0 ANI 0F0H ;NORMALIZE TO LINE START
07EB 6F MOV L,A
07EC 11BF00 LXI D,PSIZE*16-1
07EF 19 DAD D
07F0 225F0F DISP2: SHLD DISMAX
; DISPLAY MEMORY FROM DISLOC TO DISMAX
DISP3: CALL CRLF
CALL BREAK ;BREAK KEY?
JNZ START ;STOP CURRENT EXPANSION
LHLD DISLOC
SHLD TDISP
CALL PADDR ;PRINT LINE ADDRESS
DISP4: CALL BLANK
MOV A,M ;GET NEXT DATA BYTE
CALL PBYTE ;PRINT BYTE
INX H
CALL DISCOM ;COMPARE H,L WITH DISMAX
JC DISCH ;CARRY SET IF H,L > DISMAX

```

COPYRIGHT © 1980
DIGITAL RESEARCH, INC.
P. O. BOX 579
PACIFIC GROVE, CA 93950
SERIAL # _____

```

0813 7D      MOV      A,L      ;CHECK FOR LINE OVERFLOW
0814 E60F    ANI       OFH
0816 C20508  JNZ      DISP4    ;JUMP FOR ANOTHER BYTE

```

```

;
DISCH:      ;DISPLAY AREA IN CHARACTER FORM

```

```

0819 225DOF  SHLD     DISLOC  ;UPDATE FOR NEXT WRITE
081C 2A610F  LHL      TDISP
081F EB      XCHG
0820 CDC50B  CALL     BLANK

```

```

;
DISCH0:     LDAX     D      ;GET BYTE
           CALL     PGRAPH  ;PRINT IF GRAPHIC CHARACTER

```

```

0827 13      INX      D
0828 2A5DOF  LHL      DISLOC  ;COMPARE FOR END OF LINE
082B 7D      MOV      A,L
082C 93      SUB      E
082D C22308  JNZ      DISCH0
0830 7C      MOV      A,H
0831 92      SUB      D
0832 C22308  JNZ      DISCH0

```

```

;
;
DROP THRU AT END OF CHARACTERS

```

```

0835 2A5DOF  LHL      DISLOC
0838 CD450C  CALL     DISCOM  ;END OF DISPLAY?
083B DAFE06  JC       START

```

```

;
;
NO, CONTINUE WITH NEXT LINE
JMP      DISP3

```

```

;
;
FILL MEMORY AREA WITH FIXED DATA ELEMENT

```

```

;
SCAN3:      ;SCAN THREE EXPN'S FOR FILL AND MOVE

```

```

0841 CD900C  CALL     SCANEXP
0844 FE03    CPI      3
0846 C2AB0B  JNZ      CERROR
0849 CD660C  CALL     GETVAL
084C E5      PUSH     H
084D CD660C  CALL     GETVAL
0850 E5      PUSH     H
0851 CD660C  CALL     GETVAL
0854 D1      POP      D
0855 C1      POP      B      ;BC,DE,HL
0856 C9      RET

```

```

;
BCDE:      ;COMPARE BC > DE (CARRY GEN'D IF TRUE)

```

```

0857 7B      MOV      A,E
0858 91      SUB      C
0859 7A      MOV      A,D
085A 98      SBB      B
085B C9      RET

```

```

;
FILL:

```

```

085C CD4108  CALL     SCAN3  ;EXPRESSIONS SCANNED BC , DE , HL
085F 7C      MOV      A,H    ;MUST BE ZERO
0860 B7      ORA      A

```

COPYRIGHT © 1980
 DIGITAL RESEARCH, INC.
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SERIAL # _____

```

0861 C2AB0B      JNZ      CERROR
0864 CD5708      FILL0:  CALL    BCDE      ;END OF FILL?
0867 DAFE06      JC       START
086A 7D          MOV      A,L          ;DATA
086B 02          STAX     B            ;TO MEMORY
086C 03          INX      B            ;NEXT TO FILL
086D C36408      JMP      FILL0

```

```

;
: GO COMMAND WITH OPTIONAL BREAKPOINTS

```

GOTO:

0870	CD150C	CALL	CRLF	;READY FOR GO.
0873	CD900C	CALL	SCANEXP	;0,1, OR 2 EXPS
0876	CD660C	CALL	GETVAL	
0879	E5	PUSH	H	;START ADDRESS
087A	CD660C	CALL	GETVAL	
087D	E5	PUSH	H	;BKPT1
087E	CD660C	CALL	GETVAL	
0881	44	MOV	B,H	;BKPT2
0882	4D	MOV	C,L	
0883	D1	POP	D	;BKPT1
0884	E1	POP	H	;GOTO ADDRESS

;
GOPR:

```

0885 F3          DI
0886 CAA108      JZ      GOP1      ;NO BREAK POINTS
0889 DA8F08     JC      GOP0

```

```

;      SET PC
      SHLD     PLOC      ;INTO MACHINE STATE

```

```

GOPO:      ;SET BREAKS
          ANI      7FH      ;CLEAR , BIT
          DCR      A        ;IF 1 THEN SKIP (2,3 IF BREAKPOINTS)
          JZ       GOP1
          CALL     SETBK    ;BREAK POINT FROM D,E
          DCR      A
          JZ       GOP1

```

: SECOND BREAK POINT

```

089C 59      MOV     E,C
089D 50      MOV     D,B      ;TO D,E
089E CDB208  CALL    SETBK    :SECOND BREAK POINT SET

```

```

;
GOP1:    ;RESTORE MACHINE STATE AND START IT

```

08A1	31AF0F	LXI	SP, STACK-12	
08A4	D1	POP	D	
08A5	C1	POP	B	
08A6	F1	POP	PSW	
08A7	E1	POP	H	;SP IN HL
08A8	F9	SPHL		
08A9	2AB90F	LHLD	PLOC	;PC IN HL
08AC	E5	PUSH	H	;INTO USER'S STACK
08AD	2AB70F	LHLD	HLOC	;HL RESTORED
08B0	FB	EI		
08B1	C9	RET		

```
;
SETBK:  ;SET BREAK POINT AT LOCATION D,E
```

08B2 F5 PUSH PSW

```

08B3 C5          PUSH      B
08B4 214F0F      LXI        H,BREAKS          ;NUMBER OF BREAKS SET SO FAR
08B7 7E          MOV        A,M
08B8 34          INR         M          ;COUNT BREAKS UP
08B9 B7          ORA         A          ;ONE SET ALREADY?
08BA CACD08      JZ          SETBKO
;
08BD 23          INX         H
08BE 7E          MOV        A,M          ;CHECK = ADDRESSES
08BF 23          INX         H
08C0 46          MOV        B,M          ;CHECK HO ADDRESS
08C1 23          INX         H
;
08C2 BB          CMP        E          ;LOW =?
08C3 C2CD08      JNZ         SETBKO
08C6 78          MOV        A,B
08C7 BA          CMP        D          ;HIGH =?
08C8 C2CD08      JNZ         SETBKO
;
08CB 7E          MOV        A,M          ;GET DATA BYTE
08CC 12          STAX        D          ;PUT BACK INTO CODE
08CD 23          SETBKO: INX        H          ;ADDRESS FIELD
08CE 73          MOV        M,E          ;LSB
08CF 23          INX         H
08D0 72          MOV        M,D          ;MSB
08D1 23          INX         H          ;DATA FIELD
08D2 1A          LDAX        D          ;GET BYTE FROM PROGRAM
08D3 77          MOV        M,A          ;TO BREAKS VECTOR
08D4 3EFF        MVI        A,RSTIN      ;RESTART INSTRUCTION
08D6 12          STAX        D          ;TO CODE
08D7 C1          POP         B
08D8 F1          POP         PSW
08D9 C9          RET

```

HEXARI:

HEXADECIMAL ARITHMETIC

Address	Operation	Comments
08DA	CD900C	
08DD	FE02	
08DF	C2AB0B	
08E2	CD660C	
08E5	E5	
08E6	CD660C	
08E9	D1	
08EA	E5	
08EB	CD150C	
08EE	19	
08EF	CD2E0C	
08F2	CDC50B	
08F5	E1	
08F6	AF	
08F7	95	
08F8	6F	
08F9	3E00	
08FB	9C	

Operation	Comments
CALL	SCANEXP
CPI	2
JNZ	CERROR
CALL	GETVAL ;FIRST VALUE TO H,L
PUSH	H
CALL	GETVAL ;SECOND VALUE TO H,L
POP	D ;FIRST VALUE TO D,E
PUSH	H ;SAVE A COPY OF SECOND VAALUE
CALL	CRLF ;NEW LINE
DAD	D ;SUM IN H,L
CALL	PADDR
CALL	BLANK
POP	H ;RESTORE SECOND VALUE
XRA	A ;CLEAR ACCUM FOR SUBTRACTION
SUB	L
MOV	L,A ;BACK TO L
MVI	A,0 ;CLEAR IT AGAIN
SBB	H

SERIAL # _____

COPYRIGHT © 1980
DIGITAL RESEARCH, INC.
P. O. BOX 579
PACIFIC GROVE, CA 93950

SERIAL # _____

```

08FC 67      MOV      H,A
08FD 19      DAD      D      ;DIFFERENCE IN HL
08FE CD2E0C  CALL     PADDR
0901 C3FE06  JMP      START

```

```

;
; SET INPUT FILE CONTROL BLOCK (AT 5CH) TO SIMULATE CONSOLE COMMAND
INFCB:

```

```

;      FILL FCB AT 5CH
0904 AF      XRA      A
0905 327C00  STA      FCB+FCR ;CLEAR CURRENT RECORD
0908 325C00  STA      FCB      ;CLEAR DISK NUMBER
090B CDDDOB  CALL     GNC      ;CHARACTER IN A
090E 0E09    MVI      C,9      ;FILE NAME LENGTH+1
0910 215D00  LXI      H,FCB+FFN ;START OF NAME

```

```

;
FLP:      ;FILL NAME

```

```

0913 77      MOV      M,A
0914 23      INX      H
0915 0D      DCR      C
0916 CAABOB  JZ        CERROR ;FILE NAME TOO LONG.

```

```

;
0919 CDDDOB  CALL     GNC      ;READ NEXT CHAR
091C FE2E    CPI      '.'
091E CA2609  JZ        FLB      ;FOUND ., BLANK OUT
;
NOT ., MAY BE CR
0921 FE0D    CPI      CR
0923 C21309  JNZ      FLP      ;FOR ANOTHER STORE

```

```

;
; NAME FILLED, EXTEND WITH BLANKS

```

```

FLB:      DCR      C
0926 0D      JZ        TFT
0927 CA3009  MVI      M,' '
092A 3620    INX      H
092C 23      JMP      FLB
092D C32609

```

```

;
; BLANKS FILLED, SCAN FILE TYPE IF '.' FOUND

```

```

TFT:      MVI      C,4
0930 0E04    CPI      '.'
0932 FE2E    JNZ      FLB1     ;ENDED WITH . OR CR
0934 C24B09  ;FILL REMAINDER WITH BLANKS

```

```

;
; SCAN FILE TYPE

```

```

LXI      H,FCB+FFT
0937 216500

```

```

;
FLP1:

```

```

CALL     GNC
093A CDDDOB  CPI      CR
093D FE0D    JZ        FLB1
093F CA4B09  MOV      M,A
0942 77      INX      H
0943 23      DCR      C
0944 0D      JZ        CERROR ;TOO LONG
0945 CAABOB  JMP      FLP1
0948 C33A09

```

```

;
; FILL WITH BLANKS

```

```

FLB1:     DCR      C
094B 0D      JZ        FLZ
094C CA5509  MVI      M,' '
094F 3620

```

COPYRIGHT © 1980
 DIGITAL RESEARCH, INC.
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SERIAL # _____

```

0951 23          INX      H
0952 C34B09      JLP      FLB1
;
;      ZERO THE EXTENT
0955 3600      FLZ:     MVI      M,0
0957 C3FE06      JMP      START
;
;      MOVE MEMORY
MOVE:
095A CD4108      CALL     SCAN3    ;BC,DE,HL
MOVE0:           ;HAS B,C PASSED D,E?
095D CD5708      CALL     BCDE
0960 DAFE06      JC      START    ;END OF MOVE
0963 0A          LDAX     B        ;CHAR TO ACCUM
0964 03          INX      B        ;NEXT TO GET
0965 77          MOV      M,A      ;MOVE IT TO MEMORY
0966 23          INX      H
0967 C35D09      JMP      MOVE0    ;FOR ANOTHER

```

```

;
;      READ FILES (HEX OR COM)
;

```

```

QHEX:           ;HEX FILE IF ZERO AT END

```

```

096A 216500      LXI      H,FCB+FFT
096D 7E          MOV      A,M
096E E67F      ANI      07FH      ;MASK HIGH ORDER BIT
0970 FE48      CPI      'H'
0972 C0          RNZ
0973 23          INX      H
0974 7E          MOV      A,M
0975 E67F      ANI      07FH      ;MASK HIGH ORDER BIT
0977 FE45      CPI      'E'
0979 C0          RNZ
097A 23          INX      H
097B 7E          MOV      A,M
097C E67F      ANI      07FH      ;MASK HIGH ORDER BIT
097E FE58      CPI      'X'
0980 C9          RET

```

```

;
COMLOAD:        ;COMPARE HL > MLOAD

```

```

0981 EB          XCHG      ;H,L TO D,E
0982 2A870F      LHLD     MLOAD    ;MLOAD TO H,L
0985 7D          MOV      A,L      ;MLOAD LSB
0986 93          SUB      E
0987 7C          MOV      A,H
0988 9A          SBB      D        ;MLOAD-OLDHL GENS CARRY IF HL>MLOAD
0989 EB          XCHG
098A C9          RET

```

```

;
CKMLOAD:        ;CHECK FOR HL > MLOAD AND SET MLOAD IF SO

```

```

098B CD8109      CALL     COMLOAD ;CARRY IF HL>MLOAD
098E D0          RNC
098F 22870F      SHLD     MLOAD    ;CHANGE IT
0992 C9          RET

```

```

;
CHKDIS:         ;CHECK FOR DISASM PRESENT

```

```

0993 E5          PUSH     H

```

COPYRIGHT © 1980
 DIGITAL RESEARCH, INC.
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SERIAL # _____

```

0994 210000      LXI      H,MODBAS      ;ENTRY POINT
0997 CD8109      CALL     COMLOAD
099A E1          POP      H
099B C9          RET

```

```

;
READ:

```

```

099C CD900C      CALL     SCANEXP
099F 210000      LXI      H,0
09A2 CAAC09      JZ       READN
09A5 3D          DCR      A      ;ONE EXPRESSION?
09A6 C2AB0B      JNZ      CERROR
09A9 CD660C      CALL     GETVAL      ;EXPRESSION TO H,L
09AC E5          READN:  PUSH     H      ;SAVE IT FOR BELOW
09AD CD6B07      RINIT:  CALL     OPN      ;OPEN INPUT FILE
09B0 FEFF        CPI      255
09B2 CAAB0B      JZ       CERROR
;               CONTINUE IF FILE OPEN WENT OK
;               DISK FILE OPENED AND INITIALIZED
;
;               CHECK FOR 'HEX' FILE AND LOAD DIRECT TIL EOF
09B5 CD6A09      CALL     QHEX      ;LOOK FOR 'HEX'
09B8 CAE109      JZ       HREAD

```

```

;               COM FILE, LOAD WITH OFFSET GIVEN BY PUSHED REGISTER H
09BB E1          POP      H
09BC 110001      LXI      D,100H      ;BASE OF TRANSIENT AREA
09BF 19          DAD      D

```

```

;               REG H HOLDS LOAD ADDRESS

```

```

LCOMO:           ;LOAD COM FILE
09C0 E5          PUSH     H      ;SAVE DMA ADDRESS
09C1 115C00      LXI      D,DFCB
09C4 0E14        MVI      C,RDF      ;READ SECTOR
09C6 CDA206      CALL     TRAPAD
09C9 E1          POP      H
09CA B7          ORA      A      ;SET FLAGS TO CHECK RETURN CODE
09CB C2460A      JNZ      RLIFT

```

```

;               MOVE FROM 80H TO LOAD ADDRESS IN H,L

```

```

09CE 118000      LXI      D,DBF
09D1 0E80        MVI      C,80H      ;BUFFER SIZE
09D3 1A          LCOM1:  LDAX     D      ;LOAD NEXT BYTE
09D4 13          INX      D
09D5 77          MOV      M,A      ;STORE NEXT BYTE
09D6 23          INX      H
09D7 0D          DCR      C
09D8 C2D309      JNZ      LCOM1

```

```

;               LOADED, CHECK ADDRESS AGAINST MLOAD

```

```

09DB CD8B09      CALL     CKMLOAD
09DE C3C009      JMP      LCOMO

```

```

;

```

```

;

```

```

;               OTHERWISE ASSUME HEX FILE IS BEING LOADED

```

```

09E1 CD740B      HREAD:  CALL     DISKR      ;NEXT CHAR TO ACCUM
09E4 FE1A        CPI      DEOF      ;PAST END OF TAPE?
09E6 CAAB0B      JZ       CERROR      ;FOR ANOTHER COMMAND
09E9 DE3A        SBI      ':'
09EB C2E109      JNZ      HREAD      ;LOOKING FOR START OF RECORD

```

COPYRIGHT © 1980
DIGITAL RESEARCH, INC.
P. O. BOX 579
PACIFIC GROVE, CA 93950

SERIAL # _____

```

;
; START FOUND, CLEAR CHECKSUM
09EE 57      MOV     D,A
09EF E1      POP     H
09F0 E5      PUSH    H
09F1 CD260A  CALL    RBYTE
09F4 5F      MOV     E,A      ;SAVE LENGTH
09F5 CD260A  CALL    RBYTE    ;HIGH ORDER ADDR
09F8 F5      PUSH    PSW
09F9 CF260A  CALL    RBYTE    ;LOW ORDER ADDR
09FC C1      POP     B
09FD 4F      MOV     C,A
09FE 09      DAD     B      ;BIASED ADDR IN H
09FF 7B      MOV     A,E      ;CHECK FOR LAST RECORD
0A00 B7      ORA     A
0A01 C20C0A  JNZ     RDTYPE
; END OF TAPE, SET LOAD ADDRESS
0A04 60      MOV     H,B
0A05 69      MOV     L,C
0A06 22B90F  SHLD    PLOC      ;SET PC VALUE
0A09 C3460A  JMP     RLIFT    ;FOR ANOTHER COMMAND
;
RDTYPE:
0A0C CD260A  CALL    RBYTE      ;RECORD TYPE = 0
;
; LOAD RECORD
RED1:
0A0F CD260A  CALL    RBYTE
0A12 77      MOV     M,A
0A13 23      INX     H
0A14 1D      DCR     E
0A15 C20F0A  JNZ     RED1      ;FOR ANOTHER BYTE
; OTHERWISE AT END OF RECORD - CHECKSUM
0A18 CD260A  CALL    RBYTE
0A1B F5      PUSH    PSW      ;FOR CHECKSUM CHECK
0A1C CD8B09  CALL    CKMLOAD ;CHECK AGAINST MLOAD
0A1F F1      POP     PSW
0A20 C2AB0B  JNZ     CERROR ;CHECKSUM ERROR
0A23 C3E109  JMP     HREAD    ;FOR ANOTHER RECORD
;
RBYTE:      ;READ ONE BYTE FROM BUFF AT WBP TO REG-A
; COMPUTE CHECKSUM IN REG-D
0A26 C5      PUSH    B
0A27 E5      PUSH    H
0A28 D5      PUSH    D
;
0A29 CD740B  CALL    DISKR    ;GET ONE MORE CHARACTER
0A2C CD590C  CALL    HEXCON   ;CONVERT TO HEX (OR ERROR)
;
; SHIFT LEFT AND MASK
0A2F 07      RLC
0A30 07      RLC
0A31 07      RLC
0A32 07      RLC
0A33 E6F0    ANI     OFOH
0A35 F5      PUSH    PSW      ;SAVE FOR A FEW STEPS
0A36 CD740B  CALL    DISKR

```

COPYRIGHT © 1980

DIGITAL RESEARCH, INC.

P. O. BOX 579

PACIFIC GROVE, CA 93950

SERIAL # _____

```

0A39 CD590C      CALL    HEXCON
;
;      OTHERWISE SECOND NIBBLE OK, SO MERGE
0A3C C1          POP     B      ;PREVIOUS NIBBLE TO REG-B
0A3D B0          ORA      B
0A3E 47          MOV     B,A    ;VALUE IS NOW IN B TEMPORARILY
0A3F D1          POP     D      ;CHECKSUM
0A40 82          ADD     D      ;ACCUMULATING
0A41 57          MOV     D,A    ;BACK TO CS
;
;      ZERO FLAG REMAINS SET
0A42 78          MOV     A,B    ;BRING BYTE BACK TO ACCUMULATOR
0A43 E1          POP     H
0A44 C1          POP     B      ;BACK TO INITIAL STATE WITH ACCUM SET
0A45 C9          RET
;
RLIFT:          ;LIFT HEAD ON DISK BEFORE RETURNING
0A46 0E0C        MVI     C,LIFT
0A48 CDA206      CALL    TRAPAD
;      'NEXT' 'PC'
0A4B 216F0A      LXI     H,LMSG ;LOAD MESSAGE
0A4E 7E          RLIO:    MOV     A,M
0A4F B7          ORA      A      ;LAST CHAR?
0A50 CA5A0A      JZ      RLI1
0A53 CDC70B      CALL    PCHAR
0A56 23          INX     H      ;NEXT CHAR
0A57 C34E0A      JMP     RLIO
0A5A CD150C      RLI1:    CALL    CRLF
0A5D 2A870F      LHLD    MLOAD
0A60 CD2E0C      CALL    PADDR
0A63 CDC50B      CALL    BLANK
0A66 2AB90F      LHLD    PLOC
0A69 CD2E0C      CALL    PADDR
0A6C C3FE06      JMP     START
0A6F 0D0A4E4558LMSG: DB     CR,LF,'NEXT PC',0
;
;      SET MEMORY COMMAND
;
;
SETMEM:          ;ONE EXPRESSION EXPECTED
0A7A CD900C      CALL    SCANEXP ;SETS FLAGS
0A7D 3D          DCR     A      ;ONE EXPRESSION ONLY
0A7E C2AB0B      JNZ     CERROR
0A81 CD660C      CALL    GETVAL ;START ADDRESS IS IN H,L
0A84 CD150C      SETM0:   CALL    CRLF ;NEW LINE
0A87 E5          PUSH    H      ;SAVE CURRENT ADDRESS
0A88 CD2E0C      CALL    PADDR ;PRINTED
0A8B CDC50B      CALL    BLANK ;SEPARATOR
0A8E E1          POP     H      ;GET DATA
0A8F 7E          MOV     A,M
0A90 E5          PUSH    H      ;SAVE ADDRESS TO FILL
0A91 CD050C      CALL    PBYTE ;PRINT BYTE
0A94 CDC50B      CALL    BLANK ;ANOTHER SEPARATOR
0A97 CDB60B      CALL    GETBUFF ;FILL INPUT BUFFER
0A9A CDDD0B      CALL    GNC    ;MAY BE EMPTY (NO CHANGE)
0A9D E1          POP     H      ;RESTORE ADDRESS TO FILL
0A9E FE0D        CPI     CR
0AA0 CABBOA      JZ      SETM1
0AA3 FE2E        CPI     ' '

```

COPYRIGHT © 1980
 DIGITAL RESEARCH, INC.
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SERIAL # _____

```

OAA5 CAFE06      JZ      START
                  ;      DATA IS BEING CHANGED
OAA8 E5          PUSH    H      ;SAVE ADDR TO FILL
OAA9 CD930C      CALL    SCANEX ;FIRST CHARACTER ALREADY SCANNED
OAA8 3D          DCR      A      ;ONE ITEM?
OAA8 C2AB0B      JNZ     CERROR ;MORE THAN ONE
OAB0 CD660C      CALL    GETVAL ;VALUE TO H,L
OAB3 7C          MOV     A,H
OAB4 B7          ORA     A      ;HO ZERO?
OAB5 C2AB0B      JNZ     CERROR ;DATA IS IN L
OAB8 7D          MOV     A,L
OAB9 E1          POP     H      ;RESTORE DATA VALUE
OABA 77          MOV     M,A
OABB 23          SETM1: INX    H      ;NEXT ADDRESS READY
OABC C3840A      JMP     SETMO
                  ;
                  ;      UNTRACE MODE
UNTRACE:
OABF AF          XRA     A      ;CLEAR TRACE MODE FLAG
OAC0 C3C50A      JMP     ETRACE
                  ;
                  ;      START TRACE
OAC3 3EFF        TRACE: MVI    A,OFFH ;SET TRACE MODE FLAG
ETRACE:
OAC5 324C0F      STA     TMODE
OAC8 CD900C      CALL    SCANEXP
OACB 210000      LXI     H,0
OACE CADE0A      JZ      TRACO
                  ;      MUST BE T OR TN (N NOT 0)
OAD1 3D          DCR      A      ;COUNT MUST BE ONE
OAD2 C2AB0B      JNZ     CERROR
OAD5 CD660C      CALL    GETVAL ;GET VALUE TO HL
OAD8 7D          MOV     A,L      ;CHECK FOR ZERO
OAD9 B4          ORA     H
OADA CAAB0B      JZ      CERROR
OADD 2B          DCX     H      ;TRACE VALUE - 1
OADE 224D0F      TRACO: SHLD   TRACER
OAE1 CD440D      CALL    DSTATE ;STARTING STATE IS DISPLAYED
OAE4 C38508      JMP     GOPR    ;SETS BREAKPOINTS AND STARTS EXECUTION
                  ;
                  ;      EXAMINE AND MODIFY CPU REGISTERS.
EXAMINE:
OAE7 CDDDOB      CALL    GNC     ;CR?
OAEA FE0D        CPI     CR
OAE8 C2F50A      JNZ     EXAMO
OAEF CD440D      CALL    DSTATE ;DISPLAY CPU STATE
OAF2 C3FE06      JMP     START
                  ;
EXAMO:           ;REGISTER CHANGE OPERATION
OAF5 010B00      LXI     B,PVAL+1 ;B=0,C=PVAL (MAX REGISTER NUMBER)
                  ;      LOOK FOR REGISTER MATCH IN RVECT
OAF8 21B30D      LXI     H,RVECT
OAFB BE          EXAM1: CMP     M      ;MATCH IN RVECT?
OAF8 CA080B      JZ      EXAM2
OAFF 23          INX     H      ;NEXT RVECT
OB00 04          INR     B      ;INCREMENT COUNT

```

COPYRIGHT © 1980
 DIGITAL RESEARCH, INC.
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SERIAL # _____

```

OB01 0D          DCR      C          ;END OF RVECT?
OB02 C2FB0A      JNZ      EXAM1
;               NO MATCH
OB05 C3AB0B      JMP      CERROR
;
EXAM2:           ;MATCH IN RVECT, B HAS REGISTER NUMBER
OB08 CDDDOB      CALL     GNC
OB0B FE0D        CPI      CR          ;ONLY CHARACTER?
OB0D C2AB0B      JNZ      CERROR
;
;               WRITE CONTENTS, AND GET ANOTHER BUFFER
OB10 C5          PUSH     B          ;SAVE COUNT
OB11 CD150C      CALL     CRLF       ;NEW LINE FOR ELEMENT
OB14 CD1A0D      CALL     DELT       ;ELEMENT WRITTEN
OB17 CDC50B      CALL     BLANK
OB1A CDB60B      CALL     GETBUFF    ;FILL COMMAND BUFFER
OB1D CD900C      CALL     SCANEXP    ;GET INPUT EXPRESSION
OB20 B7          ORA      A          ;NONE?
OB21 CAFE06      JZ       START
OB24 3D          DCR      A          ;MUST BE ONLY ONE
OB25 C2AB0B      JNZ      CERROR
OB28 CD660C      CALL     GETVAL     ;VALUE IS IN H,L
OB2B C1          POP      B          ;RECALL REGISTER NUMBER
;               CHECK CASES FOR FLAGS, REG-A, OR DOUBLE REGISTER
OB2C 78          MOV      A,B
OB2D FE05        CPI      AVAL
OB2F D2590B      JNC      EXAM4
;               SETTING FLAGS, MUST BE ZERO OR ONE
OB32 7C          MOV      A,H
OB33 B7          ORA      A
OB34 C2AB0B      JNZ      CERROR
OB37 7D          MOV      A,L
OB38 FE02        CPI      2
OB3A D2AB0B      JNC      CERROR
;               0 OR 1 IN H,L REGISTERS - GET CURRENT FLAGS AND MASK POSITION
OB3D CDE30C      CALL     FLGSHF
;               SHIFT COUNT IN C, D,E ADDRESS FLAG POSITION
OB40 67          MOV      H,A        ;FLAGS TO H
OB41 41          MOV      B,C        ;SHIFT COUNT TO B
OB42 3EFE        MVI      A,0FEH    ;11111110 IN ACCUM TO ROTATE
OB44 CD530B      CALL     LROTATE    ;ROTATE REG-A LEFT
OB47 A4          ANA      H          ;MASK ALL BUT ALTERED BIT
OB48 41          MOV      B,C        ;RESTORE SHIFT COUNT TO B
OB49 67          MOV      H,A        ;SAVE MASKED FLAGS
OB4A 7D          MOV      A,L        ;0/1 TO LSB OF ACCUM
OB4B CD530B      CALL     LROTATE    ;ROTATED TO CHANGED POSITION
OB4E B4          ORA      H          ;RESTORE ALL OTHER FLAGS
OB4F 12          STAX     D          ;BACK TO MACHINE STATE
OB50 C3FE06      JMP      START     ;FOR ANOTHER COMMAND
;
LROTATE:         ;LEFT ROTATE FOR FLAG SETTING
;               PATTERN IS IN REGISTER A, COUNT IN REGISTER B
OB53 05          DCR      B
OB54 C8          RZ              ;ROTATE COMPLETE
OB55 07          RLC              ;END-AROUND ROTATE
OB56 C3530B      JMP      LROTATE

```

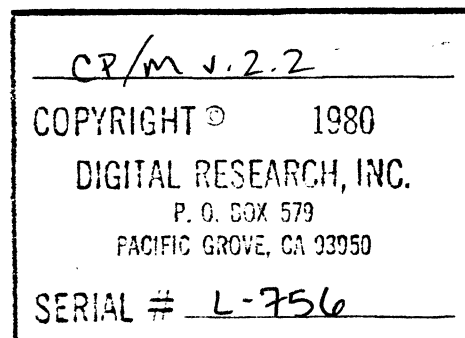
COPYRIGHT © 1980
DIGITAL RESEARCH, INC.
P. O. BOX 579
PACIFIC GROVE, CA 93950

SERIAL # _____

```

;
EXAM4: ;MAY BE ACCUMULATOR CHANGE
0B59 C2690B JNZ EXAM5
; MUST BE BYTE VALUE
0B5C 7C MOV A,H
0B5D B7 ORA A
0B5E C2AB0B JNZ CERROR
0B61 7D MOV A,L ;GET BYTE TO STORE
0B62 21B40F LXI H,ALOC ;A REG LOCATION IN MACHINE STATE
0B65 77 MOV M,A ;STORE IT AWAY
0B66 C3FE06 JMP START
;
EXAM5: ;MUST BE DOUBLE REGISTER PAIR
0B69 E5 PUSH H ;SAVE VALUE
0B6A CD010D CALL GETDBA ;DOUBLE ADDRESS TO HL
0B6D D1 POP D ;VALUE TO D,E
0B6E 73 MOV M,E
0B6F 23 INX H
0B70 72 MOV M,D ;ALTERED MACHINE STATE
0B71 C3FE06 JMP START
;
DISKR: ;DISK READ
0B74 E5 PUSH H
0B75 D5 PUSH D
0B76 C5 PUSH B
;
RDI: ;READ DISK INPUT
0B77 3A5B00 LDA DBP
0B7A E67F ANI 7FH
0B7C CA940B JZ NDI ;GET NEXT DISK INPUT RECORD
;
; READ CHARACTER
RDC:
0B7F 1600 MVI D,0
0B81 5F MOV E,A
0B82 218000 LXI H,DBF
0B85 19 DAD D
0B86 7E MOV A,M
0B87 FE1A CPI DEOF
0B89 CAA60B JZ DEF ;END OF FILE
0B8C 215B00 LXI H,DBP
0B8F 34 INR M
0B90 B7 ORA A
0B91 C3A70B JMP RRET
;
NDI: ;NEXT BUFFER IN
0B94 0E14 MVI C,RDF
0B96 115C00 LXI D,DFCB
0B99 CDA206 CALL TRAPAD
0B9C B7 ORA A
0B9D C2A60B JNZ DEF
;
; BUFFER READ OK
0BA0 325B00 STA DBP ;STORE 00H
0BA3 C37F0B JMP RDC
;

```



```

DEF:      ;SET CARRY AND RETURN (END FILE)
          STC
OBA6 37
RRET:
OBA7 C1   POP    B
OBA8 D1   POP    D
OBA9 E1   POP    H
OBAA C9   RET

;
CERROR:   ;ERROR IN COMMAND
          CALL    CRLF
          MVI     A,'?'
          CALL    PCHAR
          JMP     START

;
; SUBROUTINES
GETBUFF:  ;FILL COMMAND BUFFER AND SET POINTERS
          MVI     C,GETF ;GET BUFFER FUNCTION
          LXI     D,COMLEN;START OF COMMAND BUFFER
          CALL    TRAPAD ;FILL BUFFER
          LXI     H,COMBUF;NEXT TO GET
          SHLD    NEXTCOM
          RET

;
BLANK:
OBC5 3E20 MVI     A,' '

;
PCHAR:    ;PRINT CHARACTER TO CONSOLE
          PUSH    H
          PUSH    D
          PUSH    B
          MOV     E,A
          MVI     C,COF
          CALL    TRAPAD
          POP     B
          POP     D
          POP     H
          RET

;
TRANS:    ;
          ;
          TRANSLATE TO UPPER CASE
          CPI     7FH ;RUBOUT?
          RZ
          CPI     ('A' OR 0100000B) ;UPPER CASE A
          RC
          ANI     1011111B ;CLEAR UPPER CASE BIT
          RET

;
GNC:
          ;
          GET NEXT BUFFER CHARACTER FROM CONSOLE
          PUSH    H ;SAVE FOR REUSE LOCALLY
          LXI     H,CURLN
          MOV     A,M
          ORA     A ;ZERO?
          MVI     A,CR
          JZ      GNCRET ;RETURN WITH CR IF EXHAUSTED
          DCR     M ;CURLN=CURLN-1

```

COPYRIGHT © 1980
 DIGITAL RESEARCH, INC.
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SERIAL # _____

```

OBE9 2A630F      LHLD      NEXTCOM
OBEC 7E          MOV       A,M      ;GET NEXT CHARACTER
OBED 23          INX        H      ;NEXTCOM=NEXTCOM+1
OBEE 22630F      SHLD      NEXTCOM ;UPDATED
OBF1 CDD40B      CALL      TRANS
OBF4 E1          GNCRET: POP      H      ;RESTORE ENVIRONMENT
OBF5 C9          RET

;
PNIB:            ;PRINT NIBBLE IN LO ACCUM
OBF6 FE0A        CPI        10
OBF8 D2000C      JNC        PNIBH    ;JUMP IF A-F
OBF9 C630        ADI        '0'
OBF9 C3C70B      JMP        PCHAR    ;RET THRU PCHAR
OC00 C637        PNIBH: ADI        'A'-10
OC02 C3C70B      JMP        PCHAR

;
PBYTE:           PUSH      PSW      ;SAVE A COPY FOR LO NIBBLE
OC05 F5          RAR
OC06 1F          RAR
OC07 1F          RAR
OC08 1F          RAR
OC09 1F          RAR
OC0A E60F        ANI        OFH      ;MASK HO NIBBLE TO LO NIBBLE
OC0C CDF60B      CALL      PNIB
OC0F F1          POP        PSW      ;RECALL BYTE
OC10 E60F        ANI        OFH
OC12 C3F60B      JMP        PNIB

;
CRLF:            ;CARRIAGE RETURN LINE FEED
OC15 3E0D        MVI        A,CR
OC17 CDC70B      CALL      PCHAR
OC1A 3E0A        MVI        A,LF
OC1C C3C70B      JMP        PCHAR

;
BREAK:           ;CHECK FOR BREAK KEY
OC1F C5          PUSH      B
OC20 D5          PUSH      D
OC21 E5          PUSH      H
OC22 0E0B        MVI        C,CHKIO
OC24 CDA206      CALL      TRAPAD
OC27 E601        ANI        1B
OC29 E1          POP        H
OC2A D1          POP        D
OC2B C1          POP        B
OC2C C9          RET

;
PADDX:           ;SAME AS PADDR, EXCEPT PRINT VALUE IN D,E
OC2D EB          XCHG

;
PADDR:           ;PRINT THE ADDRESS VALUE IN H,L
OC2E 7C          MOV        A,H
OC2F CD050C      CALL      PBYTE
OC32 7D          MOV        A,L
OC33 C3050C      JMP        PBYTE

;
PGRAPH:          ;PRINT GRAPHIC CHARACTER IN REG-A OR '.' IF NOT
OC36 FE7F        CPI        7FH

```

COPYRIGHT © 1980
 DIGITAL RESEARCH, INC.
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SERIAL # _____

```

0C38 D2400C      JNC      PPERIOD
0C3B FE20        CPI
0C3D D2C70B      JNC      PCHAR

```

PPERIOD:

```

0C40 3E2E        MVI      A, ' '
0C42 C3C70B      JMP      PCHAR

```

;

DISCOM: ;COMPARE H,L AGAINST DISMAX. CARRY SET IF HL > DISMAX AND

```

0C45 EB          XCHG
0C46 2A5F0F      LHLD     DISMAX
0C49 7D          MOV      A,L
0C4A 93          SUB      E
0C4B 6F          MOV      L,A      ;REPLACE FOR ZERO TESTS LATER
0C4C 7C          MOV      A,H
0C4D 9A          SBB      D
0C4E EB          XCHG
0C4F C9          RET

```

;

DELIM: ;CHECK FOR DELIMITER CHARACTER

```

0C50 FE0D        CPI      CR
0C52 C8          RZ
0C53 FE2C        CPI      ' '
0C55 C8          RZ
0C56 FE20        CPI
0C58 C9          RET

```

;

HEXCON: ;CONVERT ACCUMULATOR TO PURE BINARY FROM EXTERNAL HEX

```

0C59 D630        SUI      '0'
0C5B FE0A        CPI      10
0C5D D8          RC          ;MUST BE 0-9
0C5E C6F9        ADI      ('0'-'A'+10) AND OFFH
0C60 FE10        CPI      16
0C62 D8          RC          ;MUST BE 0-15
0C63 C3AB0B      JMP      CERROR ;BAD HEX DIGIT

```

;

GETVAL: ;GET NEXT EXPRESSION VALUE TO H,L (POINTER IN D,E ASSUMED)

```

0C66 EB          XCHG
0C67 5E          MOV      E,M
0C68 23          INX      H
0C69 56          MOV      D,M
0C6A 23          INX      H
0C6B EB          XCHG
0C6C C9          RET

```

;

GETEXP: ;GET HEX VALUE TO D,E

```

0C6D EB          XCHG
0C6E 210000      LXI      H,0

```

GETEXP0:

```

0C71 CD590C      CALL     HEXCON
0C74 29          DAD      H      ;*2
0C75 29          DAD      H      ;*4
0C76 29          DAD      H      ;*8
0C77 29          DAD      H      ;*16
0C78 B5          ORA      L      ;HL=HL+HEX
0C79 6F          MOV      L,A
0C7A CDDD0B      CALL     GNC

```

COPYRIGHT © 1980
DIGITAL RESEARCH, INC.

P. O. BOX 579
PACIFIC GROVE, CA 93950

SERIAL # _____

```

0C7D CD500C      CALL    DELIM    ;DELIMITER?
0C80 C2710C      JNZ     GETEXPO
0C83 EB          XCHG
0C84 C9          RET

```

```

;
SCSTORE:         ;STORE D,E TO H,L AND INCREMENT ADDRESS

```

```

0C85 73          MOV     M,E
0C86 23          INX     H
0C87 72          MOV     M,D
0C88 23          INX     H
0C89 E5          PUSH    H
0C8A 21560F      LXI     H,EXPLIST
0C8D 34          INR     M      ;COUNT NUMBER OF EXPN'S
0C8E E1          POP     H
0C8F C9          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

```

```

0C90 CDDDOB      CALL    GNC
SCANEX:         ;ENTER HERE IF CHARACTER ALREADY SCANNED
0C93 21560F      LXI     H,EXPLIST
0C96 3600        MVI     M,0      ;ZERO EXPRESSIONS
0C98 23          INX     H      ;READY TO FILL EXPRESSION LIST
0C99 FE0D        CPI     CR      ;END OF LINE?
0C9B CAD50C      JZ      SCANRET

```

```

;
;               NOT CR, MUST BE DIGIT OR COMMA

```

```

0C9E FE2C        CPI     ,
0CA0 C2AE0C      JNZ     SCANE0
;               MARK AS COMMA
0CA3 3E80        MVI     A,80H
0CA5 32560F      STA     EXPLIST
0CA8 110000      LXI     D,0
0CAB C3B10C      JMP     SCANE1

```

```

;
SCANE0:         ;NOT CR OR COMMA

```

```

0CAE CD6D0C      CALL    GETEXP ;EXPRESSION TO D,E
0CB1 CD850C      SCANE1: CALL    SCSTORE ;STORE THE EXPRESSION AND INCREMENT H,L
0CB4 FE0D        CPI     CR
0CB6 CAD50C      JZ      SCANRET
0CB9 CDDDOB      CALL    GNC
0CBC CD6D0C      CALL    GETEXP
0CBF CD850C      CALL    SCSTORE
;               SECOND DIGIT SCANNED

```

```

0CC2 FE0D        CPI     CR
0CC4 CAD50C      JZ      SCANRET
0CC7 CDDDOB      CALL    GNC
0CCA CD6D0C      CALL    GETEXP
0CCD CD850C      CALL    SCSTORE
0CD0 FE0D        CPI     CR
0CD2 C2AB0B      JNZ     CERROR

```

```

SCANRET:

```

```

0CD5 11560F      LXI     D,EXPLIST
0CD8 1A          LDAX    D
0CD9 FE81        CPI     81H

```

```

;LOOK AT COUNT
;LOAD COUNT TO ACC
; , WITHOUT B?

```

COPYRIGHT © 1980
 DIGITAL RESEARCH, INC.
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SERIAL # _____

```

OCDB CAABOB      JZ      CERROR
OCDE 13          INX      D          ;READY TO EXTRACT EXPN'S
OCDF B7          ORA      A          ;ZERO FLAG MAY BE SET
OCEO 07          RLC
OCE1 0F          RRC          ;SET CARRY IF HO BIT SET (,B)
OCE2 C9          RET          ;WITH FLAGS SET

```

```

;
;
;
;
;
;
;
;

```

SUBROUTINES FOR CPU STATE DISPLAY

```

FLGSHF: ;SHIFT COMPUTATION FOR FLAG GIVEN BY REG-B
;      REG A CONTAINS FLAG UPON EXIT (UNSHIFTED)
;      REG C CONTAINS NUMBER OF SHIFTS REQUIRED+1
;      REGS D,E CONTAIN ADDRESS OF FLAGS IN TEMPLATE

```

```

OCE3 E5          PUSH     H
OCE4 21C30D      LXI      H,FLGTAB      ;SHIFT TABLE
OCE7 58          MOV      E,B
OCE8 1600        MVI      D,0
OCEA 19          DAD      D
OCEB 4E          MOV      C,M          ;SHIFT COUNT TO C
OCEC 21B30F      LXI      H,FLOC      ;ADDRESS OF FLAGS
OCEF 7E          MOV      A,M          ;TO REG A
OCF0 EB          XCHG          ;SAVE ADDRESS
OCF1 E1          POP      H
OCF2 C9          RET

```

```

;

```

```

GETFLG: ;GET FLAG GIVEN BY REG-B TO REG-A AND MASK
CALL     FLGSHF ;BITS TO SHIFT IN REG-A

```

```

OCF3 CDE30C      CALL     FLGSHF ;BITS TO SHIFT IN REG-A
OCF6 0D          GETFLO: DCR      C
OCF7 CAFE0C      JZ      GETFL1
OCFA 1F          RAR
OCFB C3F60C      JMP      GETFLO
OCFE E601        GETFL1: ANI      1B
OD00 C9          RET

```

```

;

```

```

GETDBA: ;GET DOUBLE BYTE ADDRESS CORRESPONDING TO REG-A TO HL

```

```

OD01 D606        SUI      BVAL      ;NORMALIZE TO 0,1,...
OD03 21BE0D      LXI      H,RINX    ;INDEX TO STACKED VALUES
OD06 5F          MOV      E,A      ;INDEX TO E
OD07 1600        MVI      D,0      ;DOUBLE PRECISION
OD09 19          DAD      D      ;INDEXED INTO VECTOR
OD0A 5E          MOV      E,M      ;OFFSET TO E
OD0B 16FF        MVI      D,OFFH    ;-1
OD0D 21BB0F      LXI      H,STACK
OD10 19          DAD      D      ;HL HAS BASE ADDRESS
OD11 C9          RET

```

```

;

```

```

GETDBL: ;GET DOUBLE BYTE CORRESPONDING TO REG-A TO HL

```

```

OD12 CD010D      CALL     GETDBA ;ADDRESS OF ELT IN HL
OD15 5E          MOV      E,M      ;LSB
OD16 23          INX      H
OD17 56          MOV      D,M      ;MSB
OD18 EB          XCHG          ;BACK TO HL
OD19 C9          RET

```

```

;

```

```

DELT: ;DISPLAY CPU ELEMENT GIVEN BY COUNT IN REG-B, ADDRESS IN H,L
MOV      A,M      ;GET CHARACTER

```

```

OD1A 7E

```

COPYRIGHT © 1980
DIGITAL RESEARCH, INC.
P. O. BOX 579
PACIFIC GROVE, CA 93950
SERIAL #

```

OD1B CDC70B      CALL    PCHAR    ;PRINT IT
OD1E 78          MOV     A,B      ;GET COUNT
OD1F FE05        CPI     AVAL     ;PAST A?
OD21 D22B0D      JNC     DELTO    ;JMP IF NOT FLAG
;
;
;      DISPLAY FLAG
OD24 CDF30C      CALL    GETFLG   ;FLAG TO REG-A
OD27 CDF60B      CALL    PNIB
OD2A C9          RET
;
;
DELTO:           ;NOT FLAG, DISPLAY = AND DATA
OD2B F5          PUSH    PSW
OD2C 3E3D        MVI     A,'='
OD2E CDC70B      CALL    PCHAR
OD31 F1          POP     PSW
OD32 C23D0D      JNZ     DELT1    ;JUMP IF NOT REG-A
;
;
;      REGISTER A, DISPLAY BYTE VALUE
OD35 21B40F      LXI     H,ALOC
OD38 7E          MOV     A,M
OD39 CD050C      CALL    PBYTE
OD3C C9          RET
;
;
DEL1:            ;DOUBLE BYTE DISPLAY
OD3D CD120D      CALL    GETDBL   ;TO H,L
OD40 CD2E0C      CALL    PADDR   ;PRINTED
OD43 C9          RET
;
;
DSTATE:         ;DISPLAY CPU STATE
OD44 21B30D      LXI     H,RVECT  ;REGISTER VECTOR
OD47 0600        MVI     B,0      ;REGISTER COUNT
OD49 CD150C      CALL    CRLF
OD4C C5          DSTAO:  PUSH    B
OD4D E5          PUSH    H
OD4E CD1A0D      CALL    DELT     ;ELEMENT DISPLAYED
OD51 E1          POP     H        ;RVECT ADDRESS RESTORED
OD52 C1          POP     B        ;COUNT RESTORED
OD53 04          INR     B        ;NEXT COUNT
OD54 23          INX     H        ;NEXT REGISTER
OD55 78          MOV     A,B      ;LAST COUNT?
OD56 FE0B        CPI     PVAL+1
OD58 D2660D      JNC     DSTA1    ;JMP IF PAST END
OD5B FE05        CPI     AVAL     ;BLANK AFTER?
OD5D DA4C0D      JC      DSTAO
;
;      YES, BLANK AND GO AGAIN
OD60 CDC50B      CALL    BLANK
OD63 C34C0D      JMP     DSTAO
;
;
;      READY TO SEND DECODED INSTRUCTION
DSTA1:
OD66 CDC50B      CALL    BLANK
OD69 CD850E      CALL    NBRK     ;COMPUTE BREAKPOINTS IN CASE OF TRACE
OD6C F5          PUSH    PSW      ;SAVE EXPRESSION COUNT - B,C AND D,E HAVE BPTS
OD6D D5          PUSH    D        ;SAVE BP ADDRESS
OD6E C5          PUSH    B        ;SAVE AUX BREAKPOINT
OD6F CD9309      CALL    CHKDIS   ;CHECK TO SEE IF DISASSEMBLER IS HERE

```

COPYRIGHT © 1980
 DIGITAL RESEARCH, INC.
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SERIAL # _____

```

0D72 D2860D      JNC      DCHEX      ;DISPLAY HEX IF NOT
;               DISASSEMBLE CODE
0D75 2AB90F      LHL D      PLOC      ;GET CURRENT PC
0D78 220C00      SHLD     DISPC      ;SET DISASSM PC
0D7B 211000      LXI      H,DISPG;PAGE MODE = OFFH TO TRACE
0D7E 36FF        MVI      M,OFFH
0D80 CD0600      CALL     DISEN
0D83 C3AF0D      JMP      DSTRET
;
DCHEX:           ;DISPLAY HEX
0D86 2B          DCX      H           ;POINT TO LAST TO WRITE
0D87 225F0F      SHLD     DISMAX      ;SAVE FOR COMPARE BELOW
0D8A 2AB90F      LHL D      PLOC      ;START ADDRESS OF TRACE
0D8D 7E          MOV      A,M         ;GET OPCODE
0D8E CD050C      CALL     PBYTE
0D91 23          INX      H           ;READY FOR NEXT BYTE
0D92 CD450C      CALL     DISCOM      ;ZERO SET IF ONE BYTE TO PRINT, CARRY IF NO MORE
0D95 DAAF0D      JC       DSTRET
0D98 F5          PUSH     PSW         ;SAVE RESULT OF ZERO TEST
0D99 CDC50B      CALL     BLANK       ;SEPARATOR
0D9C F1          POP      PSW         ;RECALL ZERO TEST
0D9D B3          ORA      E           ;ZERO TEST
0D9E CAAB0D      JZ       DSTA2
;               DISPLAY DOUBLE BYTE
0DA1 5E          MOV      E,M
0DA2 23          INX      H
0DA3 56          MOV      D,M
0DA4 EB          XCHG
0DA5 CD2E0C      CALL     PADDR      ;PRINT ADDRESS
0DA8 C3AF0D      JMP      DSTRET
;
DSTA2:           ;PRINT BYTE VALUE
0DAB 7E          MOV      A,M
0DAC CD050C      CALL     PBYTE
DSTRET:
0DAF C1          POP      B           ;AUX BREAKPOINT
0DB0 D1          POP      D           ;RESTORE BREAKPOINT
0DB1 F1          POP      PSW        ;RESTORE COUNT
0DB2 C9          RET
;
;               DATA VECTORS FOR CPU DISPLAY
0DB3 435A4D4549RVECT: DB      'CZMEIABDHSP'
0DBE F6          RINX:  DB      (BLOC-STACK) AND OFFH ;LOCATION OF BC
0DBF F4          DB      (DLOC-STACK) AND OFFH ;LOCATION OF DE
0DC0 FC          DB      (HLOC-STACK) AND OFFH ;LOCATION OF HL
0DC1 FA          DB      (SLOC-STACK) AND OFFH ;LOCATION OF SP
0DC2 FE          DB      (PLOC-STACK) AND OFFH ;LOCATION OF PC
;               FLGTAB ELEMENTS DETERMINE SHIFT COUNT TO SET/EXTRACT FLAGS
0DC3 0107080305FLGTAB: DB      1,7,8,3,5 ;CY, ZER, SIGN, PAR, IDCY
;
CLRTRACE:        ;CLEAR THE TRACE FLAG
0DC8 210000      LXI      H,0
0DCB 224D0F      SHLD     TRACER
0DCE C9          RET
;
BREAKP:          ;ARRIVE HERE WHEN PROGRAMMED BREAK OCCURS

```

COPYRIGHT © 1980
 DIGITAL RESEARCH, INC.
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SERIAL # _____

```

ODCF F3          DI
ODD0 22B70F      SHLD  HLOC    ;HL SAVED
ODD3 E1          POP  H        ;RECALL RETURN ADDRESS
ODD4 2B          DCX  H        ;DECREMENT FOR RESTART
ODD5 22B90F      SHLD  PLOC
;               DAD SP BELOW DESTROYS CY, SO SAVE AND RECALL
ODD8 F5          PUSH PSW      ;INTO USER'S STACK
ODD9 210200      LXI  H,2      ;BIAS SP BY 2 BECAUSE OF PUSH
ODDC 39          DAD  SP       ;SP IN HL
ODDD F1          POP  PSW      ;RESTORE CY AND FLAGS
ODDE 31B70F      LXI  SP,STACK-4;LOCAL STACK
ODE1 E5          PUSH H        ;SP SAVED
ODE2 F5          PUSH PSW
ODE3 C5          PUSH B
ODE4 D5          PUSH D
;               MACHINE STATE SAVED, CLEAR BREAK POINTS
ODE5 2AB90F      LHL D PLOC    ;CHECK FOR RST INSTRUCTION
ODE8 7E          MOV  A,M      ;OPCODE TO A
ODE9 FEFF        CPI  RSTIN
;               SAVE CONDITION CODES FOR LATER TEST
ODEB F5          PUSH PSW
;               SAVE PLOC FOR LATER INCREMENT OR DECREMENT
ODEC E5          PUSH H
;
;               CLEAR BREAKPOINTS WHICH ARE PENDING
ODED 214F0F      LXI  H,BREAKS
ODF0 7E          MOV  A,M
ODF1 3600        MVI  M,0      ;SET TO ZERO BREAKS
ODF3 B7          CLERO: ORA  A   ;ANY MORE?
ODF4 CA040E      JZ    CLER1
ODF7 3D          DCR  A
ODF8 47          MOV  B,A      ;SAVE COUNT
ODF9 23          INX  H        ;ADDRESS OF BREAK
ODFA 5E          MOV  E,M      ;LOW ADDR
ODFB 23          INX  H
ODFC 56          MOV  D,M      ;HIGH ADDR
ODFD 23          INX  H
ODFE 7E          MOV  A,M      ;INSTRUCTION
ODFF 12          STAX D        ;BACK TO PROGRAM
OE00 78          MOV  A,B      ;RESTORE COUNT
OE01 C3F30D      JMP  CLERO
;
CLER1:           ;CLEARED, CONTINUE TRACING, OR STOP EXECUTION
OE04 E1          POP  H        ;RESTORE PLOC
OE05 F1          POP  PSW      ;RESTORE CONDITION FLAGS
OE06 CA280E      JZ    BREAKO  ;BRANCH IF PROGRAMMED INTERRUPT
;
;               MUST BE FRONT PANEL INTERRUPT, CHECK IF IN BDOS
OE09 23          INX  H        ;DON'T DECREMENT ON PANEL INTERRUPT
OE0A 22B90F      SHLD PLOC     ;RESTORE TO NEXT LOGICAL INSTRUCTION
OE0D EB          XCHG          ;TO D,E FOR COMPARE
OE0E 21A806      LXI  H,TRAPJMP+1
OE11 4E          MOV  C,M      ;LOW BDOS ADDR
OE12 23          INX  H
OE13 46          MOV  B,M      ;HIGH BDOS ADDR
OE14 CD5708      CALL BCDE     ;CY IF BDOS>PLOC

```

COPYRIGHT © 1980
 DIGITAL RESEARCH, INC.
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SERIAL # _____

```

OE17 DA280E      JC      BREAK0 ;BRANCH IF PLOC <= BDOS
;
;      IN THE BDOS, DON'T BREAK UNTIL THE RETURN OCCURS
OE1A CDC80D      CALL     CLRTRACE;CLEAR TRACE FLAGS
OE1D 2A4A0F      LHL D    RETLOC ;TRAPPED RETLOC ON ENTRY TO DOS
OE20 EB          XCHG     ;TO D,E READY FOR BREAKPOINT
OE21 3E82        MVI      A,82H ;LOOKS LIKE G,BBBD
OE23 B7          ORA      A ;SETS FLAGS
OE24 37          STC      ;SUBSEQUENT TEST FOR CY
OE25 C38508      JMP      GOPR ;START PROGRAM EXECUTION, WITH BREAKPOINT
;
BREAK0: ;NORMAL BREAKPOINT
OE28 FB          EI
OE29 2A4D0F      LHL D    TRACER
OE2C 7C          MOV      A,H
OE2D B5          ORA      L
OE2E CA4E0E      JZ       STOPEX
;
;      TRACE IS ON
OE31 2B          DCX      H
OE32 224D0F      SHLD     TRACER
OE35 CD1F0C      CALL     BREAK ;BREAK KEY DEPRESSED?
OE38 C24E0E      JNZ      STOPEX
OE3B 3A4C0F      LDA      TMODE ;TRACE MODE T IF OFFH
OE3E B7          ORA      A
OE3F C2480E      JNZ      BREAK1
;      NOT TRACING, BUT MONITORING, SO SET BREAKPOINTS
OE42 CD850E      CALL     NBRK
OE45 C38508      JMP      GOPR
;
BREAK1: ;TRACING AND MONITORING
OE48 CD440D      CALL     DSTATE ;STATE DISPLAYED, CHECK FOR BREAKPOINTS
OE4B C38508      JMP      GOPR ;STARTS EXECUTION
;
STOPEX:
OE4E CDC80D      CALL     CLRTRACE ;TRACE FLAGS GO TO ZERO
OE51 3E2A        MVI      A,'*'
OE53 CDC70B      CALL     PCHAR
OE56 2AB90F      LHL D    PLOC
;      CHECK TO ENSURE DISASSEMBLER IS PRESENT
OE59 CD9309      CALL     CHKDIS
OE5C D2620E      JNC      STOPO
OE5F 220C00      SHLD     DISPC
OE62 CD2E0C      STOPO:  CALL     PADDR
OE65 2AB70F      LHL D    HLOC
OE68 225D0F      SHLD     DISLOC
OE6B C3FE06      JMP      START
;
CAT: ;DETERMINE OPCODE CATEGORY - CODE IN REGISTER B
;      D,E CONTAIN DOUBLE PRECISION CATEGORY NUMBER ON RETURN
OE6E 110D00      LXI      D,OPMAX ;D=0,E=OPMAX
OE71 212F0F      LXI      H,OPLIST
OE74 7E          CATO:  MOV      A,M ;MASK TO A
OE75 A0          ANA      B ;MASK OPCODE FROM B
OE76 23          INX      H ;READY FOR COMPARE
OE77 BE          CMP      M ;SAME AFTER MASK?

```

COPYRIGHT © 1980
 DIGITAL RESEARCH, INC.
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SERIAL # _____

```

0E78 23      INX      H      ;READY FOR NEXT COMPARE
0E79 CA810E   JZ       CAT1   ;EXIT IF COMPARED OK
0E7C 14      INR      D      ;UP COUNT IF NOT MATCHED
0E7D 1D      DCR      E      ;FINISHED?
0E7E C2740E   JNZ      CAT0
0E81 5A      CAT1:  MOV    E,D    ;E IS CATEGORY NUMBER
0E82 1600     MVI      D,0    ;DOUBLE PRECISION
0E84 C9      RET

```

```

;
NBRK: ;FIND NEXT BREAK POINT ADDRESS
;
; UPON RETURN, REGISTER A IS SETUP AS IF USER TYPED G,B1,B2 OR
; G,B1 DEPENDING UPON OPERATOR CATEGORY. B,C CONTAINS SECOND BP,
; D,E CONTAINS PRIMARY BP. HL ADDRESS NEXT OPCODE BYTE

```

```

0E85 2AB90F   LHLD     PLOC
0E88 46      MOV      B,M      ;GET OPERATOR
0E89 23      INX      H      ;HL ADDRESS BYTE FOLLOWING OPCODE
0E8A E5      PUSH     H      ;SAVE IT FOR LATER
0E8B CD6E0E   CALL     CAT      ;DETERMINE OPERATOR CATEGORY
0E8E 21490F   LXI      H,CATNO ;SAVE CATEGORY NUMBER
0E91 73      MOV      M,E
0E92 219C0E   LXI      H,CATTAB;CATEGORY TABLE BASE
0E95 19      DAD      D      ;INXED
0E96 19      DAD      D      ;INXED*2
0E97 5E      MOV      E,M      ;LOW BYTE TO E
0E98 23      INX      H
0E99 56      MOV      D,M      ;HIGH BYTE TO D
0E9A EB      XCHG
0E9B E9      PCHL
0E9C B80E     CATTAB: DW      JMPOP ;JUMP INTO TABLE
0E9E E00E     DW      CCOP  ;JUMP OPERATOR
0EAO B80E     DW      JMPOP ;JUMP CONDITIONAL
0EA2 E00E     DW      CCOP  ;CALL OPERATOR (TREATED AS JMP)
0EA4 BE0E     DW      CCOP  ;CALL CONDITIONAL
0EA6 F20E     DW      RETOP ;RETURN FROM SUBROUTINE
0EA8 040F     DW      RSTOP ;RESTART
0EAA 260F     DW      PCOP  ;PCHL
0EAC 260F     DW      IMOP  ;SINGLE PRECISION IMMEDIATE (2 BYTE)
0EAE 230F     DW      IMOP  ;ADI ... CPI
0EB0 230F     DW      DIMOP ;DOUBLE PRECISION IMMEDIATE (3 BYTES)
0EB2 190F     DW      DIMOP ;LHLD ... STA
0EB4 260F     DW      RCOND ;RETURN CONDITIONAL
;
; NEXT DW MUST BE THE LAST IN THE SEQUENCE
0EB6 140F     DW      IMOP  ;IN/OUT
;
;
JMPOP: ;GET OPERAND FIELD, CHECK FOR BDOS
0EB8 CDCE0E   CALL     GETOPA ;GET OPERAND ADDRESS TO D,E AND COMPARE WITH BDOS
0EBB C2290F   JNZ      ENDOP ;TREAT AS SIMPLE OPERATOR IF NOT BDOS
;
; OTHERWISE, TREAT AS A RETURN INSTRUCTION
0EBE CDD90E   RETOP:  CALL     GETSP ;ADDRESS AT STACKTOP TO D,E
0EC1 C3290F   JMP      ENDOP ;TREAT AS SIMPLE OPERATOR
;
;
CBDOS: ;COMPARE D,E WITH BDOS ADDRESS, RETURN ZERO FLAG IF EQUAL
0EC4 3AA806   LDA      TRAPJMP+1
0EC7 BB      CMP      E
0EC8 C0      RNZ
0EC9 3AA906   LDA      TRAPJMP+2

```

COPYRIGHT © 1980
DIGITAL RESEARCH, INC.
P. O. BOX 579
PACIFIC GROVE, CA 93950
SERIAL # _____

```

OECC BA      CMP      D
OECD C9      RET

;
GETOPA: ;GET OPERAND ADDRESS AND COMPARE WITH BDOS
OECE C1      POP      B      ;GET RETURN ADDRESS
OECF E1      POP      H      ;GET OPERAND ADDRESS
OED0 5E      MOV      E,M
OED1 23      INX      H
OED2 56      MOV      D,M
OED3 23      INX      H
OED4 E5      PUSH     H      ;UPDATED PC INTO STACK
OED5 C5      PUSH     B      ;RETURN ADDRESS TO STACK
OED6 C3C40E  JMP      CBDOS ;RETURN THROUGH CBDOS WITH ZERO FLAG SET

;
GETSP: ;GET RETURN ADDRESS FROM USER'S STACK TO D,E
OED9 2AB50F  LHLD     SLOC
OEDC 5E      MOV      E,M
OEDD 23      INX      H
OEDE 56      MOV      D,M
OEDF C9      RET

;
CCOP: ;CALL CONDITIONAL OPERATOR
OEE0 CDCE0E  CALL     GETOPA ;GET OPERAND ADDRESS TO D,E / COMPARE WITH BDOS
OEE3 CAED0E  JZ       CCOP1

;
NOT THE BDOS, BREAK AT OPERAND ADDRESS AND NEXT ADDRESS
OEE6 C1      POP      B      ;NEXT ADDRESS TO B,C
OEE7 C5      PUSH     B      ;BACK TO STACK
OEE8 3E02    MVI      A,2    ;TWO BREAKPOINTS
OEEA C32B0F  JMP      RETCAT ;RETURN FROM NBRK

;
CCOP1: ;BREAK ADDRESS AT NEXT LOCATION ONLY, WAIT FOR RETURN FROM BDOS
OEED D1      POP      D
OEEE D5      PUSH     D      ;BACK TO STACK
OEEF C3290F  JMP      ENDOP ;ONE BREAKPOINT ADDRESS

;
RSTOP: ;RESTART INSTRUCTION - CHECK FOR RST 7
OEF2 78      MOV      A,B
OEF3 FEFF    CPI      RSTIN ;RESTART INSTRUCTION USED FOR SOFT INT
OEF5 C2FC0E  JNZ      RSTO

;
;
SOFT RST, NO BREAK POINT SINCE IT WILL OCCUR IMMEDIATELY
OEF8 AF      XRA      A
OEF9 C32D0F  JMP      RETCAT1 ;ZERO ACCUMULATOR
OEF3 E638    RSTO:   ANI      111000B ;GET RESTART NUMBER
OEFE 5F      MOV      E,A
OEFF 1600    MVI      D,0    ;DOUBLE PRECISION BREAKPOINT TO D,E
OF01 C3290F  JMP      ENDOP

;
PCOP: ;PCHL
OF04 2AB70F  LHLD     HLOC
OF07 EB      XCHG     ;HL VALUE TO D,E FOR BREAKPOINT
OF08 CDC40E  CALL     CBDOS ;BDOS VALUE?
OF0B C2290F  JNZ      ENDOP
;
PCHL TO BDOS, USE RETURN ADDRESS
OF0E C3BE0E  JMP      RETOP
;

```

COPYRIGHT © 1980
 DIGITAL RESEARCH, INC.
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SERIAL # _____

```

OF11 C3290F      JMP      ENDOP
;
SIMOP:  ;SIMPLE OPERATOR, USE STACKED PC
OF14 D1          POP      D
OF15 D5          PUSH     D
OF16 C3290F      JMP      ENDOP
;
RCOND:  ;RETURN CONDITIONAL
OF19 CDD90E      CALL     GETSP  ;GET RETURN ADDRESS FROM STACK
OF1C C1          POP      B      ;B,C ALTERNATE LOCATION
OF1D C5          PUSH     B      ;REPLACE IT
OF1E 3E02        MVI      A,2
OF20 C32B0F      JMP      RETCAT ;TO SET FLAGS AND RETURN
;
DIMOP:  ;DOUBLE PRECISION IMMEDIATE OPERATOR
OF23 D1          POP      D
OF24 13          INX      D      ;INCREMENTED ONCE, DROP THRU FOR ANOTHER
OF25 D5          PUSH     D      ;COPY BACK
;
IMOP:    ;SINGLE PRECISION IMMEDIATE
OF26 D1          POP      D
OF27 13          INX      D
OF28 D5          PUSH     D
;
ENDOP:  ;END OPERATOR SCAN
OF29 3E01        MVI      A,1  ;SINGLE BREAKPOINT
RETCAT:  ;RETURN FROM NBRK
OF2B 3C          INR      A      ;COUNT UP FOR G,...
OF2C 37          STC
RETCAT1:
OF2D E1          POP      H      ;RECALL NEXT ADDRESS
OF2E C9          RET

```

COPYRIGHT © 1
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 9

COPYRIGHT © 1980
DIGITAL RESEARCH, INC.
P. O. BOX 579
PACIFIC GROVE, CA 93950

SERIAL #

```

;          OPCODE CATEGORY TABLES
OF2F FFC3      OPLIST: DB      1111$1111B,      1100$0011B      ;0 JMP
OF31 C7C2      DB      1100$0111B,      1100$0010B      ;1 JCOND
OF33 FFCB      DB      1111$1111B,      1100$1101B      ;2 CALL
OF35 C7C4      DB      1100$0111B,      1100$0100B      ;3 CCOND
OF37 FFC9      DB      1111$1111B,      1100$1001B      ;4 RET
OF39 C7C7      DB      1100$0111B,      1100$0111B      ;5 RST 0..7
OF3B FFE9      DB      1111$1111B,      1110$1001B      ;6 PCHL
OF3D C706      DB      1100$0111B,      0000$0110B      ;7 MVI
OF3F C7C6      DB      1100$0111B,      1100$0110B      ;8 ADI...CPI
OF41 CF01      DB      1100$1111B,      0000$0001B      ;9 LXI
OF43 E722      DB      1110$0111B,      0010$0010B      ;10 LHLD SHLD LDA STA
OF45 C7C0      DB      1100$0111B,      1100$0000B      ;11 RCOND
OF47 F7D3      DB      1111$0111B,      1101$0011B      ;IN OUT
000D =          OPMAX      EQU      ($-OPLIST)/2
;
OF49          CATNO: DS      1          ;CATEGORY NUMBER SAVED IN NBRK
OF4A          RETLOC: DS      2          ;RETURN ADDRESS TO USER FROM BDOS
OF4C          TMODE: DS      1          ;TRACE MODE
OF4D          TRACER: DS      2          ;TRACE COUNT
OF4F          BREAKS: DS      7          ;#BREAKS/BKPT1/DAT1/BKPT2/DAT2

```

```

0F56      EXPLIST:DS      7      ;COUNT+(EXP1)(EXP2)(EXP3)
0F5D      DISLOC: DS      2      ;DISPLAY LOCATION
0F5F      DISMAX: DS      2      ;MAX VALUE FOR CURRENT DISPLAY
0F61      TDISP: DS      2      ;TEMP 16 BIT LOCATION
0F63      NEXTCOM:DS      2      ;NEXT LOCATION FROM COMMAND BUFFER
0F65 20    COMLEN: DB      CSIZE   ;MAX COMMAND LENGTH
0F66      CURLEN: DS      1      ;CURRENT COMMAND LENGTH
0F67      COMBUF: DS      CSIZE   ;COMMAND BUFFER
0F87      MLOAD: DS      2      ;MAX LOAD ADDRESS
0F89      DS              SSIZE   ;STACK AREA

      STACK:
0FB9 =    PLOC      EQU      STACK-2 ;PC IN TEMPLATE
0FB7 =    HLOC      EQU      STACK-4 ;HL
0FB5 =    SLOC      EQU      STACK-6 ;SP
0FB4 =    ALOC      EQU      STACK-7 ;A
0FB3 =    FLOC      EQU      STACK-8 ;FLAGS
0FB1 =    BLOC      EQU      STACK-10 ;BC
0FAF =    DLOC      EQU      STACK-12;D,E

      ;
0FBB 00    NOP
0FBC      END

```

;FOR RELOCATION BOUNDARY

COPYRIGHT © 1980
 DIGITAL RESEARCH, INC.
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SERIAL # _____

0FB4 ALOC	0009 ASSEM	077E ASSM	0005 AVAL	0857 BCDE
1686 BDOS	0005 BDOSE	06AA BEGIN	0BC5 BLANK	0FB1 BLOC
0E28 BREAKO	0E48 BREAK1	0686 BREAKA	0C1F BREAK	0DCF BREAKP
0F4F BREAKS	0006 BVAL	0E6E CAT	0E74 CATO	0E81 CAT1
0F49 CATNO	0E9C CATTAB	0EC4 CBDOS	0EE0 CCOP	0EED CCOP1
0BAB CERROR	0993 CHKDIS	000B CHKIO	0001 CIF	098B CKMLOAD
0DF3 CLERO	0E04 CLER1	0DC8 CLRTRACE	0002 COF	0F67 COMBUF
0F65 COMLEN	0981 COMLOAD	000D CR	0C15 CRLF	0020 CSIZE
0F66 CURLEN	0080 DBF	005B DBP	0D86 DCHEx	0000 DEBUG
0BA6 DEF	0C50 DELIM	0D1A DELT	0D2B DELTO	003D DELT1
0680 DEMON	001A DEOF	005C DFCB	0F23 DIMOP	070D DISASMOK
0819 DISCH	0823 DISCHO	0C45 DISCOM	0006 DISEN	0003 DISIN
0B74 DISKR	0F5D DISLOC	0F5F DISMAX	07D5 DISPO	07E5 DISP1
07F0 DISP2	07F3 DISP3	0805 DISP4	000C DISPC	0010 DISPG
07C6 DISPLAY	000E DISPM	0FAF DLOC	001A DMAF	0D4C DSTAO
0D66 DSTA1	0DAB DSTA2	0D44 DSTATE	0DAF DSTRET	0007 DVAL
0F29 ENDOP	0AC5 ETRACE	0AF5 EXAMO	0AFB EXAM1	0B08 EXAM2
0B59 EXAM4	0B69 EXAM5	0AE7 EXAMINE	0F56 EXPLIST	0000 FALSE
005C FCB	0020 FCR	0000 FDN	0001 FFN	0009 FFT
085C FILL	0864 FILL0	0926 FLB	094B FLB1	0CE3 FLGSHF
0DC3 FLGTAB	0021 FLN	0FB3 FLOC	0913 FLP	093A FLP1
0955 FLZ	000F FRC	000C FRL	0BB6 GETBUFF	0D01 GETDBA
0D12 GETDBL	0C6D GETEXP	0C71 GETEXPO	000A GETF	0CF6 GETFLO
0CFE GETFL1	0CF3 GETFLG	0ECE GETOPA	0ED9 GETSP	0C66 GETVAL
0BDD GNC	0BF4 GNCRET	088F GOP0	08A1 GOP1	0885 GOPR
0870 GOTO	08DA HEXARI	0C59 HEXCON	0FB7 HLOC	09E1 HREAD
0008 HVAL	0007 IDS	0F26 IMOP	0904 INFCB	0EB8 JMPOP
0737 JMPTAB	0797 LASSM	09C0 LCOM0	09D3 LCOM1	000A LF
000C LIFT	0A6F LMSG	0005 LOF	0B53 LROTATE	0F87 MLOAD
0000 MODBAS	095A MOVE	095D MOVE0	0E85 NBRK	0B94 NDI
0F63 NEXTCOM	000F OPF	0F2F OPLIST	000D OPMAX	076B OPN
0C2E PADDR	0C2D PADDX	0C05 PBYTE	0100 PCBASE	0BC7 PCHAR
0F04 PCOP	0C36 PGRAPH	0FB9 PLOC	0BF6 PNIB	0C00 PNIBH
0004 POF	0C40 PPERIOD	000C PSIZE	000A PVAL	096A QHEX
0A26 RBYTE	0F19 RCOND	0B7F RDC	0014 RDF	0B77 RDI
0A0C RDTYPE	099C READ	09AC READN	0A0F RED1	FFFF RELOC
0F2B RETCAT	0F2D RETCAT1	0F4A RETLOC	0EBE RETOP	0003 RIF
09AD RINIT	0DBE RINX	0A4E RLIO	0A5A RLI1	0A46 RLIFT
0BA7 RRET	0EFC RSTO	00FF RSTIN	0038 RSTLOC	0007 RSTNUM
0EF2 RSTOP	0DB3 RVECT	0841 SCAN3	0CAE SCANE0	0CB1 SCANE1
0C93 SCANEX	0C90 SCANEXP	0CD5 SCANRET	0C85 SCSTORE	08B2 SETBK
08CD SETBKO	0A84 SETMO	0ABB SETM1	0A7A SETMEM	0F14 SIMOP
0FB5 SLOC	07BD SPAGO	07BB SPAGE	0100 SPBASE	0032 SSIZE
0FBB STACK	06FE START	0E62 STOP0	0E4E STOPEX	0009 SVAL
0F61 TDISP	0930 TFT	0F4C TMODE	0ADE TRACO	0AC3 TRACE
0F4D TRACER	0BD4 TRANS	06A2 TRAPAD	06A7 TRAPJMP	FFFF TRUE
0ABF UNTRACE				

COPYRIGHT © 1980
 DIGITAL RESEARCH, INC.
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SERIAL # _____

