

```

*****
* COMPUTE MONITOR, VERSION III.6
* REQUIRES SBC-880 CPU BOARD
* AND UFDC-1 FLOPPY DISK CONTROLLER
* PROVIDES FOR USE OF CT256K-1
* DYNAMIC MEMORY BOARD
* PROVIDES MULTIPLE FORMAT CAPABILITY
* IBM COMPATIBLE FORMATS
* 8" OR 5 1/4" DISK
* SINGLE OR DOUBLE DENSITY
* SINGLE OR DOUBLE SIDE
* 128, 256, 512, OR 1024 BYTE SECTORS
* 1K, 2K, OR 4K ALLOCATION BLOCK SIZE
* VARIABLE SECTOR SKEW
* REQUIRES 2.2 CPM
* CONFIGURED FOR 60K OF RAM
* 2.2 CPM BIOS CONTAINED IN ROM AND
* DISK PARAMETERS ARE OBTAINED FROM
* TABLES ON DISK
* PRINT DRIVER FOR CENTRONIX PARALLEL
* INTERFACE AND CONSOLE DRIVER FOR
* 9600 BAUD TERMINAL ARE IN BIOS
* SBC880 PROCESSOR BOARD SWITCH SETTINGS
* SWITCH 1 POS 1,3,6 OFF
* SWITCH 2 POS 2,4,5 ON
* SWITCH 3 POS 1,2,3,4,6 OFF
* SWITCH 4 POS 1,2,3,4,5 OFF
* SWITCH 5 POS 6 ON
* WRITTEN BY R. D. CATILLER
* COPYRIGHT 1982 (C) COMPUTIME
*****
.PHEX
.XLINK
.PABS
F800 = OF800H ;MONITOR BASE ADDRESS.
F7FF = BASE-1 ;MONITOR STACK
CONSTANTS FOR MONITOR
CR = 0DH ;ASCII CARRIAGE RET
LF = 0AH ;ASCII LINE FEED
I/O PORTS ON CPU BOARD
T0 = 02BH
T1 = T0+1
T2 = T0+2
TCTL = T0+3
INOUT = T0+4
CONDDTA = T0+6
*****
002F *****
002F COMPUTE MONITOR, VERSION III.6
008B AND UFDC-1 FLOPPY DISK CONTROLLER
0089 PROVIDES FOR USE OF CT256K-1
008A DYNAMIC MEMORY BOARD
008B PROVIDES MULTIPLE FORMAT CAPABILITY
008B IBM COMPATIBLE FORMATS
008B 8" OR 5 1/4" DISK
008B SINGLE OR DOUBLE DENSITY
008B SINGLE OR DOUBLE SIDE
0016 128, 256, 512, OR 1024 BYTE SECTORS
003C 1K, 2K, OR 4K ALLOCATION BLOCK SIZE
A000 VARIABLE SECTOR SKEW
D400 REQUIRES 2.2 CPM
DC06 CONFIGURED FOR 60K OF RAM
EA00 2.2 CPM BIOS CONTAINED IN ROM AND
0001 DISK PARAMETERS ARE OBTAINED FROM
0002 TABLES ON DISK
EA03 PRINT DRIVER FOR CENTRONIX PARALLEL
009C INTERFACE AND CONSOLE DRIVER FOR
009C 9600 BAUD TERMINAL ARE IN BIOS
009D SBC880 PROCESSOR BOARD SWITCH SETTINGS
009E SWITCH 1 POS 1,3,6 OFF
009E SWITCH 2 POS 2,4,5 ON
009E SWITCH 3 POS 1,2,3,4,6 OFF
009E SWITCH 4 POS 1,2,3,4,5 OFF
009E SWITCH 5 POS 6 ON
009B WRITTEN BY R. D. CATILLER
009B COPYRIGHT 1982 (C) COMPUTIME
*****
MEMORY USED BY MONITOR
F800 OF800H ;MONITOR BASE ADDRESS.
F800 BASE-1 ;MONITOR STACK
CONSTANTS FOR MONITOR
CR = 0DH ;ASCII CARRIAGE RET
LF = 0AH ;ASCII LINE FEED
I/O PORTS ON CPU BOARD
T0 = 02BH
T1 = T0+1
T2 = T0+2
TCTL = T0+3
INOUT = T0+4
CONDDTA = T0+6
*****
002F *****
002F COMPUTE MONITOR, VERSION III.6
008B AND UFDC-1 FLOPPY DISK CONTROLLER
0089 PROVIDES FOR USE OF CT256K-1
008A DYNAMIC MEMORY BOARD
008B PROVIDES MULTIPLE FORMAT CAPABILITY
008B IBM COMPATIBLE FORMATS
008B 8" OR 5 1/4" DISK
008B SINGLE OR DOUBLE DENSITY
008B SINGLE OR DOUBLE SIDE
0016 128, 256, 512, OR 1024 BYTE SECTORS
003C 1K, 2K, OR 4K ALLOCATION BLOCK SIZE
A000 VARIABLE SECTOR SKEW
D400 REQUIRES 2.2 CPM
DC06 CONFIGURED FOR 60K OF RAM
EA00 2.2 CPM BIOS CONTAINED IN ROM AND
0001 DISK PARAMETERS ARE OBTAINED FROM
0002 TABLES ON DISK
EA03 PRINT DRIVER FOR CENTRONIX PARALLEL
009C INTERFACE AND CONSOLE DRIVER FOR
009C 9600 BAUD TERMINAL ARE IN BIOS
009D SBC880 PROCESSOR BOARD SWITCH SETTINGS
009E SWITCH 1 POS 1,3,6 OFF
009E SWITCH 2 POS 2,4,5 ON
009E SWITCH 3 POS 1,2,3,4,6 OFF
009E SWITCH 4 POS 1,2,3,4,5 OFF
009E SWITCH 5 POS 6 ON
009B WRITTEN BY R. D. CATILLER
009B COPYRIGHT 1982 (C) COMPUTIME
*****
MEMORY USED BY MONITOR
F800 OF800H ;MONITOR BASE ADDRESS.
F800 BASE-1 ;MONITOR STACK
CONSTANTS FOR MONITOR
CR = 0DH ;ASCII CARRIAGE RET
LF = 0AH ;ASCII LINE FEED
I/O PORTS ON CPU BOARD
T0 = 02BH
T1 = T0+1
T2 = T0+2
TCTL = T0+3
INOUT = T0+4
CONDDTA = T0+6
*****
CONCTL = TO+7
CONSTS = CONCTL
I/O PORTS ON CT256K MEMORY BOARD
MAPO = 8BH ;MEMORY MAP REGISTERS
MAP1 = 89H
MAP2 = 8AH
MAP3 = 8BH
MEMSTS = 8BH ;MEMORY STATUS PORT
DISK DEFINES
VERS = 22 ;CP/M VERSION NUMBER
MSIZE = 60 ;CP/M MEMORY SIZE
BIAS = (MSIZE-20)*1024
CCP = 3400H+BIAS ;BASE OF CCP
BDO5 = CCP+806H ;BASE OF BDO5
BDO6 = CCP+1600H ;BASE OF BDO6
WRALL = 0 ;WRT TO ALOC BLK
WRDIR = 1 ;WRT TO DIRECTORY
WRUAL = 2 ;WRT TO UNALOC BLK
WBOOT = BIOS+3 ;WARM BOOT ENTRY
DISK I/O PORTS
DSTAT = 9CH ;DISK STATUS PORT
DCMID = DSTAT ;DISK COMMAND PORT
DTRCK = DSTAT+1 ;DISK TRACK PORT
DSCTR = DSTAT+2 ;DISK SECTOR PORT
DDATA = DSTAT+3 ;DISK DATA PORT
DFLAG = 9BH ;STATUS REGISTER 1
DCNTL = DFLAG ;CONTROL REGISTER 1
MEMORY USED BY CPM
CURDSK = 4 ;CURRENT DISK DRIVE
TBUFF = 80H ;DEFAULT CPM BUFFER
PROGRAM CODE BEGINS:
.LOC
BASE
;LET US BEGIN
JMP BEGIN ;RESET JUMP LATCH
MONITOR SIGN-ON MESSAGE
.MSG1 .BYTE CR,LF
.MSG2 .ASCII 'COMPUTIME III.6'
.MSG3 .ASCII 'COMPUTIME III.6'
MEMORY PARITY ERROR MESSAGE

```

```

F8D7 C690 ADI 90H
F8D9 DAA
F8DA ACE40
F8DC DAA
F8DD MOV C,A
F8DE RET

; MAIN KEYBOARD ROUTINE
;
; MAIN: CALL CI
; MOV C,A
; JMPR CO

; MAIN PARAMETER GETTING ROUTINE
;
; CLEAR HL
; GET INPUT
; SAVE IT
; TEST FOR SPACE
; RETURN IF SPACE
; TEST FOR COMMA
; RETURN IF COMMA
; TEST FOR CR
; RETURN IF CR
; TEST < 0
; INPUT ERROR
; TEST IF > F
; INPUT ERROR
; TEST FOR NUMBER
; GO SAVE NUMBER
; ADJUST LETTER
; TEST FOR . THRU 9
; INPUT ERROR
; SHIFT HL 1 DIGIT

; GETS START & END ADDRESS AND DETERMINES LENGTH
;
; RANGE: CALL GPARAM
; CPI CR
; JZ ERROR
; MOV D,H
; MOV E,L
; CALL GPARAM
; PUSH H
; ORA A
; DSBC
; MOV B,H
; MOV C,L
; MOV H,D
; MOV L,E
; POP D
; RET

; RANGE2: CALL RANGE
; CPI CR

F922 CD F99A
F925 4F
F926 18A3

F928 21 0000
F92B CD F922
F92E 47
F92F FE20
F931 CB
F932 FE2C
F934 CB
F935 FE0D
F937 CB
F938 D630
F93A DA FBAE
F93D FE17
F93F D2 FBAE
F942 FE0A
F944 3807
F946 D607
F948 FE0A
F94A DA FBAE
F94E 29
F94E 29
F94F 29
F950 29
F951 B5
F952 6F
F953 18D6

F955 CD F928
F958 FE0D
F95A CA FBAE
F95D 54
F95E 5D
F95F CD F928
F962 E5
F963 B7
F964 ED52
F966 44
F967 4D
F968 62
F969 6B
F96A D1
F96B C9

F96C CD F955
F96F FE0D

; CONSOLE CARRIAGE RETURN &
; LINE FEED ROUTINE.
;
; CRLF: PUSH H
; MVI B,2
; CALL MESSG
; POP H
; RET

; CONSOLE STATUS TEST ROUTINE.
;
; CSTS: IN MEMSTS
; ANI OCOH
; CPI 80H
; JRNZ CSTS1
; LDA PARFLG
; CPI 0
; JRNZ CSTS1
; INR A
; STA PARFLG
; PUSH H
; PUSH B
; LXI H,MSG1
; MVI B,MSG1
; CALL MESSG
; POP B
; POP H
; IN CONCTL
; ANI 02H
; MVI A,OFFH
; RET

; PRINT H&L ON CONSOLE
;
; DISPHL: MOV A,H
; CALL DISP8
; MOV A,L
; DISP8: PUSH PSW
; RRC
; RRC
; RRC
; CALL HTA2
; POP PSW
; CALL HTA
; JMPR CO

; RANGE: CALL RANGE
; CPI CR
; RANGE2: CALL RANGE
; CPI CR

F99A CD F99A
F99B 4F
F99C 18A3

F99D 21 0000
F99E CD F922
F99F 47
F9A0 FE20
F9A1 CB
F9A2 FE2C
F9A3 CB
F9A4 FE0D
F9A5 CB
F9A6 D630
F9A7 DA FBAE
F9A8 FE17
F9A9 D2 FBAE
F9AA FE0A
F9AB 3807
F9AC D607
F9AD FE0A
F9AE DA FBAE
F9AF 29
F9B0 29
F9B1 29
F9B2 29
F9B3 B5
F9B4 6F
F9B5 18D6

F9B6 CD F928
F9B7 FE0D
F9B8 CA FBAE
F9B9 54
F9BA 5D
F9BB CD F928
F9BC E5
F9BD B7
F9BE ED52
F9BF 44
F9C0 4D
F9C1 62
F9C2 6B
F9C3 D1
F9C4 C9

F9C5 CD F955
F9C6 FE0D

```

Address	Code	Label	Comments	Input	Output	Errors	Registers	Stack	Control
F971	C2 F8AE	JNZ	ERROR	INPUT ERROR	F9C0	2F	C,LL		
F974	C9	RET			F9C1	C9			
F975	CD F955	RANGE3:	RANGE	GET 2 PARAMETERS	F9C2	CD F96C	DISP1:		
F978	FE0D	CPI	CR	TEST FOR CR	F9C5	7D	DISP2:		
F97A	CA F8AE	JZ	ERI DR	INPUT ERROR	F9C6	E6F0	DISP3:		
F97D	22 F3DC	SHLD	HL5TR	SAVE START	F9C8	6F			
F980	C5	PUSH	B	SAVE BC	F9C9	7B			
F981	CD F928	CALL	GPARAM	GET 3RD PARAMETER	F9CA	E60F			
F984	C1	POP	B	RESTRE BC	F9CC	2803			
F985	FE0D	CPI	CR	TEST FOR CR	F9CE	13			
F987	C2 F8AE	JNZ	ERROR	INPUT ERROR	F9CF	18F8			
F98A	C9	RET			F9D1	CD F8C3			
F98B	E5	PUSH	H	SAVE HL	F9D4	CD F8C9			
F98C	B7	ORA	A	HL - DE	F9D7	7E			
F98D	ED52	DSBC	D	RESTORE HL	F9D8	CD F914			
F98F	E1	POP	H	RETURN FLAGS	F9DB	7D			
F990	C9	RET			F9DC	E60F			
F991	CD F975	CALL	RANGE3	GET 3 PARAMETERS	F9DE	FE0F			
F994	54	MOV	D,H	DEST TO DE	F9E0	2803			
F995	5D	MOV	E,L	SOURCE TO HL	F9E2	23			
F996	2A F3DC	LHLD	HL5TR	BC = LENGTH	F9E5	18EF			
F999	C9	RET			F9E8	CD F8C9			
F99A	CD F8E7	CALL	CSTS	TEST FOR INPUT	F9EB	7D			
F99D	B7	ORA	A		F9EE	E6F0			
F99E	28FA	JRZ	CI		F9EF	7E			
F9A0	DB2E	IN	CONDTA		F9F0	E67F			
F9A2	E67F	ANI	7FH		F9F2	FE20			
F9A4	C9	RET			F9F4	3002			
F9A5	DB2C	IN	INOUT		F9F6	3E2E			
F9A7	E601	ANI	1		F9F8	FE7C			
F9A9	20FA	JRNZ	PRINT		F9FA	30FA			
F9AB	79	MOV	A,C		F9FC	4F			
F9AC	F680	ORI	80H		F9FD	CD F8CB			
F9AE	D32C	OUT	INOUT		FA00	23			
F9B0	E67F	ANI	7FH		FA01	7D			
F9B2	D32C	OUT	INOUT		FA02	E60F			
F9B4	F680	ORI	80H		FA04	20E9			
F9B6	D32C	OUT	INOUT		FA06	CD F98B			
F9B8	C9	RET			FA09	20C6			
F9B9	DB2C	IN	INOUT		FA0B	C3 F85A			
F9BB	E601	ANI	1		FA0E	CD F975			
F9BD	3EFF	MVI	A,OFFH		FA11	7D			
F9BF	C8	RZ			FA12	2A F3DC			
F9C0	2F				FA15	77			
F9C1	C9				FA16	54			
F9C2	CD F96C				FA17	5D			
F9C5	7D				FA18	13			
F9C6	E6F0								
F9C8	6F								
F9C9	7B								
F9CA	E60F								
F9CC	2803								
F9CE	13								
F9CF	18F8								
F9D1	CD F8C3								
F9D4	CD F8C9								
F9D7	7E								
F9D8	CD F914								
F9DB	7D								
F9DC	E60F								
F9DE	FE0F								
F9E0	2803								
F9E2	23								
F9E5	18EF								
F9E8	CD F8C9								
F9EB	7D								
F9EE	E6F0								
F9EF	7E								
F9F0	E67F								
F9F2	FE20								
F9F4	3002								
F9F6	3E2E								
F9F8	FE7C								
F9FA	30FA								
F9FC	4F								
F9FD	CD F8CB								
FA00	23								
FA01	7D								
FA02	E60F								
FA04	20E9								
FA06	CD F98B								
FA09	20C6								
FA0B	C3 F85A								
FA0E	CD F975								
FA11	7D								
FA12	2A F3DC								
FA15	77								
FA16	54								
FA17	5D								
FA18	13								

```

FA19          QB          ;ADJUST LENGTH
FA1A          EDB0        ;WRITE DATA
FA1C          C3 F85A    ;GOTO USER PROGRAM WITH OPTIONAL BREAKPOINT

;GOTO USER PROGRAM WITH OPTIONAL BREAKPOINT
;
;
;GOTO: CALL PARAM1
;PCHL: PCHL
;
;EXTENDED MEMORY TEST
;
;TEST: CALL RANGE2
;MOV B,H
;MVI C,L
;MVI A,O
;
;LOOP: STAI
;FILLIT: LDAI
;XRA XRA
;MOV MOV
;INX H
;MOV A,H
;CMP D
;JRNZ FILLIT
;MOV A,L
;CMP E
;JRNZ FILLIT
;MOV H,B
;MOV L,C
;
;TEST1: LDAI
;XRA XRA
;XRA XRA
;CMP CMP
;CNZ MERR
;INX H
;MOV A,H
;CMP D
;JRNZ TEST1
;MOV A,L
;CMP E
;JRNZ TEST1
;MOV H,B
;MOV L,C
;IN CONCTL
;ANI O2H
;JNZ START
;LDAI A
;INR JMPR
;MERR: PUSH B
;PUSH PSW
;CALL CRLFHL
;POP PSW
;CALL DISPB

FA23          CD F96C
FA26          44
FA27          4D
FA28          3E00
FA2A          ED47
FA2C          ED57
FA2E          AD
FA2F          AC
FA30          77
FA31          23
FA32          7C
FA33          BA
FA34          20F6
FA36          7D
FA37          8B
FA38          20F2
FA3A          60
FA3B          69
FA3C          ED57
FA3E          AD
FA3F          AC
FA40          BE
FA41          C4 F85B
FA44          23
FA45          7C
FA46          BA
FA47          20F3
FA49          7D
FA4A          8B
FA4B          20EF
FA4D          60
FA4E          69
FA4F          DB2F
FA51          E602
FA53          C2 F85A
FA56          ED57
FA58          3C
FA59          18CF
FA5B          C5
FA5C          F5
FA5D          CD F8C3
FA60          F1
FA61          CD F914

DCX B
LDIR
JMP START

CALL PARAM1
PCHL

;EXTENDED MEMORY TEST
;TEST: CALL RANGE2
;MOV B,H
;MVI C,L
;MVI A,O
;
;LOOP: STAI
;FILLIT: LDAI
;XRA XRA
;MOV MOV
;INX H
;MOV A,H
;CMP D
;JRNZ FILLIT
;MOV A,L
;CMP E
;JRNZ FILLIT
;MOV H,B
;MOV L,C
;
;TEST1: LDAI
;XRA XRA
;XRA XRA
;CMP CMP
;CNZ MERR
;INX H
;MOV A,H
;CMP D
;JRNZ TEST1
;MOV A,L
;CMP E
;JRNZ TEST1
;MOV H,B
;MOV L,C
;IN CONCTL
;ANI O2H
;JNZ START
;LDAI A
;INR JMPR
;MERR: PUSH B
;PUSH PSW
;CALL CRLFHL
;POP PSW
;CALL DISPB

CD F980
E9

CD F980
E9

CD F96C
44
4D
3E00
ED47
ED57
AD
AC
77
23
7C
BA
20F6
7D
8B
20F2
60
69
ED57
AD
AC
BE
C4 F85B
23
7C
BA
20F3
7D
8B
20EF
60
69
DB2F
E602
C2 F85A
ED57
3C
18CF
C5
F5
CD F8C3
F1
CD F914

FA64          CD F8C7
FA67          7E
FA68          CD F914
FA6B          C1
FA6C          C9

FA6D          CD F991
FA70          EDB0
FA72          C3 F85A

FA75          CD F980
FA78          4D
FA79          ED78
FA7B          F5
FA7C          CD F8DF
FA7F          F1
FA80          CD F914
FA83          C3 F85A

FA86          CD F96C
FA89          4D
FA8A          ED59
FA8C          C3 F85A

FA8F          21 F033
FA92          11 EA00
FA95          01 0033
FA98          C9

FA99          21 F8B7
FA9C          11 0000
FA9F          01 000B
FAA2          EDB0
FAA4          21 FB64
FAA7          CD FA92
FAA9          EDB0
FAAC          31 F7FF
FAAF          21 EA33
FAB2          11 EA34
FAB5          01 09F1
FAB8          3600
FABA          EDB0
FABC          CD FABF
FABF          EB
FAC0          EDB0

;ADJUST LENGTH
;WRITE DATA
;GOTO USER PROGRAM WITH OPTIONAL BREAKPOINT
;GOTO USER PROGRAM
;GET 2 PARAMETERS
;SAVE START IN BC
;CLEAR I
;BUILD DATA
;WRITE DATA
;NEXT DATA
;TEST FOR END
;CONTINUE WRITING
;TEST FOR END
;CONTINUE WRITING
;RESTORE START
;BUILD DATA
;COMPARE IT
;DISPLAY ERRORS
;TEST FOR END
;CONTINUE TEST
;TEST FOR END
;CONTINUE TEST
;RESTORE START
;TEST KEYBOARD
;ABORT IF KEY PRESSED
;INCREMENT TALLY
;ANOTHER PASS
;SAVE BC
;SAVE DATA
;DISPLAY ADDRESS
;DISPLAY DATA

;GOTO USER PROGRAM WITH OPTIONAL BREAKPOINT
;
;
;GOTO: CALL PARAM1
;PCHL: PCHL
;
;EXTENDED MEMORY TEST
;
;TEST: CALL RANGE2
;MOV B,H
;MVI C,L
;MVI A,O
;
;LOOP: STAI
;FILLIT: LDAI
;XRA XRA
;MOV MOV
;INX H
;MOV A,H
;CMP D
;JRNZ FILLIT
;MOV A,L
;CMP E
;JRNZ FILLIT
;MOV H,B
;MOV L,C
;
;TEST1: LDAI
;XRA XRA
;XRA XRA
;CMP CMP
;CNZ MERR
;INX H
;MOV A,H
;CMP D
;JRNZ TEST1
;MOV A,L
;CMP E
;JRNZ TEST1
;MOV H,B
;MOV L,C
;IN CONCTL
;ANI O2H
;JNZ START
;LDAI A
;INR JMPR
;MERR: PUSH B
;PUSH PSW
;CALL CRLFHL
;POP PSW
;CALL DISPB

CD F980
E9

CD F96C
44
4D
3E00
ED47
ED57
AD
AC
77
23
7C
BA
20F6
7D
8B
20F2
60
69
ED57
AD
AC
BE
C4 F85B
23
7C
BA
20F3
7D
8B
20EF
60
69
DB2F
E602
C2 F85A
ED57
3C
18CF
C5
F5
CD F8C3
F1
CD F914

CALL SPACE
MOV A,H
CALL DISP8
POP B
RET

;MOVE BLOCK OF MEMORY
;
;MOVE: CALL SDL
;LDIR
;JMP START

;INPUT DATA FROM AN I/O PORT
;
;INPUT: CAL PFRAM1
;MOV C,L
;INP A
;PUSH PSW
;CALL CRLF
;POP PSW
;CALL DISP8
;JMP START

;OUTPUT DATA TO AN I/O PORT
;
;OUTPUT: CALL RANGE2
;MOV C,L
;OUTP E
;JMP START

;GET READY TO MOVE BIOS VECTORS
;
;BIOSMV: LXI H,DIRBF
;BIOSM1: LXI D,BIOS
;BIOSM2: LXI B,BCDL
;RET

;BOOT CPM AND RUN
;
;BOOT: LXI H,TPLT
;LXI D,0
;LXI B,B
;LDIR
;LXI H,BIOSCD
;CALL BIOSM1
;LDIR
;LXI SP,STACK
;LXI H,BEGDAT
;LXI D,BEGDAT+1
;LXI B,DATSIZ
;MVI M,0
;LDIR BIOSMV
;CALL XCHG
;LDIR BIOSM1

;RESTORE STACK
;CLEAR SCRATCH RAM
;SAVE BIOS VECTORS

;DISPLAY A SPACE
;GET MEM DATA
;DISPLAY IT
;RESTORE BC
;CONTINUE TESTING

;SRC, DEST, LENGH
;DO MOVE

;GET PARAMETER
;PUT IO ADR IN C
;INPUT DATA TO A
;SAVE IT
;DISPLAY CRLF
;GET DATA
;DISPLAY IT

;GET 2 PARAMETERS
;OUTPUT DATA

;GET READY TO MOVE BIOS VECTORS
;BIOSMV: LXI H,DIRBF
;BIOSM1: LXI D,BIOS
;BIOSM2: LXI B,BCDL
;RET

;BOOT CPM AND RUN
;BOOT: LXI H,TPLT
;LXI D,0
;LXI B,B
;LDIR
;LXI H,BIOSCD
;CALL BIOSM1
;LDIR
;LXI SP,STACK
;LXI H,BEGDAT
;LXI D,BEGDAT+1
;LXI B,DATSIZ
;MVI M,0
;LDIR BIOSMV
;CALL XCHG
;LDIR BIOSM1

;RESTORE STACK
;CLEAR SCRATCH RAM
;SAVE BIOS VECTORS

```

```

FAC2 21 FFBF LXI H,EPBASE
FAC5 11 F3DE LXI D,DPBASE
FAC8 01 0040 LXI B,EPLGTH
FACB EDB0 LDJR
FACD LXI 21 D400
FAD0 22 F3BD SHLD DMAAD
FAD3 3E02 MVI A,2
FAD5 32 F3B9 STA SECTOR
FADB AF F3D2 STA LUNIT
FADC CD FCD9 CALL LOGDS2
FADF C2 F85A JNZ CTBLP
FAE2 DD2A F3DB D02A F3DB
FAE6 3E31 MVI A,49
FAE8 32 F3D5 STA BTSEC
FAEB 1619 MVI D,25
FAED DB98 IN DFLAG
FAEF E604 ANI 4
FAF1 2802 JRZ WBOOT8
FAF3 1611 MVI D,17
FAF5 3A F3D5 STA BTSEC
FAF8 92 F3D5 STA D
FAF9 32 F3D5 STA BTSEC
FAFC 060A MVI B,10
FAFE C5 PUSH TST128
FAFF CD FF8B CALL DSDTR
FB02 D39E OUT SETUP
FB04 CD FF96 CALL DMAAD
FB07 D5 LHL D
FB08 2A F3BD LDA SECTOR
FB0E E680 ANI BOH
FB10 3E98 MVI A,98H
FB12 2802 JRZ BTRW3
FB14 3E9A MVI A,9AH
FB16 32 F3BB STA CMND
FB19 D39C OUT DCMD
FB1B 01 809F LXI B,(128*256)+DDATA
FB1E EDB2 INIR
FB20 15 DCR
FB21 20FB JRNZ
FB23 0632 MVI B,50
FB25 10FE BWTDLY
FB27 3E00 MVI A,ODOH
FB29 D39C OUT DCMD
FB2B 060A MVI B,10
FB2D 10FE MVI FRC1
FB2F DB9C DSTAT
FB31 D1 POP
FB32 CD FF31 CALL EOJ
FB35 E69C ANI 9CH
FB37 C1 POP
FB38 2808 JRZ WBOOT8
FB3A 10C2 JRZ BRWAGN
FB3C CD FC78 DERR: CALL

```

```

;SET UP CPM TABLES
FB3F FB8A C3 FB5A
FB42 22 F3BD 22 F38D
FB45 3A F3D5 3A F3D5
FB48 B7
FB49 CA FB72
FB4B DDCB124E
FB4C 3EB1
FB50 2008 3EB1
FB52 0658
FB54 CD FF23
FB56 21 F3BB
FB59 34
FB5C 3E01
FB5D 32 F3B9
FB5F 3A F3D5
FB62 DD8E00
FB65 37
FB68 DA FAF5
FB69 DD5600
FB6C C3 FAF5
FB6F C3 FAF5
FB72 CD FAFB
FB75 EDB0
FB77 21 0080
FB7A 22 F3D6
FB7D 3A 0004
FB80 4F
FB81 C3 D400

```

```

;SET DMA ADR
;SECTOR = 2
;LOG ON DRIVE A
;LOG ON ERROR
;GET TBL POINTER
;NO OF SECS
18" SECS ON TRK 0
18" OR 5 1/4" ?
18"
;SUBTRACT SECTORS
;ADJUST SECTOR ADR
;SAVE FOR RETRY
;GET DMA ADR
;TEST FOR SIDE 1
;READ COMMAND
;SIDE 0
;SELECT SIDE 1
;LOOP CONTROL
;FORCE INT COMMAND
;EXECUTE IT
;DELAY
;READ STATUS
;RESTORE LENGTH
;DONE, NO ERRORS
;RETRY
;DISPLAY ERROR

```

```

;UPDATE DMAAD
;ALL SECS DONE?
;DONE
;TEST FOR DOUBLE SIDE
;SIDE 1 SELECT
;DOUBLE SIDE
;STEP IN 1 TRACK
;TRK + 1
;SECTOR = 1
;BTSEC > SEC/TRK?
;REMAINING SECS
;DO THE REST
;SECS PER TRK
;DO NEXT TRK
;RESTORE BIOS VECTORS
;DEFAULT CPM BUFFER
;SET DMA ADDRESS
;LOG-ON DSK
;SEND TO CPM
;GO TO CP/M
;COLD BOOT
;WARM BOOT
;CONSOLE STATUS
;CONSOLE INPUT
;CONSOLE OUTPUT
;LIST DEVICE
;PUNCH DEVICE
;READER DEVICE
;MOVE HEAD TO TRK 0
;SELECT DISK
;SET TRACK NUMBER
;SET SECTOR NUMBER
;SET DMA ADDRESS
;READ DISK
;WRITE DISK
;LIST STATUS
;SECTOR TRANSLATE

```

```

START DMAAD
BTSEC LDA
A ORA
GOCPM JZ
1,18(X) BIT
A,81H MVI
WBOOT9 JRNZ
B,58H MVI
EOJA CALL
H,TRACK LXI
M INR
A,1 MVI
SECTOR STA
BTSEC LDA
O(X) CMP
D,A MOV
WBOOT8 JC
D,O(X) MOV
WBOOT8 JMP
BIOSMV CALL
LDIR
H,TBUF LXI
SEKDMA SHLD
CURDSK LDA
C,A MOV
CCP JMP
BOOT WBOOT
CSTS
CI
CO
CO
CI
HOME
SELDSK
SETTRK
SETSEC
SETDMA
HREAD
HWRITE
PST9
SECTRN
.-BIOSCD
BCDL
;JUMPS AT START OF MEMORY AND IOBYTE
;TFLT:
;IOBYT:
JMP .BYTE
JMP .BYTE
JMP O
JMP BDO5

```

```

;MONITOR BIOS JUMP VECTORS
;BIOSCD:
JMP FA99
JMP FAAC
JMP FB87
JMP FB8A
JMP FB8D
JMP FB90
JMP FB93
JMP FB96
JMP FB99
JMP FB9C
JMP FB9F
JMP FBA2
JMP FBA5
JMP FBAB
JMP FBAB
JMP FBAE
JMP FBB1
JMP FBB4
JMP 0033
JMP EA03
JMP BD
JMP 00
JMP C3 DC06

```



FC90	1A	LDA	D	GET DATA	FCED	JECO	MVI	A,OCOH	STORE SELECT CODE
FC91	13	INX	D	STEP TO NEXT	FCEF	B1	ORA	C	SEE IF NEW DISK
FC92	CD F914	DISPB	D	DISPLAY DATA	FCFO	32 F3C0	STA	HSELCD	
FC95	CD FBC9	SPACE	H	DISPLAY SPACE	FCF3	21 F3B3	LXI	H,DKSEL	
FC98	E1	CALL	H	GET HL	FCF6	0600	MVI	B,0	
FC99	C1	POP	B	GET BC	FCF8	09	DAD	B	
FC9A	OD	DCR	C	TALLY - 1	FCF9	7E	MOV	A,M	
FC9B	20E9	JRNZ	C	CONTINUE	FCFA	B7	ORA	A	NOT NEW DISK
FC9D	0603	MVI	B,3	DISPLAY NAME	FCFB	202D	JRNZ	DBPARM	
FC9F	CD F8B8	CALL	MESG1		FCFD	2F	CMA		
FCA2	CD FBC9	CALL	SPACE	DISPLAY SPACE	FCFE	77	MOV	M,A	MARK AS USED
FCA5	2A F3BD	LHLD	DMAAD	LAST PARAMETER	FCFF	3A F3C0	LDA	HSELCD	GET SEL CODE
FCA8	CD F90F	CALL	DISPHL	DISPLAY IT	FD02	D39B	OUT	DCNTL	SEL DRIVE
FCAE	E1	POP	H	RESTORE REGISTERS	FD04	CD FF21	CALL	EOJB	RESTORE DRIVE
FCAC	D1	POP	D		FD07	E680	ANI	BOH	TEST FOR NOT REA
FCAD	C1	FOP	B		FD09	C2 FC78	JNZ	DERR1	NOT RDY ERROR
FCAE	3A F3BA	LDA	STATUS		FD0C	060A	MVI	B,10	RETRY COUNT
FCB1	B7	ORA	A		FD0E	C5	PUSH	B	
FCB2	C9	RET			FD0F	3E01	MVI	A,1	SELECT SEC 1
					FD11	D39E	OUT	D,CTR	
					FD13	57	MOV	D,A	SEC SIZE=128
					FD14	2A F3D8	LHLD	CTBLP	GET SELECT CODE
					FD17	3A F3C0	LDA	HSELCD	
					FD1A	D39B	OUT	DCNTL	
					FD1C	3E8C	MVI	A,8CH	
					FD1E	CD FEFD	CALL	RDAT	
					FD21	C1	POP	B	
					FD22	B7	ORA	A	
					FD23	2805	DBPARM	JRZ	DBPARM
					FD25	10E7	DJNZ	NTAGN	
					FD27	C3 FC78	JMP	DERR1	RETRY
					FD2A	DD4E0F	MOV	L,15(X)	DISP PARAMETERS
					FD2D	DD4610	MOV	H,16(X)	GET BYTES/SEC
					FD30	29	DAD	H	
					FD31	7C	MOV	A,H	CPM SEC/PHY SEC
					FD32	32 F3C1	STA	HSTBLK	
					FD35	3D	DCR	A	
					FD36	32 F3C3	STA	SECMASK	SECTOR MASK
					FD39	DD4602	MOV	B,2(X)	GET BLK SHIFT FA
					FD3C	3E01	MVI	A,1	CPM ALOC SIZE/12
					FD3E	07	RLC		
					FD3F	10FD	DJNZ	BLKCAL	
					FD41	32 F3C2	STA	BLKCNT	
					FD44	3A F3B7	LDA	DISKNO	GET DISK NO
					FD47	21 F3D2	LXI	H,LUNIT	POINT TO LAST UN
					FD4A	BE	CMP	M	SEE IF SAME
					FD4B	77	MOV	M,A	SAVE THIS UNIT
					FD4C	CB	RZ		
					FD4D	CD FF96	CALL	SETUP	READ ADR COMMAND
					FD50	21 F41E	LXI	H,IDSV	
					FD53	1601	MVI	D,1	
					FD55	3E4	MVI	A,OC4H	
					FD57	D39C	OUT	DCMMD	
					FD59	32 F3BB	STA	CMND	
					FD5C	01 069F	LXI	B,(6*256)+DDATA	

SELECT DISK GIVEN BY REGISTER C

FCB3	21 0000	SELDSK1	LXI	H,0	ERROR RETURN CODE				
FCB6	79	MOV	A,C						
FCB7	32 F3C9	STA	SEKDSK		STORE DSK NO.				
FCB8	FE04	CPI	4		MUST BE 0,1,2,OR 3				
FCBC	D0	RNC	X		INVALID DSK NO.				
FCBD	DDE5	PUSH	X		LOG ON DISK				
FCBF	E5	PUSH	H						
FCC0	CD FCD9	CALL	LOGDS2						
FCC3	E1	POP	H						
FCC4	C0	POP	X		ERROR RETURN				
FCC5	DDE1	RNZ	SEKDSK		GET DISK NO.				
FCC7	3A F3C9	LDA	L,A						
FCCA	6F	MOV	H,B						
FCCB	60	MOV	H,B						
FCCC	29	DAD	H						
FCCD	29	DAD	H						
FCCF	29	DAD	H						
FCCF	29	DAD	H						
FCD0	EB	XCHG	H,DPBASE						
FCD1	21 F3DE	LXI	D		HL=DPBASE+(DISKNO*16)				
FCD4	19	DAD							
FCD5	C9	RET							
FCD6	3A F3B7	LOGDSK1	LDA	DISKNO					
FCD9	4F	LOGDSK2	MOV	C,A					
FCDA	3C	INR	A						
FCDB	DD21 EE33	LXI	X,DP0						
FCDF	11 0080	LXI	D,128						
FCE2	3D	DABL1	A	DABLD2					
FCE3	2804	JRZ	D	DABLD1					
FCE5	DD19	DAD	D	DABLD1					
FCE7	1BF9	JMPR	D	CTBLP					
FCE9	DD22 F3D8	DABL2	SIXD						





```

FE16          UNACNT          STA          32 F3C5          ;SET RSFLAG          L          ;BUF SEL BITS X 12
FE19          A              INR          3C          ;NEED PREREAD        H,A
FE1A          RSFLAG         STA          32 F3D0          ;ADD BUFFER INDEX    L,O
FE1D          LXI          SHLD          21 EA33          ;LOG ON SEKDSK      L
FE20          DDE5          PUSH          22 F38D          ;COMPUTE PHY SEC ADR H,DBUF
FE23          DDE5          LDA          DDE5          ;GET SIDE SELECT    D,DBUF
FE25          3A F3C9        CALL          3A F3C9        ;SAVE IT            D
FE28          CD FCD9        LDA          CD FCD9        ;MASK OFF SIDE SEL B,12B
FE2B          3A F3CB        MOV          3A F3CB        ;RSFLAG SET?       READOP
FE2E          4F          MOV          4F          ;YES DO READ       A
FE2F          E6B0          ANI          E6B0          ;NO PENDING WRT    RWMOVE
FE31          STA          STA          32 F3CD          ;MASK BUF SEL BITS A,1
FE34          79          MOV          79          ;HOST WRITTEN?    HSTWRT
FE35          ANI          ANI          7FH          ;WRITE HOST BUF    XCHG
FE37          4F          MOV          4F          ;GET SET TO FILL BUF   LDA
FE38          3A F3C1        LDA          3A F3C1        ;PHYSICAL SEC ADR  LDA
FE3B          0D          DCR          0D          ;1ST SEC = 0?      DCR
FE3C          1F          RAR          1F          ;1ST SEC = 1       RAR
FE3D          3804          JRC          3804          ;1ST SEC = 0       JRC
FE3F          C839          SRLR         C839          ;RSFLAG SET?       SRLR
FE41          18F9          JMPR         18F9          ;YES DO READ       JMPR
FE43          0C          INR          0C          ;HOST WRITTEN?    INR
FE44          LDA          LDA          3A F3CD          ;WRITE HOST BUF    LDA
FE47          01          ORA          01          ;GET SET TO FILL BUF   ORA
FE48          STA          STA          32 F3CC          ;PHYSICAL SEC ADR  STA
FE4B          LXI          H,HSTACT   LXI          21 F3CF          ;1ST SEC = 0?     LXI
FE4E          MOV          MOV          7E          ;1ST SEC = 1       MOV
FE4F          MVI          MVI          3601          ;1ST SEC = 0       MVI
FE51          07          ORA          07          ;RSFLAG SET?       ORA
FE52          2816          JRZ          2816          ;HOST WRITTEN?    JRZ
FE54          2A F3B7        LHL          2A F3B7        ;WRITE HOST BUF    LHL
FE57          3A F3CC        LDA          3A F3CC        ;GET SET TO FILL BUF   LDA
FE5A          4F          MOV          4F          ;PHYSICAL SEC ADR  MOV
FE5B          3A F3B9        LDA          3A F3B9        ;1ST SEC = 0?     LDA
FE5E          CD FECB        CALL          CD FECB        ;1ST SEC = 1       CALL
FE61          2825          JRZ          2825          ;1ST SEC = 0       JRZ
FE63          3A F3CE        LDA          3A F3CE        ;RSFLAG SET?       LDA
FE66          07          DRA          07          ;HOST WRITTEN?    DRA
FE67          C4 FED9        CNZ          C4 FED9        ;WRITE HOST BUF    CNZ
FE6A          2A F3C9        LHL          2A F3C9        ;GET SET TO FILL BUF   LHL
FE6D          22 F387        SHLD         22 F387        ;PHYSICAL SEC ADR  SHLD
FE70          3A F3CC        LDA          3A F3CC        ;1ST SEC = 0?     LDA
FE73          DDCB1256        BIT          DDCB1256        ;1ST SEC = 1       BIT
FE77          JRZ          JRZ          2801          ;1ST SEC = 0       JRZ
FE79          3D          DCR          3D          ;RSFLAG SET?       DCR
FE7A          32 F389        STA          32 F389        ;YES DO READ       STA
FE7D          3A F3D0        LDA          3A F3D0        ;NO PENDING WRT    LDA
FE80          07          DRA          07          ;MASK BUF SEL BITS DRA
FE81          C4 FED5        CNZ          C4 FED5        ;GET HOST BUF     CNZ
FE84          XRA          XRA          A              ;WRITE HOST BUF    XRA
FE85          STA          STA          32 F3CE          ;GET SET TO FILL BUF   STA
FE8B          3A F3CB        LDA          3A F3CB        ;PHYSICAL SEC ADR  LDA
FE8C          3D          DCR          3D          ;1ST SEC = 0?     DCR
FE8D          6F          MOV          6F          ;1ST SEC = 1       MOV
FE90          3A F3C3        LDA          3A F3C3        ;RSFLAG SET?       LDA
FE91          FE90          FE90          FE90          ;WRITE HOST BUF    FE90
FE92          FE92          FE92          FE92          ;GET SET TO FILL BUF   FE92
FE94          FE94          FE94          FE94          ;PHYSICAL SEC ADR  FE94
FE96          FE96          FE96          FE96          ;1ST SEC = 0?     FE96
FE98          FE98          FE98          FE98          ;1ST SEC = 1       FE98
FE9B          FE9B          FE9B          FE9B          ;RSFLAG SET?       FE9B
FE9C          FE9C          FE9C          FE9C          ;YES DO READ       FE9C
FE9E          FE9E          FE9E          FE9E          ;NO PENDING WRT    FE9E
FEA0          FE9E          FE9E          FE9E          ;MASK BUF SEL BITS FEA0
FEA3          FE9E          FE9E          FE9E          ;GET HOST BUF     FEA3
FEA6          FE9E          FE9E          FE9E          ;WRITE HOST BUF    FEA6
FEA9          FE9E          FE9E          FE9E          ;GET SET TO FILL BUF   FEA9
FEAB          FE9E          FE9E          FE9E          ;PHYSICAL SEC ADR  FEAB
FEAE          FE9E          FE9E          FE9E          ;1ST SEC = 0?     FEAE
FEAF          FE9E          FE9E          FE9E          ;1ST SEC = 1       FEAF
FEB0          FE9E          FE9E          FE9E          ;RSFLAG SET?       FEB0
FEB3          FE9E          FE9E          FE9E          ;YES DO READ       FEB3
FEB6          FE9E          FE9E          FE9E          ;NO PENDING WRT    FEB6
FEB9          FE9E          FE9E          FE9E          ;MASK BUF SEL BITS FEB9
FEBB          FE9E          FE9E          FE9E          ;GET HOST BUF     FEBB
FEBE          FE9E          FE9E          FE9E          ;WRITE HOST BUF    FEBE
FEC0          FE9E          FE9E          FE9E          ;GET SET TO FILL BUF   FEC0
FEC2          FE9E          FE9E          FE9E          ;PHYSICAL SEC ADR  FEC2
FEC3          FE9E          FE9E          FE9E          ;1ST SEC = 0?     FEC3
FEC4          FE9E          FE9E          FE9E          ;1ST SEC = 1       FEC4
FEC5          FE9E          FE9E          FE9E          ;RSFLAG SET?       FEC5
FEC6          FE9E          FE9E          FE9E          ;YES DO READ       FEC6
FEC7          FE9E          FE9E          FE9E          ;NO PENDING WRT    FEC7
FEC8          FE9E          FE9E          FE9E          ;MASK BUF SEL BITS FEC8
FEC9          FE9E          FE9E          FE9E          ;GET HOST BUF     FEC9
FECA          FE9E          FE9E          FE9E          ;WRITE HOST BUF    FECA
FECB          FE9E          FE9E          FE9E          ;GET SET TO FILL BUF   FECB
FECF          FE9E          FE9E          FE9E          ;PHYSICAL SEC ADR  FECF
FED0          FE9E          FE9E          FE9E          ;1ST SEC = 0?     FED0
FED2          FE9E          FE9E          FE9E          ;1ST SEC = 1       FED2
FED3          FE9E          FE9E          FE9E          ;RSFLAG SET?       FED3
FED4          FE9E          FE9E          FE9E          ;YES DO READ       FED4
FED5          FE9E          FE9E          FE9E          ;NO PENDING WRT    FED5
FED6          FE9E          FE9E          FE9E          ;MASK BUF SEL BITS FED6
FED7          FE9E          FE9E          FE9E          ;GET HOST BUF     FED7
FED8          FE9E          FE9E          FE9E          ;WRITE HOST BUF    FED8
FED9          FE9E          FE9E          FE9E          ;GET SET TO FILL BUF   FED9
FEDA          FE9E          FE9E          FE9E          ;PHYSICAL SEC ADR  FEDA
FEDD          FE9E          FE9E          FE9E          ;1ST SEC = 0?     FEDD
FEDE          FE9E          FE9E          FE9E          ;1ST SEC = 1       FEDE
FEE0          FE9E          FE9E          FE9E          ;RSFLAG SET?       FEE0
FEE3          FE9E          FE9E          FE9E          ;YES DO READ       FEE3
FEE6          FE9E          FE9E          FE9E          ;NO PENDING WRT    FEE6
FEE9          FE9E          FE9E          FE9E          ;MASK BUF SEL BITS FEE9
FEEA          FE9E          FE9E          FE9E          ;GET HOST BUF     FEEA
FEEB          FE9E          FE9E          FE9E          ;WRITE HOST BUF    FEEB
FEED          FE9E          FE9E          FE9E          ;GET SET TO FILL BUF   FEED
FEF0          FE9E          FE9E          FE9E          ;PHYSICAL SEC ADR  FEF0
FEF2          FE9E          FE9E          FE9E          ;1ST SEC = 0?     FEF2
FEF5          FE9E          FE9E          FE9E          ;1ST SEC = 1       FEF5
FEF6          FE9E          FE9E          FE9E          ;RSFLAG SET?       FEF6
FEF7          FE9E          FE9E          FE9E          ;YES DO READ       FEF7
FEF8          FE9E          FE9E          FE9E          ;NO PENDING WRT    FEF8
FEF9          FE9E          FE9E          FE9E          ;MASK BUF SEL BITS FEF9
FEFA          FE9E          FE9E          FE9E          ;GET HOST BUF     FEFA
FEFB          FE9E          FE9E          FE9E          ;WRITE HOST BUF    FEFB
FEFC          FE9E          FE9E          FE9E          ;GET SET TO FILL BUF   FEFC
FEFD          FE9E          FE9E          FE9E          ;PHYSICAL SEC ADR  FEFD
FEFE          FE9E          FE9E          FE9E          ;1ST SEC = 0?     FEFE
FEFF          FE9E          FE9E          FE9E          ;1ST SEC = 1       FEFF

```

```

FF99  B7  ORA  A  3E04  MVI  A,4
FFA0  7B  MOV  A,E  2801  JRZ  RDWRT1
FFA1  2813  WRDAT  A  AF  XRA  A
FFA2  D39C  OUT  DCMDM  C688  ADI  8BH
FFA3  32 F38B  OUT  CMND  4F  MOV  C,A
FFA4  01 809F  STA  B, (12*256)+DDATA  LDA  TRACK
FFA5  EDB2  INIR  D  1601  ORA  A
FFA6  15  DCR  D  2808  MVI  D,1
FFA7  20F8  JRNZ  READ1  JRZ  RDWRT2
FFA8  CD FF31  CALL  EOJ  DD60F  MOV  L,15(X)
FFA9  E69C  ANI  9CH  DD6610  MOV  H,16(X)
FFA0  C9  RET  FF7C  DAD  H
FFB0  F620  ORI  20H  FF7D  MOV  D,H
FFB1  D39C  OUT  DCMDM  2A F38D  DMAAD
FFB2  33 F38B  STA  CMND  3A F389  LDA  SECTOR
FFB3  01 809F  LXI  B, (128*256)+DDATA  E680  ANI  80H
FFB4  EDB3  OUTIR  D  79  MOV  A,C
FFB5  15  DCR  D  FF7E  RZ  2
FFB6  20F8  JRNZ  JMPR  FF7F  ORI  2
FFB7  1810  MVI  B,8  FF80  CMP  A
FFB8  0608  IN  DFLAG  FF81  RET
FFB9  D89B  ANI  3  FF82  LDA  SECTOR
FFB0  E603  ORA  B  FF83  BIT  3,18(X)
FFB1  80  STA  CMND  FF84  RZ  7FH
FFB2  32 F38B  OUT  STA  CMND  C9
FFB3  D39C  OUT  DCMDM  3A F389  LDA  SECTOR
FFB4  0680  MVI  B,128  FF85  BIT  3,18(X)
FFB5  10FE  SKDLY:  D  79  MOV  A,C
FFB6  DB9C  IN  DSTAT  FF86  RZ  2
FFB7  E601  ANI  1  FF87  ORI  2
FFB8  20FA  JRNZ  EOJ  FF88  CMP  A
FFB9  CD FFB7  CALL  DWAIT  FF89  RET
FFBA  DB9C  IN  DSTAT  FF8A  LDA  SECTOR
FFBB  32 F38A  STA  STATUS  FF8B  BIT  3,18(X)
FFBC  E6FC  ANI  OFCH  FF8C  RZ  7FH
FFBD  C9  RET  FF8D  ANI  RET
FFBE  CD FCD6  CALL  LOGDSK  FF8E  BIT  7,18(X)
FFBF  C0  RNZ  FF8F  RES  6,A
FFC0  CD FF8B  CALL  TST128  FF90  BIT  7,M
FFC1  D39E  OUT  DSCTR  FF91  SET  5,A
FFC2  DB9D  IN  DTRCK  FF92  SET  DCNTL
FFC3  4F  MOV  C,A  FF93  OUT  RET
FFC4  3A F38B  LDA  TRACK  FF94  BIT  7,18(X)
FFC5  B9  CMP  C  FF95  RES  6,A
FFC6  280A  JRZ  RDWRT  FF96  BIT  7,M
FFC7  D39F  OUT  DDATA  FF97  SET  5,A
FFC8  0618  MVI  B,18H  FF98  SET  DCNTL
FFC9  CD FF23  CALL  EOJA  FF99  OUT  RET
FFCA  E680  ANI  80H  FF9A  LDA  HSELCD
FFCB  C0  RNZ  FF9B  ANI  7FH
FFCC  CD FF96  CALL  SETUP  FF9C  OUT  DCNTL
FFCD  DB9B  IN  DFLAG  FF9D  LDA  HSELCD
FFCE  E610  ANI  10H  FF9E  ANI  7FH
FFCF  F611  RET  FF9F  OUT  DCNTL
FFD0  F612  RET  FF9F  RET
FFD1  F613  RET  FF9F  RET
FFD2  F614  RET  FF9F  RET
FFD3  F615  RET  FF9F  RET
FFD4  F616  RET  FF9F  RET
FFD5  F617  RET  FF9F  RET
FFD6  F618  RET  FF9F  RET
FFD7  F619  RET  FF9F  RET
FFD8  F61A  RET  FF9F  RET
FFD9  F61B  RET  FF9F  RET
FFDA  F61C  RET  FF9F  RET
FFDB  F61D  RET  FF9F  RET
FFDC  F61E  RET  FF9F  RET
FFDD  F61F  RET  FF9F  RET
FFDE  F620  RET  FF9F  RET
FFDF  F621  RET  FF9F  RET
FFE0  F622  RET  FF9F  RET
FFE1  F623  RET  FF9F  RET
FFE2  F624  RET  FF9F  RET
FFE3  F625  RET  FF9F  RET
FFE4  F626  RET  FF9F  RET
FFE5  F627  RET  FF9F  RET
FFE6  F628  RET  FF9F  RET
FFE7  F629  RET  FF9F  RET
FFE8  F62A  RET  FF9F  RET
FFE9  F62B  RET  FF9F  RET
FFEA  F62C  RET  FF9F  RET
FFEB  F62D  RET  FF9F  RET
FFEC  F62E  RET  FF9F  RET
FFED  F62F  RET  FF9F  RET
FFEE  F630  RET  FF9F  RET
FFEF  F631  RET  FF9F  RET
FF90  F632  RET  FF9F  RET
FF91  F633  RET  FF9F  RET
FF92  F634  RET  FF9F  RET
FF93  F635  RET  FF9F  RET
FF94  F636  RET  FF9F  RET
FF95  F637  RET  FF9F  RET
FF96  F638  RET  FF9F  RET
FF97  F639  RET  FF9F  RET
FF98  F63A  RET  FF9F  RET
FF99  F63B  RET  FF9F  RET
FF9A  F63C  RET  FF9F  RET
FF9B  F63D  RET  FF9F  RET
FF9C  F63E  RET  FF9F  RET
FF9D  F63F  RET  FF9F  RET
FF9E  F640  RET  FF9F  RET
FF9F  F641  RET  FF9F  RET

```

```

FFBF                                EPBASE:
FFBF                                !DISK PARAMETER HEADER FOR DISK 00
FFC3                                WORD TD0,0
FFC7                                WORD 0,0
FFCB                                WCTD DIRBF,DPO
FFC7                                WCTD CHK00,ALLO0
FFC7                                !DISK PARAMETER HEADER FOR DISK 01
FFC7                                WORD TD1,0
FFD3                                WORD 0,0
FFD7                                WORD DIRBF,DP1
FFDB                                WORD CHK01,ALLO1
FFDF                                !DISK PARAMETER HEADER FOR DISK 02
FFE3                                WORD 0,0
FFE7                                WORD DIRBF,DP2
FFE7                                WORD CHK02,ALLO2
FFE7                                !DISK PARAMETER HEADER FOR DISK 03
FFE7                                WORD TD3,0
FFE7                                WORD 0,0
FFF3                                WORD DIRBF,DP3
FFF7                                WORD CHK03,ALLO3
FFFB                                WORD EPBASE = -.EPBASE
0040                                EPLGTH =

EA33                                !SCRATCH RAM AREA FOR BDOS USE
EA33                                !
EA33                                !BIOS+BCDL
EA33                                BEGDAT =
EA33                                .LOC BEGDAT
EA33                                DBUF: .BLKB 1024
EA47                                DPO: .BLKB 20
EA47                                TD0: .BLKB 108
EA47                                DP1: .BLKB 20
EA47                                TD1: .BLKB 108
EA47                                DP2: .BLKB 20
EA47                                TD2: .BLKB 108
EA47                                DP3: .BLKB 20
EA47                                TD3: .BLKB 108
EA47                                DIRBF: .BLKB 128
EA47                                ALLO0: .BLKB 128
EA47                                ALLO1: .BLKB 128
EA47                                ALLO2: .BLKB 128
EA47                                ALLO3: .BLKB 128
EA47                                CHK00: .BLKB 64
EA47                                CHK01: .BLKB 64
EA47                                CHK02: .BLKB 64
EA47                                CHK03: .BLKB 64
EA47                                DSKSEL: .BLKB 4
EA47                                DISKNO: .BLKB 1
EA47                                TRACK: .BLKB 1
EA47                                SECTOR: .BLKB 1
EA47                                STATUS: .BLKB 1
EA47                                CMND: .BLKB 1
EA47                                RWFLG: .BLKB 1
EA47                                DMAAD: .BLKB 2
EA47                                PARFLG: .BLKB 1
EA47                                HSELCD: .BLKB 1

EA33                                !SCRATCH DIRECTORY AREA
EA33                                !ALLOCATION VECTOR 0
EA33                                !ALLOCATION VECTOR 1
EA33                                !ALLOCATION VECTOR 2
EA33                                !ALLOCATION VECTOR 3
EA33                                !CHECK VECTOR 0
EA33                                !CHECK VECTOR 1
EA33                                !CHECK VECTOR 2
EA33                                !CHECK VECTOR 3
EA33                                !DISK ACCESS TABLE
EA33                                !DISK PARAMETERS

EA33                                !PAR ERROR FLAG
EA33                                !HOST SELECT CODE

```

```

F3C1                                HSTBLK: .BLKB 1
F3C2                                BLKCNT: .BLKB 1
F3C3                                SECHSK: .BLKB 1
F3C4                                WRTYPE: .BLKB 1
F3C5                                UNACNT: .BLKB 1
F3C6                                UNADSK: .BLKB 1
F3C7                                UNATRK: .BLKB 1
F3C8                                UNASEC: .BLKB 1
F3C9                                SEKDSK: .BLKB 1
F3CA                                SEKTRK: .BLKB 1
F3CB                                SEKSEC: .BLKB 1
F3CC                                SEKHSK: .BLKB 1
F3CD                                SIDSEL: .BLKB 1
F3CE                                HSTWRT: .BLKB 1
F3CF                                HSTACT: .BLKB 1
F3D0                                RSFLAG: .BLKB 1
F3D1                                READOP: .BLKB 1
F3D2                                LUNIT: .BLKB 1
F3D3                                EREFLAG: .BLKB 1
F3D4                                RFLAG: .BLKB 1
F3D5                                BTSEC: .BLKB 1
F3D6                                SEKDMA: .BLKB 2
F3D8                                CTBLP: .BLKB 2
F3DA                                UTBLP: .BLKB 2
F3DC                                HLSTR: .BLKB 2
F3DE                                DPBASE: .BLKB 2
F41E                                IDSV: .BLKB 6
09F1                                !
                                !DATSIZ = .-BEGDAT
                                !END OF PROGRAM
                                !.END

```

```

                                !CPM SEC/PHY SEC
                                !REC/CPMBLK
                                !SECTOR MASK
                                !WRITE TYPE
                                !UNALOC REC COUNT
                                !LAST UNALOC DISK
                                !LAST UNALOC TRACK
                                !LAST UNALOC SECTOR
                                !SEEK DISK
                                !SEEK TRACK
                                !SEEK SECTOR
                                !PHYSICAL SEC ADR
                                !SIDE SELECT STORAGE
                                !HOST WRITTEN FLAG
                                !HOST ACTIVE FLAG
                                !READ SECTOR FLAG
                                !I-READ, O-WRITE
                                !LAST SELECTED DRIVE
                                !ERROR FLAG
                                !CPM READ FLAG
                                !NO OF BOOT SECS
                                !SEEK DMA ADDRESS
                                !CURRENT TBL ADR
                                !UNALOC TBL ADR
                                !CPM DISK PARAMETER HEA
                                !SIZE OF SCRATCH
                                !BEGDAT
                                !END OF PROGRAM
                                !.END

```