

```

1
2      title "CCP/M-86 Loader Program and BIOS"
3      ;*****
4      ; The Loader consists of the three modules:
5      ; Loader BDDS (the file LBDDOS.H86), the Loader
6      ; Program and the Loader BIOS. This
7      ; module contains both the Loader Program
8      ; and the Loader BIOS.
9      ;
10     ; The Loader resides in sectors 2-8 on the first track
11     ; of an IBM PC floppy diskette. The Loader is
12     ; brought into memory by the boot Sector which
13     ; resides in sector 1 of track 0. The boot Sector
14     ; is brought into memory by the IBM PC's ROM
15     ; monitor.
16     ;
17     ; The Loader Program opens the file "CCPM.SYS" using the
18     ; Loader BDDS and Loader BIOS,
19     ; and then reads it into memory. The DS register is set
20     ; to the start of the CCPM DATA area, and a JMPF to the
21     ; first byte of the CCPM code is executed.
22     ;
23     ; The first 128 byte record of the CCPM.SYS file is a header
24     ; with the following format:
25     ;
26     ; +---+---+---+---+---+---+---+---+
27     ; |TYPE| LEN | ABS | MIN | MAX |
28     ; +---+---+---+---+---+---+---+---+
29     ;
30     ; type      rb      1      ;seg type
31     ; len       rw      1      ;length
32     ; abs       dw      1      ;absolute segment address for LOADER
33     ; min       rw      1      ;minimum mem
34     ; max       rw      1      ;max mem needed
35     ;
36     ; The code is expected first and then the data within CCPM.SYS
37     ; This header record is constructed automatically by the
38     ; GENCCPM utility. See the variables declared at "SEC1:"
39     ; where the first sector of CCPM.SYS will be read.
40     ;
41     ; Loader may be read into any segment by the boot
42     ; Sector that does not overlap the area of memory
43     ; the system image in CCPM.SYS will occupy.
44     ;
45     ;*****
46
47     0000      false      equ      0
48     FFFF      true       equ      not false
49
50     000D      cr         equ      0dh
51     000A      lf         equ      0ah
52
53     0000      lddos_offset equ      0000H      ;offset of Loader BDDS

```

CCP/M-86 2.0 V.2
 COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # CC-104

```

54
55      0900      ldr_offset      equ      0900h      ;offset of Loader BIOS
56
57      0001      codetype       equ      1          ; code type CMD header
58      0002      datatype       equ      2          ; data type CMD header
59
60      ; dosos function numbers
61
62      000E      seldskf         equ      14
63      000F      openf          equ      15
64      0014      readsf         equ      20
65      001A      setdmaf        equ      26
66      0020      setuserf       equ      32
67      002C      setmultcntf    equ      44
68      0033      setdmabf       equ      51
69
70
71      ;*****
72      ;*
73      ;*      CCPMLDR starts here
74      ;*
75      ;*****
76
77      cseg
78      org      ldr_offset
79
80      0900 E96303      0C66      jmp init
81      0903 E9D700      090D      jmp entry
82      0906 E90100      090A      jmp loadp
83
84      0909 00          ldusr     db      0          ;load user is 0
85
86
87      loadp:          ; loader entry from BIOS init
88      ;-----
89      ;      entry:  CH = boot user number
90      ;              CL = boot disk number
91      ;              DS,ES,SS = CS
92
93      090A BA540D          mov dx,offset signon
94      090D E8B800      09C8      call msg
95
96      0910 2E8A160909      mov dl,ldusr
97      0915 D120CDE0      mov cl,setuserf 1 int 224      ;set user number
98
99      0919 BAC70D          mov dx,offset ccpmfcb
100     091C B10FCDE0      mov cl,openf 1 int 224          ;open CCPM.SYS file
101     0920 3CFF7506      092A      cmp al,255 ! jne perr      ;insure good file
102     0924 BA680D          mov dx,offset nofile
103     0927 E97500      099F      jmp stop
104     perr:
105     092A 8CDA          mov dx,ds
106     092C B133CDE0      mov cl,setdmabf 1 int 224      ;set DMA segment address

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```

107
108
109 0930 BAE80D      mov dx,offset sect1
110 0933 B11ACDE0    mov cl,setdmaf 1 int 224      ;set DMA offset address
111
112 0937 B201        mov dl,1
113 0939 B12CCDE0    mov cl,setmultcntf 1 int 224    ;set Multi-sector count to 1
114
115 093D BAC70D      mov dx,offset ccpmfcb
116 0940 B114CDE0    mov cl,readsf 1 int 224      ; read first record
117
118                ; the following
119                ; "commented out" code
120                ; can be used to
121                ; perform error checking
122
123                ; cmp ctype,codetype      ; code should be first
124                ; jnz badhdr
125                ; cmp dtype,datatype      ; then data
126                ; jnz badhdr
127                ; mov ax,cldseg           ; code abs + code length
128                ; add ax,clen             ; should be = to data abs
129                ; cmp ax,dldseg 1 jnz badhdr
130                ; add ax,dlen 1 cmp ax,cldseg ; check for wrap around
131                ; ja hdrok
132                ;badhdr:
133                ;     mov dx,offset rerr
134                ;     jmp stop
135                ;hdrok:
136 0944 BA9C0D      mov dx,offset csegment
137 0947 E88100      call msg                      ; put memory map on console
138 094A A1E80D      mov ax,cldseg
139 094D E85400      call phex                      ; print base code segment
140 0950 BA800D      mov dx,offset dsegment
141 0953 E87500      call msg                      ; print base data segment
142 0956 A1F40D      mov ax,dldseg
143 0959 E84800      call phex
144
145 095C BA8000      mov dx,128
146 095F B12CCDE0    mov cl,setmultcntf 1 int 224    ; set multi-sector count to 128
147 0963 BA0000      mov dx,0
148 0966 B11ACDE0    mov cl,setdmaf 1 int 224      ; set DMA offset to 0
149 096A 8B16E80D    mov dx,cldseg                ; initial DMA segment
150
151                readit1:
152 096E 52          push dx                      ; save dma segment
153 096F B133CDE0    mov cl,setdmabf 1 int 224      ; set DMA segment for disk io
154 0973 BAC70D      mov dx,offset ccpmfcb
155 0976 B114CDE0    mov cl,readsf 1 int 224      ; next 128 sector read
156 097A 5A          pop dx                     ; restore dma segment
157 097B 81C20004    add dx,8*128                ; increment dma segment
158 097F 3C01740A    cmp al,01H 1 je done          ; check for EOF
159 0983 3C0074E7    cmp al,0 1 je readit1         ; check for good write
159 0987 BA830D      mov dx,offset rerr          ; print READ ERROR message

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```

160
161 098A E91200      099F      jmp stop
162      done:
163 098D EB3800      09CB      call pcrLf      ; and a crlf
164
165 0990 C706E90D0000      mov clen,0      ; set CCPM offset to 0
166
167 0996 8E1EF40D      mov ds,dldseg      ; CCP/M data segment
168 099A 2EFF2E90D      jmpf cs:dword ptr clen      ; leap to CCPM initialization
169
170      ;*****
171      ;*
172      ;*      subroutines
173      ;*
174      ;*****
175
176      stop:
177      ;----
178 099F EB2900      09CB      call msg
179 09A2 FAF4      cli ; hlt
180
181
182      phex:
183      ;----
184 09A4 B90404      mov cx,0404h      ; 4 in both CH and CL
185
186      lhex:
187 09A7 D3C0      rol ax,cl      ; rotate left 4
188 09A9 5150      push cx ; push ax      ; save crucial registers
189 09AB E80700      09B5      call pnib      ; print hex nibble
190 09AE 5859      pop ax ; pop cx      ; restore registers
191 09B0 FEC075F3      09A7      dec ch ; jnz lhex      ; and loop four times
192 09B4 C3      ret
193
194      pnib:
195 09B5 240F3C09      and al,0fh ; cmp al,9      ;print low nibble in AL as hex char
196 09B9 7704      09BF      ja p10      ;above 9 ?
197 09BB 0430      add al,"0"      ;digit
198 09BD EB02      09C1      jmps prn
199 09BF 0437      p10:      add al,"A"-10      ;char a-e
200 09C1 8AD0      prn:      mov dl,al
201
202      putchar:
203      ;-----
204 09C3 8ACA      mov cl,dl
205 09C5 E92900      09F1      jmp io_conout
206
207      pcrLf:
208      ;-----
209 09C8 BAC40D      mov dx,offset crlf      ;print carriage return, line feed
210
211      msg:
212      ;---
213 09CB 8BDA      mov bx,dx      ;print msg starting at dx until ;
214      msg1:

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```
213
214 09CD 8A17          mov dl,[bx]
215 09CF 80FA247408    cmp dl,'$' l je msg2
216 09D4 53            push bx
217 09D5 E8EBFF        call putchar
218 09D8 58            pop bx
219 09D9 43            inc bx
220 09DA EBF1          jmps msg1
221                      msg2:
222 09DC C3            ret
223
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____

```

224      eject
225      ;*****
226      ;
227      ;
228      ;
229      ;          L O A D E R
230      ;
231      ;          B I O S - 8 6
232      ;          =====
233      ;
234      ;          CP/M-86 3.0 or Concurrent CP/M-86 2.0
235      ;
236      ;          Boot Loader I/O System
237      ;          for the
238      ;          IBM Personal Computer
239      ;
240      ;          Copyright (c) 1982
241      ;          Digital Research, Inc.
242      ;          box 579, Pacific Grove
243      ;          California, 93950
244      ;
245      ;          (Permission is hereby granted to use or
246      ;          abstract the following program in the
247      ;          implementation of Concurrent CP/M-86,
248      ;          CP/M, MP/M or CP/Net for the 8086 or 8088
249      ;          micro-processor.)
250      ;
251      ;*****
252      ;
253      ;          Register usage for BIOS interface routines:
254      ;
255      ;          Entry:  AL = function # (in entry)
256      ;                   CX = entry parameter
257      ;                   DX = entry parameter
258      ;                   DS = LDRDOS data segment
259      ;
260      ;          Exit:   AX = return
261      ;                   BX = AX (in exit)
262      ;          ALL SEGMENT REGISTERS PRESERVED:
263      ;          CS,DS,ES,SS must be preserved though call
264      ;
265      ;          Notes: flag set and the far jump to the dispatcher are
266      ;                   the only legal Operating System "entry" points
267      ;                   for an interrupt routine)
268      ;
269      ;                   changes have been made in the
270      ;                   register conventions from
271      ;                   the CP/M-86 BIOS and the MP/M-86 BIOS.
272      ;
273      ;*****
274      ;*****
275      ;
276      ;

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

*SER. # _____

```

277
278      ;*          IBM PC SOFTWARE INTERRUPT STRUCTURE          *
279      ;*
280      ;*****
281
282      0008      tick_interrupt      equ      08h
283      0009      keyboard_interrupt equ      09h
284      000E      disk_interrupt     equ      0Eh
285      00E0      os_interrupt       equ      224      ;Loader BIOS entry
286      00E1      debugger_interrupt equ      225      ;debugger entry to O.S.
287
288      ;*****
289      ;*
290      ;*          INTERFACE TO LOADER BIOS
291      ;*
292      ;*****
293
294      ;=====
295      ;=====
296      entry:    ;arrive here from JMP at
297      ;=====      ;03H in BIOS code segment
298      ;=====
299
300      ;      entry:  AL = function number
301      ;
302      ;      exit:   AX = BX = return
303      ;
304      ;      ALL SEGMENT REGISTERS PRESERVED:
305      ;      CS,DS,ES,SS must be preserved though call
306
307      ;      Note: no alteration of stack is allowed during entry except
308      ;      for the return address caused by the "call function_table[0x1]"
309      ;      instruction.
310
311      09DD FC      cld                      ;set the direction flag
312      09DE 32E4    xor ah,ah
313      09E0 00E0    shl al,1                ;multiply by 2
314      09E2 8808    mov bx,ax               ;put in indirect register
315      09E4 FF97C50C call function_table[0x1]    ;no range checking needed
316      09E8 8CDB    mov bx,ax               ;only called by loader BIOS
317      09EA CB      retf                    ;return to loader BIOS
318
319      09EB C3      io_ret:                 ret
320
321      ;-----
322      i_disk:
323      ;-----
324      09EC 8020    mov al,pic_nseoi        ;signal end of interrupt
325      09EE E620    out pic_even_port,al    ;to 8259
326      09F0 CF      iret                    ;note we destroyed AL
327
328      ;=====
329      io_conout:

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```

330
331 ;=====
332 ; entry: CL = character to output
333 ;         DL = device number
334 ; exit:   None
335 ;         ALL SEGMENT REGISTERS PRESERVED:
336 ;         CS,DS,ES,SS must be preserved though call
337
338 ; Put character in screen and update cursor position
339 ; BX = screen structure
340
341 09F1 8BE10C      mov bx,offset ss0
342 09F4 8B3F      mov di,ss_cursor[bx]      ;cursor offset in bytes
343 09F6 80F90D      cmp cl,cr
344 09F9 7423      je carriage_return
345 09FB 80F90A      cmp cl,lf
346 09FE 7428      je linefeed
347 0A00 06      push es
348 0A01 880080      mov ax,bw_video_seg
349 0A04 8EC0      mov es,ax      ;segment of foreground screen
350 0A06 8AC1      mov al,cl
351 0A08 E407      mov ah,07h      ;default attribute
352 0A0A AB      stosw      ;update and don't touch the attribute
353 0A0B 07      pop es      ;DI incremented by 2
354 0A0C 807F074F      cmp ss_column[bx],columns_per_screen - 1
355 0A10 7506      jne inc_col
356 0A12 E80900      call carriage_return
357 0A15 E91000      jmp line_feed
358
359 0A18 FE4707      inc_col:      inc ss_column[bx]      ;DI = next data, attribute
360 0A1b 893F      mov ss_cursor[bx],di      ;save new cursor position
361 0A1D C3      ret
362
363 ;-----
364 carriage_return:
365 ;-----
366 ; entry:  BX = screen structure
367 ; exit:   BX preserved
368
369 0A1E 33D2      xor dx,dx      ;AX = 0
370 0A20 865707      xchg di,ss_column[bx]      ;set cursor position to beginning of
371 0A23 D1E2      shl dx,1      ;times 2 for data and attribute
372 0A25 2917      sub ss_cursor[bx],dx      ;set new cursor position
373 0A27 C3      ret
374
375 ;-----
376 line_feed:
377 ;-----
378 ; entry:  BX = screen structure
379 ; exit:   BX preserved
380
381 ; This routine assumes the loader will never
382 ; need to scroll the screen

```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____

```

383
384
385 0A28 FE4706          inc ss_row[ebx]
386 0A28 8107A000       add ss_cursor[ecx],columns_per_screen*2
387 0A2F C3             ret
388
389
390 *****
391 ;*
392 ;*          6845 CRT CONTROLLER PORT AND COMMAND EQUATES
393 ;*
394 *****
395
396
397 ; The IBM PC's monochrome memory mapped video display begins
398 ; at paragraph 0B000H. It represents a screen 80 X 25.
399 ; Each video character requires a word value, the low byte
400 ; is the ASCII code (characters codes > 126 are also displayed)
401 ; and the high byte is an attribute byte. The 25th line
402 ; is reserved by this BIOS as a status line.
403
404
405 0384                bw_card          equ      003b4h
406
407 0029                video_on         equ      00029h
408 0021                video_off        equ      00021h
409
410 000A                cursor_start     equ      10
411 000B                cursor_end       equ      11
412 000C                display_start_hi equ      12
413 000D                display_start_low equ     13
414 000E                cursor_hi        equ      14
415 000F                cursor_low       equ      15
416 0010                light_pen_hi     equ      16
417 0011                light_pen_low    equ      17
418
419
420 *****
421 ;*
422 ;*          SCREEN PARAMETERS
423 ;*
424 *****
425
426 0018                rows_per_screen  equ      24
427 0050                columns_per_screen equ     80
428 0780                screen_siz       equ      rows_per_screen * columns_per_screen
429 ;in words
430 B000                bw_video_seg      equ      0b000h
431 ;segment address of
432 0F00                bw_video_status_line equ     screen_siz * 2
433 ;byte offset of status line
434
435 *****

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

```

436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488

```

```

;
;
;
;
;*****
;
;      Each virtual console has a structure of the following
;      format associated with it. (SS = Screen Structure)
;      The data in this structure is dependent on the type of screen
;      supported and any escape sequence handling in io_conout.
;      Note: ss_cursor, ss_row, ss_column are relative to 0 and are
;      word pointers, i.e., if ss_cursor is 1 then it refers to
;      bytes 2 and 3 in the video RAM or a background screen's
;      data area.
;
ss_cursor      equ      word ptr 0           ;points at data/attrib
ss_escape      equ      word ptr ss_cursor + word ;escape routine to return to
ss_screen_seg  equ      word ptr ss_escape + word ;data for screen image
ss_row         equ      byte ptr ss_screen_seg + word ;current row
ss_column      equ      byte ptr ss_row + byte   ;current col

;
;      DISK I/O
;      -----
;
;*****
;
;      8237 DIRECT MEMORY ACCESS CONTROLLER PORT AND COMMANDS
;
;*****
;
dma_c0_address equ      000h           ;8237 channel 0 address      rw
dma_c0_count   equ      001h           ;8237 channel 0 transfer count  rw
dma_c1_address equ      002h           ;8237 channel 1 address      rw
dma_c1_count   equ      003h           ;8237 channel 1 transfer count  rw
dma_c2_address equ      004h           ;8237 channel 2 address      rw
dma_c2_count   equ      005h           ;8237 channel 2 transfer count  rw
dma_c3_address equ      006h           ;8237 channel 3 address      rw
dma_c3_count   equ      007h           ;8237 channel 3 transfer count  rw
dma_stat_reg   equ      008h           ;8237 status register        ro
dma_cmd_reg    equ      dma_stat_reg    ;8237 command register        ro
dma_requ_reg   equ      dma_stat_reg+1  ;8237 software dma request     ro
dma_bmsk_reg   equ      dma_stat_reg+2  ;8237 binary channel mask     ro
dma_mode_reg   equ      dma_stat_reg+3  ;8237 mode register           ro
dma_cbpf       equ      dma_stat_reg+4  ;8237 clear byte pointer f/f   ro
dma_temp_reg   equ      dma_stat_reg+5  ;8237 temporary register      ro
dma_clear      equ      dma_stat_reg+5  ;8237 master clear            ro
dma_mask_reg   equ      dma_stat_reg+7  ;8237 linear channel mask     ro

dma_page_c1    equ      080h           ;a16 to a20 for channel 1
dma_page_fdc   equ      081h           ;a16 to a20 for channel 2
dma_page_c3    equ      082h           ;a16 to a20 for channel 3

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P.O. BOX 579

PACIFIC GROVE, CA 93950

SER # _____

```

489
490 ; The following labels define single mode, address increment
491 ; auto-initialization disable, read or write using channel 2
492
493 004A dma_mode_write_fdc equ 01001010b
494 0046 dma_mode_read_fdc equ 01000110b
495
496 0002 dma_bmsk_fdc equ 00000010b ;binary channel mask for disk
497
498
499 ; DMA channel assignments
500
501 ;channel 0 dynamic memory refresh
502 ;channel 1
503 ;channel 2 floppy disk controller
504 ;channel 3
505
506
507 ;*****
508 ;*
509 ;* FLOPPY DISK DRIVER EQUATES
510 ;*
511 ;*****
512
513
514 ; The following equates are set to the size of a double density,
515 ; single sided 5 1/4" floppy.
516
517 0200 bytes_per_sector equ 512
518 0008 sectors_per_track equ 8 ;1 to 8
519 1000 bytes_per_track equ sectors_per_track * bytes_per_sector
520 0028 tracks_per_disk equ 40 ;0 to 39
521 8000 bytes_per_disk equ tracks_per_disk * bytes_per_track
522
523
524 ; The following equates are for the INTEL 8272 Floppy Disk
525 ; Controller.
526
527 03F4 fdc_stat equ 03f4h ;status port for the disk controller
528 03F5 fdc_data equ fdc_stat+1 ;data port for the disk controller
529 03F2 fdc_port equ 03f2h ;all bits clear on channel reset
530
531 ;7 6 5 4 3 2 1 0
532 ;| | | | | \ /
533 ;| | | | | |
534 ;| | | | | | drive select 00=a,01=b,10=c,11=d
535 ;| | | | | fdc reset#
536 ;| | | | int & dmaeq enable
537 ;d c b a motor on
538
539 000C fdc_on equ 00001100b ;mask to keep the 8272 unreset
540 00FC fdc_no_motor equ 11111100b ;mask for no motors
541 0066 fdc_read_cmd equ 01100110b ;mfM, skip deleted data, read

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```

542
543 0045 fdc_write_cmd equ 01000101b ;imr, write
544 004D fdc_format_cmd equ 01001101b ;imr, format
545 000F fdc_seek_cmd equ 00001111b ;seek
546 0007 fdc_recal_cmd equ 00000111b ;home to track 0
547 0008 fdc_si_cmd equ 00001000b ;sense interrupt status
548 0003 fdc_spec_cmd equ 00000011b ;specify
549
550 0080 fdc_ready equ 10000000b ;mask for transfer ready
551
552 00CF fdc_spec_1 equ 11001111b ;srt=0c, hd unload=0f first specify byte
553 0003 fdc_spec_2 equ 00000011b ;hd load=1, mode=DMA second specify byte
554
555 0002 f_bytes equ 2 ;magic number for 512 bytes per sector
556 0008 f_sectors equ 8 ;sectors per track
557 003A f_gap equ 03ah ;magic number for format gap
558 00E5 f_filler equ 0e5h ;fill character
559
560 0002 r_bytes equ 2 ;magic number for 512 bytes
561 0008 r_sectors equ 8
562 002A r_gap equ 02ah
563 00Ff r_util equ 0ffh

```

```

564
565 ; Equates for parameter passing for read and write from the
566 ; BIOS.

```

```

567
568 ; At the disk read and write function entries,
569 ; all disk I/O parameters are on the stack.
570 ; The stack at these entries appears as
571 ; follows:

```

```

572 ;
573 ; +-----+-----+
574 ; +14 | DRV | MCNT | Drive and Multi sector count
575 ; +-----+-----+
576 ; +12 | TRACK | Track number
577 ; +-----+-----+
578 ; +10 | SECTOR | Physical sector number
579 ; +-----+-----+
580 ; +8 | DMA_SEG | DMA segment
581 ; +-----+-----+
582 ; +6 | DMA_OFF | DMA offset
583 ; +-----+-----+
584 ; +4 | RET_SEG | BIOS return segment
585 ; +-----+-----+
586 ; +2 | RET_OFF | BIOS return offset
587 ; +-----+-----+
588 ; SP+0 | RET_ADR | Return address to BIOS ENTRY routine
589 ; +-----+-----+
590 ;

```

```

591 ; These parameters may be indexed and modified
592 ; directly on the stack by the BIOS read and write routines
593 ; They will be removed by the BIOS when the BIOS completes
594 ; the read/write function and returns to the BIOS.

```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____

```

595
596
597
598     000E             drive    equ     byte ptr 14[ebp]
599     000F             mcnt     equ     byte ptr 15[ebp]
600     000C             track    equ     word ptr 12[ebp]
601     000A             sector   equ     word ptr 10[ebp]
602     0008             dma_seg  equ     word ptr 8[ebp]
603     0006             dma_off  equ     word ptr 6[ebp]
604
605
606             ;          Some equates in the Disk Parameter Header (DPH)
607             ;          and the Disk Parameter Block.
608
609     0000             xlt       equ     0           ;translation table offset in DPH
610     0008             dpb       equ     8           ;disk parameter block offset in DPH
611     0000             spt       equ     0           ;sectors per track offset in DPH
612     000F             psh       equ     15          ;physical shift factor offset in DPB
613
614

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

```

615             ;*****
616             ;*
617             ;*          DISK DRIVER ROUTINES
618             ;*
619             ;*****
620
621
622             ;=====
623             io_sel disk:      ; Function 7: Select Disk
624             ;=====
625             ;          entry:  CL = disk to be selected
626             ;                  DL = 00h if disk has not been previously selected
627             ;                  = 01h if disk has been previously selected
628             ;          exit:   AX = 0 if illegal disk
629             ;                  = offset of DPH relative from
630             ;                  BIOS Data Segment
631             ;          ALL SEGMENT REGISTERS PRESERVED:
632             ;          CS,DS,ES,SS must be preserved though call
633
634     0A30 33C0             xor ax,ax                ;get ready for error(s)
635     0A32 80F900           cmp cl,0                ;is it a A: ?
636     0A35 7503             jne sel_ret              ;if not just exit
637     0A37 8B0900           mov ax,offset dph0
638
639     0A3A C3              sel_ret:                  ret
640
641             ;=====
642             io_read:        ; Function 11: Read sector
643             ;=====
644             ; Reads the sector on the current disk, track and
645             ; sector into the current DMA buffer.
646
647             ;          entry:  parameters on stack

```

```

648
649 ; exit: AL = 00 if no error occurred
650 ; AL = 01 if an error occurred
651 ; AL = 0ffh if density change detected
652 ; ALL SEGMENT REGISTERS PRESERVED:
653 ; CS,DS,ES,SS must be preserved though call
654
655 0A3B 88EC mov bp,sp ;set BP for indexing into IOPB
656 ; mov dma_mode_storage,dma_mode_read_fdc
657 ; mov fdc_rw_cnd,fdc_read_cnd ;loader always reads
658 0A3D E80700 0A47 call flp_io
659 0A40 8001 mov al,1
660 0A42 7502 0A46 jnz i_read_ret ;zero flag is set with successful
661 0A44 FEC8 dec al ;I/O
662 i_read_ret:
663 0A46 C3 ret
664
665 flp_io:
666 ;-----
667 ; entry: parameters on stack
668 ; exit: zero flag set if no error
669
670 0A47 B80800 mov ax,sectors_per_track;max sectors - starting sector =
671 0A4A 28450A sub ax,sector ;sectors left on track
672 0A4D 32FF xor bh,bh
673 0A4F 8A5E0F mov bl,mcnt ;multi sector count is byte variable
674 0A52 38C3 cmp ax,bx ;sectors left on track, sectors requested
675 0A54 7603 0A59 jbe flp_track ;transfer to end of track
676 0A56 8A450F mov al,mcnt ;transfer to before end of track
677
678 flp_track:
679 0A59 28460F sub mcnt,al ;AL = # sectors to R/W on this iteration
680 ;through FLP_IO,
681 0A5C A2040D mov track_mcnt,al ;sectors to R/W on current track
682
683 ;check for 64K page overlap
684 0A5F 8AE0 mov ah,al ;shl al,8 (* 256)
685 0A61 32C0 xor al,al
686 0A63 D0E4 shl ah,1 ;* 512
687 0A65 50 push ax ;how many bytes to R/W on cur trk
688
689 0A66 8B4608 mov ax,dma_seg ;compute new 20 bit DMA addr
690 0A69 8B5E06 mov bx,dma_off
691 0A6C E89400 0B03 call comp_dma ;returns AX = low 16 bits of 20 bit adr
692 0A6F F7D0 not ax ;how many bytes left in this 64K page
693
694 0A71 5B pop bx ;BX=bytes to R/W on current track
695 0A72 38C3 cmp ax,bx ;does this transfer fit in 64K page
696 0A74 7203 0A79 jb flp_end_64K
697 0A76 E95D00 0A06 jmp flp_pg_ok
698
699 flp_end_64K:
700 ;read to end of 64K page and then

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

701
702                                     ;read spanning sector locally
703 0A79 8AC4          mov al,an        ;how many sectors fit in this page ?
704 0A7B 32E4          xor ah,an        ;bytes left in 64K page
705 0A7D 01E3          shr ax,1        ;divided by 512
706 0A7F 85C0          test ax,ax      ;if 0, no sectors fit
707 0A81 741E          jz flp_rwlocal  ;in the rest of this 64K page
708
709 0A83 A2030D         mov num_sec,al   ;sectors that fit in end 64K page
710 0A86 2806040D       sub track_mcnt,al ;track_mcnt always > AL
711 0A8A E88E00         call flp_phys_io ;read/write them, DMA already computed
712 0A8D 7403          jz flp_end64k_ok
713 0A8F E97000         jmp flp_ret    ;return if zero reset on error
714
715 flp_end64k_ok:
716 0A92 33C0          xor ax,ax
717 0A94 A0030D         mov al,num_sec   ;compute new DMA offset
718 0A97 01460A         add sector,ax   ;still on the same track
719 0A9A 86E0          xchg ah,al       ;shl ax,8 (* 256)
720 0A9C D0E4          shl ah,1        ;* bytes_per_sector (512)
721 0A9E 014506         add dma_off,ax  ;64K wrap around is legal
722
723 flp_rwlocal:
724 0AA1 BB7010         mov bx,local_buf ;read into BIOS data segment the spanning
725 0AA4 8C38          mov ax,ds        ;sector
726 0AA6 E85A00         call comp_dma   ;compute 20 bit local DMA addr
727                                     ;
728                                     ;
729                                     ;
730                                     ;
731                                     ;
732                                     ;
733                                     ;
734                                     ;
735                                     ;
736                                     ;
737                                     ;
738                                     ;
739                                     ;
740                                     ;
741 flp_local:
742
743 0AA9 C606030D01     mov num_sec,1   ;read/write one sector
744 0AAE E86A00         call flp_phys_io
745 0AB1 754F          jnz flp_ret      ;return zero flag reset
746 0AB3 FF460AFE0E04   inc sector ; dec track_mcnt
747 0D
748
749                                     ;
750                                     ;
751                                     ;
752 0ABA 8B7E06         mov di,dma_off   ;move the sector to user's area
753 0ABD BE7010         mov si,local_buf ;if reading

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

```

754
755 OAC0 068E4608          push es 1 mov es,dma_seg
756 OAC4 B90001          mov cx,bytes_per_sector/2
757 OAC7 F3A5            rep movsw
758 OAC9 897E06          mov dma_off,di          ;update DMA offset
759 OACC 07              pop es
760
761                      flp_local_done:
762 OACD 8B4508          mov ax,dma_seg          ;compute new 20 bit DMA addr
763 OAD0 8B5E06          mov dx,dma_off          ;for next FDC read/write
764 OAD3 E82900          call comp_dma
765
766                      flp_pg_ok:
767 OAD6 A0040D          mov al,track_mcnt          ;read will not cross 64K boundary
768 OAD9 84C07413        test al,al 1 jz nxt_track          ;could be 0 if we just read locally
769 OADD A2030D          mov num_sec,al          ;read the rest from this track
770 OAE0 E83800          call flp_phys_io          ;DMA is already computed
771 OAE3 751D          jnz flp_ret          ;return zero flag reset on error
772 OAE5 33C0          xor ax,ax
773 OAE7 8A26030D        mov ah,num_sec          ;shl num_sec,8 (* 256)
774 OAE8 00E4          shl ah,1          ;* 512
775 OAE0 014506          add dma_off,ax
776
777                      nxt_track:
778 OAF0 32C0          xor al,al
779 OAF2 38450F          cmp mcnt,al
780 OAF5 740B          jz flp_ret          ;return successful with zero flag set
781 OAF7 FF460C          inc track
782 OAF8 C7450A0000      mov sector,0
783 OAFF E945FF          jmp flp_io
784
785 O802 C3          flp_ret: ret
786
787                      comp_dma:          ;Compute 20 bit address from offset, segment
788                      ;-----
789
790                      ; entry: AX = segment
791                      ;        BX = offset
792                      ; exit:  AX = low 16 bits
793                      ;        CH = highest 4 bits of address, always less than 16 -
794                      ;        no megabyte wrap around
795
796                      ;
797                      ; The BIOS variables DMA_LOW16 and DMA_HIGH4 are
798                      ; set by this routine. These variables are transferred
799                      ; to the floppy disk controller by the routine DMA_SET_UP.
800
801 O803 B104D3C0        mov cl,4 1 rol ax,cl          ;make paragraphs into bytes
802 O807 8AE824F0        mov ch,al 1 and al,0F0h          ;save high 4 bits, 0 low 4 bits
803 O80B 03C3          add ax,bx          ;add byte offset
804 O80D 80D50080E50F    adc ch,0 1 and ch,0Fh          ;add in the carry, page is less than
805 O813 A3050D          mov dma_low16,ax          ;16
806 O816 882E070D        mov dma_high4,ch
807 O81A C3          ret

```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____

```

807
808
809      flp_phys_io:
810      ;-----
811      ;      entry:  num_sec = number of sectors to read in this
812      ;              operation
813      ;              disk parameters on the stack
814      ;      exit:   zero flag set if ok
815
816      ;      Perform physical actions to read/write floppy diskette
817
818      0d1b c605f10c02      mov d_a_number,r_bytes
819      0b20 c606f20c08      mov d_a_eot,r_sectors
820      0b25 c606f30c2a      mov d_a_gpl,r_gap
821      0b2a c606f40cff      mov d_a_dtl,r_dtl
822
823      0b2f e82700      0b59      call motor_and_recal      ;make sure motor is on
824      0b32 c606ea0c05      mov recals,max_recals
825
826      0b37 c606e90c03      recal_loop:
827                          mov retries,max_retries
828                          retry_loop:
829      0b3c e85000      0b8f      call seek
830      0b3f e89100      0b03      call dma_setup
831      0b42 e8b700      0bfc      call fdc_read_write
832      0b45 7411      0b58      jz phys_ret      ;if errors
833      0b47 fe0ee90c      dec retries      ;attempt retries
834      0b4d 75ef      0b3c      jnz retry_loop
835      0b4d e82800      0b78      call recal
836      0b50 fe0eea0c      dec recals
837      0b54 75e1      0b37      jnz recal_loop
838      0b56 0c01      or al,1      ;reset zero flag
839      0b58 c3      phys_ret:      ;and return error
840                          ret
841
842      motor_and_recal:
843      ;-----
844      ;      entry:  none
845      ;      exit:   none
846
847      ;      Turn on the motor if off. Also do RECAL operation if
848      ;      motor is off. Note: loader does not turn off the motor.
849
850      0b59 8010      mov al,010h      ;pick up a bit for motor a
851      0b5b 8a4e0e      mov cl,drive      ;fetch the binary drive code
852      0b5e d2e0      shl al,cl      ;make it a bit for motor x
853      0b60 8a25080d      mov ah,motor_flags      ;fetch the motor bits
854      0b64 84e0      test ah,al      ;check to see if its on
855      0b66 750f      0b77      jnz motor_on_done      ;yes then leap
856      0b68 0c0c      or al,fdc_on      ;mask in the no reset,enable interrupt
857      0b6a 0a460e      or al,drive      ;mask in the drive
858      0b6d a2080d      mov motor_flags,al      ;save for later
859      0b70 8af203      mov dx,fdc_port      ;point to the port number
860      0b73 ee      out dx,al      ;select & motor on

```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____

```

860
861 0B74 E80100      0B78      call recal          ;back to track zero
862                      motor_on_done:          ;when we first turn on the motor
863 0B77 C3          ret
864
865                      recal:
866                      ;-----
867                      ;      entry:  none
868                      ;      exit:   none
869
870                      ;      Move the head to home on the selected disk drive.
871
872 0B78 C605E80C02    mov disk_arguments,2        ;specify number of arguments
873 0B7D C606EC0C07    mov d_a_command,fdc_recall_cmd
874 0B82 8A450E        mov al,drive                ;get current disk
875 0B85 A2E00C        mov d_a_drive,al            ;set up the command block
876 0B88 E8A400        call fdc_command_put        ;go send the command block
877 0B8B F4            hlt                          ;wait for FDC interrupt
878 0B8C E98A00        0C19    jmp sense_interrupt ;required after RECAL command
879                      ;ret
880
881                      seek:
882                      ;-----
883                      ;      entry:  TRACK to seek to on the stack relative to BP
884                      ;      exit:   none
885
886 0B8F C606E80C03    mov disk_arguments,3        ;request 3 byte command transfer
887 0B94 C606EC0C0F    mov d_a_command,fdc_seek_cmd
888 0B99 320B          xor bl,bl                    ;select head 0
889 0B9B 8B450C        mov ax,track                ;get track off stack
890 0B9E 3C28          cmp al,40                    ;test for off side 1
891 0BA0 7208          0BAA    jb side_ok            ;if single sided then leap
892 0BA2 B301          mov bl,1                     ;else get a head 1
893 0BA4 B44F          mov ah,79                     ;else get back track bias
894 0BA6 2AE0          sub ah,al                     ;compute new track number
895 0BA8 86E0          xchg ah,al                    ;and get it into al
896
897 0BAA 8B1EEF0C      side_ok:    mov d_a_head,bl        ;set up the head
898 0BAE 0203          add bl,bl                      ;multiply by 2
899 0BB0 020B          add bl,bl                      ;multiply by 4
900 0BB2 A2EE0C        mov d_a_cylinder,al          ;set up the track number
901 0BB5 0A5E0E        or bl,drive                  ;bits 0,1 are drive number
902 0BB8 8B1EEB0C      mov d_a_drive,bl            ;set up disk number
903 0BBC 8B450A        mov ax,sector                ;get sector off stack
904 0BBF 40            inc ax                        ;sectors are relative to 1
905 0BC0 A2F00C        mov d_a_record,al            ;and set it up
906 0BC3 0205030D      add al,num_sec               ;compute new end of track
907 0BC7 FEC8          dec al
908 0BC9 A2F20C        mov d_a_eot,al               ;save it
909 0BCC E85000        0C2F    call fdc_command_put    ;send the command block
910 0BCF F4            hlt                          ;wait for FDC interrupt
911 0BD0 E94600        0C19    jmp sense_interrupt    ;required after SEEK command
912                      ;ret

```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____

```

913
914
915 dma_setup:
916 ;-----
917 ; entry: DMA_MODE_STORAGE, DMA_LOW16, DMA_HIGH4 set up
918 ; exit: none
919
920 ; Set the DMA device up for a read/write operation.
921 ; The current DMA command word must in DMA_MODE_STORAGE.
922 ; DMA_LOW16 and DMA_HIGH4 are the twenty bit starting address.
923 ; The read/write operation cannot cross a physical 64K boundary.
924
925 0BD3 E60C out dma_cbpf,al ;reset the byte pointer
926
927 0BD5 A0020D mov al,dma_mode_storage ;get the mode byte
928 0BD8 E603 out dma_mode_reg,al ;set the mode
929 0BD4 A1050D mov ax,dma_low16 ;low 16 bits of 20 bit DMA address
930 0BD0 E604 out dma_c2_address,al ;send low 8 bits
931 0BD5 8AC4 mov al,ah
932 0BE1 E604 out dma_c2_address,al ;send next 8 bits
933 0BE3 A0070D mov al,dma_high4 ;high 4 bits of 20 bit DMA address
934 0BE6 E681 out dma_page_fdc,al
935 0BE8 33C0 xor ax,ax
936 0BEA 8A26030D mov ah,num_sec ;shl num_sec,8 (* 256)
937 0BE4 D0E4 shl ah,1 ;*512
938 0BF0 48 dec ax ;0 relative
939 0BF1 E605 out dma_c2_count,al ;set up the low byte
940 0BF3 8AC4 mov al,ah ;get the low byte
941 0BF5 E605 out dma_c2_count,al ;and the high byte
942 0BF7 8002 mov al,dma_bmsk_fdc ;get the binary channel mask
943 0BF9 E60A out dma_bmsk_reg,al ;enable the disk channel
944 0BF8 C3 ret
945
946 fdc_read_write:
947 ;-----
948
949 ; entry: DMA device set up, head positioned on correct
950 ; track
951 ; exit: zero flag set if successful
952
953 ; Send read or write command to 8272 FDC
954
955 0BFC C606EB0C09 mov disk_arguments,9 ;9 byte command for read or write
956 0C01 A0010D mov al,fdc_rw_cmdu ;get read or write command
957 0C04 A2EC0C mov d_a_command,al ;put it in the command string
958 0C07 E82500 call fdc_command_put ;send the command to the FDC
959 0C0A F4 hlt ;wait for FDC interrupt
960 0C0B C606F50C07 mov disk_results,7 ;7 byte result transfer
961 0C10 E83400 call fdc_status_get ;get the result bytes
962 0C13 F605F60CC0 test d_r_st0,0C0H ;test status register 0
963 0C18 C3 ret ;return zero flag set on success
964
965 sense_interrupt:

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

966
967
968 ;-----
969 ; Sense interrupt command on the FDC. It is called
970 ; after a recal or a seek to test for seek complete.
971 0C19 C606E80C01      mov disk_arguments,1      ;only one byte of command
972 0C1E C606EC0C08      mov d_a_command,fdc_si_cmd    ;sense interrupt command
973 0C23 E80900      0C2F call fdc_command_put    ;send the command to the FDC
974 0C26 C606F50C02      mov disk_results,2        ;2 bytes are returned
975 0C28 E81900      0C47 call fdc_status_get    ;get the 2 result bytes
976 0C2E C3          ret
977
978 fdc_command_put:
979 ;-----
980 ; entry: DISK_ARGUMENT array set up
981 ; exit: none
982
983 ; Send the command block in the DISK_ARGUMENTS table to
984 ; the 8272 FDC.
985 ; The number of commands to write to the FDC is the
986 ; first item in the table.
987
988 0C2F BAF403      mov dx,fdc_stat      ;point to the i/o port
989 0C32 BEEB0C      mov si,offset disk_arguments ;point to the table of arguments
990 0C35 FC          cld                  ;make sure we go forward
991 0C36 AC          lodsb                ;get the length of the arguments table
992 0C37 8AC3        mov cl,al           ;get it into the count register
993 0C39 2AE0        sub ch,ch           ;zero the high byte
994
995 fdc_command_loop:
996 0C3B EC          in al,dx             ;get the current control byte
997 0C3C A880        test al,fdc_ready    ;if not ok to send next byte
998 0C3E 74FB        jz fdc_command_loop ;then loop waiting
999 0C40 42          inc dx              ;point at the data port
1000 0C41 AC          lodsb               ;else get the byte to send
1001 0C42 EE          out dx,al           ;send it
1002 0C43 4A          dec dx              ;point back at the status port
1003 0C44 E2F5        loop fdc_command_loop ;if not last byte then loop
1004 0C46 C3          ret                ;else were all done
1005
1006 fdc_status_get:
1007 ;-----
1008 ; entry: number of results in 1st byte of DISK_RESULTS array
1009 ; exit: none
1010
1011 ; Get the status information from the 8272
1012 ; FDC and place them in the table at DISK_RESULTS.
1013 ; The first byte in the table is the number of results
1014 ; to read from the FDC
1015
1016 0C47 06          push es              ;save UOA
1017 0C48 BAF403      mov dx,fdc_stat      ;point at the status port
1018 0C4B 8CD8        mov ax,ds           ;get our data segment

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

1019
1020 0C4D 8EC0      mov es,ax      ;into the extra segment
1021 0C4F BFF50C    mov di,offset disk_results + 1 ;point to where to put the data
1022 0C52 FC        cld          ;make sure we go forward
1023 0C53 8A0EF50C  mov cl,disk_results ;fetch the number of expected results
1024 0C57 2AE3      sub ch,ch     ;zero the high byte
1025
1026      fdc_status_loop:
1027 0C59 EC        in al,dx      ;get the current control byte
1028 0C5A A880      test al,fdc_ready ;if not ok to read next byte
1029 0C5C 74FB      jz fdc_status_loop ;then loop waiting
1030 0C5E 42        inc dx       ;point at the data port
1031 0C5F EC        in al,dx     ;get the byte
1032 0C60 AA        stosb        ;put it in the structure
1033 0C61 4A        dec dx       ;point back at the status port
1034 0C62 E2F5      loop fdc_status_loop ;if not last then loop
1035 0C64 07        pop es       ;restore UDA
1036 0C65 C3      ret          ;else return
1037
1038
1039
1040      ;          INIT ROUTINE
1041
1042      ;*****
1043      ;*
1044      ;*      8259 PROGRAMMABLE INTERRUPT CONTROLLER COMMANDS
1045      ;*      AND PORTS
1046      ;*
1047      ;*****
1048
1049 0020      pic_even_port      equ      020h      ;port 0
1050 0021      pic_odd_port      equ      021h      ;bit 0 is AO of 8259 instructions
1051
1052 0020      pic_nseoi         equ      020h      ;non specific end of interrupt
1053
1054      ;*****
1055      ;*
1056      ;*      BIOS INITIALIZATION ROUTINE
1057      ;*
1058      ;*****
1059
1060      ;track buffer
1061 0012      memory_size      equ      12h      ;ROM interrupt to get memory size
1062 0011      equipment        equ      11h      ;ROM interrupt to get equipment word
1063
1064      ;
1065      ;      The following routine is used to initialize any required
1066      ;      data areas and alter any peripheral chip programming when
1067      ;      starting up CCP/M-86. This code is called once from the
1068      ;      SUP(ERVISOR) after calling the SUP has called the RTN,
1069      ;      RTN, CIO, MEM, BIOS initialization routines and before the
1070      ;      SUP has created the RSP processes.
1071      ;      This code can be placed in an BIOS data area if the BIOS is
1072      ;      8080 model (mixed code and data). Usually, overlaying the

```

CCP/M-86 2-D V.2
 COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # CC-104

```

1072
1073 ; initialization code with a data area is done after the BIOS
1074 ; has been debugged.
1075
1076
1077 ;====
1078 ;====
1079 init: ;arrive here from the JNP
1080 ;==== ;at 0 in BIOS code segment
1081 ;====
1082
1083 0C66 FAFD cli l cld ;Supervisor restores DS,ES and int
1084 ;224 after on INIT call
1085
1086 ; Disk I/O initialization
1087
1088
1089 set_up_interrupts:
1090 0C68 B0B0 mov al,10111101b ;enable diskette and keyboard
1091 0C6A E621 out pic_odd_port,al ;interrupts only
1092 ;and mask off the rest
1093
1094 ; mov disk_arguments,3 ;send specify command to the
1095 ; mov d_a_command,fdc_spec_cmd ;FDC (Intel 3272 or NEC 765)
1096 ; mov d_a_drive,0d0h ;step rate=3ms, head unload=0ms
1097 ; mov d_a_cylinder,20h ;head load=2ms, DMA mode true
1098 ; call fdc_command_put ;head unload and head load times
1099 ;are meaningless on mini floppies
1100 ;where the head is loaded
1101 ;when the motor is turned on
1102
1103 0C6C 8CDB mov ax,ds
1104 0C6E 01064E0D add data_buf_seg,ax ;fix up data buffer segment
1105
1106 0C72 0633C0 push es l xor ax,ax
1107 0C75 8EC0 xov es,ax
1108 0C77 BF3800 mov di,disk_interrupt*4 ;point at vector location
1109 0C7A B8EC09 mov ax,offset i_disk
1110 0C7D AB stow ;save and inc DI
1111 0C7E 8CC8 mov ax,cs
1112 0C80 AB stow
1113 0C81 07 pop es
1114
1115 set_up_video:
1116 0C82 BAB403 mov dx,bw_card ;get the video chip port
1117 0C85 BEB50C mov si,offset bw_table ;initialization commands
1118 0C88 B91000 mov cx,length bw_table ;how many commands
1119 0C8B E81300 call video_init ;send commands to port
1120 ;color board is similar ...
1121 0C8E 06 push es
1122 0C8F BF00B0 mov di,bw_video_seg ;clear the video RAM
1123 0C92 8EC7 mov es,di
1124 0C94 33FF xor di,di

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```

1125
1126 0C96 B9D007      mov cx,screen_siz + columns_per_screen
1127                                     ;screen length in words plus
1128                                     ;status line
1129 0C99 B82007      mov ax,0720h      ;7 = default attribute
1130                                     ;20 = ASCII space
1131 0C9C F3AB        rep stosw
1132 0C9E 07          pop ax
1133
1134 0C9F FB          sti
1135 0CA0 CB          retf              ;initialization done
1136
1137 ;-----
1138 video_init:
1139 ;-----
1140 0CA1 320B        xor bl,bl
1141 video_init_1:
1142 0CA3 8AC3FEC3    mov al,bl | inc bl
1143 0CA7 EE42        out dx,al | inc dx
1144 0CA9 ACEE        lodsb | out dx,al
1145 0CAB 4A          dec dx |
1146 0CAC E2F5        loop video_init_1
1147 0CAE 83C204      add dx,4
1148 0CB1 B029        mov al,video_on
1149 0CB3 EE          out dx,al
1150 0CB4 C3          ret
1151
1152
1153 0CB5             last_code_offset equ offset $
1154 DSEG
1155 ORG last_code_offset
1156
1157 ; Loader BIOS data
1158
1159 0CB5 6150520F1906  bw_table      db 61h,50h,52h,0fh,19h,06h,19h,19h,02h,0dh,0bh,0ch,0,0,0,0
1160 19190200080C
1161 00000000
1162
1163 function_table:
1164 0CC5 EB09        dw io_ret      ; 0 console status
1165 0CC7 EB09        dw io_ret      ; 1 console input
1166 0CC9 F109        dw io_conout   ; 2 console output
1167 0CCB EB09        dw io_ret      ; 3 list status
1168 0CCD EB09        dw io_ret      ; 4 list output
1169 0CCF EB09        dw io_ret      ; 5 auxillary input
1170 0CD1 EB09        dw io_ret      ; 6 auxillary out
1171 0CD3 EB09        dw io_ret      ; 7 switch screen
1172 0CD5 EB09        dw io_ret      ; 8 update or print new status
1173 0CD7 300A        dw io_sel disk ; 9 select disk
1174 0CD9 380A        dw io_read     ;10 read logical sector
1175 0CDB EB09        dw io_ret      ;11 write logical sector
1176 0CDD EB09        dw io_ret      ;12 flush buffers
1177 0CDF EB09        dw io_ret      ;13 poll device

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

```

1178
1179
1180
1181                                ;short screen structure
1182                                ;for IO_CONOUT
1183
1184    OCE1 000000000000    ss0      dw      0,0,0    ;cursor, escape, screen_size
1185    OCE7 0000            db      0,0    ;row,column
1186
1187
1188                                ; Floppy Disk Driver data
1189
1190
1191                                max_retries equ      3
1192    OCE9 00            retries db      0    ;retry counter
1193                                max_recals equ      5
1194    OCEA 00            recals db      0
1195
1196                                ; The following 2 tables are used to issue commands to,
1197                                ; and read results from the 8272 FDC. The first entry
1198                                ; in each table is the number of bytes to send or receive
1199                                ; from the device
1200
1201    OCEB 00            disk_arguments db      0    ;number of arguments to send
1202    OCEC 00            d_a_command db      0    ;command read/write
1203    OCED 00            d_a_drive db      0    ;drive select & head select
1204    OCEE 00            d_a_cylinder db      0    ;cylinder to read/write
1205    OCEF 00            d_a_head db      0    ;head
1206    OCF0 00            d_a_record db      0    ;sector
1207    OCF1 02            d_a_number db      2    ;magic number for 512 bytes/sector
1208    OCF2 08            d_a_eot db      sectors_per_track ;end of track sector number
1209    OCF3 2A            d_a_gpl db      02ah    ;inter sector gap length
1210    OCF4 FF            d_a_dtl db      0ffh    ;data length
1211
1212    OCF5 00            disk_results db      0    ;number of bytes to read
1213    OCF6 00            d_r_st0 db      0    ;status byte 0
1214    OCF7 00            d_r_st1 db      0    ;status byte 1
1215    OCF8 00            d_r_st2 db      0    ;status byte 2
1216    OCF9 00            d_r_cylinder db      0    ;cylinder we are on now
1217    OCFA 00            d_r_head db      0
1218    OCFB 00            d_r_record db      0
1219    OCFC 00            d_r_number db      0    ;number of sectors read
1220
1221    OCFD 00            f_a_bytes db      0
1222    OCFE 00            f_a_sectors db      0
1223    OCFF 00            f_a_gap db      0
1224    0D00 00            f_a_filler db      0
1225
1226    0D01 66            fdc_rw_cmd db      fdc_read_cmd ;read or write command
1227    0D02 46            dma_mode_storage db      dma_mode_read_fdc
1228
1229    0D03 01            num_sec db      1    ;num sectors to read/write
1230                                ;in one call to controller

```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER # _____

```

1231
1232
1233 0004      track_mcnt      rc      1      ;multi sector count on
1234                      ;current track
1235 0005      dma_low16      rw      1      ;20 bit address storage
1236 0007      dma_high4      rb      1
1237
1238 0008 00      motor_flags      db      0      ;last state of the motor bits
1239
1240      ;      Disk Parameter Header
1241
1242 0009 0000      dph0      dw      0      ;translate table
1243 0008 000000000000      dw      0,0,0      ;scratch area
1244 0011 1000      dw      dph0      ;disk parm block
1245 0013 0000      dw      0      ;check
1246 0015 0000      dw      0      ;alloc vectors
1247 0017 2E0D      dw      dir_bcb_hdr      ;dir buff cntnl blk
1248 0019 4100      dw      data_bcb_hdr      ;data buff cntnl blk
1249 0018 0000      dw      0      ;hash table segment
1250
1251      ;      Disk Parameter Block
1252
1253 001D 0800      dpb0      dw      8      ;sectors per track
1254 001F 03      db      3      ;block shift
1255 0020 07      db      7      ;block mask
1256 0021 00      db      0      ;extnt mask
1257 0022 9800      dw      155      ;disk size in 1k blocks
1258                      ;less offset track(s)
1259 0024 3F00      dw      63      ;directory max
1260 0026 C0      db      11000000b      ;alloc0
1261 0027 00      db      0      ;alloc1
1262 0028 1000      dw      16      ;check size
1263 002A 0100      dw      1      ;offset
1264 002C 02      db      2      ;phys sec shift
1265 002D 03      db      3      ;phys sec mask
1266
1267 002E 310D      dir_bcb_hdr      dw      dir_bcb
1268 0030 01      db      1
1269
1270 0031 FF      dir_bcb      db      0ffh      ;drive
1271 0032 000000      db      0,0,0      ;record
1272 0035 0000      db      0,0      ;wflg,seg
1273 0037 00000000      dw      0,0      ;track,sector
1274 0038 700E      dir_buf_off      dw      dir_buf      ;buffer offset
1275 003D 00000000      dw      0,0      ;link,pdadr
1276
1277 0041 440D      data_bcb_hdr      dw      data_bcb
1278 0043 01      db      1
1279
1280 0044 FF      data_bcb      db      0ffh      ;drive
1281 0045 000000      db      0,0,0      ;record
1282 0048 0000      db      0,0      ;wflg,seg
1283 004A 00000000      dw      0,0      ;track,sector

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```

1284
1285 004E 2701      data_buf_seg  dw      data_buf;buffer segment, fixed up by LETOS init:
1286 0050 00000000      dw      0,0      ;link,pdadr
1287
1288
1289      ;      Loader Program data
1290
1291 0054 4343502F4D2D      signon      db      "CCP/M-86 Loader 2.04"
1292      3836204C6F61
1293      64657220322E
1294      3024
1295 0068 00044E6F2043      nofile      db      cr,lf,"No CCPM.SYS File On Disk!"
1296      4350402E5359
1297      532046696C65
1298      204F6E204469
1299      736824
1300 0083 000A4572726F      rerr      db      cr,lf,"Error Reading CCPM.SYS!"
1301      722052656164
1302      696E67204343
1303      50402E535953
1304      24
1305 009C 000A436F6465      csegment      db      cr,lf,"Code Paragraph = $"
1306      205061726167
1307      726170682030
1308      2024
1309 00B0 000A44617461      usegment      db      cr,lf,"Data Paragraph = $"
1310      205061726167
1311      726170682030
1312      2024
1313 00C4 000A24      crlf      db      cr,lf,"$"
1314
1315 00C7 014343504020      ccpmfcb db      1,"CCPM      ", "SYS",0,0,0,0
1316      202020535953
1317      00000000
1318 00D7 000000000000      db      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1319      000000000000
1320      00000000
1321
1322 00E7 00      db      0
1323
1324
1325      ;      Note the Loader Program and Loader BIOS uninitialized
1326      ;      data areas are combined to save space.
1327
1328      ;      Unitialized data for Loader Program
1329
1330
1331      ;read first CMD header of
1332      ;CCPM.SYS file here
1332 00E8      ctype      rb      1      ;type
1333 00E9      clen      rw      1      ;length
1334 00EB      cldseg      rw      1      ;abs
1335 00ED      cmin      rw      1      ;minimum
1336 00EF      cmax      rw      1      ;maximum

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

1337
1338   0DF1          dtype    rb      1
1339   0DF2          dlen     rw      1
1340   0DF4          dldseq   rw      1
1341   0DF6          dmin     rw      1
1342   0DF8          dmax     rw      1
1343
1344          org      offset ctype
1345   0DE8          sect1    rb      128          ;reserve space for the sector
1346
1347
1348          ;          Unitialized data for Loader BIOS
1349
1350
1351   0E70          dir_buf    equ      ((offset $) + 15) and 0fff0h
1352
1353          ;          Sector for reads that span 64K boundaries;
1354          ;          this sector can not cross a 64K boundary
1355
1356   1070          local_buf  equ      dir_buf + 512
1357
1358   0127          data_buf   equ      (local_buf + 512)/16
1359
1360
1361  END OF ASSEMBLY.  NUMBER OF ERRORS:  0.  USE FACTOR: 16%

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

BWCARD	0384 N	405#	1116				
BWTABLE	0CB5 V	1117	1118	1159#			
BWVIDEOSEG	3000 N	348	430#	1122			
BWVIDEOSTATUSLIN	0F00 N	432#					
BYTESPERDISK	8000 W	521#					
BYTESPERSECTOR	0200 N	517#	519	756			
BYTESPERTRACK	1000 N	519#	521				
CARRIAGERETURN	0A1E L	344	356	364#			
CCPMFCB	0DC7 V	99	115	153	1315#		
CLDSEG	0DEB V	138	149	1334#			
CLEN	0DE9 V	165	168	1333#			
CMAx	0DEF V	1336#					
CMIN	0DED V	1335#					
CUDETYPE	0U01 N	57#					
COLUMNSPERSCREEN	0050 N	354	386	427#	428	1126	
COMPMA	0B03 L	691	726	764	787#		
CR	000D N	50#	343	1295	1300	1305	1309 1313
CRLF	0DC4 V	207	1313#				
CS	SKEG V	168	1111				
CSEGMENT	0D9C V	136	1305#				
CTYPE	0DE8 V	1332#	1344				
CURSOREND	0006 N	411#					
CURSORMI	000E N	414#					
CURSLOW	000F N	415#					
CURSORSTART	000A N	410#					
DACOMMAND	0CEC V	873	887	957	972	1202#	
DACYLINDER	0CEE V	909	1204#				
DADRIVE	0CED V	875	902	1203#			
DADTL	0CF4 V	821	1210#				
DAEOT	0CF2 V	819	908	1208#			
DAGPL	0CF3 V	820	1209#				
DAHEAD	0CEF V	897	1205#				
DANUMBER	0CF1 V	818	1207#				
DARECORD	0CF0 V	905	1206#				
DATABCB	0D44 V	1277	1280#				
DATABCBHDR	0D41 V	1248	1277#				
DATABUF	0127 N	1285	1358#				
DATABUFSEG	0D4E V	1104	1285#				
DATATYPE	0002 N	50#					
DEBUGGERINTERRUPT	00E1 N	286#					
DIRBCB	0D31 V	1267	1270#				
DIRBCBHDK	0D2E V	1247	1267#				
DIRBUF	0E70 N	1274	1351#	1356			
DIRBUFOFF	0D38 V	1274#					
DISKARGUMENTS	0CEB V	872	886	955	971	989	1201#
DISKINTERRUPT	000E N	284#	1108				
DISKRESULTS	0CF5 V	960	974	1021	1023	1212#	
DISPLAYSTARHI	000C N	412#					
DISPLAYSTARTLOW	000D N	413#					
DLDSEG	0DF4 V	142	167	1340#			
DLEN	0DF2 V	1339#					
DMABMSKFOC	0002 N	496#	942				
DMABMSKREG	000A N	477#	943				

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 F. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

DMAC0ADDRESS	0000 N	466#																		
DMAC0COUNT	0001 N	467#																		
DMAC1ADDRESS	0002 N	468#																		
DMAC1COUNT	0003 N	469#																		
DMAC2ADDRESS	0004 N	470#	930	932																
DMAC2COUNT	0005 N	471#	939	941																
DMAC3ADDRESS	0006 N	472#																		
DMAC3COUNT	0007 N	473#																		
DMACBPF	000C N	479#	925																	
DMACLEAR	000D N	491#																		
DMACHDREG	0008 N	475#																		
DMAHIGH4	0D07 V	805	933	1236#																
DMALOW16	0D05 V	804	929	1235#																
DMAMASKREG	000F N	482#																		
DMAMODERLADFDC	0046 N	494#	1227																	
DMAMODEREG	000B N	478#	928																	
DMAMODESTORAGE	0D02 V	927	1227#																	
DMAMODEWRITEFDC	004A N	493#																		
DMADFF	0006 V	603#	690	721	752	758	763	775												
DMAPAGEC1	0080 N	484#																		
DMAPAGEC3	0082 N	486#																		
DMAPAGEFDC	0081 N	485#	934																	
DMAREQUIREG	0009 N	476#																		
DMASEG	0008 V	602#	689	755	762															
DMASETUP	0B03 L	829	915#																	
DMASIATREG	0008 N	474#	475	476	477	478	479	480	481	482										
DMATEMPREG	000D N	480#																		
DMAX	0DF8 V	1342#																		
DMIN	0DF6 V	1341#																		
DONE	098D L	157	162#																	
DPB	0008 N	610#																		
DPB0	0D1D V	1244	1253#																	
DPH0	0D09 V	637	1242#																	
DRCYLINDER	0CF9 V	1216#																		
DRHEAD	0CFA V	1217#																		
DRIVE	000E V	598#	850	855	874	901														
DRNUMBER	0CFC V	1219#																		
DRRECORD	0CFB V	1218#																		
DRST0	0CF6 V	962	1213#																	
DRST1	0CF7 V	1214#																		
DRST2	0CF8 V	1215#																		
DS	SREG V	105	167	725	1018	1103														
DSEGMENT	0DB0 V	140	1309#																	
DTYPE	0DF1 V	1338#																		
ENTRY	09DD L	81	296#																	
EQUIPMENT	0011 N	1062#																		
ES	SREG V	347	349	353	755	755	759	1016	1020	1035	1106									
		1107	1113	1121	1123	1132														
FABYTES	0CFD V	1221#																		
FAPILLER	0D00 V	1224#																		
FAGAP	0CFE V	1223#																		
FALSE	0000 N	47#	48																	
FASECTORS	0CFE V	1222#																		

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

FBYTES	0002 N	555#							
FDCCOMMANDLOOP	0C3B L	995#	998	1003					
FDCCOMMANDPUT	0C2F L	876	909	958	973	978#			
FDCDATA	03F5 N	528#							
FDCFORMATCMD	0040 N	544#							
FDCNUMOTOR	00FC N	540#							
FDCON	000C N	539#	855						
FDCPORT	03F2 N	529#	858						
FDCREADCMD	0066 N	541#	1226						
FDCREADWRITE	0BFC L	830	946#						
FDCREADY	0080 N	550#	597	1028					
FDCRECALCMD	0007 N	546#	873						
FDCRWCMD	0001 V	956	1226#						
FDCSEEKCMD	000F N	545#	887						
FDCSICMD	0008 N	547#	972						
FDCSPEC1	00CF N	552#							
FDCSPEC2	0003 N	553#							
FDCSPEC3CMD	0003 N	548#							
FDCSTAT	03F4 N	527#	528	988	1017				
FDCSTATUSGET	0C47 L	961	975	1006#					
FDCSTATUSLOOP	0C59 L	1026#	1029	1034					
FDCWRITECMD	0045 N	543#							
FFILLER	00E5 N	558#							
FGAP	003A N	557#							
FLPEND64K	0A79 L	696	699#						
FLPEND64KOK	0A92 L	712	715#						
FLPID	0A47 L	658	665#	783					
FLPLCALDONE	0ACD L	761#							
FLPPGOK	0AD6 L	697	766#						
FLPPHYSIO	0B18 L	711	744	770	809#				
FLPRET	0B02 L	713	745	771	780	784#			
FLPRWLOCAL	0AA1 L	707	723#						
FLPTRACK	0A59 L	675	678#						
FSECTORS	0008 N	556#							
FUNCTIONTABLE	0CC5 L	314	1163#						
IDISK	09EC L	322#	1109						
INCCOL	0A18 L	355	358#						
INIT	0C66 L	80	1079#						
IUCONOUT	09F1 L	203	329#	1166					
IDREAD	0A38 L	642#	1174						
IDRET	09EB L	318#	1164	1165	1167	1168	1169	1170	1171
		1176	1177					1172	1175
IUSELDSK	0A30 L	623#	1173						
IREADRET	0A46 L	660	662#						
KEYBOARDINTERRUP	0009 N	283#							
LASTCODEOFFSET	0CB5 N	1153#	1155						
LD8DOSOFFSET	0000 N	53#							
LDKOFFSET	0900 N	55#	78						
LDUSR	0909 V	84#	96						
LF	000A N	51#	345	1295	1300	1305	1309	1313	
LHEX	09A7 L	185#	190						
LIGHTPENHI	0010 N	416#							
LIGHTPENLOW	0011 N	417#							

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

LINEFEED	0A28 L	346	357	376#					
LOADP	090A L	82	87#						
LOCALBUF	1070 H	724	753	1356#	1358				
MAXRECALLS	0005 H	824	1193#						
MAXRETRIES	0003 H	826	1191#						
MCNT	000F V	599#	673	676	679	779			
MEMORYSIZE	0012 N	1061#							
MOTORANDRECAL	0B59 L	823	841#						
MOTORFLAGS	0D08 V	852	857	1238#					
MOTORUNDONE	0B77 L	854	862#						
MSG	09C8 L	94	137	141	178	209#			
MSG1	09CD L	212#	220						
MSG2	09DC L	215	221#						
NOFILE	0D68 V	102	1295#						
NUMSEC	0D03 V	709	717	743	769	773	906	936	1229#
NXITRACK	0AF0 L	768	777#						
OPENF	000F N	63#	100						
OSINTERRUPT	00E0 N	285#							
P10	098F L	194	197#						
PCRLF	09C8 L	163	205#						
PERR	092A L	101	104#						
PHEX	09A4 L	139	143	182#					
PHYSRET	0B58 L	831	838#						
PTCEVENPORT	0020 N	325	1049#						
PICNSEOI	0020 H	324	1052#						
PICODDPORT	0021 N	1050#	1091						
PNIB	09B5 L	188	192#						
PRN	09C1 L	196	198#						
PSH	000F N	612#							
PUTCHAR	09C3 L	200#	217						
RBYTES	0002 N	560#	818						
ROIL	00FF N	563#	821						
READIT1	096E L	150#	158						
READSF	0014 N	64#	116	154					
RECAL	0B78 L	834	861	865#					
RECALLLOOP	0B37 L	825#	836						
RECALLS	0CEA V	824	835	1194#					
RERR	0D83 V	159	1300#						
RETRIES	0CE9 V	826	832	1192#					
RETRYLOOP	0B3C L	827#	833						
RGAP	002A N	562#	820						
ROWSPERSCREEN	0018 N	426#	428						
RSECTORS	0008 N	561#	819						
SCREENSIZ	0780 N	428#	432	1126					
SECI	0DE8 V	109	1345#						
SECTOR	000A V	601#	671	718	746	782	903		
SECTURSPERTRACK	0008 N	518#	519	670	1208				
SEEK	0B8F L	828	881#						
SELOSKF	000E N	62#							
SELRET	0A3A L	636	638#						
SENSEINTERRUPT	0C19 L	878	911	965#					
SETDMA8F	0033 N	68#	106	152					
SETDMAF	001A N	65#	110	148					

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

SETMULTCNTF	0020 N	67#	113	146			
SETUPINTERRUPTS	0C68 L	1089#					
SETUPVIDEO	0C82 L	1115#					
SETUSERF	0020 N	60#	97				
SIDEOK	0BAA L	891	896#				
SIGNON	0D54 V	95	1291#				
SPT	0000 N	611#					
SS	SREG V						
SSU	0CE1 V	341	1184#				
SSCOLUMN	0007 N	354	359	370	455#		
SSCURSOR	0000 N	342	360	372	386	451#	452
SSESCAPE	0002 N	452#	453				
SSROW	0006 N	385	454#	455			
SSSCREENSEG	0004 N	453#	454				
STOP	099F L	103	161	176#			
TRKINTERRUPT	0008 N	282#					
TRACK	000C V	600#	781	889			
TRACKCNT	0D04 V	681	710	746	767	1233#	
TRACKSPERDISK	0028 N	520#	521				
TRUE	FFFF N	48#					
VIDEOINIT	0CA1 L	1119	1138#				
VIDEOINITL	0CA3 L	1141#	1146				
VIDEOOFF	0021 N	408#					
VIDEOON	0029 N	407#	1148				
XLT	0000 N	609#					

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____