

```

*****
*
* CP/M VERS 2.2 COLD START LOADER.
*
* WRITTEN BY BOBBY DALE GIFFORD.
* 3/17/80
*
* THE FOLLOWING ROUTINES WILL BOOT CP/M FROM THE DISK JOCKEY
* 2D (ALL REVS AND MODELS), OR FROM THE DISK JOCKEY HARD
* DISK CONTROLLER.
*
* THE COLD BOOT LOADER (SECTOR 1, TRACK 0) IS LOADED
* INTO THE RAM OF THE CONTROLLER BY THE COLD BOOT ROUTINE OF
* THE FIRMWARE. THE FIRST THING THE BOOT DOES IS TO LOAD INTO
* THE CONTROLLER RAM, A VERSION OF THE DISK JOCKEY 2D FIRMWARE.
* FROM THEN ON, ALL CALLS TO THE FIRMWARE WILL ACTUALLY BE
* DIRECTED TO THE DISK JOCKEY RAM. THE NEXT PROCESS IS TO LOAD
* IN A BOOT ROUTINE WHICH CAN LOAD IN ALL OF CP/M. THIS IS
* DONE BY DETERMINING THE SIZE OF THE SECTORS ON TRACK 1, AND
* USING THIS INFORMATION TO LOAD IN THE PROPER BOOT INTO 80H.
*
* THE FOLLOWING TABLES EXPLAIN THE ORDER OF SECTOR LOADING FOR
* EACH OF THE DIFFERENT SECTOR SIZES. AN ENTRY OF -----
* REPRESENTS A WRAP BACK AROUND (NEGATIVE DMA ADJUSTMENT).
* AN ENTRY FLAGGED WITH ASTRICKS REPRESENTS A PARTIAL SECTOR
* LOAD.

```

ALL SECTOR SIZES:

TRACK 0	SECTOR 1	ADDRESS
0	8	11 2C00H
0	10	13 2D00H
0	12	15 2E00H
0	14	17 2F00H
0	16	19 3000H
0	18	21 2780H
0	20	23 2880H
0	22	25 2980H
0	24	27 2A80H
0	26	29 2B80H
0	9	12 2C80H
0	11	14 2D80H
0	13	16 2E80H
0	15	18 2F80H
0	17	20 2700H
0	19	22 2800H
0	21	24 2900H
0	23	26 2A00H
0	25	28 2B00H

SINGLE DENSITY
TRACK 0
172 BYTE SECTORS

THE FOLLOWING DEPEND ON THE SECTOR SIZE, ALL SECTORS ARE FROM TRACK 1.

256	512	1024	
SEC	ADDRESS	SEC	ADDRESS
1	2C00H start	1	2C00H start
3	2E00H	3	3000H
		1	start 2C00H
		3	3400H

ABOUT & PRN

FLOPPIES A, B

HD C, D, E

1 LOAD DJ FIRMWARE

2 DETERMINE SIZE OF SECTORS ON TRACK 1

3 LOAD PROPER BOOT INTO 80H

4 LOAD CP/M

2/1/82

NOTE: ABOUT & 5B, ASM/PRN IS THIS EXCEPT
H/D IS "FAST" 9/8/82

```

* 5 3000H      5 3400H      5 3C00H      *
* 7 3200H      7 3800H      ** 7 47FF end 4400H      *
* 9 3400H      9 3C00H      -----      *
* 11 3600H     11 4000H     2 2 3000H      *
* 13 3800H     13 4400H     4 4 3800H      *
* 15 3A00H     -----      6 6 4000H      *
* 17 3C00H      2 2 2E00H      *
* 19 3E00H      4 3200H      *
* 21 4000H      6 3600H      *
* 23 4200H      8 3A00H      *
* 25 4400H      10 3E00H      *
* -----      12 4200H      *
* 2 2D00H      DISCUS M10, M20, M26      *
* 4 2F00H      3 E400H (IF 2D IS PRESENT)      *
* 6 3100H      4 E600H (IF 2D IS PRESENT)      *
* 8 3300H      ** 20 16 4500H      *
* 10 3500H     5 2700H      *
* 12 3700H     6 2900H      *
* 14 3900H     7 2B00H      *
* 16 3B00H     8 2D00H      *
* 18 3D00H     9 2F00H      *
* 20 3F00H     10 3100H      *
* 22 4100H     11 3300H      *
* 24 4300H     12 3500H      *
* 26 4500H     13 3700H      *
* -----      14 3900H      *
* -----      15 3B00H      *
* -----      16 3D00H      *
* -----      17 3E00H      *
* -----      18 4100H      *
* -----      19 4300H      *
* -----      *

```

5 3C00H *
 ** 7 47FF end 4400H *
 ----- *
 2 2 3000H *
 4 4 3800H *
 6 6 4000H *

512 bytes
 not used

I don't understand. address is E700!
 actual load address is E700?
 why load E400-E6FF?

END
 END
 END

END 46FF

256 bytes
 not used

TITLE '*** Cold Boot Loader for CP/M Ver. 2.2 ***'

```

0038 = MSIZE EQU 56 ;MEMORY SIZE OF TARGET CP/M
9000 = BIAS EQU (MSIZE-20)*1024 ;MEMORY OFFSET FROM 20K SYSTEM
B700 = CCP EQU 2700H+BIAS ;CONSOLE COMMAND PROCESSOR
CD00 = BIOS EQU CCP+1600H ;CBIOS ADDRESS
000A = RETRIES EQU 10 ;MAXIMUM # OF DISK RETRIES

```

LOOKS LIKE (COULD) MANUALLY CODE
 THESE FOR ADDING LESS THAN 1K TO TPA

 * THE FOLLOWING EQUATES SET UP THE RELATIONSHIP BETWEEN THE *
 * 2D FLOPPIES AND THE HARD DISK CONTROLLERS. *

```

0000 = FIRST EQU 0 ;0 = FLOPPIES ARE A-D DRIVES AND  

; HARD DISK ARE E-P  

; 1 = HARD DISKS ARE A-L DRIVES AND  

; FLOPPIES ARE M-P  

0001 = MAXHD EQU 1 ;SET TO NUMBER OF HARD DISKS

```

0002 = MAXFLOP EQU 2 ;SET TO NUMBER OF FLOPPIES

 *
 * THE FOLLOWING EQUATES ARE FOR THE DISKUS HARD DISK IF WANTED. *
 *

```

    IF      (MAXHD NE 0) AND FIRST ;WANT HARD DISK INCLUDED ?
    HDORG   EQU      50H           ;HARD DISK CONTROLLER
    HDSTAT  EQU      HDORG        ;HARD DISK STATUS
    HDCNTL  EQU      HDORG        ;HARD DISK CONTROL
    HDDATA  EQU      HDORG+3      ;HARD DISK DATA
    HDFUNC  EQU      HDORG+2      ;HARD DISK FUNCTION
    HDCMND  EQU      HDORG+1      ;HARD DISK COMMAND
    HDRESLT EQU      HDORG+1      ;HARD DISK RESULT
    RETRY   EQU      2           ;RETRY BIT OF RESULT
    TKZ     EQU      1           ;TRACK ZERO BIT OF STATUS
    OPDONE  EQU      2           ;OPERATION DONE BIT OF STATUS
    COMPLT  EQU      4           ;COMPLETE BIT OF STATUS
    TMOUT   EQU      8           ;TIME OUT BIT OF STATUS
    WFAULT  EQU      10H        ;WRITE FAULT BIT OF STATUS
    DRVRDY  EQU      20H        ;DRIVE READY BIT OF STATUS
    INDX    EQU      40H        ;INDEX BIT OF STATUS
    PSTEP   EQU      4           ;STEP BIT OF FUNCTION
    NSTEP   EQU      0FBH       ;STEP BIT MASK OF FUNCTION
    HDRLEN  EQU      4           ;SECTOR HEADER LENGTH
    SECLN   EQU      512        ;SECTOR DATA LENGTH
    WENABL  EQU      0FH        ;WRITE ENABLE
    WRESET  EQU      0BH        ;WRITE RESET OF FUNCTION
    SCENBL  EQU      5           ;CONTROLLER CONTROL
    DSKCLK  EQU      7           ;DISK CLOCK FOR CONTROL
    MDIR    EQU      0F7H       ;DIRECTION MASK FOR FUNCTION
    NULL    EQU      0FCH       ;NULL COMMAND
    IDBUFF  EQU      0           ;INITIALIZE DATA COMMAND
    ISBUFF  EQU      8           ;INITIALIZE HEADER COMMAND
    RSECT   EQU      1           ;READ SECTOR COMMAND
    WSECT   EQU      5           ;WRITE SECTOR COMMAND
    ENDIF
    
```

*only if boot FA HD
(HD=A)*

 *
 * COLD BOOT LOADER COMMON TO ALL SECTOR SIZES. *
 * THIS SECTOR IS LOADED INTO MEMORY AT E700H IN A STANDARD *
 * CONFIGURATION. IT IS RESPONSIBLE FOR READING MOST OF TRACK 0 *
 * INTO MEMORY ON COLD BOOTS. *
 *

```

E000 = ORIGIN EQU 0E000H
E400 = DJRAM EQU ORIGIN+400H
E400 = DJBOOT EQU DJRAM
    ENDIF
    
```

IF (MAXFLOP NE 0)

```

2900 =      OFFSETB EQU 900H-ORIGIN
           ELSE
           OFFSETB EQU 0
           ENDIF

           IF NOT FIRST
E02D =      PUTDEN EQU ORIGIN+02DH ;SET DENSITY ROUTINE ON DISK JOCKEY 2D
E012 =      PUTDMA EQU ORIGIN+12H ;DISK JOCKEY 2D SET DMA ADDRESS ROUTINE
E027 =      GETSTAT EQU ORIGIN+27H ;DISK STATUS ROUTINE ON DISK JOCKEY 2D
E00F =      PUTSEC EQU ORIGIN+0FH ;DISK JOCKEY 2D SET SECTOR ROUTINE
E00C =      PUTTRK EQU ORIGIN+0CH ;DISK JOCKEY 2D SET TRACK ROUTINE
E009 =      PUTHOM EQU ORIGIN+9H ;DISK JOCKEY 2D TRACK 0 SEEK
E015 =      DOREAD EQU ORIGIN+15H ;DISK JOCKEY 2D READ ROUTINE
E02A =      BOTERR EQU ORIGIN+2AH ;DISK JOCKEY 2D FLASH ERROR LIGHT ROUTINE

E000      ORG ORIGIN ;DISK JOCKEY 2D RAM

0700 #      DIFF SET ORIGIN+700H-$ ;OFFSET TO BOOT LOADER ADDRESS

E000 3180E7 LXI SP,STAC+DIFF
E003 3E06 FIRMLOD MVI A,6 ;PREVIOUS SECTOR #
E004 = NEWSEC EQU $-1
E005 3C INR A ;UPDATE SECTOR #
E006 3C INR A
E007 FE1B CPI 27 ;TEST IF ALL DONE
E009 CA00BC JZ CCP+500H
E00C DA11E7 JC NOWRAP+DIFF ;TEST IF WRAP AROUND
E00F D613 SUI 19
E011 3204E7 NOWRAP STA NEWSEC+DIFF ;SAVE THE UPDATED SECTOR #
E014 4F MOV C,A
E015 CD0FE0 CALL PUTSEC ;SET UP THE SECTOR
E018 2100BB LXI H,CCP+400H ;PREVIOUS DMA ADDRESS
E019 = NEWDMA EQU $-2
E01B 110001 LXI D,100H ;UPDATE DMA ADDRESS
E01E 19 DAD D
E01F 7C MOV A,H
E020 FEC0 CPI (CCP+980H)/100H
E022 DA32E7 JC NOWRP+DIFF
E025 C22EE7 JNZ WRP+DIFF
E028 7D MOV A,L
E029 FE80 CPI (CCP+980H) MOD 100H
E02B DA32E7 JC NOWRP+DIFF
E02E 1180F6 WRP LXI D,-980H
E031 19 DAD D
E032 2219E7 NOWRP SHLD NEWDMA+DIFF ;SAVE THE UPDATED DMA ADDRESS
E035 44 MOV B,H
E036 4D MOV C,L
E037 CD12E0 CALL PUTDMA ;SET UP THE NEW DMA ADDRESS
E03A 01000A LXI B,RETRIES*100H+0;MAXIMUM # OF ERRORS
E03D C5 FREAD PUSH B
E03E CD0CE0 CALL PUTTRK ;SET UP THE PROPER TRACK
E041 CD15E0 CALL DOREAD ;READ THE SECTOR
E044 C1 POP B
E045 D203E7 JNC FIRMLOD+DIFF ;CONTINUE IF NO ERROR
E048 05 DCR B
E049 C23DE7 JNZ FREAD+DIFF ;KEEP TRYING IF ERROR

```

10
51

```

E04C C32AE0      JMP      BOTERR      ;TO MANY ERRORS, FLASH THE LIGHT
E04F             DS        80H-($ MOD 80H)
E080 =          STAC     EQU      $
    
```

```

*****
*
* THE FOLLOWING EQUATES RELATE TO THE THINKER TOYS 2D CONTROLLER*
* IF THE CONTROLLER IS NON STANDARD (0E000H) ONLY THE ORIGIN *
* EQUATE NEED BE CHANGED. THIS VERSION OF THE CBIOS WILL WORK *
* WITH 2D CONTROLLER BOARDS REV 0, 1, 3, 3.1, 4. *
*
*****
    
```

```

E403 =          DJCIN   EQU      DJRAM+3H      ;DISK JOCKEY 2D CHARACTER INPUT ROUTINE
E406 =          DJCOUT  EQU      DJRAM+6H      ;DISK JOCKEY 2D CHARACTER OUTPUT ROUTINE
E409 =          DJHOME  EQU      DJRAM+9H      ;DISK JOCKEY 2D TRACK ZERO SEEK
E40C =          DJTRK   EQU      DJRAM+0CH     ;DISK JOCKEY 2D TRACK SEEK ROUTINE
E40F =          DJSEC   EQU      DJRAM+0FH     ;DISK JOCKEY 2D SET SECTOR ROUTINE ←
E412 =          DJDMA   EQU      DJRAM+012H   ;DISK JOCKEY 2D SET DMA ADDRESS
E415 =          DJREAD  EQU      DJRAM+15H    ;DISK JOCKEY 2D READ ROUTINE
E418 =          DJWRITE EQU      DJRAM+18H    ;DISK JOCKEY 2D WRITE ROUTINE
E41B =          DJSEL   EQU      DJRAM+1BH    ;DISK JOCKEY 2D SELECT DRIVE ROUTINE
E424 =          DJDMAST EQU      DJRAM+24H    ;DISK JOCKEY 2D DMA STATUS
E427 =          DJSTAT  EQU      DJRAM+27H    ;DISK JOCKEY 2D STATUS ROUTINE
E42A =          DJERR   EQU      DJRAM+2AH    ;DISK JOCKEY 2D ERROR, BLINK LED
E42D =          DJDEN   EQU      DJRAM+2DH    ;DISK JOCKEY 2D SET DENSITY ROUTINE
E421 =          DJTSTAT EQU      DJRAM+21H    ;DISK JOCKEY 2D TERMINAL STATUS ROUTINE
E430 =          DJSIDE  EQU      DJRAM+30H    ;DISK JOCKEY 2D SET SIDE ROUTINE
    
```

```

*****
*
* THE FOLLOWING THREE SECTORS OF CODE RESIDE AT 80H. THERE IS *
* ONE SECTOR FOR EACH OF THE POSSIBLE SECTOR SIZES (256,512, *
* 1024). EACH SECTOR IS RESPONSIBLE FOR PERFORMING A COLD BOOT *
* FOR THE SPECIFIED SECTOR SIZE. *
*
*****
    
```

```

2000 #          DIFF    SET      80H-$
T0 59 E080 310001      LXI      SP,CSTK256+DIFF ;SET UP STACK AT END OF THIS SECTOR
E083 01011A        LXI      B,26*100H+1 ;B = SECTOR COUNT, C = SECTOR #
E086 C5           CLOD256 PUSH    B ;SAVE SECTOR AND COUNT
E087 CD0FE4        CALL    DJSEC ;SET THE NEXT SECTOR TO READ
E08A 2100BA        LXI      H,CCP+300H ;GET DMA ADDRESS (SELF MODIFYING)
E08B =            CDMA256 EQU    $-2 ;STORAGE FOR PREVIOUS DMA ADDRESS
E08D 110002        LXI      D,200H ;OFFSET TO NEW DMA ADDRESS
E090 19           DAD     D ;ADD IN OFFSET, HL = NEW DMA ADDRESS
E091 228B00        SHLD   CDMA256+DIFF ;SAVE NEW DMA ADDRESS
E094 44           MOV     B,H ;PUT DMA ADDRESS INTO BC
E095 4D           MOV     C,L
E096 CD12E4        CALL    DJDMA ;SET THE DMA ADDRESS
E099 CDB600        CALL    CRD256+DIFF ;ATTEMPT A READ
E09C C1           POP     B ;RECOVER SECTOR NUMBER AND COUNT
; B = COUNT, C = NUMBER
E09D 05           DCR     B ;UPDATE SECTOR COUNT
    
```

```

E09E CA00CD      JZ      BIOS      ;ALL DONE ?
E0A1 3E02        MVI      A,2        ;SECTOR UPDATE
E0A3 81          ADD      C          ;ADD IN THE SECTOR SKEW FACTOR
E0A4 4F          MOV      C,A        ;PUT NEW SECTOR BACK INTO C
E0A5 FE1B        CPI      27        ;PAST THE END OF THE TRACK ?
E0A7 DA8600      JC      CLOD256+DIFF ;TAKE JUMP IF NOT PAST END OF TRACK
E0AA D619        SUI      25        ;PERFORM A NEGATIVE SECTOR ADJUSTMENT
E0AC 4F          MOV      C,A        ;PUT NEW SECTOR IN C
E0AD 2100BB      LXI      H,CCP+400H ;NEGATIVE DMA ADJUSTMENT
E0B0 228B00      SHLD    CDMA256+DIFF ;SAVE THE NEW DMA ADDRESS
E0B3 C38600      JMP      CLOD256+DIFF ;CONTINUE READING
    
```

```

*****
*
* CRD256 DOES THE ACTUAL READ FROM THE CONTROLLER, THE DMA
* ADDRESS AND SECTOR # HAVE ALREADY BEEN SET UP.
*
*****
    
```

```

E0B6 01010A      CRD256 LXI      B,RETRIES*100H+1 ;MAXIMUM # OF ATTEMPTS
E0B9 C5          CR256  PUSH    B          ;SAVE ERROR COUNT
E0BA CD0CE4      CALL    DJTRK      ;INITIALIZE THE TRACK
E0BD CD15E4      CALL    DJREAD     ;ATTEMPT THE READ
E0C0 C1          POP     B          ;RESTORE THE ERROR COUNT
E0C1 D0          RNC     ;RETURN IF NO ERROR
E0C2 05          DCR     B          ;UPDATE ERROR COUNT
E0C3 C2B900      JNZ    CR256+DIFF ;TRY AGAIN IF NOT TO MANY ERRORS
E0C6 C32AE4      JMP     DJERR      ;GO AND FLASH THE LIGHT ON CONTROLLER
    
```

```

E0C9            DS      80H-($ MOD 80H)
E100 =          CSTK256 EQU $
    
```

```

*****
*
* THE NEXT LOADS CP/M FROM A 512 BYTE SECTOR DISKETTE.
*
*****
    
```

```

1F80 #          DIFF   SET      80H-$
T0 517
E100 310001      LXI      SP,CSTK512+DIFF ;SET UP STACK AT END OF THIS SECTOR
E103 01010D      LXI      B,13*100H+1    ;B = SECTOR COUNT, C = SECTOR #
E106 C5          CLOD512 PUSH    B          ;SAVE SECTOR AND COUNT
E107 CD0FE4      CALL    DJSEC      ;SET THE NEXT SECTOR TO READ
E10A 2100B8      LXI      H,CCP+100H   ;GET DMA ADDRESS (SELF MODIFYING)
E10B =          CDMA512 EQU    $-2    ;STORAGE FOR PREVIOUS DMA ADDRESS
E10D 110004      LXI      D,400H      ;OFFSET TO NEW DMA ADDRESS
E110 19          DAD     D          ;ADD IN OFFSET, HL = NEW DMA ADDRESS
E111 228B00      SHLD    CDMA512+DIFF ;SAVE NEW DMA ADDRESS
E114 44          MOV     B,H        ;PUT DMA ADDRESS INTO BC
E115 4D          MOV     C,L
E116 CD12E4      CALL    DJDMA      ;SET THE DMA ADDRESS
E119 CDB600      CALL    CRD512+DIFF ;ATTEMPT A READ
E11C C1          POP     B          ;RECOVER SECTOR NUMBER AND COUNT
                        ?      B = COUNT, C = NUMBER
E11D 05          DCR     B          ;UPDATE SECTOR COUNT
    
```

```

E11E CA00CD      JZ      BIOS      ;ALL DONE ?
E121 3E02      MVI      A,2      ;SECTOR UPDATE
E123 81        ADD      C      ;ADD IN THE SECTOR SKEW FACTOR
E124 4F        MOV      C,A      ;PUT NEW SECTOR BACK INTO C
E125 FE0E      CPI      14      ;PAST THE END OF THE TRACK ?
E127 DA8600    JC      CLOD512+DIFF ;TAKE JUMP IF NOT PAST END OF TRACK
E12A D60D      SUI      13      ;PERFORM A NEGATIVE SECTOR ADJUSTMENT
E12C 4F        MOV      C,A      ;PUT NEW SECTOR IN C
E12D 2100BA    LXI      H,CCP+300H ;NEGATIVE DMA ADJUSTMENT
E130 228B00    SHLD     CDMA512+DIFF ;SAVE THE NEW DMA ADDRESS
E133 C38600    JMP      CLOD512+DIFF ;CONTINUE READING
    
```

*
* CRD512 DOES THE ACTUAL READ FROM THE CONTROLLER, THE DMA
* ADDRESS AND SECTOR # HAVE ALREADY BEEN SET UP.
*

```

E136 01010A    CRD512 LXI      B,RETRIES*100H+1 ;MAXIMUM # OF ATTEMPTS
E139 C5        CR512  PUSH     B      ;SAVE ERROR COUNT
E13A CD0CE4    CALL     DJTRK   ;INITIALIZE THE TRACK
E13D CD15E4    CALL     DJREAD  ;ATTEMPT THE READ
E140 C1        POP      B      ;RESTORE THE ERROR COUNT
E141 D0        RNC      ;RETURN IF NO ERROR
E142 05        DCR      B      ;UPDATE ERROR COUNT
E143 C2B900    JNZ     CR512+DIFF ;TRY AGAIN IF NOT TO MANY ERRORS
E146 C32AE4    JMP     DJERR   ;GO AND FLASH THE LIGHT ON CONTROLLER
    
```

```

E149          DS      80H-($ MOD 80H)
E180 =        CSTK512 EQU $
    
```

*
* THE NEXT SECTOR LOADS CP/M FROM A 1024 BYTE SECTOR DISKETTE.
*

```

1F00 #        DIFF   SET      80H-$
E180 310031    LXI      SP,CSTK124+DIFF ;SET UP STACK AT END OF THIS SECTOR
E183 010701    LXI      B,1*100H+7     ;B = SECTOR COUNT, C = SECTOR #
E186 CDAD00    CALL     CLOD124+DIFF   ;LOAD SECTOR 5 INTO CCP
E189 2100D4    LXI      H,CCP+1D00H    ;DESTINATION OF MOVE
E18C 1100BC    LXI      D,CCP+500H     ;SOURCE OF MOVE
E18F 010002    LXI      B,200H
E192 1A        CMOV124 LDAX     D      ;GET A BYTE OF SOURCE
E193 77        MOV      M,A      ;MOVE IT
E194 23        INX     H      ;BUMP DESTINATION
E195 13        INX     D      ;BUMP SOURCE
E196 0D        DCR      C      ;ALL DONE WITH THIS PAGE ?
E197 C29200    JNZ     CMOV124+DIFF
E19A 05        DCR      B
E19B C29200    JNZ     CMOV124+DIFF
E19E 2100B4    LXI      H,CCP-300H     ;INITIAL DMA ADDRESS
E1A1 22B200    SHLD     CDMA124+DIFF
    
```

T0
S25

1X100H+7 = 0107

```

E1A4 010106      LXI      B,6*100H+1      ;B = SECTOR COUNT, C = SECTOR #
E1A7 CDAD00      CALL     CLOD124+DIFF
E1AA C300CD      JMP      BIOS
E1AD C5          CLOD124 PUSH     B              ;SAVE SECTOR AND COUNT
E1AE CD0FE4      CALL     DJSEC.            ;SET THE NEXT SECTOR TO READ
E1B1 2100B4      LXI      H,CCP-300H      ;GET DMA ADDRESS (SELF MODIFYING)
E1B2 =          CDMA124 EQU     $-2          ;STORAGE FOR PREVIOUS DMA ADDRESS
E1B4 110008      LXI      D,800H         ;OFFSET TO NEW DMA ADDRESS
E1B7 19          DAD      D              ;ADD IN OFFSET, HL = NEW DMA ADDRESS
E1B8 22B200      SHLD    CDMA124+DIFF    ;SAVE NEW DMA ADDRESS
E1BB 44          MOV      B,H            ;PUT DMA ADDRESS INTO BC
E1BC 4D          MOV      C,L
E1BD CD12E4      CALL     DJDMA           ;SET THE DMA ADDRESS
E1C0 CDDB00      CALL     CRD124+DIFF    ;ATTEMPT A READ
E1C3 C1          POP      B              ;RECOVER SECTOR NUMBER AND COUNT
                        ;      B = COUNT, C = NUMBER
E1C4 05          DCR      B              ;UPDATE SECTOR COUNT
E1C5 C8          RZ
                        ;ALL DONE ?
E1C6 3E02      MVI      A,2           ;SECTOR UPDATE
E1C8 81          ADD      C              ;ADD IN THE SECTOR SKEW FACTOR
E1C9 4F          MOV      C,A           ;PUT NEW SECTOR BACK INTO C
E1CA FE07      CPI      7              ;PAST THE END OF THE TRACK ?
E1CC DAAD00      JC      CLOD124+DIFF    ;TAKE JUMP IF NOT PAST END OF TRACK
E1CF D605      SUI      5              ;PERFORM A NEGATIVE SECTOR ADJUSTMENT
E1D1 4F          MOV      C,A           ;PUT NEW SECTOR IN C
E1D2 2100B8      LXI      H,CCP+100H     ;NEGATIVE DMA ADJUSTMENT
E1D5 22B200      SHLD    CDMA124+DIFF    ;SAVE THE NEW DMA ADDRESS
E1D8 C3AD00      JMP      CLOD124+DIFF    ;CONTINUE READING
    
```

```

*****
*
* RD124 DOES THE ACTUAL READ FROM THE CONTROLLER, THE DMA
* ADDRESS AND SECTOR # HAVE ALREADY BEEN SET UP.
*
*****
    
```

```

E1DB 01010A      CRD124 LXI      B,RETRIES*100H+1 ;MAXIMUM # OF ATTEMPTS
E1DE C5          CR124  PUSH     B              ;SAVE ERROR COUNT
E1DF CD0CE4      CALL     DJTRK          ;INITIALIZE THE TRACK
E1E2 CD15E4      CALL     DJREAD        ;ATTEMPT THE READ
E1E5 C1          POP      B              ;RESTORE THE ERROR COUNT
E1E6 D0          RNC
                        ;RETURN IF NO ERROR
E1E7 05          DCR      B              ;UPDATE ERROR COUNT
E1E8 C2DE00      JNZ     CR124+DIFF     ;TRY AGAIN IF NOT TO MANY ERRORS
E1EB C32AE4      JMP      DJERR          ;GO AND FLASH THE LIGHT ON CONTROLLER
    
```

```

E1EE          DS      80H-($ MOD 80H)
E200 =          CSTK124 EQU     $
    
```

```

*****
*
* THE NEXT THREE SECTORS OF CODE ALSO RESIDE AT 80H. THERE IS
* ONE SECTOR FOR EACH OF THE POSSIBLE SECTOR SIZES (256,512,
* 1024). EACH SECTOR IS RESPONSIBLE FOR PERFORMING A WARM BOOT
* FOR THE SPECIFIED SECTOR SIZE.
*
*****
    
```


* THE FOLLOWING TABLE SHOWS HOW SECTORS ARE READ IN, SKEWING *
 * OF THE SECTORS IS NECESSARY BECAUSE SEQUENTIAL SECTORS CAN *
 * NOT BE READ WITHOUT WAITING ONE COMPLETE REVOLUTION BETWEEN *
 * EACH ONE. ENTRIES OF ---- REPRESENT A WRAP AROUND (NEGATIVE *
 * DMA ADJUSTMENT). AN ENTRY FLAGGED WITH ** REPRESENTS ONLY A *
 * PARTIAL LOAD FROM THAT SECTOR. *
 *

256		512		1024	
SEC	ADDRESS	SEC	ADDRESS	SEC	ADDRESS
* 1	3100H	1	3100H	1	3100H
* 3	3300H	3	3500H	3	3900H
* 5	3500H	5	3900H	** 5	4100H
* 7	3700H	7	3D00H	-----	-----
* 9	3900H	** 9	4100H	2	3500H
* 11	3B00H	-----	-----	4	3D00H
* 13	3D00H	2	3300H		
* 15	3F00H	4	3700H		
* 17	4100H	6	3B00H		
* ----	----	8	3F00H		
* 2	3200H				
* 4	3400H				
* 6	3600H				
* 8	3800H				
* 10	3A00H				
* 12	3C00H				
* 14	3E00H				
* 16	4000H				

1E80 #	DIFF	SET	80H-\$
E200 310001		LXI	SP,WSTK256+DIFF ;SET UP STACK AT END OF THIS SECTOR
E203 010111		LXI	B,17*100H+1 ;B = SECTOR COUNT, C = SECTOR #
E206 C5	WLOD256	PUSH	B ;SAVE SECTOR AND COUNT
E207 CD0FE4		CALL	DJSEC ;SET THE NEXT SECTOR TO READ
E20A 2100BA		LXI	H,CCP+300H ;GET DMA ADDRESS (SELF MODIFYING)
E20B =	WDMA256	EQU	\$-2 ;STORAGE FOR PREVIOUS DMA ADDRESS
E20D 110002		LXI	D,200H ;OFFSET TO NEW DMA ADDRESS
E210 19		DAD	D ;ADD IN OFFSET, HL = NEW DMA ADDRESS
E211 228B00		SHLD	WDMA256+DIFF ;SAVE NEW DMA ADDRESS
E214 44		MOV	B,H ;PUT DMA ADDRESS INTO BC
E215 4D		MOV	C,L
E216 CD12E4		CALL	DJDMA ;SET THE DMA ADDRESS
E219 CDB600		CALL	WRD256+DIFF ;ATTEMPT A READ
E21C C1		POP	B ;RECOVER SECTOR NUMBER AND COUNT
			; B = COUNT, C = NUMBER
E21D 05		DCR	B ;UPDATE THE SECTOR COUNT
E21E CA03CD		JZ	BIOS+3 ;ALL DONE ?
E221 3E02		MVI	A,2 ;SECTOR UPDATE
E223 81		ADD	C ;ADD IN THE SECTOR SKEW FACTOR
E224 4F		MOV	C,A ;PUT NEW SECTOR BACK INTO C
E225 FE13		CPI	19 ;PAST THE END OF THE TRACK ?
E227 DA8600		JC	WLOD256+DIFF ;TAKE JUMP IF NOT PAST END OF TRACK
E22A D611		SUI	17 ;PERFORM A NEGATIVE SECTOR ADJUSTMENT
E22C 4F		MOV	C,A ;PUT NEW SECTOR IN C

T0
533

```

E22D 2100BB      LXI      H,CCP+400H      ;NEGATIVE DMA ADJUSTMENT
E230 228B00      SHLD     WDMA256+DIFF    ;SAVE THE NEW DMA ADDRESS
E233 C38600      JMP      WLOD256+DIFF    ;CONTINUE READING
    
```

```

*****
*
* WRD256 DOES THE ACTUAL READ FROM THE CONTROLLER, THE DMA
* ADDRESS AND SECTOR # HAVE ALREADY BEEN SET UP.
*
*****
    
```

```

E236 01010A      WRD256 LXI      B,RETRIES*100H+1 ;MAXIMUM # OF ATTEMPTS
E239 C5          WR256  PUSH     B          ;SAVE ERROR COUNT
E23A CD0CE4      CALL     DJTRK        ;INITIALIZE THE TRACK
E23D CD15E4      CALL     DJREAD       ;ATTEMPT THE READ
E240 C1          POP      B          ;RESTORE THE ERROR COUNT
E241 D0          RNC      ;RETURN IF NO ERROR
E242 05          DCR      B          ;UPDATE ERROR COUNT
E243 C2B900      JNZ     WR256+DIFF    ;TRY AGAIN IF NOT TO MANY ERRORS
E246 C32AE4      JMP     DJERR         ;GO AND FLASH THE LIGHT ON CONTROLLER
    
```

```

E249            DS      80H-($ MOD 80H)
E280 =          WSTK256 EQU    $
    
```

```

*****
*
* DISK JOCKEY 2D CP/M FROM A 512 BYTE SECTOR DISKETTE.
*
*****
    
```

```

1E00 #          DIFF     SET      80H-$
    
```

<p><i>TP</i> <i>511</i></p>	<pre> E280 310001 LXI SP,WSTK512+DIFF ;SET UP STACK AT END OF THIS SECTOR E283 010901 LXI B,1*100H+9 ;B = SECTOR COUNT, C = SECTOR # E286 CDA800 CALL WLOD512+DIFF ;LOAD SECTOR 9 INTO CCP E289 2100CC LXI H,CCP+1500H ;DESTINATION OF MOVE E28C 1100BC LXI D,CCP+500H ;SOURCE OF MOVE E28F 0E00 MVI C,0 E291 1A MOV512 LDAX D ;GET A BYTE OF SOURCE E292 77 MOV M,A ;MOVE IT E293 23 INX H ;BUMP DESTINATION E294 13 INX D ;BUMP SOURCE E295 0D DCR C ;ALL DONE WITH THIS PAGE ? E296 C29100 JNZ MOV512+DIFF ;INITIAL DMA ADDRESS E299 2100BA LXI H,CCP+300H ;INITIAL DMA ADDRESS E29C 22AD00 SHLD WDMA512+DIFF ;B = SECTOR COUNT, C = SECTOR # E29F 010208 LXI B,8*100H+2 ;B = SECTOR COUNT, C = SECTOR # E2A2 CDA800 CALL WLOD512+DIFF E2A5 C303CD JMP BIOS+3 E2A8 C5 WLOD512 PUSH B ;SAVE SECTOR AND COUNT E2A9 CD0FE4 CALL DJSEC ;SET THE NEXT SECTOR TO READ E2AC 2100B8 LXI H,CCP+100H ;GET DMA ADDRESS (SELF MODIFYING) E2AD = WDMA512 EQU \$-2 ;STORAGE FOR PREVIOUS DMA ADDRESS E2AF 110004 LXI D,400H ;OFFSET TO NEW DMA ADDRESS E2B2 19 DAD D ;ADD IN OFFSET, HL = NEW DMA ADDRESS E2B3 22AD00 SHLD WDMA512+DIFF ;SAVE NEW DMA ADDRESS </pre>
---------------------------------	--

```

E2B6 44      MOV      B,H          ;PUT DMA ADDRESS INTO BC
E2B7 4D      MOV      C,L
E2B8 CD12E4  CALL     DJDMA         ;SET THE DMA ADDRESS
E2BB CDD600  CALL     WRD512+DIFF      ;ATTEMPT A READ
E2BE C1      POP      B          ;RECOVER SECTOR NUMBER AND COUNT
                        ;      B = COUNT, C = NUMBER
E2BF 05      DCR      B          ;UPDATE SECTOR COUNT
E2C0 C8      RZ
E2C1 3E02    MVI      A,2         ;SECTOR UPDATE
E2C3 81      ADD      C          ;ADD IN THE SECTOR SKEW FACTOR
E2C4 4F      MOV      C,A        ;PUT NEW SECTOR BACK INTO C
E2C5 FE0A    CPI      10        ;PAST THE END OF THE TRACK ?
E2C7 DAA800  JC       WLOD512+DIFF    ;TAKE JUMP IF NOT PAST END OF TRACK
E2CA D609    SUI      9            ;PERFORM A NEGATIVE SECTOR ADJUSTMENT
E2CC 4F      MOV      C,A        ;PUT NEW SECTOR IN C
E2CD 2100B8  LXI      H,CCP+100H    ;NEGATIVE DMA ADJUSTMENT
E2D0 22AD00  SHLD   WDMA512+DIFF      ;SAVE THE NEW DMA ADDRESS
E2D3 C3A800  JMP      WLOD512+DIFF    ;CONTINUE READING
    
```

```

*****
*
* WRD512 DOES THE ACTUAL READ FROM THE CONTROLLER, THE DMA
* ADDRESS AND SECTOR # HAVE ALREADY BEEN SET UP.
*
*****
    
```

```

E2D6 01010A WRD512 LXI      B,RETRIES*100H+1 ;MAXIMUM # OF ATTEMPTS
E2D9 C5      WR512  PUSH   B          ;SAVE ERROR COUNT
E2DA CD0CE4  CALL   DJTRK   ;INITIALIZE THE TRACK
E2DD CD15E4  CALL   DJREAD  ;ATTEMPT THE READ
E2E0 C1      POP    B          ;RESTORE THE ERROR COUNT
E2E1 D0      RNC
E2E2 05      DCR    B          ;UPDATE ERROR COUNT
E2E3 C2D900  JNZ    WR512+DIFF ;TRY AGAIN IF NOT TO MANY ERRORS
E2E6 C32AE4  JMP    DJERR   ;GO AND FLASH THE LIGHT ON CONTROLLER
    
```

```

E2E9      DS      80H-($ MOD 80H)
E300 =    WSTK512 EQU  $
    
```

```

*****
*
* THE NEXT SECTOR LOADS CP/M FROM A 1024 BYTE SECTOR DISKETTE.
*
*****
    
```

```

1D80 #      DIFF  SET      80H-$
E300 310001 LXI      SP,WSTK124+DIFF ;SET UP STACK AT END OF THIS SECTOR
E303 010501 LXI      B,1*100H+5      ;B = SECTOR COUNT, C = SECTOR #
E306 CDA800 CALL   WLOD124+DIFF    ;LOAD SECTOR 6 INTO CCP
E309 2100CC LXI      H,CCP+1500H    ;DESTINATION OF MOVE
E30C 1100BC LXI      D,CCP+500H    ;SOURCE OF MOVE
E30F 0E00    MVI      C,0
E311 1A      MOV124 LDAX   D          ;GET A BYTE OF SOURCE
E312 77      MOV      M,A        ;MOVE IT
E313 23      INX      H          ;BUMP DESTINATION
    
```

TP
549

```

E314 13      INX      D      ;BUMP SOURCE
E315 0D      DCR      C      ;ALL DONE WITH THIS PAGE ?
E316 C29100  JNZ      MOV124+DIFF
E319 2100B8  LXI      H,CCP+100H    ;INITIAL DMA ADDRESS
E31C 22AD00  SHLD     WDMA124+DIFF
E31F 010204  LXI      B,4*100H+2    ;B = SECTOR COUNT, C = SECTOR #
E322 CDA800  CALL     WLOD124+DIFF
E325 C303CD  JMP      BIOS+3
E328 C5      WLOD124 PUSH    B      ;SAVE SECTOR AND COUNT
E329 CD0FE4  CALL     DJSEC          ;SET THE NEXT SECTOR TO READ
E32C 2100B4  LXI      H,CCP-300H    ;GET DMA ADDRESS (SELF MODIFYING)
E32D =      WDMA124 EQU     $-2    ;STORAGE FOR PREVIOUS DMA ADDRESS
E32F 110008  LXI      D,800H        ;OFFSET TO NEW DMA ADDRESS
E332 19      DAD      D      ;ADD IN OFFSET, HL = NEW DMA ADDRESS
E333 22AD00  SHLD     WDMA124+DIFF  ;SAVE NEW DMA ADDRESS
E336 44      MOV      B,H          ;PUT DMA ADDRESS INTO BC
E337 4D      MOV      C,L
E338 CD12E4  CALL     DJDMA          ;SET THE DMA ADDRESS
E33B CDD600  CALL     WRD124+DIFF   ;ATTEMPT A READ
E33E C1      POP      B      ;RECOVER SECTOR NUMBER AND COUNT
                        ;      B = COUNT, C = NUMBER
E33F 05      DCR      B      ;UPDATE SECTOR COUNT
E340 C8      RZ          ;ALL DONE ?
E341 3E02    MVI      A,2          ;SECTOR UPDATE
E343 81      ADD      C      ;ADD IN THE SECTOR SKEW FACTOR
E344 4F      MOV      C,A        ;PUT NEW SECTOR BACK INTO C
E345 FE06    CPI      6          ;PAST THE END OF THE TRACK ?
E347 DAA800  JC       WLOD124+DIFF ;TAKE JUMP IF NOT PAST END OF TRACK
E34A D605    SUI      5          ;PERFORM A NEGATIVE SECTOR ADJUSTMENT
E34C 4F      MOV      C,A        ;PUT NEW SECTOR IN C
E34D 2100B4  LXI      H,CCP-300H    ;NEGATIVE DMA ADJUSTMENT
E350 22AD00  SHLD     WDMA124+DIFF  ;SAVE THE NEW DMA ADDRESS
E353 C3A800  JMP      WLOD124+DIFF  ;CONTINUE READING

```

```

*****
*
* WR124 DOES THE ACTUAL READ FROM THE CONTROLLER, THE DMA
* ADDRESS AND SECTOR # HAVE ALREADY BEEN SET UP.
*
*****

```

```

E356 01010A WRD124 LXI      B,RETRIES*100H+1 ;MAXIMUM # OF ATTEMPTS
E359 C5      WR124  PUSH    B      ;SAVE ERROR COUNT
E35A CD0CE4  CALL     DJTRK        ;INITIALIZE THE TRACK
E35D CD15E4  CALL     DJREAD       ;ATTEMPT THE READ
E360 C1      POP      B      ;RESTORE THE ERROR COUNT
E361 D0      RNC          ;RETURN IF NO ERROR
E362 05      DCR      B      ;UPDATE ERROR COUNT
E363 C2D900  JNZ      WR124+DIFF   ;TRY AGAIN IF NOT TO MANY ERRORS
E366 C32AE4  JMP      DJERR        ;GO AND FLASH THE LIGHT ON CONTROLLER

```

```

E369      DS      80H-($ MOD 80H)
E380 =      WSTK124 EQU     $

```

```

*****
*

```

* THE NEXT SECTOR OF CODE RESIDES AT CCP+500H. IT'S TASK IS TO *
 * MOVE THE FIRMWARE CODE INTO THE DISK JOCKEY RAM, THEN *
 * LOADING A SECTOR INTO 80H WHICH WILL LOAD THE REST OF CP/M. *
 * THE SECTOR LOADED AT 80H IS DEPENDENT ON THE SECTOR SIZE *
 * OF THE DISKETTE BEING BOOTED FROM. *
 * *

70
551

D880 #	DIFF	SET	CCP+500H-\$;USED TO RELOCATE THIS SECTOR OF CODE
E380 C306BC		JMP	DOCOLD+DIFF	;JUMP TO COLD BOOT PORTION
E383 C324BC		JMP	DOWARM+DIFF	;JUMP TO WARM BOOT PORTION
E386 3180BC	DOCOLD	LXI	SP,STK+DIFF	;SET UP INITIAL STACK AT END OF THIS SECTOR
E389 2100E4		LXI	H,DJRAM	;DESTINATION POINTER
E38C 1180BC		LXI	D,STK+DIFF	;SOURCE POINTER
E39F 010003		LXI	B,300H	;LENGTH OF TRANSFER
E392 1A	MLOOP	LDAX	D	;GET ONE BYTE OF SOURCE
E393 77		MOV	M,A	;PUT AT DESTINATION
E394 23		INX	H	;BUMP DESTINATION
E395 13		INX	D	;BUMP SOURCE
E396 0B		DCX	B	;UPDATE COUNT OF BYTES TO MOVE
E397 78		MOV	A,B	;TEST IF ALL DONE
E398 B1		ORA	C	
E399 C212BC		JNZ	MLOOP+DIFF	;CONTINUE MOVING NEW FIRMWARE
E39C CD00E4		CALL	DJBOOT	;INITIALIZE THE NEW FIRMWARE
E39F 3E01		MVI	A,1	
E3A1 3246BC		STA	BOTBIAS+DIFF	;SET UP FOR PROPER SECTOR SELECT
E3A4 3180BC	DOWARM	LXI	SP,STK+DIFF	
E3A7 0E01		MVI	C,1	;SET THE DENSITY TO DOUBLE
E3A9 CD2DE4		CALL	DJDEN	
E3AC 0E01		MVI	C,1	;SET UP TO READ SECTOR 1 ON TRACK 1
E3AE 79		MOV	A,C	
E3AF 3264BC		STA	TRKNUM+DIFF	;SET TRACK
E3B2 CD0FE4		CALL	DJSEC	;SET SECTOR
E3B5 0180BC		LXI	B,STK+DIFF	;SET THE DMA ADDRESS
E3B8 CD12E4		CALL	DJDMA	
E3BB CD63BC		CALL	REED+DIFF	;READ THE SECTOR INTO MEMORY AT ; END OF THIS SECTOR
E3BE CD27E4		CALL	DJSTAT	;DETERMINE THE SECTOR SIZE
E3C1 E60C		ANI	0CH	;STRIP OFF UNWANTED BITS
E3C3 1F		RAR		;FORM THE DESIRED SECTOR FOR COLD BOOT
E3C4 1F		RAR		; BASED ON THE LENGTH OF THE
E3C5 C604		ADI	4	; SECTORS ON THIS DISKETTE
E3C6 =	BOTBIAS	EQU	\$-1	
E3C7 4F		MOV	C,A	;PREPARE TO READ THE COLD BOOT
E3C8 CD0FE4		CALL	DJSEC	;SET UP THE SECTOR
E3CB AF		XRA	A	;TRACK 0
E3CC 3264BC		STA	TRKNUM+DIFF	
E3CF 018000		LXI	B,80H	;COLD BOOT LOADS AT 80H
E3D2 C5		PUSH	B	;USED AS JUMP ADDRESS TO COLD BOOT--
E3D3 CD12E4		CALL	DJDMA	
E3D6 0E00		MVI	C,0	;DENSITY ON TRACK 0 IS SINGLE
E3D8 CD2DE4		CALL	DJDEN	

```

E3DB CD63BC      CALL    REED+DIFF      ;READ IN THE COLD BOOT
E3DE 0E01        MVI      C,1          ;SET THE DENSITY BACK TO DOUBLE
E3E0 C32DE4      JMP      DJDEN           ;
                                     ;GO TO THE COLD BOOT <-----
    
```

```

*****
*
* REED DOES THE ACTUAL READ FROM THE CONTROLLER, THE DMA
* ADDRESS AND SECTOR # HAVE ALREADY BEEN SET UP.
*
*****
    
```

```

E3E3 01000A      REED    LXI      B,RETRIES*100H+0 ;MAXIMUM # OF ATTEMPTS
E3E4 =          TRKNUM  EQU      $-2          ;STORAGE FOR TRACK NUMBER
E3E6 C5         REED1   PUSH     B              ;SAVE ERROR COUNT
E3E7 CD0CE4      CALL    DJTRK          ;INITIALIZE THE TRACK
E3EA CD15E4      CALL    DJREAD         ;ATTEMPT THE READ
E3ED C1         POP     B              ;RESTORE THE ERROR COUNT
E3EE D0         RNC      ;RETURN IF NO ERROR
E3EF 05         DCR     B              ;UPDATE ERROR COUNT
E3F0 C266BC      JNZ     REED1+DIFF       ;TRY AGAIN IF NOT TO MANY ERRORS
E3F3 C32AE4      JMP     DJERR          ;GO AND FLASH THE LIGHT ON CONTROLLER

E3F6           DS      80H-($ MOD 80H)
E400 =         STK    EQU      $
                ELSE
    
```

```

*****
*
* COLD BOOT LOADER FOR DISCUS M10, M20, M26.
*
*****
    
```

```

                IF      MAXFLOP NE 0
                ORG     ORIGIN          ;ORG PROGRAM AT FLOPPY ORIGIN
                ELSE
                ORG     100H            ; OR ELSE AT 100H
                ENDIF
DIFF           SET     100H-$

BOOTHD        LXI     SP,CSTKHD+DIFF    ;SET UP STACK AT END OF THIS SECTOR
                LXI     B,1*100H+20    ;B = SECTOR COUNT, C = SECTOR #
                CALL    CLODHD+DIFF    ;LOAD SECTOR 20 INTO CCP
                LXI     H,CCP+1E00H    ;DESTINATION OF MOVE
                LXI     D,CCP          ;SOURCE OF MOVE
                MVI     C,0

CMOVHD        LDAX   D              ;GET A BYTE OF SOURCE
                MOV     M,A          ;MOVE IT
                INX    H              ;BUMP DESTINATION
                INX    D              ;BUMP SOURCE
                DCR    C              ;ALL DONE WITH THIS PAGE ?
                JNZ    CMOVHD+DIFF
                LXI    H,CCP-200H     ;INITIAL DMA ADDRESS
                SHLD   CDMAHD+DIFF
                LXI    B,15*100H+5    ;B = SECTOR COUNT, C = SECTOR #
                CALL   CLODHD+DIFF
    
```

512

```

                IF      MAXFLOP NE 0
MVI            C,17          ;CHECK IF DISCUS 2D IS PRESENT
MVI            A,(JMP)      ;SHOULD BE 17 JUMPS IN THE JUMP TABLE
LXI            H,ORIGIN
CLOP           CMP          M
JNZ            BIOS        ;NOT 17 JUMPS, DON'T READ IN THE RAMWARE
INX            H           ;SKIP OVER THE JUMP INSTRUCTION
INX            H
INX            H
DCR            C           ;UPDATE JUMP COUNTER
JNZ            CLOP+DIFF   ;CONTINUE CHECKING
LXI            H,DJRAM-200H ;LOAD IN THE DISK JOCKEY 2D RAMWARE
SHLD          CDMAHD+DIFF
LXI            B,2*100H+3
CALL          CLODHD+DIFF
ENDIF
CLODHD        JMP          BIOS      ;GO TO CP/M
PUSH          B           ;SAVE SECTOR AND COUNT
MOV           A,C
STA           HDSEC+DIFF
LXI            H,CCP-200H   ;GET DMA ADDRESS (SELF MODIFYING)
CDMAHD       EQU          $-2     ;STORAGE FOR PREVIOUS DMA ADDRESS
LXI            D,200H      ;OFFSET TO NEW DMA ADDRESS
DAD           D           ;ADD IN OFFSET, HL = NEW DMA ADDRESS
SHLD          CDMAHD+DIFF   ;SAVE NEW DMA ADDRESS
CALL          CRDHD+DIFF   ;ATTEMPT A READ
POP           B           ;RECOVER SECTOR NUMBER AND COUNT
                ;         B = COUNT, C = NUMBER
DCR            B           ;UPDATE SECTOR COUNT
RZ            ;ALL DONE ?
INR            C
JMP           CLODHD+DIFF   ;CONTINUE READING

```

```

*****
*
* RDHD DOES THE ACTUAL READ FROM THE CONTROLLER, THE DMA
* ADDRESS AND SECTOR # HAVE ALREADY BEEN SET UP.
*
*****

```

```

CRDHD        LXI            B,RETRIES*100H+1 ;MAXIMUM # OF ATTEMPTS
CRHD         PUSH          B           ;SAVE ERROR COUNT
CALL          HDREAD+DIFF   ;ATTEMPT THE READ
POP          B           ;RESTORE THE ERROR COUNT
RNC          ;RETURN IF NO ERROR
DCR            B           ;UPDATE ERROR COUNT
JNZ          CRHD+DIFF     ;TRY AGAIN IF NOT TO MANY ERRORS
JMP          $           ;DYNAMIC ERROR HALT

HDREAD       CALL          HDPREP+DIFF  ;PREPARE THE SECTOR HEADER IMAGE
RC           ;ERROR EXIT
MVI            A,RSECT     ;READ SECTOR COMMAND
OUT          HDCMND
CALL          PROCESS+DIFF  ;PROCESS THE READ
RC           ;ERROR EXIT

```

```

XRA      A                ;POINTER TO DATA BUFFER
OUT      HDCMND
MVI      B,SECLN/4        ;NUMBER OF BYTES TO READ
LHLD    CDMAHD+DIFF      ;GET DESTINATION OF DATA
IN       HDDATA          ;TWO DUMMY DATA BYTES
IN       HDDATA
RTLOOP  IN       HDDATA    ;MOVE FOUR BYTES
MOV      M,A            ;BYTE ONE
INX     H
IN       HDDATA          ;BYTE TWO
MOV      M,A
INX     H
IN       HDDATA          ;BYTE THREE
MOV      M,A
INX     H
IN       HDDATA          ;BYTE FOUR
MOV      M,A
INX     H
DCR     B                ;UPDATE BYTE COUNT
JNZ     RTLOOP+DIFF
RET

PROCESS IN       HDSTAT    ;WAIT FOR COMMAND TO FINISH
MOV      B,A
ANI     OPDONE
JZ      PROCESS+DIFF
MVI     A,DSKCLK          ;TURN ON DISK CLOCK
OUT     HDCNTL
IN      HDSTAT
ANI     TMOUT             ;TIMED OUT ?
STC
RNZ
IN      HDRESLT
ANI     RETRY             ;ANY RETRIES ?
STC
RNZ
XRA     A                ;NO ERROR EXIT
RET

HDPREP  IN       HDSTAT    ;IS DRIVE READY ?
ANI     DRVRDY
STC
RNZ
MVI     A,ISBUFF         ;INITIALIZE POINTER TO HEADER BUFFER
OUT     HDCMND
MVI     A,NULL
OUT     HDFUNC           ;SELECT DRIVE A
XRA     A
OUT     HDDATA           ;FORM HEAD BYTE
OUT     HDDATA           ;FORM TRACK BYTE
MVI     A,0              ;FORM SECTOR BYTE
HDSEC   EQU     $-1
OUT     HDDATA
MVI     A,80H            ;FORM KEY
OUT     HDDATA
MVI     A,DSKCLK        ;TURN ON DISK CLOCK

```



```

OUT    HDCNTL
MVI    A,WENABL    ;WRITE ENABLE ON
OUT    HDCNTL
RET
    
```

```

ORG    BOOTHHD+200H-2
    
```

```

CSTKHD EQU    $
        DW    BOOTHHD+DIFF
        DS    200H
        ENDIF
    
```

```

*****
*
* DISK JOCKEY 2D FIRMWARE REVISION 3.1 AND 4.0
* BY GEORGE MORROW
*
* THE FOLLOWING FIRMWARE IS LOADED INTO MEMORY AND THEN MOVED
* INTO THE CONTROLLER RAM.
*
*****
    
```

```

E000 = ROM    IF    MAXFLOP NE 0
E400 = RAM    EQU    EQU    ORIGIN
        EQU    ORIGIN+400H
    
```

```

E3F8 = IO    EQU    ROM+3F8H
E3F8 = UDATA EQU    IO
E3F9 = DCMD  EQU    IO+1
E3F9 = DSTAT EQU    DCMD
E3FA = DREG  EQU    IO+2
E3FA = USTAT EQU    DREG
E3FC = CMDREG EQU    IO+4
E3FC = CSTAT EQU    CMDREG
E3FD = TRKREG EQU    IO+5
E3FE = SECREG EQU    IO+6
E3FF = DATREG EQU    IO+7
*
*
    
```

```

0080 = RCMD  EQU    200Q
00A0 = WCMD  EQU    240Q
0004 = HEAD  EQU    4
0010 = LOAD  EQU    20Q
0001 = DENSTY EQU    1
0018 = ULOAD EQU    30Q
0004 = RSTBIT EQU    4
0002 = ACCESS EQU    2
0020 = READY EQU    40Q
0010 = INDEX EQU    20Q
00C4 = RACMD EQU    304Q
00D0 = CLRCMD EQU    320Q
001D = SVCMD EQU    35Q
0018 = SKCMD EQU    30Q
0009 = HCMD  EQU    11Q
0004 = ISTAT EQU    4
0008 = OSTAT EQU    10Q
    
```

```

0008 =      DSIDE  EQU    10Q
0004 =      TZERO  EQU     4
0003 =      MDINT  EQU     3
001E =      LIGHT  EQU   36Q
003E =      NOLITE EQU   76Q

```

```

*
*

```

```

E400 C333E4  DBOOT  JMP    BOOT
E403 C303E0  TERMIN JMP    ORIGIN+3
E406 C306E0  TRMOUT JMP    ORIGIN+6
E409 C3FEE4  TKZERO JMP    HOME
E40C C332E5  TRKSET JMP    SEEK
E40F C325E5  SETSEC JMP    SECSET
E412 C3E5E4  SETDMA JMP    DMA
E415 C384E5  DREAD  JMP    READ
E418 C363E5  DWRITE JMP    WRITE
E41B C3DAE4  SELDRV JMP    DRIVE
E41E C31EE0  TPANIC JMP    ORIGIN+1EH
E421 C321E0  TSTAT  JMP    ORIGIN+21H
E424 C3D2E4  DMAST  JMP    DMSTAT
E427 C3AEE4  STATUS JMP    DISKST
E42A C32AE0  DSKERR JMP    ROM+52Q
E42D C3CAE7  SETDEN JMP    DENFIX
E430 C37FE6  SETSID JMP    SIDEFX

```

```

*
*

```

```

E433 79      BOOT  MOV    A,C
E434 32EBE6      STA    DISK
E437 3268E4      STA    BDISK
E43A 3E7F      MVI    A,7FH
E43C 07      DSRT  RLC
E43D 0D      DCR    C
E43E F23CE4      JP     DSRT
E441 32EAE6      STA    DRVSEL
E444 3267E4      STA    BDRVSEL
          IF
          CALL  FIXIO
          ENDF

```

```

E447 3AF4E3      LDA    IO-4      ;TEST FOR MODEL A OR B
E44A FEC9      CPI    (RET)
E44C CA8EE4      JZ     MODELA
E44F 1100E0      LXI    D,ORIGIN
E452 2100E4      LXI    H,RAM
E455 0E33      MVI    C,BOOT-DBOOT ;COPY PROM JUMP TABLE INTO RAM
          IF
          CALL  MODELM
          MVI    A,7FH
          CALL  SDSEL
          MVI    A,CLRCMD
          STA    CMDREG
          LDA    BDRVSEL
          ORI    60H
          ANI    7FH
          CALL  SDSEL
          MVI    A,9H
          CALL  SCBITS

```

```

MVI    A,(JZ)
STA    INDX1
STA    INDX3
MVI    A,(JNZ)
STA    INDX2
CALL   MEASUR
XCHG
SHLD   BTIMER
LXI    D,BTBLE
LXI    H,ORIGIN+7E3H
MVI    C,23
ENDIF
E457 1A      MODEL M LDAX   D
E458 77      MOV    M,A
E459 13      INX   D
E45A 23      INX   H
E45B 0D      DCR   C
E45C C257E4 JNZ   MODEL M
E45F C9      RET

E460 00      BTBLE  DB    0
E461 00180000 BTIMER DW   1800H,0
E465 0800    DB    8,0
E467 7E      BDRVSEL DB   7EH
E468 00080009FFBDISK DB  0,8,0,9,0FFH,9,0FFH,9,0FFH,9,0,1,0

E477 2A07E0  FIXIO  LHLD   ROM+7      ;FIND THE 2ND
E47A 23      INX   H                ; BYTE OF INPUT ROUTINE
E47B 110400  LXI   D,4                ;OFFSET
E47E 7E      MOV   A,M                ;GET ADDR OF USTAT
E47F 21EAE7  LXI   H,SDSEL+1           ;I/O ROUTINES
E482 77      MOV   M,A                ;STORE USTAT ADDR
E483 19      DAD   D                ;INCREMENT MEM ADDR
E484 77      MOV   M,A                ;STORE USTAT ADDR
E485 19      DAD   D                ;INCREMENT MEM ADDR
E486 EE03    XRI   3                ;SWITCH THE ADDR
E488 77      MOV   M,A                ;STORE DSTAT ADDR
E489 19      DAD   D                ;INCREMENT MEM ADDR
E48A 77      MOV   M,A                ;STORE DATAT ADDR
E48B 19      DAD   D                ;INCREMENT MEM ADDR
E48C 77      MOV   M,A                ;STORE DSTAT ADDR
E48D C9      RET

E48E =      MODEL A EQU    $
IF          NOT FIRST
E48E 0E00    MVI   C,0                ;COPY LAST PAGE OF RAM
E490 CD57E4 CALL   MODEL M
ENDIF
E493 CD77E4 CALL   FIXIO
E496 3E03    MVI   A,3
E498 CDF1E7 CALL   SCBITS
E49B 3ED0    MVI   A,CLRCMD           ;1791 RESET
E49D 32FCE3 STA   CMDREG                ; COMMAND
E4A0 3AEAE6 LDA   DRVSEL           ;INITIALIZE 1791
E4A3 CDE9E7 CALL   SDSEL                ; CONTROL BITS
E4A6 CDADE7 CALL   MEASUR

```

```

E4A9 EB          XCHG
E4AA 22E5E6      SHLD   TIMER
E4AD C9          RET

*
*
DISKST
E4AE 3AFEE3      LDA     SECREG      ;GET CURRENT
E4B1 47          MOV     B,A        ; SECTOR NO IN B
E4B2 3AFDE3      LDA     TRKREG     ;GET CURRENT
E4B5 4F          MOV     C,A        ; TRACK NO IN C
E4B6 3AF6E6      LDA     DCREG     ;GET CURRENT
E4B9 2F          CMA
E4BA E601        ANI     1          ; DENSITY IN
E4BC 0F          RRC          ; THE MSB
E4BD 57          MOV     D,A        ;POSITION
E4BE 3AF7E6      LDA     SIDE       ;SAVE IN D
E4C1 07          RLC
E4C2 07          RLC          ;PUT THE
E4C3 07          RLC          ; SIDE
E4C4 82          ADD     D          ; SELECT
E4C5 57          MOV     D,A        ; FLAG
E4C6 3AFDE6      LDA     SECLN     ; IN BIT
E4C9 07          RLC          ; POSITION 6
E4CA 07          RLC          ;PUT THE
E4CB 82          ADD     D          ; SECTOR LENGTH
E4CC 57          MOV     D,A        ; CODE P BITS
E4CD 3AECE6      LDA     CDISK     ; 2 & 3
E4D0 82          ADD     D          ;PUT THE CURRENT
E4D1 C9          RET          ; DISK NO IN BITS
                        ; 0 & 1

*
*
DMSTAT
E4D2 E5          PUSH    H          ;SAVE THE H-L PAIR
E4D3 2AE7E6      LHL    DMAADR     ;H-L PAIR
E4D6 44          MOV     B,H        ;MOVE THE DMA
E4D7 4D          MOV     C,L        ;ADDR TO B-C
E4D8 E1          POP     H          ;RECOVER H-L
E4D9 C9          RET

*
*
DRIVE
E4DA 3EFC        MVI     A,374Q     ;TEST FOR THE
E4DC 81          ADD     C          ; NEW DRIVE NUMBER
E4DD 3E10        MVI     A,20Q     ;LESS THAN 4
E4DF D8          RC
E4E0 79          MOV     A,C        ;STORE THE NEW
E4E1 32EBE6      STA     DISK     ;DRIVE IN DISK
E4E4 C9          RET

*
*
DMA
E4E5 210820      LXI     H,8-ROM    ;TEST THE
E4E8 09          DAD     B          ; DMA ADDRESS
E4E9 D2F7E4      JNC    DMASET    ; FOR CONFLICT
E4EC 21001C      LXI     H,-RAM    ; WITH THE I/O
E4EF 09          DAD     B          ; ON THE DJ/2D

```

```

E4F0 DAF7E4      JC      DMASET      ;CONTROLLER
E4F3 37          STC
E4F4 3E10        MVI      A,200
E4F6 C9          RET

DMASET

E4F7 60          MOV      H,B          ;GET THE DMA ADDR
E4F8 69          MOV      L,C          ;TO THE H-L PAR
E4F9 22E7E6     SHLD   DMAADR       ;STORE
E4FC AF          XRA      A          ;CLEAR THE ERROR
E4FD C9          RET          ; FLAG AND RETURN

*
*
HOME

E4FE CD00E7     CALL   HDLOAD        ;LOAD THE HEAD
E501 D8         RC          ;NOT READY ERROR
E502 CD14E5     CALL   HENTRY       ;MOVE THE HEAD
E505 F5         PUSH  PSW         ;SAVE THE FLAGS
E506 9F         SBB      A          ;UPDATE THE
E507 32F9E6     STA      TRACK       ; TRACK
E50A 32FDE3     STA      TRKREG      ; REGISTERS
E50D AF         XRA      A          ;SET THE NOT
E50E 32EDE6     STA      TZFLAG     ; VERIFIED FLAG
E511 C3CAE5     JMP      LEAVE+1     ;UNLOAD THE HEAD

HENTRY

E514 AF         XRA      A          ;UPDATE
E515 32E9E6     STA      HDFLAG     ; FLAGS
E518 210000     LXI      H,0        ;TIME OUT CONSTANT
E51B 3E09        MVI      A,HCMD     ;DO THE HOME
E51D CD7DE7     CALL   CENTRY       ; COMMAND
E520 E604        ANI      TZERO     ;TRACK ZERO BIT
E522 C0         RNZ
E523 37         STC          ;ERROR FLAG
E524 C9         RET

*
*
SECSET

E525 AF         XRA      A          ;TEST FOR
E526 B1         ORA      C          ; SECTOR ZERO
E527 37         STC          ;ERROR FLAG
E528 C8         RZ
E529 79         MOV      A,C          ;TEST FOR
E52A FE1B       CPI      27        ; SECTOR
E52C 3F         CMC          ;TOO LARGE
E52D D8         RC
E52E 32F8E6     STA      SECTOR    ;SAVE
E531 C9         RET

*
*
SEEK

E532 79         MOV      A,C          ;TEST FOR
E533 FE4D       CPI      77        ; TRACK
E535 3F         CMC          ; TOO LARGE
E536 D8         RC
E537 32F9E6     STA      TRACK    ;SAVE
E53A C9         RET

```

```

*
ISSUE
E53B 32E4E6      STA      ECOUNT+1      ;UPDATE COUNT
E53E CDADE7      CALL     MEASUR        ;FIND THE INDEX
E541 0E01        MVI      C,1          ;START W/SECTOR 1

ISLOOP
E543 79          MOV      A,C          ;INITIALIZE THE
E544 32FEE3      STA      SECREG       ; SECTOR REGISTER
E547 3AF8E6      LDA      SECTOR      ;TEST FOR
E54A B9          CMP      C            ; TARGET SECTOR
E54B C8          RZ
E54C 3E80        MVI      A,RCMD       ;DO A FAKE
E54E CD78E7      CALL     COMAND       ; READ COMMAND
E551 DAC7E5      JC      PLEASE      ;ABORT ON ERROR
E554 0C          INR      C            ;INCREMENT SECTOR NO.
E555 C343E5      JMP      ISLOOP

COMNDP
E558 32FCE3      STA      CMDREG       ;DO THE COMMAND
E55B 48          MOV      C,B          ;INITIALIZE BLOCK COUNT
E55C 11FFE3      LXI     D,DATREG     ;DATA REGISTER
E55F 2AE7E6      LHLD   DMAADR       ;TRANSFER ADDRESS
E562 C9          RET

WRITE
E563 CDD4E5      CALL     PREP        ;PREPARE FOR WRITE
E566 DAC9E5      JC      LEAVE      ;ABORT OPERATION

WRENTY
E569 3EA0        MVI      A,WCMD       ;START A WRITE
E56B CD58E5      CALL     COMNDP     ; SECTOR OPERATION

WRLOOP
E56E 7E          MOV      A,M          ;LOAD 1ST BYTE OF DATA
E56F 23          INX     H            ;ADVANCE POINTER
E570 12          STAX   D            ;WRITE 1ST BYTE OF DATA
E571 7E          MOV      A,M          ;LOAD 2ND BYTE OF DATA
E572 23          INX     H            ;ADVANCE POINTER
E573 12          STAX   D            ;WRITE 2ND BYTE OF DATA
E574 7E          MOV      A,M          ;LOAD 3RD BYTE OF DATA
E575 23          INX     H            ;ADVANCE POINTER
E576 12          STAX   D            ;WRITE 3RD BYTE OF DATA
E577 0D          DCR      C            ;REDUCE BLOCK COUNT
E578 7E          MOV      A,M          ;LOAD 4TH BYTE OF DATA
E579 23          INX     H            ;ADVANCE POINTER
E57A 12          STAX   D            ;WRITE 4TH BYTE OF DATA
E57B C26EE5      JNZ     WRLOOP       ;WRITE NEXT 4 BYTES
E57E 2169E5      LXI     H,WRENTY    ;RETURN ENTRY ADDRESS
E581 C3A2E5      JMP     CBUSY

READ
E584 CDD4E5      CALL     PREP        ;PREPARE FOR READ
E587 DAC9E5      JC      LEAVE      ;ABORT OPERATION

RDENTRY
E58A 3E80        MVI      A,RCMD       ;START A READ
E58C CD58E5      CALL     COMNDP     ; SECTOR OPERATION

RDLOOP
E58F 1A          LDAX   D            ;READ 1ST BYTE

```

```

E590 77      MOV      M,A      ;STORE 1ST BYTE
E591 23      INX      H        ;ADVANCE POINTER
E592 1A      LDAX    D        ;READ 2ND BYTE
E593 77      MOV      M,A      ;STORE 2ND BYTE
E594 23      INX      H        ;ADVANCE POINTER
E595 1A      LDAX    D        ;READ 3RD BYTE
E596 77      MOV      M,A      ;STORE 3RD BYTE
E597 23      INX      H        ;ADVANCE POINTER
E598 0D      DCR      C        ;REDUCE BLOCK COUNT
E599 1A      LDAX    D        ;READ 4TH BYTE
E59A 77      MOV      M,A      ;STORE 4TH BYTE
E59B 23      INX      H        ;ADVANCE POINTER
E59C C28FE5  JNZ      RDLOOP   ;READ NEXT 4 BYTES
E59F 218AE5  LXI      H,RDENTRY ;RETURN ENTRY ADDRESS

```

CBUSY

```

E5A2 E5      PUSH    H        ;SAVE RETURN ADDRESS
E5A3 21FCE3  LXI    H,CSTAT  ;WAIT FOR THE 1791
E5A6 CD87E7  CALL   BUSY     ; TO FINISH COMMAND
E5A9 E65F    ANI    137Q    ;ERROR BIT MASK
E5AB CAC8E5  JZ     LEAVE-1  ; TEST
E5AE FE10    CPI    10H     ;PREMATURE INTERRUPT
E5B0 C2C7E5  JNZ    PLEAVE   ;OTHER ERROR TYPE
E5B3 3AE3E6  LDA    ECOUNT   ;DECREMENT ERROR
E5B6 3D      DCR    A        ; COUNT NUMBER 1
E5B7 FABEE5  JM     STEST    ;HARD INTERRUPT ERROR
E5BA 32E3E6  STA    ECOUNT   ;UPDATE COUNT
E5BD C9      RET                    ;DO OPERATION OVER

```

STEST

```

E5BE 3AE4E6  LDA    ECOUNT+1 ;DECREMENT ERROR
E5C1 3D      DCR    A        ; COUNT NUMBER 2
E5C2 F23BE5  JP     ISSUE    ;ISSUE A COMMAND
E5C5 3E10    MVI    A,10H   ;IRRECOVERABLE ERROR

```

PLEAVE

```

E5C7 37      STC                    ;ERROR FLAG
E5C8 E1      POP    H        ;ADJUST THE STACK

```

LEAVE

```

E5C9 F5      PUSH    PSW     ;SAVE THE FLAGS
E5CA 3AF6E6  LDA    DCREG    ;1791 CONTROL BITS
E5CD EE10    XRI    LOAD     ;TOGGLE THE
E5CF CDF1E7  CALL   SCBITS   ; HEAD LOAD BITS
E5D2 F1      POP    PSW     ;RECOVER THE FLAGS
E5D3 C9      RET

```

*

*

PREP

```

E5D4 CD00E7  CALL   HDLOAD   ;LOAD THE HEAD
E5D7 D8      RC                    ;DISK NOT READY?
E5D8 3AFDE3  LDA    TRKREG   ;GET THE OLD TRK
E5DB 3C      INR    A        ;TEST FOR HEAD
E5DC CC14E5  CZ     HENTRY   ; NOT CALIBRATED
E5DF D8      RC                    ;SEEK ERROR?
E5E0 21FDE3  LXI    H,TRKREG ;PRESENT TRK
E5E3 3AF9E6  LDA    TRACK    ;THE NEW TRACK
E5E6 BE      CMP    M        ;TEST FOR HEAD MOTION
E5E7 23      INX    H        ;ADVANCE TO THE
E5E8 23      INX    H        ; DATA REGISTER

```

```

E5E9 77      MOV      M,A          ;SAVE THE NEW TRK
E5EA 79      MOV      A,C          ;TURN OFF DATA
E5EB CDF1E7  CALL     SCBITS         ; ACCESS CONTROL BIT
E5EE CA08E6  JZ       TVERIFY         ;TEST FOR SEEJ
E5F1 AF      XRA       A          ;FORCE A READ
E5F2 32E9E6  STA      HDFLAG         ; HEADER OPERATION
E5F5 CDF5E7  CALL     LDSTAT         ;GET THE
E5F8 E608    ANI      DSIDE        ; DOUBLE
E5FA 1F      RAR              ;
E5FB 1F      RAR              ; FLAG
E5FC 1F      RAR              ; TO DO 3 MS
E5FD C618    ADI      SKCMD        ; STEP OPERATION
E5FF 210000  LXI     H,0             ;DO A SEEK
E602 CD7DE7  CALL     CENTRY         ; COMMAND
E605 DA2EE6  JC       SERROR        ;SEEK ERROR?

TVERIFY
E608 3AE9E6  LDA      HDFLAG         ;GET THE FORCE
E60B B7      ORA       A          ; VERIFY TRACK FLAG
E60C C259E6  JNZ     CHKSEC         ;NO SEEK & HEAD OK
E60F 0602    MVI     B,2         ;VERIFY RETRY NO

SLOOP
E611 3E1D    MVI     A,SVCMD         ;DO A VERIFY
E613 CD78E7  CALL     COMAND         ; COMMAND
E616 E699    ANI     231Q          ;ERROR BIT MASK
E618 57      MOV     D,A          ;SAVE
E619 CA35E6  JZ      RDHDR          ;NO ERROR
E61C 3AF6E6  LDA     DCREG          ;1791 CONTROL REG
E61F EE01    XRI     DENSTY        ;FLIP THE DENSITY BIT
E621 32F6E6  STA     DCREG          ;UPDATE
E624 EE02    XRI     ACCESS        ;
E626 CDF1E7  CALL     SCBITS         ;CHANGE DENSITY
E629 05      DCR     B            ;DEC RETRY COUNT
E62A C211E6  JNZ     SLOOP          ; AND TRY AGAIN
E62D 7A      MOV     A,D          ;RESTORE ERROR BITS

SERROR
E62E 37      STC              ;ERROR FLAG
E62F F5      PUSH    PSW         ;SAVE THE STATUS
E630 CD14E5  CALL     HENTRY         ;SEEK TO TRACK 0
E633 F1      POP     PSW         ;RECOVER ERRORS
E634 C9      RET              ;

RDHDR
E635 060A    MVI     B,12Q         ;NUMBER OF RETRYS

RHLOOP
E637 11FFE3  LXI     D,DATREG        ;DATA REGISTER
E63A 21FAE6  LXI     H,TRACK+1      ;STORAGE AREA
E63D 3EC4    MVI     A,RACMD        ;DO THE READ
E63F 32FCE3  STA     CMDREG         ; HEADER COMMAND

RHL1
E642 1A      LDAX   D            ;GET A DATA BYTE
E643 77      MOV     M,A          ;STORE IN MEMORY
E644 2C      INR     L            ;INC MEM POINTER
E645 C242E6  JNZ     RHL1          ;TEST FOR MORE DATA
E648 21FCE3  LXI     H,CSTAT        ;WAIT FOR 1791
E64B CD87E7  CALL     BUSY          ;TO FINISH CMD
E64E B7      ORA     A            ;TEST FOR ERRORS
E64F CA59E6  JZ      CHKSEC        ;TRANSFER OK?

```



```

E652 05          DCR      B          ;DEC RETRY COUNT
E653 C237E6     JNZ      RHLOOP     ;TEST FOR
E656 C32EE6     JMP      SERROR     ; HARD ERROR

CHKSEC
E659 3AFDE6     LDA      SECLEN     ;GET THE SECTOR
E65C 4F         MOV      C,A          ; SIZE AND SETUP
E65D 0600       MVI      B,0          ; THE OFFSET
E65F 21DFE6     LXI      H,STABLE   ;SEC SIZE TBL
E662 09         DAD      B          ;ADD THE OFFSET
E663 3AF8E6     LDA      SECTOR     ;GET THE SECTOR
E666 47         MOV      B,A          ;SAVE IN B
E667 86         ADD      M          ;COMPARE W/TABLE ENTRY
E668 3E10       MVI      A,20Q        ;ERROR FLAG
E66A D8         RC          ;ERROR RETURN
E66B 78         MOV      A,B          ;SAVE THE SECTOR
E66C 32FEE3     STA      SECREG     ; IN SECTOR REG
E66F 3E20       MVI      A,40Q        ;128 BYTE SECTOR
E671 210505     LXI      H,505H      ;INITIALIZE
E674 22E3E6     SHLD   ECOUNT     ; ERROR COUNTS

SZLOOP
E677 0D         DCR      C          ;REDUCE SIZE COUNT
E678 47         MOV      B,A          ;SECTOR SIZE TO B
E679 F8         RM          ;RETURN ON MINUS
E67A 17         RAL      ;DOUBLE THE COUNT
E67B B7         ORA      A          ;CLEAR THE CARRY
E67C C377E6     JMP      SZLOOP

*
*
SIDEFX
E67F 79         MOV      A,C          ;GET THE SIDE BIT
E680 E601       ANI      1          ;TRIM EXCESS BITS
E682 17         RAL      ;MOVE THE BIT
E683 17         RAL      ; TO THE SIDE
E684 17         RAL      ; SELECT BIT
E685 17         RAL      ; POSITION
E686 32F7E6     STA      SIDE       ;SAVE
E689 C9         RET

TOEND
*
*
E68A           DS      300H-TOEND+DBOOT-66Q
E6CA           DS      25Q

*
*
STACK
*
*
STABLE
E6DF E5         DB      345Q
E6E0 E5         DB      345Q
E6E1 F0         DB      360Q
E6E2 F7         DB      367Q

*
*
E6E3 0000       ECOUNT DW 0          ;ERROR COUNT CELLS
E6E5 0030       TIMER  DW 3000H   ;HEAD LOAD TIME
    
```

```

E6E7 8000    DMAADR DW    2000    ;DMA ADDRESS
E6E9 00      HDFLAG DB    0        ;READ HEADER FLAG
E6EA FE      DRVSEL DB    376Q   ;DRIVE SELECT CONSTANT
E6EB 00      DISK   DB    0        ;NEW DRIVE
E6EC 08      CDISK  DB    100    ;CURRENT DRIVE
E6ED 00      TZFLAG DB    0        ;TRACK ZERO INDICATOR
E6EE 03      D0PRAM DB    3        ;DRIVE 0 PARAMETERS
E6EF FF      D0TRK  DB    377Q   ;DRIVE 0 TRACK NO
E6F0 03      D1PRAM DB    3        ;DRIVE 1 PARAMETERS
E6F1 FF      D1TRK  DB    377Q   ;DRIVE 1 TRACK NO
E6F2 03      D2PRAM DB    3        ;DRIVE 2 PARAMETERS
E6F3 FF      D2TRK  DB    377Q   ;DRIVE 2 TRACK NO
E6F4 03      D3PRAM DB    3        ;DRIVE 3 PARAMETERS
E6F5 FF      D3TRK  DB    377Q   ;DRIVE 3 TRACK NO
E6F6 03      DCREG  DB    3        ;CURRENT PARAMETERS
E6F7 00      SIDE   DB    0        ;NEW SIDE SELECT
E6F8 03      SECTOR DB    3        ;NEW SECTOR
E6F9 00      TRACK  DB    0        ;NEW TRACK
E6FA 00      TRKNO  DB    0        ;DISK
E6FB 00      SIDENO DB    0        ; SECTOR
E6FC 00      SECTNO DB    0        ; HEADER
E6FD 00      SECLN  DB    0        ; DATA
E6FE 00      CRCLO  DB    0        ; BUFFER
E6FF 00      CRCHI  DB    0

```

*
*
HDLOAD

```

E700 21EBE6 LXI    H,DISK
E703 4E      MOV    C,M        ;NEW DISK NO TO C
E704 23      INX    H
E705 5E      MOV    E,M        ;CURRENT DISK TO E
E706 71      MOV    M,C        ;UPDATE CURRENT DISK
E707 23      INX    H        ;ADDR OF DISK TABLE
E708 7B      MOV    A,E        ;TEST FOR
E709 B9      CMP    C        ; DISK CHANGE
E70A 7E      MOV    A,M        ;HEAD LOAD FLAG
E70B 3604    MVI    M,HEAD    ;UPDATE HEAD LOAD
E70D 23      INX    H        ;ADDR OF DISK TABLE
E70E CA36E7 JZ     HDCHK     ;NO DISK CHANGE?
E711 E5      PUSH   H        ;SAVE TABLE ADDRESS
E712 1600    MVI    D,0        ;SET UP THE
E714 42      MOV    B,D        ; OFFSET ADDRESS
E715 19      DAD    D        ;GET THE CURRENT
E716 19      DAD    D        ; DISK PARAMETERS
E717 3AF6E6 LDA    DCREG     ;SAVE THE
E71A 77      MOV    M,A        ;DENSITY INFO
E71B 23      INX    H        ;CURRENT TRACK
E71C 11FDE3 LXI    D,TRKREG
E71F 1A      LDAX  D        ;GET CURRENT TRK
E720 77      MOV    M,A        ;SAVE
E721 E1      POP    H        ;RECOVER TBL ADDR
E722 09      DAD    B        ;ADD THE
E723 09      DAD    B        ; OFFSET
E724 7E      MOV    A,M        ;GET CONTROL BITS
E725 32F6E6 STA    DCREG     ;UPDATE DCREG
E728 23      INX    H        ;GET THE OLD

```

```

E729 7E      MOV      A,M      ;TRACK NUMBER
E72A 12      STAX     D      ;AND UPDATE 1791
E72B 3E7F    MVI      A,177Q    ;DISK SELECT BITS

DSROT
E72D 07      RLC          ;ROTATE TO
E72E 0D      DCR      C      ; SELECT THE
E72F F22DE7  JP       DSROT    ; PROPER DRIVE
E732 32EAE6  STA      DRVSEL   ;SAVE
E735 AF      XRA      A      ;FORCE HEAD LOAD

HDCHK
E736 CDF9E7  CALL     LOADS    ;TEST FOR
E739 A6      ANA      M      ; HEAD LOADED
E73A 32E9E6  STA      HDFLAG   ;SAVE THE HEAD
E73D F5      PUSH     PSW     ; LOADED STATUS
E73E 3AEAE6  LDA      DRVSEL   ;GET CURRENT DRIVE
E741 4F      MOV      C,A     ;SAVE
E742 3AF7E6  LDA      SIDE     ;GET CURRENT SIDE
E745 2F      CMA          ;AND MERGE
E746 A1      ANA      C      ; WITH DRIVE SELECT
E747 CDE9E7  CALL     SDSEL    ;SELECT DRIVE & SIDE
E74A 3AF6E6  LDA      DCREG    ;1791 CONTROL BITS
E74D 4F      MOV      C,A     ;SAVE
E74E 3AF9E6  LDA      TRACK    ;GET THE NEW TRK
E751 D601    SUI      1        ;FORCE SINGLE
E753 9F      SBB      A      ; DENSITY
E754 3D      DCR      A      ; IF TRACK = 0
E755 2F      CMA          ;COMPLIMENT
E756 B1      ORA      C      ;MERGE W/CONTROL BITS
E757 77      MOV      M,A     ;SET 1791 CONTROL
E758 EE02    XRI      ACCESS   ;TOGGEL ACCESS BIT
E75A 4F      MOV      C,A     ;SAVE PREP ROUTINE
E75B F1      POP      PSW    ;HEAD LOAD STATUS
E75C C26AE7  JNZ     RDYCHK   ;CONDITIONALLY
E75F E5      PUSH     H      ; WAIT FOR HEAD
E760 2AE5E6  LHLD    TIMER    ; LOAD TIME OUT

TLOOP
E763 2B      DCX      H      ;COUNT DOWN
E764 7C      MOV      A,H     ; 40 MS FOR
E765 B5      ORA      L      ; HEAD LOAD
E766 C263E7  JNZ     TLOOP    ; TIME OUT
E769 E1      POP      H      ;DISK STATUS ADDR

RDYCHK
E76A 7E      MOV      A,M     ;TEST FOR
E76B E620    ANI      READY   ; DISK READY
E76D C8      RZ

UNLOAD
E76E 3AF6E6  LDA      DCREG    ;FORCE A
E771 F618    ORI      ULOAD    ; HEAD
E773 77      MOV      M,A     ; UNLOAD
E774 3E80    MVI      A,200Q   ;SET DISK
E776 37      STC          ; NOT READY
E777 C9      RET          ; ERROR FLAG

*
*
COMAND
E778 2AE5E6  LHLD    TIMER    ;GET INDEX COUNT

```

```

E77B 29      DAD      H      ; AND MULTIPLY
E77C 29      DAD      H      ; BY FOUR

                CENTRY

E77D EB      XCHG                    ;SAVE IN D-E PAIR
E77E 21FCE3  LXI      H,CSTAT ;ISSUE COMMAND
E781 77      MOV      M,A      ;TO THE 1791

                NBUSY

E782 7E      MOV      A,M      ;WAIT
E783 1F      RAR                    ; FOR THE
E784 D282E7  JNC      NBUSY      ; BUSY FLAG

                BUSY

E787 7E      MOV      A,M      ;TEST FOR
E788 1F      RAR                    ; DEVICE BUSY
E789 7E      MOV      A,M      ;RESTORE STATUS
E78A D0      RNC                    ;RETURN IF NOT BUSY
E78B 1B      DCX      D      ;TEST FOR
E78C 7A      MOV      A,D      ; TWO DISK
E78D B3      ORA      E      ; REVOLUTIONS
E78E C287E7  JNZ      BUSY      ;47 MACHINE CYCLES
E791 5E      MOV      E,M      ;SAVE ERROR CODE
E792 E5      PUSH     H      ;SAVE CMD ADDRESS
E793 23      INX      H      ;TRACK REGISTER
E794 56      MOV      D,M      ;SAVE PRESENT TRACK
E795 E3      XTHL                    ;RECOVER CMD REG.
E796 D5      PUSH     D      ;SAVE STATUS
E797 EB      XCHG                    ;ADJUST REGISTERS
E798 CDF9E7  CALL     LOADS      ;GET CONTROL REG
E79B 3AF6E6  LDA      DCREG      ;1791 CONTROL BITS
E79E EE04    XRI      RSTBIT     ;RESET THE 1791
E7A0 77      MOV      M,A      ; CONTROLLER TO
E7A1 EE04    XRI      RSTBIT     ; CLEAR FAULT
E7A3 EB      XCHG                    ;ADJUST REGISTERS
E7A4 12      STAX     D      ;START CONTROLLER
E7A5 36D0    MVI      M,CLRCMD   ;FORCE AN INTERRUPT
E7A7 D1      POP      D      ;RECOVER STATUS
E7A8 E1      POP      H      ;RECOVER TRACK REG
E7A9 72      MOV      M,D      ;RESTORE TRACK
E7AA 7B      MOV      A,E      ;RESTORE ERROR CODE
E7AB 37      STC                    ; ERROR FLAG
E7AC C9      RET

```

*
*

```

                MEASUR

E7AD 110000  LXI      D,0      ;INITIALIZE COUNT
E7B0 CDF9E7  CALL     LOADS      ;STATUS PORT
E7B3 0E10    MVI      C,INDEX   ;INDEX BIT FLAG

                INDXHI

E7B5 7E      MOV      A,M      ;WAIT FOR
E7B6 A1      ANA      C      ; INDEX
E7B7 C2B5E7  JNZ      INDXHI     ; PULSE LOW

                INDXLO

E7BA 7E      MOV      A,M      ;WAIT FOR
E7BB A1      ANA      C      ; INDEX
E7BC CABAE7  JZ      INDXLO     ; PULSE HIGH

                INDXCT

E7BF 13      INX      D      ;ADVANCE COUNT

```

```

E7C0 E3      XTHL      ;FOUR
E7C1 E3      XTHL      ; DUMMY
E7C2 E3      XTHL      ; INSTRUCTIONS
E7C3 E3      XTHL      ; FOR DELAY
E7C4 7E      MOV       A,M   ;WAIT
E7C5 A1      ANA       C     ; FOR NEXT
E7C6 C2BFE7  INDX3    JNZ     INDXCT ; LOW INDEX
E7C9 C9      RET

```

*
*
DENFIX

```

E7CA 79      MOV       A,C   ;TRIM EXCESS
E7CB E601    ANI       1     ; BITS,
E7CD 2F      CMA      ; COMPLIMENT
E7CE 47      MOV       B,A   ; B AND SAVE
E7CF 21EBE6  LXI     H,DISK ;NEW DISK
E7D2 5E      MOV       E,M   ;GET DISK NO
E7D3 1600    MVI     D,0    ;OFFSET ADDR
E7D5 23      INX     H     ;CURRENT DISK
E7D6 7E      MOV       A,M   ;MOVE TO ACC
E7D7 AB      XRA     E     ;COMPARE W/NEW
E7D8 F5      PUSH    PSW   ;SAVE STATUS
E7D9 23      INX     H     ;DISK TABLE
E7DA 23      INX     H     ; ADDRESS
E7DB 19      DAD     D     ;ADD THE
E7DC 19      DAD     D     ;OFFSET
E7DD 7E      MOV       A,M   ;GET PARAMETERS
E7DE F601    ORI     1     ;MASK OFF DENSITY
E7E0 A0      ANA     B     ;SET NEW DENSITY
E7E1 77      MOV       M,A   ;UPDATE
E7E2 F1      POP     PSW   ;CHECK FOR ND=CD
E7E3 C0      RNZ     ;NEW DISK NOT OLD
E7E4 7E      MOV       A,M   ;UPDATE CDISK
E7E5 32F6E6  STA     DCREG ; ALSO
E7E8 C9      RET

```

*
*
SDSEL

```

E7E9 32FAE3  STA     DREG   ;DRIVE SELECT REG
E7EC C9      RET

```

LUSTAT

```

E7ED 3AFAE3  LDA     USTAT  ;UART STATUS REG
E7F0 C9      RET

```

SCBITS

```

E7F1 32F9E3  STA     DCMD   ;1791 CONTROL REG
E7F4 C9      RET

```

LDSTAT

```

E7F5 3AF9E3  LDA     DSTAT  ;DRIVE STATUS REG
E7F8 C9      RET

```

LOADS

```

E7F9 21F9E3  LXI     H,DSTAT ;DRIVE STATUS REG
E7FC C9      RET

```

```

E7FD      ENDIF
          END

```

0002 ACCESS	E468 BDISK	E467 BDRVSEL	9000 BIAS	CD00 BIOS
E433 BOOT	E3C6 BOTBIAS	E02A BOTERR	E460 BTBLE	E461 BTIMER
E787 BUSY	E5A2 CBUSY	B700 CCP	E6EC CDISK	E1B2 CDMA124
E08B CDMA256	E10B CDMA512	E77D CENTRY	E659 CHKSEC	E1AD CLOD124
E086 CLOD256	E106 CLOD512	00D0 CLRCMD	E3FC CMDREG	E192 CMOV124
E778 COMAND	E558 COMNDP	E1DE CR124	E0B9 CR256	E139 CR512
E6FF CRCHI	E6FE CRCLO	E1DB CRD124	E0B6 CRD256	E136 CRD512
E3FC CSTAT	E200 CSTK124	E100 CSTK256	E180 CSTK512	E6EE D0PRAM
E6EF D0TRK	E6F0 D1PRAM	E6F1 D1TRK	E6F2 D2PRAM	E6F3 D2TRK
E6F4 D3PRAM	E6F5 D3TRK	E3FF DATREG	E400 DBOOT	E3F9 DCMD
E6F6 DCREG	E7CA DENFIX	0001 DENSTY	E6EB DISK	E4AE DISKST
E400 DJBOOT	E403 DJCIN	E406 DJCOUT	E42D DJDEN	E412 DJDMA
E424 DJDMAST	E42A DJERR	E409 DJHOME	E400 DJRAM	E415 DJREAD
E40F DJSEC	E41B DJSEL	E430 DJSIDE	E427 DJSTAT	E40C DJTRK
E421 DJTSTAT	E418 DJWRITE	E4E5 DMA	E6E7 DMAADR	E4F7 DMASET
E424 DMAST	E4D2 DMSTAT	E386 DOCOLD	E015 DOREAD	E3A4 DOWARM
E415 DREAD	E3FA DREG	E4DA DRIVE	E6EA DRVSEL	0008 DSIDE
E42A DSKERR	E72D DSROT	E43C DSRT	E3F9 DSTAT	E418 DWRITE
E6E3 ECOUNT	E003 FIRMLOD	0000 FIRST	E477 FIXIO	E03D FREAD
E027 GETSTAT	0009 HCMD	E736 HDCHK	E6E9 HDFLAG	E700 HDLOAD
0004 HEAD	E514 HENTRY	E4FE HOME	0010 INDEX	E7B7 INDX1
E7BC INDX2	E7C6 INDX3	E7BF INDXCT	E7B5 INDXHI	E7BA INDXLO
E3F8 IO	E543 ISLOOP	E53B ISSUE	0004 ISTAT	E7F5 LDSTAT
E5C9 LEAVE	001E LIGHT	0010 LOAD	E7F9 LOADS	E7ED LUSTAT
0002 MAXFLOP	0001 MAXHD	0003 MDINT	E7AD MEASUR	E392 MLOOP
E48E MODELA	E457 MODELN	E311 MOV124	E291 MOV512	0038 MSIZE
E782 NBUSY	E019 NEWDMA	E004 NEWSEC	003E NOLITE	E011 NOWRAP
E032 NOWRP	2900 OFFSETB	E000 ORIGIN	0008 OSTAT	E5C7 PLEAVE
E5D4 PREP	E02D PUTDEN	E012 PUTDMA	E009 PUTHOM	E00F PUTSEC
E00C PUTTRK	00C4 RACMD	E400 RAM	0080 RCMD	E58A RENTRY
E635 RDHDR	E58F RDLOOP	E76A RDYCHK	E584 READ	0020 READY
E3E3 REED	E3E6 REED1	000A RETRIES	E642 RHL1	E637 RHLOOP
E000 ROM	0004 RSTBIT	E7F1 SCBITS	E7E9 SDSEL	E6FD SECLN
E3FE SECREG	E525 SECSET	E6FC SECTNO	E6F8 SECTOR	E532 SEEK
E41B SELDRV	E62E SERROR	E42D SETDEN	E412 SETDMA	E40F SETSEC
E430 SETSID	E67F SIDEFX	E6F7 SIDE	E6FB SIDENO	0018 SKCMD
E611 SLOOP	E6DF STABLE	E080 STAC	E6DF STACK	E427 STATUS
E5BE STEST	E400 STK	001D SVCMD	E677 SZLOOP	E403 TERMIN
E6E5 TIMER	E409 TKZERO	E763 TLOOP	E68A TOEND	E41E TPANIC
E6F9 TRACK	E6FA TRKNO	E3E4 TRKNUM	E3FD TRKREG	E40C TRKSET
E406 TRMOUT	E421 TSTAT	E608 TVERFY	0004 TZERO	E6ED TZFLAG
E3F8 UDATA	0018 ULOAD	E76E UNLOAD	E3FA USTAT	00A0 WCMD
E32D WDMA124	E20B WDMA256	E2AD WDMA512	E328 WLOD124	E206 WLOD256
E2A8 WLOD512	E359 WR124	E239 WR256	E2D9 WR512	E356 WRD124
E236 WRD256	E2D6 WRD512	E569 WRETRY	E563 WRITE	E56E WRLOOP
E02E WRP	E380 WSTK124	E280 WSTK256	E300 WSTK512	