

```

*****
*
* 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      E700
*      0      8      2C00H
*      0     10     2D00H
*      0     12     2E00H
*      0     14     2F00H
*      0     16     3000H
*      0     18     2780H
*      0     20     2880H
*      0     22     2980H
*      0     24     2A80H
*      0     26     2B80H
*      0      9     2C80H
*      0     11     2D80H
*      0     13     2E80H
*      0     15     2F80H
*      0     17     2700H
*      0     19     2800H
*      0     21     2900H
*      0     23     2A00H
*      0     25     2B00H
*
* THE FOLLOWING DEPEND ON THE SECTOR SIZE, ALL SECTORS ARE FROM
* TRACK 1.
*
* 256          512          1024
* SEC  ADDRESS  SEC  ADDRESS  SEC  ADDRESS
* 1    2C00H    1    2C00H    1    2C00H
* 3    2E00H    3    3000H    3    3400H

```

ABOOT & b. PRN
60K DJ F000
HD = A, B, C
Floppies = D, E
2/1/82

```

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

```

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

```

```

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

```

```

0001 = FIRST EQU 1 ;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 ?
0050 = HDORG EQU 50H ;HARD DISK CONTROLLER
0050 = HDSTAT EQU HDORG ;HARD DISK STATUS
0050 = HDCNTL EQU HDORG ;HARD DISK CONTROL
0053 = HDDATA EQU HDORG+3 ;HARD DISK DATA
0052 = HDFUNC EQU HDORG+2 ;HARD DISK FUNCTION
0051 = HDCMND EQU HDORG+1 ;HARD DISK COMMAND
0051 = HDRESLT EQU HDORG+1 ;HARD DISK RESULT
0002 = RETRY EQU 2 ;RETRY BIT OF RESULT
0001 = TKZ EQU 1 ;TRACK ZERO BIT OF STATUS
0002 = OPDONE EQU 2 ;OPERATION DONE BIT OF STATUS
0004 = COMPLT EQU 4 ;COMPLETE BIT OF STATUS
0008 = TMOUT EQU 8 ;TIME OUT BIT OF STATUS
0010 = WFAULT EQU 10H ;WRITE FAULT BIT OF STATUS
0020 = DRVRDY EQU 20H ;DRIVE READY BIT OF STATUS
0040 = INDX EQU 40H ;INDEX BIT OF STATUS
0004 = PSTEP EQU 4 ;STEP BIT OF FUNCTION
00FB = NSTEP EQU 0FBH ;STEP BIT MASK OF FUNCTION
0004 = HDRLEN EQU 4 ;SECTOR HEADER LENGTH
0200 = SECLN EQU 512 ;SECTOR DATA LENGTH
000F = WENABL EQU 0FH ;WRITE ENABLE
000B = WRESET EQU 0BH ;WRITE RESET OF FUNCTION
0005 = SCENBL EQU 5 ;CONTROLLER CONTROL
0007 = DSKCLK EQU 7 ;DISK CLOCK FOR CONTROL
00F7 = MDIR EQU 0F7H ;DIRECTION MASK FOR FUNCTION
00FC = NULL EQU 0FCH ;NULL COMMAND
0000 = IDBUFF EQU 0 ;INITIALIZE DATA COMMAND
0008 = ISBUFF EQU 8 ;INITIALIZE HEADER COMMAND
0001 = RSECT EQU 1 ;READ SECTOR COMMAND
0005 = WSECT EQU 5 ;WRITE SECTOR COMMAND
    ENDIF
    
```

 *
 * 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. *
 *

```

    IF MAXFLOP NE 0
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
PUTDEN EQU    ORIGIN+02DH    ;SET DENSITY ROUTINE ON DISK JOCKEY 2D
PUTDMA EQU    ORIGIN+12H    ;DISK JOCKEY 2D SET DMA ADDRESS ROUTINE
GETSTAT EQU   ORIGIN+27H    ;DISK STATUS ROUTINE ON DISK JOCKEY 2D
PUTSEC EQU    ORIGIN+0FH    ;DISK JOCKEY 2D SET SECTOR ROUTINE
PUTTRK EQU    ORIGIN+0CH    ;DISK JOCKEY 2D SET TRACK ROUTINE
PUTHOM EQU    ORIGIN+9H     ;DISK JOCKEY 2D TRACK 0 SEEK
DOREAD EQU    ORIGIN+15H    ;DISK JOCKEY 2D READ ROUTINE
BOTERR EQU    ORIGIN+2AH    ;DISK JOCKEY 2D FLASH ERROR LIGHT ROUTINE

           ORG      ORIGIN      ;DISK JOCKEY 2D RAM

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

           LXI      SP,STAC+DIFF
FIRMLOD MVI      A,6          ;PREVIOUS SECTOR #
NEWSEC EQU      $-1
           INR      A          ;UPDATE SECTOR #
           INR      A
           CPI      27        ;TEST IF ALL DONE
           JZ       CCP+500H
           JC       NOWRAP+DIFF ;TEST IF WRAP AROUND
           SUI      19
NOWRAP STA      NEWSEC+DIFF    ;SAVE THE UPDATED SECTOR #
           MOV      C,A
           CALL     PUTSEC     ;SET UP THE SECTOR
           LXI      H,CCP+400H ;PREVIOUS DMA ADDRESS
NEWDMA EQU      $-2
           LXI      D,100H     ;UPDATE DMA ADDRESS
           DAD      D
           MOV      A,H
           CPI      (CCP+980H)/100H
           JC       NOWRP+DIFF
           JNZ      WRP+DIFF
           MOV      A,L
           CPI      (CCP+980H) MOD 100H
           JC       NOWRP+DIFF
WRP      LXI      D,-980H
           DAD      D
NOWRP SHLD     NEWDMA+DIFF     ;SAVE THE UPDATED DMA ADDRESS
           MOV      B,H
           MOV      C,L
           CALL     PUTDMA     ;SET UP THE NEW DMA ADDRESS
           LXI      B,RETRIES*100H+0;MAXIMUM # OF ERRORS
FREAD  PUSH     B
           CALL     PUTTRK     ;SET UP THE PROPER TRACK
           CALL     DOREAD     ;READ THE SECTOR
           POP      B
           JNC     FIRMLOD+DIFF ;CONTINUE IF NO ERROR
           DCR      B
           JNZ     FREAD+DIFF  ;KEEP TRYING IF ERROR

```

```

JMP BOTERR ;TO MANY ERRORS, FLASH THE LIGHT
DS 80H-($ MOD 80H)
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. *
*
*****

```

```

DJCIN EQU DJRAM+3H ;DISK JOCKEY 2D CHARACTER INPUT ROUTINE
DJCOUT EQU DJRAM+6H ;DISK JOCKEY 2D CHARACTER OUTPUT ROUTINE
DJHOME EQU DJRAM+9H ;DISK JOCKEY 2D TRACK ZERO SEEK
DJTRK EQU DJRAM+0CH ;DISK JOCKEY 2D TRACK SEEK ROUTINE
DJSEC EQU DJRAM+0FH ;DISK JOCKEY 2D SET SECTOR ROUTINE
DJDMA EQU DJRAM+012H ;DISK JOCKEY 2D SET DMA ADDRESS
DJREAD EQU DJRAM+15H ;DISK JOCKEY 2D READ ROUTINE
DJWRITE EQU DJRAM+18H ;DISK JOCKEY 2D WRITE ROUTINE
DJSEL EQU DJRAM+1BH ;DISK JOCKEY 2D SELECT DRIVE ROUTINE
DJDMAST EQU DJRAM+24H ;DISK JOCKEY 2D DMA STATUS
DJSTAT EQU DJRAM+27H ;DISK JOCKEY 2D STATUS ROUTINE
DJERR EQU DJRAM+2AH ;DISK JOCKEY 2D ERROR, BLINK LED
DJDEN EQU DJRAM+2DH ;DISK JOCKEY 2D SET DENSITY ROUTINE
DJTSTAT EQU DJRAM+21H ;DISK JOCKEY 2D TERMINAL STATUS ROUTINE
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. *
*
*****

```

```

DIFF SET 80H-$
LXI SP,CSTK256+DIFF ;SET UP STACK AT END OF THIS SECTOR
LXI B,26*100H+1 ;B = SECTOR COUNT, C = SECTOR #
CLOD256 PUSH B ;SAVE SECTOR AND COUNT
CALL DJSEC ;SET THE NEXT SECTOR TO READ
LXI H,CCP+300H ;GET DMA ADDRESS (SELF MODIFYING)
CDMA256 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 CDMA256+DIFF ;SAVE NEW DMA ADDRESS
MOV B,H ;PUT DMA ADDRESS INTO BC
MOV C,L
CALL DJDMA ;SET THE DMA ADDRESS
CALL CRD256+DIFF ;ATTEMPT A READ
POP B ;RECOVER SECTOR NUMBER AND COUNT
; B = COUNT, C = NUMBER
DCR B ;UPDATE SECTOR COUNT

```

```

JZ      BIOS      ;ALL DONE ?
MVI     A,2       ;SECTOR UPDATE
ADD     C         ;ADD IN THE SECTOR SKEW FACTOR
MOV     C,A       ;PUT NEW SECTOR BACK INTO C
CPI     27        ;PAST THE END OF THE TRACK ?
JC      CLOD256+DIFF ;TAKE JUMP IF NOT PAST END OF TRACK
SUI     25        ;PERFORM A NEGATIVE SECTOR ADJUSTMENT
MOV     C,A       ;PUT NEW SECTOR IN C
LXI     H,CCP+400H ;NEGATIVE DMA ADJUSTMENT
SHLD   CDMA256+DIFF ;SAVE THE NEW DMA ADDRESS
JMP     CLOD256+DIFF ;CONTINUE READING
    
```

```

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

```

CRD256 LXI     B,RETRIES*100H+1 ;MAXIMUM # OF ATTEMPTS
CR256  PUSH   B                 ;SAVE ERROR COUNT
        CALL  DJTRK             ;INITIALIZE THE TRACK
        CALL  DJREAD            ;ATTEMPT THE READ
        POP   B                 ;RESTORE THE ERROR COUNT
        RNC   ;RETURN IF NO ERROR
        DCR   B                 ;UPDATE ERROR COUNT
        JNZ   CR256+DIFF        ;TRY AGAIN IF NOT TO MANY ERRORS
        JMP   DJERR             ;GO AND FLASH THE LIGHT ON CONTROLLER
    
```

```

        DS   80H-($ MOD 80H)
CSTK256 EQU   $
    
```

```

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

```

DIFF   SET    80H-$

        LXI   SP,CSTK512+DIFF ;SET UP STACK AT END OF THIS SECTOR
        LXI   B,13*100H+1    ;B = SECTOR COUNT, C = SECTOR #
CLOD512 PUSH   B                 ;SAVE SECTOR AND COUNT
        CALL  DJSEC           ;SET THE NEXT SECTOR TO READ
        LXI   H,CCP+100H     ;GET DMA ADDRESS (SELF MODIFYING)
CDMA512 EQU   $-2             ;STORAGE FOR PREVIOUS DMA ADDRESS
        LXI   D,400H         ;OFFSET TO NEW DMA ADDRESS
        DAD   D               ;ADD IN OFFSET, HL = NEW DMA ADDRESS
        SHLD  CDMA512+DIFF    ;SAVE NEW DMA ADDRESS
        MOV   B,H             ;PUT DMA ADDRESS INTO BC
        MOV   C,L
        CALL  DJDMA           ;SET THE DMA ADDRESS
        CALL  CRD512+DIFF     ;ATTEMPT A READ
        POP   B               ;RECOVER SECTOR NUMBER AND COUNT
        ;     B = COUNT, C = NUMBER
        DCR   B               ;UPDATE SECTOR COUNT
    
```

```

JZ      BIOS          ;ALL DONE ?
MVI     A,2           ;SECTOR UPDATE
ADD     C              ;ADD IN THE SECTOR SKEW FACTOR
MOV     C,A           ;PUT NEW SECTOR BACK INTO C
CPI     14             ;PAST THE END OF THE TRACK ?
JC      CLOD512+DIFF ;TAKE JUMP IF NOT PAST END OF TRACK
SUI     13             ;PERFORM A NEGATIVE SECTOR ADJUSTMENT
MOV     C,A           ;PUT NEW SECTOR IN C
LXI     H,CCP+300H    ;NEGATIVE DMA ADJUSTMENT
SHLD   CDMA512+DIFF  ;SAVE THE NEW DMA ADDRESS
JMP     CLOD512+DIFF ;CONTINUE READING

```

```

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

```

```

CRD512 LXI      B,RETRIES*100H+1 ;MAXIMUM # OF ATTEMPTS
CR512  PUSH    B                ;SAVE ERROR COUNT
        CALL   DJTRK           ;INITIALIZE THE TRACK
        CALL   DJREAD          ;ATTEMPT THE READ
        POP    B                ;RESTORE THE ERROR COUNT
        RNC    ;RETURN IF NO ERROR
        DCR    B                ;UPDATE ERROR COUNT
        JNZ   CR512+DIFF       ;TRY AGAIN IF NOT TO MANY ERRORS
        JMP   DJERR            ;GO AND FLASH THE LIGHT ON CONTROLLER

```

```

        DS     80H-($ MOD 80H)
CSTK512 EQU    $

```

```

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

```

```

DIFF   SET     80H-$

        LXI   SP,CSTK124+DIFF ;SET UP STACK AT END OF THIS SECTOR
        LXI   B,1*100H+7      ;B = SECTOR COUNT, C = SECTOR #
        CALL  CLOD124+DIFF    ;LOAD SECTOR 5 INTO CCP
        LXI   H,CCP+1D00H     ;DESTINATION OF MOVE
        LXI   D,CCP+500H      ;SOURCE OF MOVE
        LXI   B,200H

CMOV124 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  CMOV124+DIFF
        DCR  B
        JNZ  CMOV124+DIFF
        LXI  H,CCP-300H        ;INITIAL DMA ADDRESS
        SHLD CDMA124+DIFF

```

```

LXI B,6*100H+1 ;B = SECTOR COUNT, C = SECTOR #
CALL CLOD124+DIFF
JMP BIOS
CLOD124 PUSH B ;SAVE SECTOR AND COUNT
CALL DJSEC ;SET THE NEXT SECTOR TO READ
LXI H,CCP-300H ;GET DMA ADDRESS (SELF MODIFYING)
CDMA124 EQU $-2 ;STORAGE FOR PREVIOUS DMA ADDRESS
LXI D,800H ;OFFSET TO NEW DMA ADDRESS
DAD D ;ADD IN OFFSET, HL = NEW DMA ADDRESS
SHLD CDMA124+DIFF ;SAVE NEW DMA ADDRESS
MOV B,H ;PUT DMA ADDRESS INTO BC
MOV C,L
CALL DJDMA ;SET THE DMA ADDRESS
CALL CRD124+DIFF ;ATTEMPT A READ
POP B ;RECOVER SECTOR NUMBER AND COUNT
; B = COUNT, C = NUMBER
DCR B ;UPDATE SECTOR COUNT
RZ ;ALL DONE ?
MVI A,2 ;SECTOR UPDATE
ADD C ;ADD IN THE SECTOR SKEW FACTOR
MOV C,A ;PUT NEW SECTOR BACK INTO C
CPI 7 ;PAST THE END OF THE TRACK ?
JC CLOD124+DIFF ;TAKE JUMP IF NOT PAST END OF TRACK
SUI 5 ;PERFORM A NEGATIVE SECTOR ADJUSTMENT
MOV C,A ;PUT NEW SECTOR IN C
LXI H,CCP+100H ;NEGATIVE DMA ADJUSTMENT
SHLD CDMA124+DIFF ;SAVE THE NEW DMA ADDRESS
JMP CLOD124+DIFF ;CONTINUE READING

```

```

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

```

```

CRD124 LXI B,RETRIES*100H+1 ;MAXIMUM # OF ATTEMPTS
CR124 PUSH B ;SAVE ERROR COUNT
CALL DJTRK ;INITIALIZE THE TRACK
CALL DJREAD ;ATTEMPT THE READ
POP B ;RESTORE THE ERROR COUNT
RNC ;RETURN IF NO ERROR
DCR B ;UPDATE ERROR COUNT
JNZ CR124+DIFF ;TRY AGAIN IF NOT TO MANY ERRORS
JMP DJERR ;GO AND FLASH THE LIGHT ON CONTROLLER

```

```

DS 80H-($ MOD 80H)
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				

```

DIFF    SET    80H-$

        LXI    SP,WSTK256+DIFF ;SET UP STACK AT END OF THIS SECTOR
        LXI    B,17*100H+1    ;B = SECTOR COUNT, C = SECTOR #
WLOD256 PUSH    B              ;SAVE SECTOR AND COUNT
        CALL   DJSEC          ;SET THE NEXT SECTOR TO READ
        LXI    H,CCP+300H    ;GET DMA ADDRESS (SELF MODIFYING)
WDMA256 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   WDMA256+DIFF  ;SAVE NEW DMA ADDRESS
        MOV    B,H           ;PUT DMA ADDRESS INTO BC
        MOV    C,L
        CALL   DJDMA         ;SET THE DMA ADDRESS
        CALL   WRD256+DIFF   ;ATTEMPT A READ
        POP    B              ;RECOVER SECTOR NUMBER AND COUNT
        ;      B = COUNT, C = NUMBER
        DCR    B              ;UPDATE THE SECTOR COUNT
        JZ     BIOS+3        ;ALL DONE ?
        MVI    A,2           ;SECTOR UPDATE
        ADD    C              ;ADD IN THE SECTOR SKEW FACTOR
        MOV    C,A           ;PUT NEW SECTOR BACK INTO C
        CPI    19            ;PAST THE END OF THE TRACK ?
        JC     WLOD256+DIFF  ;TAKE JUMP IF NOT PAST END OF TRACK
        SUI    17            ;PERFORM A NEGATIVE SECTOR ADJUSTMENT
        MOV    C,A           ;PUT NEW SECTOR IN C
    
```

```
LXI H,CCP+400H ;NEGATIVE DMA ADJUSTMENT
SHLD WDMA256+DIFF ;SAVE THE NEW DMA ADDRESS
JMP WLOD256+DIFF ;CONTINUE READING
```

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

```
WRD256 LXI B,RETRIES*100H+1 ;MAXIMUM # OF ATTEMPTS
WR256 PUSH B ;SAVE ERROR COUNT
CALL DJTRK ;INITIALIZE THE TRACK
CALL DJREAD ;ATTEMPT THE READ
POP B ;RESTORE THE ERROR COUNT
RNC ;RETURN IF NO ERROR
DCR B ;UPDATE ERROR COUNT
JNZ WR256+DIFF ;TRY AGAIN IF NOT TO MANY ERRORS
JMP DJERR ;GO AND FLASH THE LIGHT ON CONTROLLER
```

```
DS 80H-($ MOD 80H)
WSTK256 EQU $
```

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

```
DIFF SET 80H-$

LXI SP,WSTK512+DIFF ;SET UP STACK AT END OF THIS SECTOR
LXI B,1*100H+9 ;B = SECTOR COUNT, C = SECTOR #
CALL WLOD512+DIFF ;LOAD SECTOR 9 INTO CCP
LXI H,CCP+1500H ;DESTINATION OF MOVE
LXI D,CCP+500H ;SOURCE OF MOVE
MVI C,0
MOV512 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 MOV512+DIFF
LXI H,CCP+300H ;INITIAL DMA ADDRESS
SHLD WDMA512+DIFF
LXI B,8*100H+2 ;B = SECTOR COUNT, C = SECTOR #
CALL WLOD512+DIFF
JMP BIOS+3
WLOD512 PUSH B ;SAVE SECTOR AND COUNT
CALL DJSEC ;SET THE NEXT SECTOR TO READ
LXI H,CCP+100H ;GET DMA ADDRESS (SELF MODIFYING)
WDMA512 EQU $-2 ;STORAGE FOR PREVIOUS DMA ADDRESS
LXI D,400H ;OFFSET TO NEW DMA ADDRESS
DAD D ;ADD IN OFFSET, HL = NEW DMA ADDRESS
SHLD WDMA512+DIFF ;SAVE NEW DMA ADDRESS
```

```

MOV     B,H           ;PUT DMA ADDRESS INTO BC
MOV     C,L
CALL    DJDMA         ;SET THE DMA ADDRESS
CALL    WRD512+DIFF   ;ATTEMPT A READ
POP     B             ;RECOVER SECTOR NUMBER AND COUNT
                        ;      B = COUNT, C = NUMBER
DCR     B             ;UPDATE SECTOR COUNT
RZ                      ;ALL DONE ?
MVI     A,2           ;SECTOR UPDATE
ADD     C             ;ADD IN THE SECTOR SKEW FACTOR
MOV     C,A           ;PUT NEW SECTOR BACK INTO C
CPI     10            ;PAST THE END OF THE TRACK ?
JC      WLOD512+DIFF ;TAKE JUMP IF NOT PAST END OF TRACK
SUI     9             ;PERFORM A NEGATIVE SECTOR ADJUSTMENT
MOV     C,A           ;PUT NEW SECTOR IN C
LXI     H,CCP+100H    ;NEGATIVE DMA ADJUSTMENT
SHLD   WDMA512+DIFF  ;SAVE THE NEW DMA ADDRESS
JMP     WLOD512+DIFF ;CONTINUE READING

```

```

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

```

```

WRD512 LXI     B,RETRIES*100H+1 ;MAXIMUM # OF ATTEMPTS
WR512  PUSH   B                 ;SAVE ERROR COUNT
        CALL  DJTRK            ;INITIALIZE THE TRACK
        CALL  DJREAD           ;ATTEMPT THE READ
        POP   B                 ;RESTORE THE ERROR COUNT
        RNC                      ;RETURN IF NO ERROR
        DCR  B                 ;UPDATE ERROR COUNT
        JNZ  WR512+DIFF        ;TRY AGAIN IF NOT TO MANY ERRORS
        JMP  DJERR             ;GO AND FLASH THE LIGHT ON CONTROLLER

```

```

        DS   80H-($ MOD 80H)
WSTK512 EQU  $

```

```

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

```

```

DIFF   SET    80H-$

        LXI   SP,WSTK124+DIFF ;SET UP STACK AT END OF THIS SECTOR
        LXI   B,1*100H+5     ;B = SECTOR COUNT, C = SECTOR #
        CALL  WLOD124+DIFF   ;LOAD SECTOR 6 INTO CCP
        LXI   H,CCP+1500H    ;DESTINATION OF MOVE
        LXI   D,CCP+500H     ;SOURCE OF MOVE
        MVI   C,0
MOV124 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      MOV124+DIFF
      LXI      H,CCP+100H ;INITIAL DMA ADDRESS
      SHLD     WDMA124+DIFF
      LXI      B,4*100H+2 ;B = SECTOR COUNT, C = SECTOR #
      CALL     WLOD124+DIFF
      JMP      BIOS+3
WLOD124 PUSH     B      ;SAVE SECTOR AND COUNT
      CALL     DJSEC    ;SET THE NEXT SECTOR TO READ
      LXI      H,CCP-300H ;GET DMA ADDRESS (SELF MODIFYING)
WDMA124 EQU      $-2    ;STORAGE FOR PREVIOUS DMA ADDRESS
      LXI      D,800H   ;OFFSET TO NEW DMA ADDRESS
      DAD      D      ;ADD IN OFFSET, HL = NEW DMA ADDRESS
      SHLD     WDMA124+DIFF ;SAVE NEW DMA ADDRESS
      MOV      B,H     ;PUT DMA ADDRESS INTO BC
      MOV      C,L
      CALL     DJDMA    ;SET THE DMA ADDRESS
      CALL     WRD124+DIFF ;ATTEMPT A READ
      POP      B      ;RECOVER SECTOR NUMBER AND COUNT
      ;        B = COUNT, C = NUMBER
      DCR      B      ;UPDATE SECTOR COUNT
      RZ          ;ALL DONE ?
      MVI      A,2    ;SECTOR UPDATE
      ADD      C      ;ADD IN THE SECTOR SKEW FACTOR
      MOV      C,A    ;PUT NEW SECTOR BACK INTO C
      CPI      6      ;PAST THE END OF THE TRACK ?
      JC       WLOD124+DIFF ;TAKE JUMP IF NOT PAST END OF TRACK
      SUI      5      ;PERFORM A NEGATIVE SECTOR ADJUSTMENT
      MOV      C,A    ;PUT NEW SECTOR IN C
      LXI      H,CCP-300H ;NEGATIVE DMA ADJUSTMENT
      SHLD     WDMA124+DIFF ;SAVE THE NEW DMA ADDRESS
      JMP      WLOD124+DIFF ;CONTINUE READING

```

```

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

```

```

WRD124 LXI      B,RETRIES*100H+1 ;MAXIMUM # OF ATTEMPTS
WR124  PUSH     B      ;SAVE ERROR COUNT
      CALL     DJTRK   ;INITIALIZE THE TRACK
      CALL     DJREAD  ;ATTEMPT THE READ
      POP      B      ;RESTORE THE ERROR COUNT
      RNC      ;RETURN IF NO ERROR
      DCR      B      ;UPDATE ERROR COUNT
      JNZ      WR124+DIFF ;TRY AGAIN IF NOT TO MANY ERRORS
      JMP      DJERR   ;GO AND FLASH THE LIGHT ON CONTROLLER

```

```

      DS      80H-($ MOD 80H)
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. *
*
```

```
DIFF    SET    CCP+500H-$    ;USED TO RELOCATE THIS SECTOR OF CODE

        JMP    DOCOLD+DIFF    ;JUMP TO COLD BOOT PORTION
        JMP    DOWARM+DIFF    ;JUMP TO WARM BOOT PORTION

DOCOLD  LXI    SP,STK+DIFF    ;SET UP INITIAL STACK AT END OF THIS SECTOR
        LXI    H,DJRAM        ;DESTINATION POINTER
        LXI    D,STK+DIFF    ;SOURCE POINTER
        LXI    B,300H        ;LENGTH OF TRANSFER
MLOOP   LDAX   D              ;GET ONE BYTE OF SOURCE
        MOV    M,A           ;PUT AT DESTINATION
        INX   H              ;BUMP DESTINATION
        INX   D              ;BUMP SOURCE
        DCX   B              ;UPDATE COUNT OF BYTES TO MOVE
        MOV   A,B           ;TEST IF ALL DONE
        ORA   C
        JNZ   MLOOP+DIFF    ;CONTINUE MOVING NEW FIRMWARE
        CALL  DJBOOT        ;INITIALIZE THE NEW FIRMWARE
        MVI   A,1
        STA   BOTBIAS+DIFF  ;SET UP FOR PROPER SECTOR SELECT

DOWARM  LXI    SP,STK+DIFF
        MVI   C,1            ;SET THE DENSITY TO DOUBLE
        CALL  DJDEN
        MVI   C,1            ;SET UP TO READ SECTOR 1 ON TRACK 1
        MOV   A,C
        STA   TRKNUM+DIFF    ;SET TRACK
        CALL  DJSEC          ;SET SECTOR
        LXI   B,STK+DIFF    ;SET THE DMA ADDRESS
        CALL  DJDMA
        CALL  REED+DIFF      ;READ THE SECTOR INTO MEMORY AT
        ;                END OF THIS SECTOR
        CALL  DJSTAT        ;DETERMINE THE SECTOR SIZE
        ANI   0CH           ;STRIP OFF UNWANTED BITS
        RAR   ;FORM THE DESIRED SECTOR FOR COLD BOOT
        RAR   ;                BASED ON THE LENGTH OF THE
        ADI   4              ;                SECTORS ON THIS DISKETTE

BOTBIAS EQU    $-1
        MOV   C,A           ;PREPARE TO READ THE COLD BOOT
        CALL  DJSEC          ;SET UP THE SECTOR
        XRA   A              ;TRACK 0
        STA   TRKNUM+DIFF
        LXI   B,80H         ;COLD BOOT LOADS AT 80H
        PUSH B              ;USED AS JUMP ADDRESS TO COLD BOOT--
        CALL  DJDMA          ;
        MVI   C,0           ;DENSITY ON TRACK 0 IS SINGLE
        CALL  DJDEN          ;
```

```

CALL REED+DIFF ;READ IN THE COLD BOOT
MVI C,1 ;SET THE DENSITY BACK TO DOUBLE
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.
*
*****

```

```

REED LXI B,RETRIES*100H+0 ;MAXIMUM # OF ATTEMPTS
TRKNUM EQU $-2 ;STORAGE FOR TRACK NUMBER
REED1 PUSH B ;SAVE ERROR COUNT
CALL DJTRK ;INITIALIZE THE TRACK
CALL DJREAD ;ATTEMPT THE READ
POP B ;RESTORE THE ERROR COUNT
RNC ;RETURN IF NO ERROR
DCR B ;UPDATE ERROR COUNT
JNZ REED1+DIFF ;TRY AGAIN IF NOT TO MANY ERRORS
JMP DJERR ;GO AND FLASH THE LIGHT ON CONTROLLER

```

```

DS 80H-($ MOD 80H)
STK EQU $
ELSE

```

```

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

```

```

E000 IF MAXFLOP NE 0
      ORG ORIGIN ;ORG PROGRAM AT FLOPPY ORIGIN
      ELSE
      ORG 100H ; OR ELSE AT 100H
      ENDIF
2100 # DIFF SET 100H-$

E000 31FE02 BOOTHD LXI SP,CSTKHD+DIFF ;SET UP STACK AT END OF THIS SECTOR
E003 011401 LXI B,1*100H+20 ;B = SECTOR COUNT, C = SECTOR #
E006 CD4601 CALL CLODHD+DIFF ;LOAD SECTOR 20 INTO CCP
E009 2100D5 LXI H,CCP+1E00H ;DESTINATION OF MOVE
E00C 1100B7 LXI D,CCP ;SOURCE OF MOVE
E00F 0E00 MVI C,0
E011 1A CMOVHD LDAX D ;GET A BYTE OF SOURCE
E012 77 MOV M,A ;MOVE IT
E013 23 INX H ;BUMP DESTINATION
E014 13 INX D ;BUMP SOURCE
E015 0D DCR C ;ALL DONE WITH THIS PAGE ?
E016 C21101 JNZ CMOVHD+DIFF
E019 2100B5 LXI H,CCP-200H ;INITIAL DMA ADDRESS
E01C 224C01 SHLD CDMAHD+DIFF
E01F 01050F LXI B,15*100H+5 ;B = SECTOR COUNT, C = SECTOR #
E022 CD4601 CALL CLODHD+DIFF

```

```

E025 0E11          IF      MAXFLOP NE 0
E027 3EC3          MVI     C,17          ;CHECK IF DISCUS 2D IS PRESENT
E029 2100E0        MVI     A,(JMP)        ;SHOULD BE 17 JUMPS IN THE JUMP TABLE
E02C BE           CLOP    LXI     H,ORIGIN
E02D C200CD        CMP     M
E030 23           JNZ     BIOS          ;NOT 17 JUMPS, DON'T READ IN THE RAMWARE
E031 23           INX     H          ;SKIP OVER THE JUMP INSTRUCTION
E032 23           INX     H
E033 0D           DCR     C          ;UPDATE JUMP COUNTER
E034 C22C01        JNZ     CLOP+DIFF        ;CONTINUE CHECKING
E037 2100E2        LXI     H,DJRAM-200H    ;LOAD IN THE DISK JOCKEY 2D RAMWARE
E03A 224C01        SHLD   CDMAHD+DIFF
E03D 010302        LXI     B,2*100H+3
E040 CD4601        CALL   CLODHD+DIFF
                    ENDIF
E043 C300CD        JMP     BIOS          ;GO TO CP/M
E046 C5           CLODHD PUSH   B          ;SAVE SECTOR AND COUNT
E047 79           MOV     A,C
E048 32CA01        STA     HDSEC+DIFF
E04B 2100B5        LXI     H,CCP-200H    ;GET DMA ADDRESS (SELF MODIFYING)
E04C =           CDMAHD EQU     $-2    ;STORAGE FOR PREVIOUS DMA ADDRESS
E04E 110002        LXI     D,200H        ;OFFSET TO NEW DMA ADDRESS
E051 19           DAD     D          ;ADD IN OFFSET, HL = NEW DMA ADDRESS
E052 224C01        SHLD   CDMAHD+DIFF    ;SAVE NEW DMA ADDRESS
E055 CD5F01        CALL   CRDHD+DIFF    ;ATTEMPT A READ
E058 C1           POP     B          ;RECOVER SECTOR NUMBER AND COUNT
                    ;      B = COUNT, C = NUMBER
E059 05           DCR     B          ;UPDATE SECTOR COUNT
E05A C8           RZ          ;ALL DONE ?
E05B 0C           INR     C
E05C C34601        JMP     CLODHD+DIFF    ;CONTINUE READING

```

```

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

```

```

E05F 01010A        CRDHD  LXI     B,RETRIES*100H+1 ;MAXIMUM # OF ATTEMPTS
E062 C5           CRHD   PUSH   B          ;SAVE ERROR COUNT
E063 CD6F01        CALL   HDREAD+DIFF    ;ATTEMPT THE READ
E066 C1           POP     B          ;RESTORE THE ERROR COUNT
E067 D0           RNC          ;RETURN IF NO ERROR
E068 05           DCR     B          ;UPDATE ERROR COUNT
E069 C26201        JNZ     CRHD+DIFF    ;TRY AGAIN IF NOT TO MANY ERRORS
E06C C36CE0        JMP     $          ;DYNAMIC ERROR HALT

E06F CDB601        HDREAD CALL   HDPREP+DIFF    ;PREPARE THE SECTOR HEADER IMAGE
E072 D8           RC          ;ERROR EXIT
E073 3E01         MVI     A,RSECT      ;READ SECTOR COMMAND
E075 D351         OUT     HDCMND
E077 CD9C01        CALL   PROCESS+DIFF  ;PROCESS THE READ
E07A D8           RC          ;ERROR EXIT

```

```

E07B AF      XRA      A      ;POINTER TO DATA BUFFER
E07C D351    OUT      HDCMND
E07E 0680    MVI      B,SECLN/4 ;NUMBER OF BYTES TO READ
E080 2A4C01  LHL      CDMAMD+DIFF ;GET DESTINATION OF DATA
E083 DB53    IN       HDDATA ;TWO DUMMY DATA BYTES
E085 DB53    IN       HDDATA
E087 DB53    RTLOOP  IN       HDDATA ;MOVE FOUR BYTES
E089 77      MOV      M,A      ;BYTE ONE
E08A 23      INX     H
E08B DB53    IN       HDDATA ;BYTE TWO
E08D 77      MOV      M,A
E08E 23      INX     H
E08F DB53    IN       HDDATA ;BYTE THREE
E091 77      MOV      M,A
E092 23      INX     H
E093 DB53    IN       HDDATA ;BYTE FOUR
E095 77      MOV      M,A
E096 23      INX     H
E097 05      DCR      B      ;UPDATE BYTE COUNT
E098 C28701  JNZ     RTLOOP+DIFF
E09B C9      RET

E09C DB50    PROCESS IN      HDSTAT ;WAIT FOR COMMAND TO FINISH
E09E 47      MOV      B,A
E09F E602    ANI     OPDONE
E0A1 CA9C01  JZ      PROCESS+DIFF
E0A4 3E07    MVI     A,DSKCLK ;TURN ON DISK CLOCK
E0A6 D350    OUT     HDCNTL
E0A8 DB50    IN      HDSTAT
E0AA E608    ANI     TMOUT ;TIMED OUT ?
E0AC 37      STC
E0AD C0      RNZ
E0AE DB51    IN      HDRESLT
E0B0 E602    ANI     RETRY ;ANY RETRIES ?
E0B2 37      STC
E0B3 C0      RNZ
E0B4 AF      XRA     A      ;NO ERROR EXIT
E0B5 C9      RET

E0B6 DB50    HDPREP  IN      HDSTAT ;IS DRIVE READY ?
E0B8 E620    ANI     DRVRDY
E0BA 37      STC
E0BB C0      RNZ
E0BC 3E08    MVI     A,ISBUFF ;INITIALIZE POINTER TO HEADER BUFFER
E0BE D351    OUT     HDCMND
E0C0 3EFC    MVI     A,NULL
E0C2 D352    OUT     HDFUNC ;SELECT DRIVE A
E0C4 AF      XRA     A
E0C5 D353    OUT     HDDATA ;FORM HEAD BYTE
E0C7 D353    OUT     HDDATA ;FORM TRACK BYTE
E0C9 3E00    MVI     A,0 ;FORM SECTOR BYTE
E0CA =      HDSEC  EQU     $-1
E0CB D353    OUT     HDDATA
E0CD 3E80    MVI     A,80H ;FORM KEY
E0CF D353    OUT     HDDATA
E0D1 3E07    MVI     A,DSKCLK ;TURN ON DISK CLOCK

```



```

E0D3 D350      OUT      HDCNTL
E0D5 3E0F      MVI      A,WENABL      ;WRITE ENABLE ON
E0D7 D350      OUT      HDCNTL
E0D9 C9        RET
    
```

```

E1FE          ORG      BOOTHD+200H-2
    
```

```

E1FE =        CSTKHD EQU      $
E1FE 0001     DW      BOOTHD+DIFF
E200         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      EQU      MAXFLOP NE 0
E400 =        RAM      EQU      ORIGIN
                                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    100
0004 =      TZERO  EQU     4
0003 =      MDINT  EQU     3
001E =      LIGHT  EQU   360
003E =      NOLITE EQU   760
*
*
E400 C333E4  DBOOT  JMP     BOOT
E403 C303E0  TERMIN JMP    ORIGIN+3
E406 C306E0  TRMOUT JMP    ORIGIN+6
E409 C334E5  TKZERO JMP    HOME
E40C C368E5  TRKSET JMP    SEEK
E40F C35BE5  SETSEC JMP    SECSET
E412 C31BE5  SETDMA JMP    DMA
E415 C3BAE5  DREAD  JMP    READ
E418 C399E5  DWRITE JMP    WRITE
E41B C310E5  SELDRV JMP    DRIVE
E41E C31EE0  TPANIC JMP    ORIGIN+1EH
E421 C321E0  TSTAT  JMP    ORIGIN+21H
E424 C308E5  DMAST  JMP    DMSTAT
E427 C3E4E4  STATUS JMP    DISKST
E42A C32AE0  DSKERR JMP    ROM+52Q
E42D C3CAE7  SETDEN JMP    DENFIX
E430 C3B5E6  SETSID JMP    SIDEFX
*
*
E433 79      BOOT   MOV     A,C
E434 32EBE6          STA     DISK
E437 32A3E4          STA     BDISK
E43A 3E7F          MVI     A,7FH
E43C 07          DSRT   RLC
E43D 0D          DCR     C
E43E F23CE4          JP      DSRT
E441 32EAE6          STA     DRVSEL
E444 32A2E4          STA     BDRVSEL
          IF     FIRST
E447 CDB2E4          CALL    FIXIO
          ENDIF
E44A 3AF4E3          LDA     IO-4          ;TEST FOR MODEL A OR B
E44D FEC9           CPI     (RET)
E44F CAC9E4          JZ     MODELA
E452 1100E0          LXI     D,ORIGIN
E455 2100E4          LXI     H,RAM
E458 0E33           MVI     C,BOOT-DBOOT ;COPY PROM JUMP TABLE INTO RAM
          IF     FIRST
E45A CD92E4          CALL    MODELM
E45D 3E7F           MVI     A,7FH
E45F CDE9E7          CALL    SDSEL
E462 3ED0           MVI     A,CLRCMD
E464 32FCE3          STA     CMDREG
E467 3AA2E4          LDA     BDRVSEL
E46A F660           ORI     60H
E46C E67F           ANI     7FH
E46E CDE9E7          CALL    SDSEL
E471 3E09           MVI     A,9H
E473 CDF1E7          CALL    SCBITS

```

```

E476 3ECA      MVI      A,(JZ)
E478 32B7E7    STA      INDX1
E47B 32C6E7    STA      INDX3
E47E 3EC2      MVI      A,(JNZ)
E480 32BCE7    STA      INDX2
E483 CDADE7    CALL     MEASUR
E486 EB        XCHG
E487 229CE4    SHLD     BTIMER
E48A 119BE4    LXI     D,BTBLE
E48D 21E3E7    LXI     H,ORIGIN+7E3H
E490 0E17      MVI     C,23
                ENDIF
E492 1A        MODEL M LDAX     D
E493 77        MOV     M,A
E494 13        INX     D
E495 23        INX     H
E496 0D        DCR     C
E497 C292E4    JNZ     MODEL M
E49A C9        RET

E49B 00        BTBLE  DB      0
E49C 00180000  BTIMER DW     1800H,0
E4A0 0800      DB      8,0
E4A2 7E        BDRVSEL DB     7EH
E4A3 00080009FFBDISK DB     0,8,0,9,0FFH,9,0FFH,9,0FFH,9,0,1,1

E4B2 2A07E0    FIXIO  LHLD    ROM+7      ;FIND THE 2ND
E4B5 23        INX     H              ; BYTE OF INPUT ROUTINE
E4B6 110400    LXI     D,4          ;OFFSET
E4B9 7E        MOV     A,M          ;GET ADDR OF USTAT
E4BA 21EAE7    LXI     H,SDSEL+1    ;I/O ROUTINES
E4BD 77        MOV     M,A          ;STORE USTAT ADDR
E4BE 19        DAD     D          ;INCREMENT MEM ADDR
E4BF 77        MOV     M,A          ;STORE USTAT ADDR
E4C0 19        DAD     D          ;INCREMENT MEM ADDR
E4C1 EE03      XRI     3            ;SWITCH THE ADDR
E4C3 77        MOV     M,A          ;STORE DSTAT ADDR
E4C4 19        DAD     D          ;INCREMENT MEM ADDR
E4C5 77        MOV     M,A          ;STORE DATAT ADDR
E4C6 19        DAD     D          ;INCREMENT MEM ADDR
E4C7 77        MOV     M,A          ;STORE DSTAT ADDR
E4C8 C9        RET

E4C9 =         MODEL A EQU     $
                IF      NOT FIRST
                MVI     C,0      ;COPY LAST PAGE OF RAM
                CALL    MODEL M
                ENDIF

E4C9 CDB2E4    CALL    FIXIO
E4CC 3E03      MVI     A,3
E4CE CDF1E7    CALL    SCBITS
E4D1 3ED0      MVI     A,CLRCMD      ;1791 RESET
E4D3 32FCE3    STA     CMDREG        ; COMMAND
E4D6 3AEAE6    LDA     DRVSEL        ;INITIALIZE 1791
E4D9 CDE9E7    CALL    SDSEL         ; CONTROL BITS
E4DC CDADE7    CALL    MEASUR

```

```

E4DF EB          XCHG
E4E0 22E5E6     SHLD  TIMER
E4E3 C9          RET

*
*
DISKST
E4E4 3AFEE3     LDA    SECREG      ;GET CURRENT
E4E7 47         MOV    B,A          ; SECTOR NO IN B
E4E8 3AFDE3     LDA    TRKREG     ;GET CURRENT
E4EB 4F         MOV    C,A          ; TRACK NO IN C
E4EC 3AF6E6     LDA    DCREG     ;GET CURRENT
E4EF 2F         CMA           ; DENSITY IN
E4F0 E601       ANI    1          ; THE MSB
E4F2 0F         RRC           ;POSITION
E4F3 57         MOV    D,A        ;SAVE IN D
E4F4 3AF7E6     LDA    SIDE      ;PUT THE
E4F7 07         RLC           ; SIDE
E4F8 07         RLC           ; SELECT
E4F9 07         RLC           ; FLAG
E4FA 82         ADD    D          ; IN BIT
E4FB 57         MOV    D,A        ; POSITION 6
E4FC 3AFDE6     LDA    SECLN     ;PUT THE
E4FF 07         RLC           ; SECTOR LENGTH
E500 07         RLC           ; CODE P BITS
E501 82         ADD    D          ; 2 & 3
E502 57         MOV    D,A        ;
E503 3AECE6     LDA    CDISK     ;PUT THE CURRENT
E506 82         ADD    D          ; DISK NO IN BITS
E507 C9         RET           ; 0 & 1

*
*
DMSTAT
E508 E5         PUSH   H          ;SAVE THE H-L PAIR
E509 2AE7E6     LHLD  DMAADR     ;H-L PAIR
E50C 44         MOV    B,H        ;MOVE THE DMA
E50D 4D         MOV    C,L        ;ADDR TO B-C
E50E E1         POP    H          ;RECOVER H-L
E50F C9         RET

*
*
DRIVE
E510 3EFC       MVI    A,374Q     ;TEST FOR THE
E512 81         ADD    C          ; NEW DRIVE NUMBER
E513 3E10       MVI    A,200     ;LESS THAN 4
E515 D8         RC
E516 79         MOV    A,C        ;STORE THE NEW
E517 32EBE6     STA    DISK      ;DRIVE IN DISK
E51A C9         RET

*
*
DMA
E51B 210820     LXI    H,8-ROM    ;TEST THE
E51E 09         DAD    B          ; DMA ADDRESS
E51F D22DE5     JNC   DMASET     ; FOR CONFLICT
E522 21001C     LXI    H,-RAM    ; WITH THE I/O
E525 09         DAD    B          ; ON THE DJ/2D

```

```

E526 DA2DE5      JC      DMASET      ;CONTROLLER
E529 37          STC
E52A 3E10        MVI      A,20Q
E52C C9          RET

DMASET

E52D 60          MOV      H,B          ;GET THE DMA ADDR
E52E 69          MOV      L,C          ;TO THE H-L PAR
E52F 22E7E6     SHLD   DMAADR      ;STORE
E532 AF          XRA      A          ;CLEAR THE ERROR
E533 C9          RET          ; FLAG AND RETURN

*
*
HOME

E534 CD00E7     CALL   HDLOAD      ;LOAD THE HEAD
E537 D8         RC          ;NOT READY ERROR
E538 CD4AE5     CALL   HENTRY     ;MOVE THE HEAD
E53B F5         PUSH  PSW      ;SAVE THE FLAGS
E53C 9F         SBB   A          ;UPDATE THE
E53D 32F9E6     STA   TRACK      ; TRACK
E540 32FDE3     STA   TRKREG     ; REGISTERS
E543 AF         XRA   A          ;SET THE NOT
E544 32EDE6     STA   TZFLAG     ; VERIFIED FLAG
E547 C300E6     JMP   LEAVE+1     ;UNLOAD THE HEAD

HENTRY

E54A AF         XRA   A          ;UPDATE
E54B 32E9E6     STA   HDFLAG     ; FLAGS
E54E 210000     LXI  H,0         ;TIME OUT CONSTANT
E551 3E09        MVI  A,HCMD      ;DO THE HOME
E553 CD7DE7     CALL  CENTRY     ; COMMAND
E556 E604        ANI  TZERO      ;TRACK ZERO BIT
E558 C0         RNZ
E559 37         STC          ;ERROR FLAG
E55A C9         RET

*
*
SECSET

E55B AF         XRA   A          ;TEST FOR
E55C B1         ORA   C          ; SECTOR ZERO
E55D 37         STC          ;ERROR FLAG
E55E C8         RZ
E55F 79         MOV   A,C          ;TEST FOR
E560 FE1B        CPI   27         ; SECTOR
E562 3F         CMC          ;TOO LARGE
E563 D8         RC
E564 32F8E6     STA   SECTOR     ;SAVE
E567 C9         RET

*
*
SEEK

E568 79         MOV   A,C          ;TEST FOR
E569 FE4D        CPI   77         ; TRACK
E56B 3F         CMC          ; TOO LARGE
E56C D8         RC
E56D 32F9E6     STA   TRACK     ;SAVE
E570 C9         RET
    
```

*

ISSUE

```

E571 32E4E6      STA      ECOUNT+1      ;UPDATE COUNT
E574 CDADE7      CALL     MEASUR        ;FIND THE INDEX
E577 0E01        MVI      C,1          ;START W/SECTOR 1

```

ISLOOP

```

E579 79          MOV      A,C          ;INITIALIZE THE
E57A 32FEE3      STA      SECREG       ; SECTOR REGISTER
E57D 3AF8E6      LDA      SECTOR       ;TEST FOR
E580 B9          CMP      C            ; TARGET SECTOR
E581 C8          RZ
E582 3E80        MVI      A,RCMD       ;DO A FAKE
E584 CD78E7      CALL     COMAND        ; READ COMMAND
E587 DAFDE5      JC       PLEAVE       ;ABORT ON ERROR
E58A 0C          INR      C            ;INCREMENT SECTOR NO.
E58B C379E5      JMP      ISLOOP

```

COMNDP

```

E58E 32FCE3      STA      CMDREG       ;DO THE COMMAND
E591 48          MOV      C,B          ;INITIALIZE BLOCK COUNT
E592 11FFE3      LXI     D,DATREG     ;DATA REGISTER
E595 2AE7E6      LHLD   DMAADR        ;TRANSFER ADDRESS
E598 C9          RET

```

WRITE

```

E599 CD0AE6      CALL     PREP          ;PREPARE FOR WRITE
E59C DAFFE5      JC       LEAVE       ;ABORT OPERATION

```

WREENTRY

```

E59F 3EA0        MVI      A,WCMD       ;START A WRITE
E5A1 CD8EE5      CALL     COMNDP       ; SECTOR OPERATION

```

WRLOOP

```

E5A4 7E          MOV      A,M          ;LOAD 1ST BYTE OF DATA
E5A5 23          INX     H            ;ADVANCE POINTER
E5A6 12          STAX   D            ;WRITE 1ST BYTE OF DATA
E5A7 7E          MOV      A,M          ;LOAD 2ND BYTE OF DATA
E5A8 23          INX     H            ;ADVANCE POINTER
E5A9 12          STAX   D            ;WRITE 2ND BYTE OF DATA
E5AA 7E          MOV      A,M          ;LOAD 3RD BYTE OF DATA
E5AB 23          INX     H            ;ADVANCE POINTER
E5AC 12          STAX   D            ;WRITE 3RD BYTE OF DATA
E5AD 0D          DCR      C            ;REDUCE BLOCK COUNT
E5AE 7E          MOV      A,M          ;LOAD 4TH BYTE OF DATA
E5AF 23          INX     H            ;ADVANCE POINTER
E5B0 12          STAX   D            ;WRITE 4TH BYTE OF DATA
E5B1 C2A4E5      JNZ     WRLOOP        ;WRITE NEXT 4 BYTES
E5B4 219FE5      LXI     H,WREENTRY    ;RETURN ENTRY ADDRESS
E5B7 C3D8E5      JMP     CBUSY

```

READ

```

E5BA CD0AE6      CALL     PREP          ;PREPARE FOR READ
E5BD DAFFE5      JC       LEAVE       ;ABORT OPERATION

```

RDENTRY

```

E5C0 3E80        MVI      A,RCMD       ;START A READ
E5C2 CD8EE5      CALL     COMNDP       ; SECTOR OPERATION

```

RDLOOP

```

E5C5 1A          LDAX   D            ;READ 1ST BYTE

```

```

E5C6 77      MOV      M,A      ;STORE 1ST BYTE
E5C7 23      INX      H      ;ADVANCE POINTER
E5C8 1A      LDAX    D      ;READ 2ND BYTE
E5C9 77      MOV      M,A      ;STORE 2ND BYTE
E5CA 23      INX      H      ;ADVANCE POINTER
E5CB 1A      LDAX    D      ;READ 3RD BYTE
E5CC 77      MOV      M,A      ;STORE 3RD BYTE
E5CD 23      INX      H      ;ADVANCE POINTER
E5CE 0D      DCR      C      ;REDUCE BLOCK COUNT
E5CF 1A      LDAX    D      ;READ 4TH BYTE
E5D0 77      MOV      M,A      ;STORE 4TH BYTE
E5D1 23      INX      H      ;ADVANCE POINTER
E5D2 C2C5E5  JNZ      RDLOOP    ;READ NEXT 4 BYTES
E5D5 21C0E5  LXI      H,RDENTRY ;RETURN ENTRY ADDRESS

```

CBUSY

```

E5D8 E5      PUSH    H      ;SAVE RETURN ADDRESS
E5D9 21FCE3  LXI    H,CSTAT ;WAIT FOR THE 1791
E5DC CD87E7  CALL   BUSY   ; TO FINISH COMMAND
E5DF E65F    ANI    137Q   ;ERROR BIT MASK
E5E1 CAFEE5  JZ     LEAVE-1 ; TEST
E5E4 FE10    CPI    10H    ;PREMATURE INTERRUPT
E5E6 C2FDE5  JNZ    PLEAVE  ;OTHER ERROR TYPE
E5E9 3AE3E6  LDA    ECOUNT  ;DECREMENT ERROR
E5EC 3D      DCR    A      ; COUNT NUMBER 1
E5ED FAF4E5  JM     STEST   ;HARD INTERRUPT ERROR
E5F0 32E3E6  STA    ECOUNT  ;UPDATE COUNT
E5F3 C9      RET                    ;DO OPERATION OVER

```

STEST

```

E5F4 3AE4E6  LDA    ECOUNT+1 ;DECREMENT ERROR
E5F7 3D      DCR    A      ; COUNT NUMBER 2
E5F8 F271E5  JP     ISSUE   ;ISSUE A COMMAND
E5FB 3E10    MVI    A,10H  ;IRRECOVERABLE ERROR

```

PLEAVE

```

E5FD 37      STC                    ;ERROR FLAG
E5FE E1      POP    H      ;ADJUST THE STACK

```

LEAVE

```

E5FF F5      PUSH    PSW     ;SAVE THE FLAGS
E600 3AF6E6  LDA    DCREG   ;1791 CONTROL BITS
E603 EE10    XRI    LOAD    ;TOGGLE THE
E605 CDF1E7  CALL   SCBITS  ; HEAD LOAD BITS
E608 F1      POP    PSW     ;RECOVER THE FLAGS
E609 C9      RET

```

*

*

PREP

```

E60A CD00E7  CALL   HDLOAD  ;LOAD THE HEAD
E60D D8      RC                    ;DISK NOT READY?
E60E 3AFDE3  LDA    TRKREG  ;GET THE OLD TRK
E611 3C      INR    A      ;TEST FOR HEAD
E612 CC4AE5  CZ     HENTRY  ; NOT CALIBRATED
E615 D8      RC                    ;SEEK ERROR?
E616 21FDE3  LXI    H,TRKREG ;PRESENT TRK
E619 3AF9E6  LDA    TRACK   ;THE NEW TRACK
E61C BE      CMP    M      ;TEST FOR HEAD MOTION
E61D 23      INX    H      ;ADVANCE TO THE
E61E 23      INX    H      ; DATA REGISTER

```

```

E61F 77      MOV      M,A          ;SAVE THE NEW TRK
E620 79      MOV      A,C          ;TURN OFF DATA
E621 CDF1E7  CALL     SCBITS        ; ACCESS CONTROL BIT
E624 CA3EE6  JZ       TVERIFY        ;TEST FOR SEEJ
E627 AF      XRA       A          ;FORCE A READ
E628 32E9E6  STA      HDFLAG         ; HEADER OPERATION
E62B CDF5E7  CALL     LDSTAT         ;GET THE
E62E E608    ANI      DSIDE        ; DOUBLE
E630 1F      RAR              ;
E631 1F      RAR              ; FLAG
E632 1F      RAR              ; TO DO 3 MS
E633 C618    ADI      SKCMD        ; STEP OPERATION
E635 210000  LXI      H,0          ;DO A SEEK
E638 CD7DE7  CALL     CENTRY         ; COMMAND
E63B DA64E6  JC       SERROR        ;SEEK ERROR?

TVERIFY
E63E 3AE9E6  LDA      HDFLAG         ;GET THE FORCE
E641 B7      ORA       A          ; VERIFY TRACK FLAG
E642 C28FE6  JNZ     CHKSEC         ;NO SEEK & HEAD OK
E645 0602    MVI      B,2          ;VERIFY RETRY NO

SLOOP
E647 3E1D    MVI      A,SVCMD        ;DO A VERIFY
E649 CD78E7  CALL     COMAND         ; COMMAND
E64C E699    ANI      231Q         ;ERROR BIT MASK
E64E 57      MOV      D,A          ;SAVE
E64F CA6BE6  JZ       RDHDR         ;NO ERROR
E652 3AF6E6  LDA      DCREG         ;1791 CONTROL REG
E655 EE01    XRI      DENSTY        ;FLIP THE DENSITY BIT
E657 32F6E6  STA      DCREG         ;UPDATE
E65A EE02    XRI      ACCESS        ;
E65C CDF1E7  CALL     SCBITS        ;CHANGE DENSITY
E65F 05      DCR      B          ;DEC RETRY COUNT
E660 C247E6  JNZ     SLOOP         ; AND TRY AGAIN
E663 7A      MOV      A,D          ;RESTORE ERROR BITS

SERROR
E664 37      STC              ;ERROR FLAG
E665 F5      PUSH     PSW         ;SAVE THE STATUS
E666 CD4AE5  CALL     HENTRY        ;SEEK TO TRACK 0
E669 F1      POP      PSW         ;RECOVER ERRORS
E66A C9      RET              ;

RDHDR
E66B 060A    MVI      B,12Q        ;NUMBER OF RETRYS

RHLOOP
E66D 11FFE3  LXI      D,DATREG       ;DATA REGISTER
E670 21FAE6  LXI      H,TRACK+1     ;STORAGE AREA
E673 3EC4    MVI      A,RACMD        ;DO THE READ
E675 32FCE3  STA      CMDREG        ; HEADER COMMAND

RHL1
E678 1A      LDAX     D          ;GET A DATA BYTE
E679 77      MOV      M,A          ;STORE IN MEMORY
E67A 2C      INR      L          ;INC MEM POINTER
E67B C278E6  JNZ     RHL1          ;TEST FOR MORE DATA
E67E 21FCE3  LXI      H,CSTAT       ;WAIT FOR 1791
E681 CD87E7  CALL     BUSY          ;TO FINISH CMD
E684 B7      ORA       A          ;TEST FOR ERRORS
E685 CA8FE6  JZ       CHKSEC        ;TRANSFER OK?

```



```

E688 05          DCR      B          ;DEC RETRY COUNT
E689 C26DE6     JNZ      RHLOOP     ;TEST FOR
E68C C364E6     JMP      SERROR     ; HARD ERROR

                CHKSEC

E68F 3AFDE6     LDA      SECLN     ;GET THE SECTOR
E692 4F         MOV      C,A         ; SIZE AND SETUP
E693 0600       MVI      B,0         ; THE OFFSET
E695 21DFE6     LXI      H,STABLE   ;SEC SIZE TBL
E698 09         DAD      B          ;ADD THE OFFSET
E699 3AF8E6     LDA      SECTOR     ;GET THE SECTOR
E69C 47         MOV      B,A         ;SAVE IN B
E69D 86         ADD      M          ;COMPARE W/TABLE ENTRY
E69E 3E10       MVI      A,20Q       ;ERROR FLAG
E6A0 D8         RC          ;ERROR RETURN
E6A1 78         MOV      A,B         ;SAVE THE SECTOR
E6A2 32FEE3     STA      SECREG     ; IN SECTOR REG
E6A5 3E20       MVI      A,40Q       ;128 BYTE SECTOR
E6A7 210505     LXI      H,505H     ;INITIALIZE
E6AA 22E3E6     SHLD   ECOUNT     ; ERROR COUNTS

                SZLOOP

E6AD 0D         DCR      C          ;REDUCE SIZE COUNT
E6AE 47         MOV      B,A         ;SECTOR SIZE TO B
E6AF F8         RM          ;RETURN ON MINUS
E6B0 17         RAL       ;DOUBLE THE COUNT
E6B1 B7         ORA      A          ;CLEAR THE CARRY
E6B2 C3ADE6     JMP      SZLOOP

*
*
                SIDEFX

E6B5 79         MOV      A,C         ;GET THE SIDE BIT
E6B6 E601       ANI      1         ;TRIM EXCESS BITS
E6B8 17         RAL       ;MOVE THE BIT
E6B9 17         RAL       ; TO THE SIDE
E6BA 17         RAL       ; SELECT BIT
E6BB 17         RAL       ; POSITION
E6BC 32F7E6     STA      SIDE     ;SAVE
E6BF C9         RET

                TOEND
*
*
E6C0           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    200Q    ;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    10Q     ;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      SECLEN  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        ;WAIT
E7C5 A1      ANA        ; FOR NEXT
E7C6 C2BFE7  INDX3     JNZ      INDXCT ; LOW INDEX
E7C9 C9      RET        ;98 MACHINE CYCLES

```

```

*
*
DENFIX

```

```

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

```

```

*
*
SDSEL

```

```

E7E9 32FAE3  STA        ;DRIVE SELECT REG
E7EC C9      RET

```

```

LUSTAT

```

```

E7ED 3AF9E3  LDA        ;UART STATUS REG
E7F0 C9      RET

```

```

SCBITS

```

```

E7F1 32F9E3  STA        ;1791 CONTROL REG
E7F4 C9      RET

```

```

LDSTAT

```

```

E7F5 3AF9E3  LDA        ;DRIVE STATUS REG
E7F8 C9      RET

```

```

LOADS

```

```

E7F9 21F9E3  LXI        ;DRIVE STATUS REG
E7FC C9      RET

```

```

E7FD      ENDIF
          END

```

0002 ACCESS	E4A3 BDISK	E4A2 BDRVSEL	9000 BIAS	CD00 BIOS
E000 BOOTHD	E433 BOOT	E49B BTBLE	E49C BTIMER	E787 BUSY
E5D8 CBUSY	B700 CCP	E6EC CDISK	E04C CDMAHD	E77D CENTRY
E68F CHKSEC	E046 CLODHD	E02C CLOP	00D0 CLRCMD	E3FC CMDREG
E011 CMOVHD	E778 COMAND	E58E COMNDP	0004 COMPLT	E6FF CRCHI
E6FE CRCLO	E05F CRDHD	E062 CRHD	E3FC CSTAT	E1FE CSTKHD
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
E4E4 DISKST	E400 DJBOOT	E400 DJRAM	E51B DMA	E6E7 DMAADR
E52D DMASET	E424 DMAST	E508 DMSTAT	E415 DREAD	E3FA DREG
E510 DRIVE	0020 DRVRDY	E6EA DRVSEL	0008 DSIDE	0007 DSKCLK
E42A DSKERR	E72D DSROT	E43C DSRT	E3F9 DSTAT	E418 DWRITE
E6E3 ECOUNT	0001 FIRST	E4B2 FIXIO	0009 HCMD	E736 HDCHK
0051 HDCMND	0050 HDCNTL	0053 HDDATA	E6E9 HDFLAG	0052 HDFUNC
E700 HDLOAD	0050 HDORG	E0B6 HDPREP	E06F HDREAD	0051 HDRESLT
0004 HDRLEN	E0CA HDSEC	0050 HDSTAT	0004 HEAD	E54A HENTRY
E534 HOME	0000 IDBUFF	0010 INDEX	0040 INDX	E7B7 INDX1
E7BC INDX2	E7C6 INDX3	E7BF INDXCT	E7B5 INDXHI	E7BA INDXLO
E3F8 IO	0008 ISBUFF	E579 ISLOOP	E571 ISSUE	0004 ISTAT
E7F5 LDSTAT	E5FF LEAVE	001E LIGHT	0010 LOAD	E7F9 LOADS
E7ED LUSTAT	0002 MAXFLOP	0001 MAXHD	0003 MDINT	00F7 MDIR
E7AD MEASUR	E4C9 MODELA	E492 MODELM	0038 MSIZE	E782 NBUSY
003E NOLITE	00FB NSTEP	00FC NULL	2900 OFFSETB	0002 OPDONE
E000 ORIGIN	0008 OSTAT	E5FD PLEAVE	E60A PREP	E09C PROCESS
0004 PSTEP	00C4 RACMD	E400 RAM	0080 RCMD	E5C0 RENTRY
E66B RDHDR	E5C5 RDLOOP	E76A RDYCHK	E5BA READ	0020 READY
000A RETRIES	0002 RETRY	E678 RHL1	E66D RHLOOP	E000 ROM
0001 RSECT	0004 RSTBIT	E087 RTLOOP	E7F1 SCBITS	0005 SCENBL
E7E9 SDSEL	E6FD SECLN	0200 SECLN	E3FE SECREG	E55B SECSET
E6FC SECTNO	E6F8 SECTOR	E568 SEEK	E41B SELDRV	E664 SERROR
E42D SETDEN	E412 SETDMA	E40F SETSEC	E430 SETSID	E6B5 SIDEFX
E6F7 SIDE	E6FB SIDENO	0018 SKCMD	E647 SLOOP	E6DF STABLE
E6DF STACK	E427 STATUS	E5F4 STEST	001D SVCMD	E6AD SZLOOP
E403 TERMIN	E6E5 TIMER	0001 TKZ	E409 TKZERO	E763 TLOOP
0008 TMOUT	E6C0 TOEND	E41E TPANIC	E6F9 TRACK	E6FA TRKNO
E3FD TRKREG	E40C TRKSET	E406 TRMOUT	E421 TSTAT	E63E TVERFY
0004 TZERO	E6ED TZFLAG	E3F8 UDATA	0018 ULOAD	E76E UNLOAD
E3FA USTAT	00A0 WCMD	000F WENABL	0010 WFAULT	E59F WRETRY
000B WRESET	E599 WRITE	E5A4 WRLOOP	0005 WSECT	