```
                        PAGE 118,121
                        TITLE DSKETTE -- 04/21/86  DISKETTE BIOS
                        .286C
                        .LIST
                        SUBTTL (DSK1.ASM)
                        .LIST
                        ;-- INT 13 H ----------------------------------------------------------------
                        ; DISKETTE I/O
                        ;       THIS INTERFACE PROVIDES DISK ACCESS TO THE 5.25 INCH 360 KB,
                        ;       1.2 MB, 720 KB, AND 1.44 MB DISKETTE DRIVES.
                        ; INPUT
                        ;       (AH)=00H  RESET DISKETTE SYSTEM
                        ;                 HARD RESET TO NEC, PREPARE COMMAND, RECALIBRATE REQUIRED
                        ;                 ON ALL DRIVES
                        ;-------------------------------------------------------------------------------
                        ;       (AH)=01H  READ THE STATUS OF THE SYSTEM INTO (AH)
                        ;                 @DISKETTE_STATUS FROM LAST OPERATION IS USED
                        ;-------------------------------------------------------------------------------
                        ;       REGISTERS FOR READ/WRITE/VERIFY/FORMAT
                        ;       (DL) - DRIVE NUMBER (0-1 ALLOWED, VALUE CHECKED)
                        ;       (DH) - HEAD NUMBER (0-1 ALLOWED, NOT VALUE CHECKED)
                        ;       (CH) - TRACK NUMBER (NOT VALUE CHECKED)
                        ;                MEDIA      DRIVE          TRACK NUMBER
                        ;              320/360    320/360            0-39
                        ;              320/360     1.2M              0-39
                        ;               1.2M       1.2M              0-79
                        ;               720K       720K              0-79
                        ;               1.44M      1.44M             0-79
                        ;       (CL) - SECTOR NUMBER (NOT VALUE CHECKED, NOT USED FOR FORMAT)
                        ;                MEDIA      DRIVE          SECTOR NUMBER
                        ;              320/360    320/360            1-8/9
                        ;              320/360     1.2M              1-8/9
                        ;               1.2M       1.2M              1-15
                        ;               720K       720K              1-9
                        ;               1.44M      1.44M             1-18
                        ;       (AL) - NUMBER OF SECTORS (NOT VALUE CHECKED, NOT USED FOR FORMAT)
                        ;                MEDIA      DRIVE          MAX NUMBER OF SECTORS
                        ;              320/360    320/360              8/9
                        ;              320/360     1.2M                8/9
                        ;               1.2M       1.2M                15
                        ;               720K       720K                9
                        ;               1.44M      1.44M               18
                        ;
                        ;       (ES:BX) - ADDRESS OF BUFFER (NOT REQUIRED FOR VERIFY)
                        ;
                        ;-------------------------------------------------------------------------------
                        ;       (AH)=02H  READ THE DESIRED SECTORS INTO MEMORY
                        ;-------------------------------------------------------------------------------
                        ;       (AH)=03H  WRITE THE DESIRED SECTORS FROM MEMORY
                        ;-------------------------------------------------------------------------------
                        ;       (AH)=04H  VERIFY THE DESIRED SECTORS
                        ;-------------------------------------------------------------------------------
                        ;       (AH)=05H  FORMAT THE DESIRED TRACK
                        ;                 (ES:BX) MUST POINT TO THE COLLECTION OF DESIRED ADDRESS FIELDS
                        ;                 FOR THE TRACK.  EACH FIELD IS COMPOSED OF 4 BYTES, (C,H,R,N),
                        ;                 WHERE C = TRACK NUMBER, H=HEAD NUMBER, R = SECTOR NUMBER,
                        ;                 N= NUMBER OF BYTES PER SECTOR (00=128,01=256,02=512,03=1024).
                        ;                 THERE MUST BE ONE ENTRY FOR EVERY SECTOR ON THE TRACK.
                        ;                 THIS INFORMATION IS USED TO FIND THE REQUESTED SECTOR DURING
                        ;                 READ/WRITE ACCESS.
                        ;
                        ;                 PRIOR TO FORMATTING A DISKETTE, IF THERE EXISTS MORE THAN
                        ;                 ONE SUPPORTED MEDIA FORMAT TYPE WITHIN THE DRIVE IN QUESTION,
                        ;                 THEN "SET DASD TYPE" (INT 13H, AH = 17H) OR "SET MEDIA TYPE"
                        ;                 (INT 13H, AH = 18H) MUST BE CALLED TO SET THE DISKETTE TYPE
                        ;                 THAT IS TO BE FORMATED.  IF "SET DASD TYPE" OR "SET MEDIA TYPE"
                        ;                 IS NOT CALLED, THE FORMAT ROUTINE WILL ASSUME THE MEDIA FORMAT
                        ;                 TO BE THE MAXIMUM CAPACITY OF THE DRIVE.
                        ;
                        ;                 THESE PARAMETERS OF DISK_BASE MUST BE CHANGED IN ORDER TO
                        ;                 FORMAT THE FOLLOWING MEDIAS:
                        ;                 ---------------------------------------------
                        ;                 : MEDIA  :     DRIVE      : PARM 1 : PARM 2 :
                        ;                 ---------------------------------------------
                        ;                 : 320K   : 320K/360K/1.2M :  50H   :   8    :
                        ;                 : 360K   : 320K/360K/1.2M :  50H   :   9    :
                        ;                 : 1.2M   : 1.2M           :  54H   :  15    :
                        ;                 : 720K   : 720K/1.44M     :  50H   :   9    :
                        ;                 : 1.44M  : 1.44M          :  6CH   :  18    :
                        ;                 ---------------------------------------------
                        ;                 NOTES: - PARM 1 = GAP LENGTH FOR FORMAT
                        ;                        - PARM 2 = EOT (LAST SECTOR ON TRACK)
                        ;                        - DISK_BASE IS POINTED TO BY DISK POINTER LOCATED
                        ;                          AT ABSOLUTE ADDRESS 0:78.
                        ;                        - WHEN FORMAT OPERATIONS ARE COMPLETE, THE PARAMETERS
                        ;                          SHOULD BE RESTORED TO THEIR RESPECTIVE INITIAL VALUES
                        ;
                        ;-------------------------------------------------------------------------------
                        ;       (AH)=08H  READ DRIVE PARAMETERS
                        ;       REGISTERS
                        ;         INPUT
                        ;           (DL) - DRIVE NUMBER (0-1 ALLOWED, VALUE CHECKED)
                        ;         OUTPUT
                        ;           (ES:DI) POINTS TO DRIVE PARAMETERS TABLE
                        ;           (CH) - LOW ORDER 8 OF 10 BITS MAXIMUM NUMBER OF TRACKS
                        ;           (CL) - BITS 7 & 6 - HIGH ORDER TWO BITS OF MAXIMUM TRACKS
                        ;                - BITS 5 THRU 0 - MAXIMUM SECTORS PER TRACK
                        ;           (DH) - MAXIMUM HEAD NUMBER
                        ;           (DL) - NUMBER OF DISKETTE DRIVES INSTALLED
                        ;           (BH) - 0
                        ;           (BL) - BITS 7 THRU 4 - 0
                        ;                  BITS 3 THRU 0 - VALID DRIVE TYPE VALUE IN CMOS
                        ;           (AX) - 0
                        ;         UNDER THE FOLLOWING CIRCUMSTANCES:
                        ;           (1) THE DRIVE NUMBER IS INVALID,
                        ;           (2) THE DRIVE TYPE IS UNKNOWN AND CMOS IS NOT PRESENT,
                        ;           (3) THE DRIVE TYPE IS UNKNOWN AND CMOS IS BAD,
                        ;           (4) OR THE DRIVE TYPE IS UNKNOWN AND THE CMOS DRIVE TYPE IS INVALID
                        ;           THEN ES,AX,BX,CX,DH,DI=0 ; DL=NUMBER OF DRIVES.
                        ;           IF NO DRIVES ARE PRESENT THEN; ES,AX,BX,CX,DX,DI=0.
                        ;           @DSKETTE_STATUS = 0 AND CY IS RESET.
                        ;-------------------------------------------------------------------------------
                        ;       (AH)=15H READ DASD TYPE
                        ;       OUTPUT REGISTERS
```

```
                        ;       (AH) - ON RETURN IF CARRY FLAG NOT SET, OTHERWISE ERROR
                        ;               00 - DRIVE NOT PRESENT
                        ;               01 - DISKETTE, NO CHANGE LINE AVAILABLE
                        ;               02 - DISKETTE, CHANGE LINE AVAILABLE
                        ;               03 - RESERVED
                        ;       (DL) - DRIVE NUMBER (0-1 ALLOWED, VALUE CHECKED)
                        ;
                        ;--------------------------------------------------------------------------------
                        ;       (AH)=16H DISK CHANGE LINE STATUS
                        ;       OUTPUT REGISTERS
                        ;       (AH) - 00 - DISK CHANGE LINE NOT ACTIVE
                        ;              06 - DISK CHANGE LINE ACTIVE & CARRY BIT ON
                        ;       (DL) - DRIVE NUMBER (0-1 ALLOWED, VALUE CHECKED)
                        ;
                        ;--------------------------------------------------------------------------------
                        ;       (AH)=17H SET DASD TYPE FOR FORMAT
                        ;       INPUT REGISTERS
                        ;       (AL) -   00 - NOT USED
                        ;                01 - DISKETTE 320/360K IN 360K DRIVE
                        ;                02 - DISKETTE 360K IN 1.2M DRIVE
                        ;                03 - DISKETTE 1.2M IN 1.2M DRIVE
                        ;                04 - DISKETTE 720K IN 720K DRIVE
                        ;       (DL) - DRIVE NUMBER (0-1 ALLOWED, VALUE CHECKED;
                        ;              DO NOT USE WHEN DISKETTE ATTACH CARD USED)
                        ;--------------------------------------------------------------------------------
                        ;       (AH)=18H SET MEDIA TYPE FOR FORMAT
                        ;       INPUT REGISTERS
                        ;       (CH) - LOW ORDER 8 OF 10 BITS MAXIMUM NUMBER OF TRACKS
                        ;       (CL) - BITS 7 & 6 - HIGH ORDER TWO BITS OF MAXIMUM TRACKS
                        ;            - BITS 5 THRU 0 - MAXIMUM SECTORS PER TRACK
                        ;       (DL) - DRIVE NUMBER (0-1 ALLOWED, VALUE CHECKED)
                        ;       OUTPUT REGISTERS
                        ;       (ES:DI) - POINTER TO DRIVE PARAMETERS TABLE FOR THIS MEDIA TYPE,
                        ;                 UNCHANGED IF (AH) IS NON-ZERO
                        ;       (AH) - 00H, CY = 0, TRACK AND SECTORS/TRACK COMBINATION IS SUPPORTED
                        ;            - 01H, CY = 1, FUNCTION IS NOT AVAILABLE
                        ;            - 0CH, CY = 1, TRACK AND SECTORS/TRACK COMBINATION IS NOT SUPPORTED
                        ;                           OR DRIVE TYPE UNKNOWN
                        ;            - 80H, CY = 1, TIME OUT (DISKETTE NOT PRESENT)
                        ;--------------------------------------------------------------------------------
                        ;       DISK CHANGE STATUS IS ONLY CHECKED WHEN A MEDIA SPECIFIED IS OTHER
                        ;       THAN 360 KB DRIVE.  IF THE DISK CHANGE LINE IS FOUND TO BE
                        ;       ACTIVE THE FOLLOWING ACTIONS TAKE PLACE:
                        ;               ATTEMPT TO RESET DISK CHANGE LINE TO INACTIVE STATE.
                        ;               IF ATTEMPT SUCCEEDS SET DASD TYPE FOR FORMAT AND RETURN DISK
                        ;               CHANGE ERROR CODE
                        ;               IF ATTEMPT FAILS RETURN TIMEOUT ERROR CODE AND SET DASD TYPE
                        ;               TO A PREDETERMINED STATE INDICATING MEDIA TYPE UNKNOWN.
                        ;       IF THE DISK CHANGE LINE IN INACTIVE PERFORM SET DASD TYPE FOR FORMAT.
                        ;
                        ; DATA VARIABLE -- @DISK_POINTER
                        ;       DOUBLE WORD POINTER TO THE CURRENT SET OF DISKETTE PARAMETERS
                        ;--------------------------------------------------------------------------------
                        ; OUTPUT FOR ALL FUNCTIONS
                        ;       AH = STATUS OF OPERATION
                        ;               STATUS BITS ARE DEFINED IN THE EQUATES FOR @DISKETTE_STATUS
                        ;               VARIABLE IN THE DATA SEGMENT OF THIS MODULE
                        ;       CY = 0  SUCCESSFUL OPERATION (AH=0 ON RETURN, EXCEPT FOR READ DASD
                        ;               TYPE AH=(15)).
                        ;       CY = 1  FAILED OPERATION (AH HAS ERROR REASON)
                        ;       FOR READ/WRITE/VERIFY
                        ;               AL = COUNT OF SECTORS TRANSFERRED
                        ;               DS,BX,DX,CX PRESERVED
                        ;       NOTE: IF AN ERROR IS REPORTED BY THE DISKETTE CODE, THE APPROPRIATE
                        ;               ACTION IS TO RESET THE DISKETTE, THEN RETRY THE OPERATION.
                        ;               ON READ ACCESSES, NO MOTOR START DELAY IS TAKEN, SO THAT
                        ;               THREE RETRIES ARE REQUIRED ON READS TO ENSURE THAT THE
                        ;               PROBLEM IS NOT DUE TO MOTOR START-UP.
                        ;--------------------------------------------------------------------------------
                        .LIST
                        ; DISKETTE STATE MACHINE - ABSOLUTE ADDRESS 40:90 (DRIVE A) & 91 (DRIVE B)
                        .LIST
                        ;   ---------------------------------------------------------------
                        ;   |       |       |       |       |       |       |       |       |
                        ;   |   7   |   6   |   5   |   4   |   3   |   2   |   1   |   0   |
                        ;   |       |       |       |       |       |       |       |       |
                        ;   ---------------------------------------------------------------
                        ;       |       |       |       |       |       |       |       |
                        ;       |       |       |       |       |       |       ---------
                        ;       |       |       |       |       |       |
                        ;       |       |       |       |   RESERVED    |
                        ;       |       |       |       |               PRESENT STATE
                        ;       |       |       |       |       000: 360K IN 360K DRIVE UNESTABLISHED
                        ;       |       |       |       |       001: 360K IN 1.2M DRIVE UNESTABLISHED
                        ;       |       |       |       |       010: 1.2M IN 1.2M DRIVE UNESTABLISHED
                        ;       |       |       |       |       011: 360K IN 360K DRIVE ESTABLISHED
                        ;       |       |       |       |       100: 360K IN 1.2M DRIVE ESTABLISHED
                        ;       |       |       |       |       101: 1.2M IN 1.2M DRIVE ESTABLISHED
                        ;       |       |       |       |       110: RESERVED
                        ;       |       |       |       |       111: NONE OF THE ABOVE
                        ;       |       |       |       |
                        ;       |       |       |       ------> MEDIA/DRIVE ESTABLISHED
                        ;       |       |       |
                        ;       |       |       ---------------> DOUBLE STEPPING REQUIRED (360K IN 1.2M
                        ;       |       |                        DRIVE)
                        ;       |       |
                        ;       ---------------------------------> DATA TRANSFER RATE FOR THIS DRIVE:
                        ;
                        ;                                               00: 500 KBS
                        ;                                               01: 300 KBS
                        ;                                               10: 250 KBS
                        ;                                               11: RESERVED
                        ;
                        .LIST
                        ;--------------------------------------------------------------------------------
                        ; STATE OPERATION STARTED - ABSOLUTE ADDRESS 40:92 (DRIVE A) & 93 (DRIVE B)
                        ;--------------------------------------------------------------------------------
                        ; PRESENT CYLINDER NUMBER - ABSOLUTE ADDRESS 40:94 (DRIVE A) & 95 (DRIVE B)
                        ;--------------------------------------------------------------------------------
                        SUBTTL (DSK2.ASM)
```

```
                        PAGE

                        MD_STRUC        STRUC
0000 ??                 MD_SPEC1        DB      ?       ; SRT=D, HD UNLOAD=0F - 1ST SPECIFY BYTE
0001 ??                 MD_SPEC2        DB      ?       ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
0002 ??                 MD_OFF_TIM      DB      ?       ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
0003 ??                 MD_BYT_SEC      DB      ?       ; 512 BYTES/SECTOR
0004 ??                 MD_SEC_TRK      DB      ?       ; EOT ( LAST SECTOR ON TRACK)
0005 ??                 MD_GAP          DB      ?       ; GAP LENGTH
0006 ??                 MD_DTL          DB      ?       ; DTL
0007 ??                 MD_GAP3         DB      ?       ; GAP LENGTH FOR FORMAT
0008 ??                 MD_FIL_BYT      DB      ?       ; FILL BYTE FOR FORMAT
0009 ??                 MD_HD_TIM       DB      ?       ; HEAD SETTLE TIME (MILLISECONDS)
000A ??                 MD_STR_TIM      DB      ?       ; MOTOR START TIME (1/8 SECONDS)
000B ??                 MD_MAX_TRK      DB      ?       ; MAX. TRACK NUMBER
000C ??                 MD_RATE         DB      ?       ; DATA TRANSFER RATE
000D                    MD_STRUC        ENDS

= 007F                  BIT7OFF         EQU     7FH
= 0080                  BIT7ON          EQU     80H

                                PUBLIC  DISK_INT_1
                                PUBLIC  SEEK
                                PUBLIC  DSKETTE_SETUP
                                PUBLIC  DISKETTE_IO_1

                                EXTRN   CMOS_READ:NEAR
                                EXTRN   DDS:NEAR
                                EXTRN   DISK_BASE:NEAR
                                EXTRN   WAITF:NEAR


0000                    CODE    SEGMENT BYTE PUBLIC

                                ASSUME  CS:CODE,DS:DATA,ES:DATA

                        ;--------------------------------------------------------
                        ;       DRIVE TYPE TABLE                                :
                        ;--------------------------------------------------------
0000 01                 DR_TYPE         DB      01              ; DRIVE TYPE, MEDIA TABLE
0001 0012 R                             DW      OFFSET MD_TBL1
0003 82                                 DB      02+BIT7ON
0004 001F R                             DW      OFFSET MD_TBL2
0006 02                 DR_DEFAULT      DB      02
0007 002C R                             DW      OFFSET MD_TBL3
0009 03                                 DB      03
000A 0039 R                             DW      OFFSET MD_TBL4
000C 84                                 DB      04+BIT7ON
000D 0046 R                             DW      OFFSET MD_TBL5
000F 04                                 DB      04
0010 0053 R                             DW      OFFSET MD_TBL6
= 0012                  DR_TYPE_E       =$                      ; END OF TABLE
= 0006                  DR_CNT          EQU     (DR_TYPE_E-DR_TYPE)/3

                        ;--------------------------------------------------------
                        ;       MEDIA/DRIVE PARAMETER TABLES                    :
                        ;--------------------------------------------------------


                        ;--------------------------------------------------------
                        ;       360 KB MEDIA IN 360 KB DRIVE                    :
                        ;--------------------------------------------------------

0012                    MD_TBL1         LABEL BYTE
0012 DF                         DB      11011111B       ; SRT=D, HD UNLOAD=0F - 1ST SPECIFY BYTE
0013 02                         DB      2               ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
0014 25                         DB      MOTOR_WAIT      ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
0015 02                         DB      2               ; 512 BYTES/SECTOR
0016 09                         DB      09              ; EOT ( LAST SECTOR ON TRACK)
0017 2A                         DB      02AH            ; GAP LENGTH
0018 FF                         DB      0FFH            ; DTL
0019 50                         DB      050H            ; GAP LENGTH FOR FORMAT
001A F6                         DB      0F6H            ; FILL BYTE FOR FORMAT
001B 0F                         DB      15              ; HEAD SETTLE TIME (MILLISECONDS)
001C 08                         DB      8               ; MOTOR START TIME (1/8 SECONDS)
001D 27                         DB      39              ; MAX. TRACK NUMBER
001E 80                         DB      RATE_250        ; DATA TRANSFER RATE

                        ;--------------------------------------------------------
                        ;       360 KB MEDIA IN 1.2 MB DRIVE                    :
                        ;--------------------------------------------------------

001F                    MD_TBL2         LABEL BYTE
001F DF                         DB      11011111B       ; SRT=D, HD UNLOAD=0F - 1ST SPECIFY BYTE
0020 02                         DB      2               ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
0021 25                         DB      MOTOR_WAIT      ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
0022 02                         DB      2               ; 512 BYTES/SECTOR
0023 09                         DB      09              ; EOT ( LAST SECTOR ON TRACK)
0024 2A                         DB      02AH            ; GAP LENGTH
0025 FF                         DB      0FFH            ; DTL
0026 50                         DB      050H            ; GAP LENGTH FOR FORMAT
0027 F6                         DB      0F6H            ; FILL BYTE FOR FORMAT
0028 0F                         DB      15              ; HEAD SETTLE TIME (MILLISECONDS)
0029 08                         DB      8               ; MOTOR START TIME (1/8 SECONDS)
002A 27                         DB      39              ; MAX. TRACK NUMBER
002B 40                         DB      RATE_300        ; DATA TRANSFER RATE

                        ;--------------------------------------------------------
                        ;       1.2 MB MEDIA IN 1.2 MB DRIVE                    :
                        ;--------------------------------------------------------


002C                    MD_TBL3         LABEL BYTE
002C DF                         DB      11011111B       ; SRT=D, HD UNLOAD=0F - 1ST SPECIFY BYTE
002D 02                         DB      2               ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
002E 25                         DB      MOTOR_WAIT      ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
002F 02                         DB      2               ; 512 BYTES/SECTOR
0030 0F                         DB      15              ; EOT ( LAST SECTOR ON TRACK)
0031 1B                         DB      01BH            ; GAP LENGTH
0032 FF                         DB      0FFH            ; DTL
0033 54                         DB      054H            ; GAP LENGTH FOR FORMAT
0034 F6                         DB      0F6H            ; FILL BYTE FOR FORMAT
0035 0F                         DB      15              ; HEAD SETTLE TIME (MILLISECONDS)
0036 08                         DB      8               ; MOTOR START TIME (1/8 SECONDS)
0037 4F                         DB      79              ; MAX. TRACK NUMBER
0038 00                         DB      RATE_500        ; DATA TRANSFER RATE
```

```
                        ;--------------------------------------------------------
                        ;       720 KB MEDIA IN 720 KB DRIVE                    :
                        ;--------------------------------------------------------

0039                    MD_TBL4         LABEL BYTE
0039 DF                         DB      11011111B       ; SRT=D, HD UNLOAD=0F - 1ST SPECIFY BYTE
003A 02                         DB      2               ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
003B 25                         DB      MOTOR_WAIT      ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
003C 02                         DB      2               ; 512 BYTES/SECTOR
003D 09                         DB      09              ; EOT ( LAST SECTOR ON TRACK)
003E 2A                         DB      02AH            ; GAP LENGTH
003F FF                         DB      0FFH            ; DTL
0040 50                         DB      050H            ; GAP LENGTH FOR FORMAT
0041 F6                         DB      0F6H            ; FILL BYTE FOR FORMAT
0042 0F                         DB      15              ; HEAD SETTLE TIME (MILLISECONDS)
0043 08                         DB      8               ; MOTOR START TIME (1/8 SECONDS)
0044 4F                         DB      79              ; MAX. TRACK NUMBER
0045 80                         DB      RATE_250        ; DATA TRANSFER RATE

                        ;--------------------------------------------------------
                        ;       720 KB MEDIA IN 1.44 MB DRIVE                   :
                        ;--------------------------------------------------------

0046                    MD_TBL5         LABEL BYTE
0046 DF                         DB      11011111B       ; SRT=D, HD UNLOAD=0F - 1ST SPECIFY BYTE
0047 02                         DB      2               ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
0048 25                         DB      MOTOR_WAIT      ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
0049 02                         DB      2               ; 512 BYTES/SECTOR
004A 09                         DB      09              ; EOT ( LAST SECTOR ON TRACK)
004B 2A                         DB      02AH            ; GAP LENGTH
004C FF                         DB      0FFH            ; DTL
004D 50                         DB      050H            ; GAP LENGTH FOR FORMAT
004E F6                         DB      0F6H            ; FILL BYTE FOR FORMAT
004F 0F                         DB      15              ; HEAD SETTLE TIME (MILLISECONDS)
0050 08                         DB      8               ; MOTOR START TIME (1/8 SECONDS)
0051 4F                         DB      79              ; MAX. TRACK NUMBER
0052 80                         DB      RATE_250        ; DATA TRANSFER RATE

                        ;--------------------------------------------------------
                        ;       1.44 MB MEDIA IN 1.44 MB DRIVE                  :
                        ;--------------------------------------------------------

0053                    MD_TBL6         LABEL BYTE
0053 AF                         DB      10101111B       ; SRT=A, HD UNLOAD=0F - 1ST SPECIFY BYTE
0054 02                         DB      2               ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
0055 25                         DB      MOTOR_WAIT      ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
0056 02                         DB      2               ; 512 BYTES/SECTOR
0057 12                         DB      18              ; EOT ( LAST SECTOR ON TRACK)
0058 1B                         DB      01BH            ; GAP LENGTH
0059 FF                         DB      0FFH            ; DTL
005A 6C                         DB      06CH            ; GAP LENGTH FOR FORMAT
005B F6                         DB      0F6H            ; FILL BYTE FOR FORMAT
005C 0F                         DB      15              ; HEAD SETTLE TIME (MILLISECONDS)
005D 08                         DB      8               ; MOTOR START TIME (1/8 SECONDS)
005E 4F                         DB      79              ; MAX. TRACK NUMBER
005F 00                         DB      RATE_500        ; DATA TRANSFER RATE


0060                    DISKETTE_IO_1   PROC    FAR             ;>>> ENTRY POINT FOR ORG 0EC59H

0060 FB                         STI                             ; INTERRUPTS BACK ON
0061 55                         PUSH    BP                      ; USER REGISTER
0062 57                         PUSH    DI                      ; USER REGISTER
0063 52                         PUSH    DX                      ; HEAD #, DRIVE # OR USER REGISTER
0064 53                         PUSH    BX                      ; BUFFER OFFSET PARAMETER OR REGISTER
0065 51                         PUSH    CX                      ; TRACK #-SECTOR # OR USER REGISTER
0066 8B EC                      MOV     BP,SP                   ; BP    => PARAMETER LIST DEP. ON AH
                                                                ; [BP]  = SECTOR #
                                                                ; [BP+1] = TRACK #
                                                                ; [BP+2] = BUFFER OFFSET
                                                                ; FOR RETURN OF DRIVE PARAMETERS:
                                                                ; CL/[BP] = BITS 7&6 HI BITS OF MAX CYL
                                                                ;           BITS 0-5 MAX SECTORS/TRACK
                                                                ; CH/[BP+1] = LOW 8 BITS OF MAX CYL.
                                                                ; BL/[BP+2] = BITS 7-4 = 0
                                                                ;             BITS 3-0 = VALID CMOS TYPE
                                                                ; BH/[BP+3] = 0
                                                                ; DL/[BP+4] = # DRIVES INSTALLED
                                                                ; DH/[BP+5] = MAX HEAD #
                                                                ; DI/[BP+6] = OFFSET TO DISK BASE
0068 1E                         PUSH    DS                      ; BUFFER SEGMENT PARM OR USER REGISTER
0069 56                         PUSH    SI                      ; USER REGISTERS
006A E8 0000 E                  CALL    DDS                     ; SEGMENT OF BIOS DATA AREA TO DS
006D 80 FC 19                   CMP     AH,(FNC_TAE-FNC_TAB)/2  ; CHECK FOR > LARGEST FUNCTION
0070 72 02                      JB      OK_FUNC                 ; FUNCTION OK
0072 B4 14                      MOV     AH,14H                  ; REPLACE WITH KNOWN INVALID FUNCTION

0074                    OK_FUNC:
0074 80 FC 01                   CMP     AH,1                    ; RESET OR STATUS ?
0077 76 0C                      JBE     OK_DRV                  ; IF RESET OR STATUS DRIVE ALWAYS OK
0079 80 FC 08                   CMP     AH,8                    ; READ DRIVE PARMS ?
007C 74 07                      JZ      OK_DRV                  ; IF SO DRIVE CHECKED LATER
007E 80 FA 01                   CMP     DL,1                    ; DRIVES 0 AND 1 OK
0081 76 02                      JBE     OK_DRV                  ; IF 0 OR 1 THEN JUMP
0083 B4 14                      MOV     AH,14H                  ; REPLACE WITH KNOWN INVALID FUNCTION

0085                    OK_DRV:
0085 8A CC                      MOV     CL,AH                   ; CL = FUNCTION
0087 32 ED                      XOR     CH,CH                   ; CX = FUNCTION
0089 D0 E1                      SHL     CL,1                    ; FUNCTION TIMES 2
008B BB 00B5 R                  MOV     BX,OFFSET FNC_TAB       ; LOAD START OF FUNCTION TABLE
008E 03 D9                      ADD     BX,CX                   ; ADD OFFSET INTO TABLE => ROUTINE
0090 8A E6                      MOV     AH,DH                   ; AX = HEAD #,# OF SECTORS OR DASD TYPE
0092 32 F6                      XOR     DH,DH                   ; DX = DRIVE #
0094 8B F0                      MOV     SI,AX                   ; SI = HEAD #,# OF SECTORS OR DASD TYPE
0096 8B FA                      MOV     DI,DX                   ; DI = DRIVE #
0098 8A 26 0041 R               MOV     AH,@DSKETTE_STATUS      ; LOAD STATUS TO AH FOR STATUS FUNCTION
009C C6 06 0041 R 00            MOV     @DSKETTE_STATUS,0       ; INITIALIZE FOR ALL OTHERS

                        ;       THROUGHOUT THE DISKETTE BIOS, THE FOLLOWING INFORMATION IS CONTAINED IN
                        ;       THE FOLLOWING MEMORY LOCATIONS AND REGISTERS. NOT ALL DISKETTE BIOS
                        ;       FUNCTIONS REQUIRE ALL OF THESE PARAMETERS.
                        ;
```

```
                        ;               DI      : DRIVE #
                        ;               SI-HI   : HEAD #
                        ;               SI-LOW  : # OF SECTORS OR DASD TYPE FOR FORMAT
                        ;               ES      : BUFFER SEGMENT
                        ;               [BP]    : SECTOR #
                        ;               [BP+1]  : TRACK #
                        ;               [BP+2]  : BUFFER OFFSET
                        ;
                        ;       ACROSS CALLS TO SUBROUTINES THE CARRY FLAG (CY=1), WHERE INDICATED IN
                        ;       SUBROUTINE PROLOGUES, REPRESENTS AN EXCEPTION RETURN (NORMALLY AN ERROR
                        ;       CONDITION). IN MOST CASES, WHEN CY = 1, @DSKETTE_STATUS CONTAINS THE
                        ;       SPECIFIC ERROR CODE.
                                                                ; (AH) = @DSKETTE_STATUS
00A1 2E: FF 17                  CALL    WORD PTR CS:[BX]        ; CALL THE REQUESTED FUNCTION
                        ;
00A4 5E                         POP     SI                      ; RESTORE ALL REGISTERS
00A5 1F                         POP     DS
00A6 59                         POP     CX
00A7 5B                         POP     BX
00A8 5A                         POP     DX
00A9 5F                         POP     DI
00AA 8B EC                      MOV     BP,SP
00AC 50                         PUSH    AX
00AD 9C                         PUSHF
00AE 58                         POP     AX
00AF 89 46 06                   MOV     [BP+6],AX
00B2 58                         POP     AX
00B3 5D                         POP     BP
00B4 CF                         IRET
                        ;
                        ;--------------------------------------------------------------
00B5 00E7 R             FNC_TAB DW      DISK_RESET              ; AH = 00H; RESET
00B7 013C R                     DW      DISK_STATUS             ; AH = 01H; STATUS
00B9 0148 R                     DW      DISK_READ               ; AH = 02H; READ
00BB 0154 R                     DW      DISK_WRITE              ; AH = 03H; WRITE
00BD 0160 R                     DW      DISK_VERF               ; AH = 04H; VERIFY
00BF 016C R                     DW      DISK_FORMAT             ; AH = 05H; FORMAT
00C1 01C6 R                     DW      FNC_ERR                 ; AH = 06H; INVALID
00C3 01C6 R                     DW      FNC_ERR                 ; AH = 07H; INVALID
00C5 01D0 R                     DW      DISK_PARMS              ; AH = 08H; READ DRIVE PARAMETERS
00C7 01C6 R                     DW      FNC_ERR                 ; AH = 09H; INVALID
00C9 01C6 R                     DW      FNC_ERR                 ; AH = 0AH; INVALID
00CB 01C6 R                     DW      FNC_ERR                 ; AH = 0BH; INVALID
00CD 01C6 R                     DW      FNC_ERR                 ; AH = 0CH; INVALID
00CF 01C6 R                     DW      FNC_ERR                 ; AH = 0DH; INVALID
00D1 01C6 R                     DW      FNC_ERR                 ; AH = 0EH; INVALID
00D3 01C6 R                     DW      FNC_ERR                 ; AH = 0FH; INVALID
00D5 01C6 R                     DW      FNC_ERR                 ; AH = 10H; INVALID
00D7 01C6 R                     DW      FNC_ERR                 ; AH = 11H; INVALID
00D9 01C6 R                     DW      FNC_ERR                 ; AH = 12H; INVALID
00DB 01C6 R                     DW      FNC_ERR                 ; AH = 13H; INVALID
00DD 01C6 R                     DW      FNC_ERR                 ; AH = 14H; INVALID
00DF 0289 R                     DW      DISK_TYPE               ; AH = 15H; READ DASD TYPE
00E1 02AB R                     DW      DISK_CHANGE             ; AH = 16H; CHANGE STATUS
00E3 02D6 R                     DW      FORMAT_SET              ; AH = 17H; SET DASD TYPE
00E5 0337 R                     DW      SET_MEDIA               ; AH = 18H; SET MEDIA TYPE
= 00E7                  FNC_TAE EQU     $                       ; END
00E7                    DISKETTE_IO_1   ENDP
                        ;--------------------------------------------------------------
                        ; DISK_RESET  (AH = 00H)                                      :
                        ;       RESET THE DISKETTE SYSTEM.                            :
                        ;                                                             :
                        ; ON EXIT:     @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION :
                        ;--------------------------------------------------------------
00E7                    DISK_RESET      PROC    NEAR
00E7 BA 03F2                    MOV     DX,03F2H                ; ADAPTER CONTROL PORT
00EA FA                         CLI                             ; NO INTERRUPTS
00EB A0 003F R                  MOV     AL,@MOTOR_STATUS        ; GET DIGITAL OUTPUT REGISTER REFLECTION
00EE 24 3F                      AND     AL,00111111B            ; KEEP SELECTED AND MOTOR ON BITS
00F0 C0 C0 04                   ROL     AL,4                    ; MOTOR VALUE TO HIGH NIBBLE
                                                                ; DRIVE SELECT TO LOW NIBBLE
00F3 0C 08                      OR      AL,00001000B            ; TURN ON INTERRUPT ENABLE
00F5 EE                         OUT     DX,AL                   ; RESET THE ADAPTER
00F6 C6 06 003E R 00            MOV     @SEEK_STATUS,0          ; SET RECALIBRATE REQUIRED ON ALL DRIVES
00FB EB 00                      JMP     $+2                     ; WAIT FOR I/O
00FD EB 00                      JMP     $+2                     ; WAIT FOR I/O (TO INSURE MINIMUM
                                                                ;   PULSE WIDTH)
00FF 0C 04                      OR      AL,00000100B            ; TURN OFF RESET BIT
0101 EE                         OUT     DX,AL                   ; RESET THE ADAPTER
0102 FB                         STI                             ; ENABLE THE INTERRUPTS
0103 E8 0AC1 R                  CALL    WAIT_INT                ; WAIT FOR THE INTERRUPT
0106 72 2D                      JC      DR_ERR                  ; IF ERROR, RETURN IT
0108 B9 00C0                    MOV     CX,11000000B            ; CL = EXPECTED @NEC_STATUS
                        ;
010B                    NXT_DRV:
010B 51                         PUSH    CX                      ; SAVE FOR CALL
010C BB 0134 R                  MOV     AX,OFFSET DR_POP_ERR    ; LOAD NEC_OUTPUT ERROR ADDRESS
010F 50                         PUSH    AX                      ; "
0110 B4 08                      MOV     AH,08H                  ; SENSE INTERRUPT STATUS COMMAND
0112 E8 09F8 R                  CALL    NEC_OUTPUT
0115 58                         POP     AX                      ; THROW AWAY ERROR RETURN
0116 E8 0AE9 R                  CALL    RESULTS                 ; READ IN THE RESULTS
0119 59                         POP     CX                      ; RESTORE AFTER CALL
011A 72 19                      JC      DR_ERR                  ; ERROR RETURN
011C 3A 0E 0042 R               CMP     CL,@NEC_STATUS          ; TEST FOR DRIVE READY TRANSITION
0120 75 13                      JNZ     DR_ERR                  ; EVERYTHING OK
0122 FE C1                      INC     CL                      ; NEXT EXPECTED @NEC_STATUS
0124 80 F9 C3                   CMP     CL,11000011B            ; ALL POSSIBLE DRIVES CLEARED
0127 76 E2                      JBE     NXT_DRV                 ; FALL THRU IF 11000100B OR >
                        ;
0129 E8 03D8 R                  CALL    SEND_SPEC               ; SEND SPECIFY COMMAND TO NEC
                        ;
012C                    RESBAC:
012C E8 0854 R                  CALL    SETUP_END               ; VARIOUS CLEANUPS
012F 8B DE                      MOV     BX,SI                   ; GET SAVED AL TO BL
0131 8A C3                      MOV     AL,BL                   ; PUT BACK FOR RETURN
0133 C3                         RET
                        ;
0134                    DR_POP_ERR:
0134 59                         POP     CX                      ; CLEAR STACK
0135                    DR_ERR:                                 ;
0135 80 0E 0041 R 20            OR      @DSKETTE_STATUS,BAD_NEC ; SET ERROR CODE
013A EB F0                      JMP     SHORT RESBAC            ; RETURN FROM RESET
013C                    DISK_RESET      ENDP
```

```
                        ;--------------------------------------------------------
                        ; DISK_STATUS  (AH = 01H)                               :
                        ;       DISKETTE STATUS.                                :
                        ; ON ENTRY:     AH = STATUS OF PREVIOUS OPERATION       :
                        ;                                                       :
                        ; ON EXIT:      @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION :
                        ;--------------------------------------------------------
013C                    DISK_STATUS     PROC    NEAR
013C 88 26 0041 R               MOV     @DSKETTE_STATUS,AH      ; PUT BACK FOR SETUP_END
0140 E8 0854 R                  CALL    SETUP_END               ; VARIOUS CLEANUPS
0143 8B DE                      MOV     BX,SI                   ; GET SAVED AL TO BL
0145 8A C3                      MOV     AL,BL                   ; PUT BACK FOR RETURN
0147 C3                         RET
0148                    DISK_STATUS     ENDP
                        ;--------------------------------------------------------
                        ; DISK_READ  (AH = 02H)                                 :
                        ;       DISKETTE READ.                                  :
                        ; ON ENTRY:     DI      = DRIVE #                       :
                        ;               SI-HI   = HEAD #                        :
                        ;               SI-LOW  = # OF SECTORS                  :
                        ;               ES      = BUFFER SEGMENT                :
                        ;               [BP]    = SECTOR #                      :
                        ;               [BP+1]  = TRACK #                       :
                        ;               [BP+2]  = BUFFER OFFSET                 :
                        ;                                                       :
                        ; ON EXIT:      @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION :
                        ;--------------------------------------------------------
0148                    DISK_READ       PROC    NEAR
0148 80 26 003F R 7F            AND     @MOTOR_STATUS,01111111B ; INDICATE A READ OPERATION
014D B8 E646                    MOV     AX,0E646H               ; AX = NEC COMMAND, DMA COMMAND
0150 E8 04AE R                  CALL    RD_WR_VF                ; COMMON READ/WRITE/VERIFY
0153 C3                         RET
0154                    DISK_READ       ENDP
                        ;--------------------------------------------------------
                        ; DISK_WRITE  (AH = 03H)                                :
                        ;       DISKETTE WRITE.                                 :
                        ; ON ENTRY:     DI      = DRIVE #                       :
                        ;               SI-HI   = HEAD #                        :
                        ;               SI-LOW  = # OF SECTORS                  :
                        ;               ES      = BUFFER SEGMENT                :
                        ;               [BP]    = SECTOR #                      :
                        ;               [BP+1]  = TRACK #                       :
                        ;               [BP+2]  = BUFFER OFFSET                 :
                        ;                                                       :
                        ; ON EXIT:      @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION :
                        ;--------------------------------------------------------
0154                    DISK_WRITE      PROC    NEAR
0154 B8 C54A                    MOV     AX,0C54AH               ; AX = NEC COMMAND, DMA COMMAND
0157 80 0E 003F R 80            OR      @MOTOR_STATUS,10000000B ; INDICATE WRITE OPERATION
015C E8 04AE R                  CALL    RD_WR_VF                ; COMMON READ/WRITE/VERIFY
015F C3                         RET
0160                    DISK_WRITE      ENDP
                        ;--------------------------------------------------------
                        ; DISK_VERF  (AH = 04H)                                 :
                        ;       DISKETTE VERIFY.                                :
                        ; ON ENTRY:     DI      = DRIVE #                       :
                        ;               SI-HI   = HEAD #                        :
                        ;               SI-LOW  = # OF SECTORS                  :
                        ;               ES      = BUFFER SEGMENT                :
                        ;               [BP]    = SECTOR #                      :
                        ;               [BP+1]  = TRACK #                       :
                        ;               [BP+2]  = BUFFER OFFSET                 :
                        ;                                                       :
                        ; ON EXIT:      @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION :
                        ;--------------------------------------------------------
0160                    DISK_VERF       PROC    NEAR
0160 80 26 003F R 7F            AND     @MOTOR_STATUS,01111111B ; INDICATE A READ OPERATION
0165 B8 E642                    MOV     AX,0E642H               ; AX = NEC COMMAND, DMA COMMAND
0168 E8 04AE R                  CALL    RD_WR_VF                ; COMMON READ/WRITE/VERIFY
016B C3                         RET
016C                    DISK_VERF       ENDP
                        ;--------------------------------------------------------
                        ; DISK_FORMAT  (AH = 05H)                               :
                        ;       DISKETTE FORMAT.                                :
                        ; ON ENTRY:     DI      = DRIVE #                       :
                        ;               SI-HI   = HEAD #                        :
                        ;               SI-LOW  = # OF SECTORS                  :
                        ;               ES      = BUFFER SEGMENT                :
                        ;               [BP]    = SECTOR #                      :
                        ;               [BP+1]  = TRACK #                       :
                        ;               [BP+2]  = BUFFER OFFSET                 :
                        ;               @DISK_POINTER POINTS TO THE PARAMETER TABLE OF  :
                        ;                       THIS DRIVE                      :
                        ;                                                       :
                        ; ON EXIT:      @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION :
                        ;--------------------------------------------------------
016C                    DISK_FORMAT     PROC    NEAR
016C E8 040F R                  CALL    XLAT_NEW                ; TRANSLATE STATE TO PRESENT ARCH.
016F E8 059B R                  CALL    FMT_INIT                ; ESTABLISH STATE IF UNESTABLISHED
0172 80 0E 003F R 80            OR      @MOTOR_STATUS,10000000B ; INDICATE WRITE OPERATION
0177 E8 05E9 R                  CALL    MED_CHANGE              ; CHECK MEDIA CHANGE AND RESET IF SO
017A 72 3F                      JC      FM_DON                  ; MEDIA CHANGED, SKIP
017C E8 03D8 R                  CALL    SEND_SPEC               ; SEND SPECIFY COMMAND TO NEC
017F E8 063D R                  CALL    CHK_LASTRATE            ; ZF=1 ATTEMPT RATE IS SAME AS LAST RATE
0182 74 03                      JZ      FM_WR                   ; YES, SKIP SPECIFY COMMAND
0184 E8 0624 R                  CALL    SEND_RATE               ; SEND DATA RATE TO CONTROLLER
0187                    FM_WR:
0187 E8 06AD R                  CALL    FMTDMA_SET              ; SET UP THE DMA FOR FORMAT
018A 72 2F                      JC      FM_DON                  ; RETURN WITH ERROR
018C B4 4D                      MOV     AH,04DH                 ; ESTABLISH THE FORMAT COMMAND
018E E8 0700 R                  CALL    NEC_INIT                ; INITIALIZE THE NEC
0191 72 28                      JC      FM_DON                  ; ERROR - EXIT
0193 B8 01BB R                  MOV     AX,OFFSET FM_DON        ; LOAD ERROR ADDRESS
0196 50                         PUSH    AX                      ; PUSH NEC_OUT ERROR RETURN
0197 B2 03                      MOV     DL,3                    ; BYTES/SECTOR VALUE TO NEC
0199 E8 0905 R                  CALL    GET_PARM
019C E8 09F8 R                  CALL    NEC_OUTPUT
019F B2 04                      MOV     DL,4                    ; SECTORS/TRACK VALUE TO NEC
01A1 E8 0905 R                  CALL    GET_PARM
01A4 E8 09F8 R                  CALL    NEC_OUTPUT
01A7 B2 07                      MOV     DL,7                    ; GAP LENGTH VALUE TO NEC
01A9 E8 0905 R                  CALL    GET_PARM
01AC E8 09F8 R                  CALL    NEC_OUTPUT
```

```
01AF B2 08                      MOV     DL,8                    ; FILLER BYTE TO NEC
01B1 E8 0905 R                  CALL    GET_PARM
01B4 E8 09F8 R                  CALL    NEC_OUTPUT
01B7 58                         POP     AX                      ; THROW AWAY ERROR
01B8 E8 075B R                  CALL    NEC_TERM                ; TERMINATE, RECEIVE STATUS, ETC.
01BB                    FM_DON:
01BB E8 0435 R                  CALL    XLAT_OLD                ; TRANSLATE STATE TO COMPATIBLE MODE
01BE E8 0854 R                  CALL    SETUP_END               ; VARIOUS CLEANUPS
01C1 8B DE                      MOV     BX,SI                   ; GET SAVED AL TO BL
01C3 8A C3                      MOV     AL,BL                   ; PUT BACK FOR RETURN
01C5 C3                         RET
01C6                    DISK_FORMAT     ENDP
                        ;--------------------------------------------------------------------
                        ; FNC_ERR                                                          :
                        ;       INVALID FUNCTION REQUESTED OR INVALID DRIVE;               :
                        ;       SET BAD COMMAND IN STATUS.                                 :
                        ;                                                                  :
                        ; ON EXIT:       @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION :
                        ;--------------------------------------------------------------------
01C6                    FNC_ERR PROC    NEAR                    ; INVALID FUNCTION REQUEST
01C6 8B C6                      MOV     AX,SI                   ; RESTORE AL
01C8 B4 01                      MOV     AH,BAD_CMD              ; SET BAD COMMAND ERROR
01CA 88 26 0041 R               MOV     @DSKETTE_STATUS,AH      ; STORE IN DATA AREA
01CE F9                         STC                             ; SET CARRY INDICATING ERROR
01CF C3                         RET
01D0                    FNC_ERR ENDP
                        ;--------------------------------------------------------------------
                        ; DISK_PARMS  (AH = 08H)                                           :
                        ;       READ DRIVE PARAMETERS.                                     :
                        ; ON ENTRY:                                                        :
                        ;       DI = DRIVE #                                               :
                        ; ON EXIT:                                                         :
                        ;       CL/[BP]    = BITS 7 & 6 HIGH 2 BITS OF MAX CYLINDER        :
                        ;                    BITS 0-5 MAX SECTORS/TRACK                    :
                        ;       CH/[BP+1]  = LOW 8 BITS OF MAX CYLINDER                    :
                        ;       BL/[BP+2]  = BITS 7-4 = 0                                  :
                        ;                    BITS 3-0 = VALID CMOS DRIVE TYPE              :
                        ;       BH/[BP+3]  = 0                                             :
                        ;       DL/[BP+4]  = # DRIVES INSTALLED                            :
                        ;       DH/[BP+5]  = MAX HEAD #                                    :
                        ;       DI/[BP+6]  = OFFSET OF MEDIA/DRIVE PARAMETER TABLE         :
                        ;       ES         = SEGMENT OF MEDIA/DRIVE PARAMETER TABLE        :
                        ;       AX         = 0                                             :
                        ;                                                                  :
                        ; NOTE : THE ABOVE INFORMATION IS STORED IN THE USERS STACK AT     :
                        ;        THE LOCATIONS WHERE THE MAIN ROUTINE WILL POP THEM        :
                        ;        INTO THE APPROPRIATE REGISTERS BEFORE RETURNING TO THE    :
                        ;        CALLER.                                                   :
                        ;--------------------------------------------------------------------
01D0                    DISK_PARMS      PROC    NEAR
01D0 E8 040F R                  CALL    XLAT_NEW                ; TRANSLATE STATE TO PRESENT ARCH.
01D3 C7 46 02 0000              MOV     WORD PTR [BP+2],0       ; DRIVE TYPE = 0
01D8 A1 0010 R                  MOV     AX,@EQUIP_FLAG          ; LOAD EQUIPMENT FLAG FOR # DISKETTES
01DB 24 C1                      AND     AL,11000001B            ; KEEP DISKETTE DRIVE BITS
01DD B2 02                      MOV     DL,2                    ; DISKETTE DRIVES = 2
01DF 3C 41                      CMP     AL,01000001B            ; 2 DRIVES INSTALLED ?
01E1 74 06                      JZ      STO_DL                  ; IF YES JUMP

01E3 FE CA                      DEC     DL                      ; DISKETTE DRIVES = 1
01E5 3C 01                      CMP     AL,00000001B            ; 1 DRIVE INSTALLED ?
01E7 75 6A                      JNZ     NON_DRV                 ; IF NO JUMP

01E9 88 56 04           STO_DL: MOV     [BP+4],DL               ; STORE NUMBER OF DRIVES
01EC 83 FF 01                   CMP     DI,1                    ; CHECK FOR VALID DRIVE
01EF 77 66                      JA      NON_DRV1                ; DRIVE INVALID
01F1 C6 46 05 01                MOV     BYTE PTR[BP+5],1        ; MAXIMUM HEAD NUMBER = 1
01F5 E8 08EC R                  CALL    CMOS_TYPE               ; RETURN DRIVE TYPE IN AL
01F8 72 16                      JC      CHK_EST                 ; ON CMOS BAD CHECK ESTABLISHED
01FA 0A C0                      OR      AL,AL                   ; TEST FOR NO DRIVE TYPE
01FC 74 12                      JZ      CHK_EST                 ; JUMP IF SO
01FE E8 03B8 R                  CALL    DR_TYPE_CHECK           ; RTN CS:BX = MEDIA/DRIVE PARAM TBL
0201 72 0D                      JC      CHK_EST                 ; TYPE NOT IN TABLE (POSSIBLE BAD CMOS)
0203 88 46 02                   MOV     [BP+2],AL               ; STORE VALID CMOS DRIVE TYPE
0206 2E: 8A 4F 04               MOV     CL,CS:[BX].MD_SEC_TRK   ; GET SECTOR/TRACK
020A 2E: 8A 6F 0B               MOV     CH,CS:[BX].MD_MAX_TRK   ; GET MAX. TRACK NUMBER
020E EB 32                      JMP     SHORT STO_CX            ; CMOS GOOD, USE CMOS

0210                    CHK_EST:
0210 8A A5 0090 R               MOV     AH,@DSK_STATE[DI]       ; LOAD STATE FOR THIS DRIVE
0214 F6 C4 10                   TEST    AH,MED_DET              ; CHECK FOR ESTABLISHED STATE
0217 74 3E                      JZ      NON_DRV1                ; CMOS BAD/INVALID AND UNESTABLISHED

0219                    USE_EST:
0219 80 E4 C0                   AND     AH,RATE_MSK             ; ISOLATE STATE
021C 80 FC 80                   CMP     AH,RATE_250             ; RATE 250 ?
021F 75 54                      JNE     USE_EST2                ; NO, GO CHECK OTHER RATE

                        ;--- DATA RATE IS 250 KBS, TRY 360 KB TABLE FIRST

0221 B0 01                      MOV     AL,01                   ; DRIVE TYPE 1 (360KB)
0223 E8 03B8 R                  CALL    DR_TYPE_CHECK           ; RTN CS:BX = MEDIA/DRIVE PARAM TBL
0226 2E: 8A 4F 04               MOV     CL,CS:[BX].MD_SEC_TRK   ; GET SECTOR/TRACK
022A 2E: 8A 6F 0B               MOV     CH,CS:[BX].MD_MAX_TRK   ; GET MAX. TRACK NUMBER
022E F6 85 0090 R 01            TEST    @DSK_STATE[DI],TRK_CAPA ; 80 TRACK ?
0233 74 0D                      JZ      STO_CX                  ; MUST BE 360KB DRIVE

                        ;--- IT IS 1.44 MB DRIVE

0235                    PARM144:
0235 B0 04                      MOV     AL,04                   ; DRIVE TYPE 4 (1.44MB)
0237 E8 03B8 R                  CALL    DR_TYPE_CHECK           ; RTN CS:BX = MEDIA/DRIVE PARAM TBL
023A 2E: 8A 4F 04               MOV     CL,CS:[BX].MD_SEC_TRK   ; GET SECTOR/TRACK
023E 2E: 8A 6F 0B               MOV     CH,CS:[BX].MD_MAX_TRK   ; GET MAX. TRACK NUMBER

0242                    STO_CX:
0242 89 4E 00                   MOV     [BP],CX                 ; SAVE IN STACK FOR RETURN
0245 89 5E 06                   MOV     [BP+6],BX               ; ADDRESS OF MEDIA/DRIVE PARM TABLE
0248 8C C8                      MOV     AX,CS                   ; SEGMENT MEDIA/DRIVE PARAMETER TABLE
024A 8E C0                      MOV     ES,AX                   ; ES IS SEGMENT OF TABLE

024C E8 0435 R          DP_OUT: CALL    XLAT_OLD                ; TRANSLATE STATE TO COMPATIBLE MODE
024F 33 C0                      XOR     AX,AX                   ; CLEAR
0251 F8                         CLC
0252 C3                         RET
```

```
                        ;----- NO DRIVE PRESENT HANDLER

0253                    NON_DRV:
0253 C6 46 04 00                MOV     BYTE PTR [BP+4],0       ; CLEAR NUMBER OF DRIVES

0257                    NON_DRV1:
0257 81 FF 0080                 CMP     DI,80H                  ; CHECK FOR FIXED MEDIA TYPE REQUEST
025B 72 09                      JB      NON_DRV2                ; CONTINUE IF NOT REQUEST FALL THROUGH

                        ;-----  FIXED DISK REQUEST FALL THROUGH ERROR

025D E8 0435 R                  CALL    XLAT_OLD                ; ELSE TRANSLATE TO COMPATIBLE MODE
0260 8B C6                      MOV     AX,SI                   ; RESTORE AL
0262 B4 01                      MOV     AH,BAD_CMD              ; SET BAD COMMAND ERROR
0264 F9                         STC                             ; SET ERROR RETURN CODE
0265 C3                         RET

0266                    NON_DRV2:
0266 33 C0                      XOR     AX,AX                   ; CLEAR PARMS IF NO DRIVES OR CMOS BAD
0268 89 46 00                   MOV     [BP],AX                 ; TRACKS, SECTORS/TRACK = 0
026B 88 66 05                   MOV     [BP+5],AH               ; HEAD = 0
026E 89 46 06                   MOV     [BP+6],AX               ; OFFSET TO DISK_BASE = 0
0271 8E C0                      MOV     ES,AX                   ; ES IS SEGMENT OF TABLE
0273 EB D7                      JMP     SHORT DP_OUT

                        ;--- DATA RATE IS EITHER 300 KBS OR 500 KBS, TRY 1.2 MB TABLE FIRST

0275                    USE_EST2:
0275 B0 02                      MOV     AL,02                   ; DRIVE TYPE 2 (1.2MB)
0277 E8 03B8 R                  CALL    DR_TYPE_CHECK           ; RTN CS:BX = MEDIA/DRIVE PARAM TBL
027A 2E: 8A 4F 04               MOV     CL,CS:[BX].MD_SEC_TRK   ; GET SECTOR/TRACK
027E 2E: 8A 6F 0B               MOV     CH,CS:[BX].MD_MAX_TRK   ; GET MAX. TRACK NUMBER
0282 80 FC 40                   CMP     AH,RATE_300             ; RATE 300 ?
0285 74 BB                      JE      STO_CX                  ; MUST BE 1.2MB DRIVE
0287 EB AC                      JMP     SHORT PARM144           ; ELSE, IT IS 1.44MB DRIVE

0289                    DISK_PARMS      ENDP

                        ;--------------------------------------------------------
                        ; DISK_TYPE  (AH = 15H)                                 :
                        ;       THIS ROUTINE RETURNS THE TYPE OF MEDIA INSTALLED.:
                        ;  ON ENTRY:    DI = DRIVE #                            :
                        ;                                                       :
                        ;  ON EXIT:     AH = DRIVE TYPE, CY=0                   :
                        ;--------------------------------------------------------
0289                    DISK_TYPE       PROC    NEAR
0289 E8 040F R                  CALL    XLAT_NEW                ; TRANSLATE STATE TO PRESENT ARCH.
028C 8A 85 0090 R               MOV     AL,@DSK_STATE[DI]       ; GET PRESENT STATE INFORMATION
0290 0A C0                      OR      AL,AL                   ; CHECK FOR NO DRIVE
0292 74 13                      JZ      NO_DRV                  ;
0294 B4 01                      MOV     AH,NOCHGLN              ; NO CHANGE LINE FOR 40 TRACK DRIVE
0296 A8 01                      TEST    AL,TRK_CAPA             ; IS THIS DRIVE AN 80 TRACK DRIVE?
0298 74 02                      JZ      DT_BACK                 ; IF NO JUMP
029A B4 02                      MOV     AH,CHGLN                ; CHANGE LINE FOR 80 TRACK DRIVE

029C                    DT_BACK:
029C 50                         PUSH    AX                      ; SAVE RETURN VALUE
029D E8 0435 R                  CALL    XLAT_OLD                ; TRANSLATE STATE TO COMPATIBLE MODE
02A0 58                         POP     AX                      ; RESTORE RETURN VALUE
02A1 F8                         CLC                             ; NO ERROR
02A2 8B DE                      MOV     BX,SI                   ; GET SAVED AL TO BL
02A4 8A C3                      MOV     AL,BL                   ; PUT BACK FOR RETURN
02A6 C3                         RET
02A7                    NO_DRV:
02A7 32 E4                      XOR     AH,AH                   ; NO DRIVE PRESENT OR UNKNOWN
02A9 EB F1                      JMP     SHORT DT_BACK
02AB                    DISK_TYPE ENDP
                        ;--------------------------------------------------------
                        ; DISK_CHANGE  (AH = 16H)                               :
                        ;       THIS ROUTINE RETURNS THE STATE OF THE DISK CHANGE LINE.:
                        ;                                                       :
                        ; ON ENTRY:    DI = DRIVE #                            :
                        ;                                                       :
                        ; ON EXIT:     AH = @DSKETTE_STATUS                    :
                        ;                   00 - DISK CHANGE LINE INACTIVE, CY = 0 :
                        ;                   06 - DISK CHANGE LINE ACTIVE, CY = 1 :
                        ;--------------------------------------------------------
02AB                    DISK_CHANGE     PROC    NEAR
02AB E8 040F R                  CALL    XLAT_NEW                ; TRANSLATE STATE TO PRESENT ARCH.
02AE 8A 85 0090 R               MOV     AL,@DSK_STATE[DI]       ; GET MEDIA STATE INFORMATION
02B2 0A C0                      OR      AL,AL                   ; DRIVE PRESENT ?
02B4 74 19                      JZ      DC_NON                  ; JUMP IF NO DRIVE
02B6 A8 01                      TEST    AL,TRK_CAPA             ; 80 TRACK DRIVE ?
02B8 74 05                      JZ      SETIT                   ; IF SO , CHECK CHANGE LINE

02BA E8 0B28 R          DC0:    CALL    READ_DSKCHNG            ; GO CHECK STATE OF DISK CHANGE LINE
02BD 74 05                      JZ      FINIS                   ; CHANGE LINE NOT ACTIVE

02BF C6 06 0041 R 06    SETIT:  MOV     @DSKETTE_STATUS,MEDIA_CHANGE    ; INDICATE MEDIA REMOVED

02C4 E8 0435 R          FINIS:  CALL    XLAT_OLD                ; TRANSLATE STATE TO COMPATIBLE MODE
02C7 E8 0854 R                  CALL    SETUP_END               ; VARIOUS CLEANUPS
02CA 8B DE                      MOV     BX,SI                   ; GET SAVED AL TO BL
02CC 8A C3                      MOV     AL,BL                   ; PUT BACK FOR RETURN
02CE C3                         RET

02CF                    DC_NON:
02CF 80 0E 0041 R 80            OR      @DSKETTE_STATUS,TIME_OUT        ; SET TIMEOUT, NO DRIVE
02D4 EB EE                      JMP     SHORT FINIS
02D6                    DISK_CHANGE     ENDP

                        ;--------------------------------------------------------
                        ; FORMAT_SET  (AH = 17H)                                :
                        ;       THIS ROUTINE IS USED TO ESTABLISH THE TYPE OF   :
                        ;       MEDIA TO BE USED FOR THE FOLLOWING FORMAT OPERATION.:
                        ;                                                       :
                        ; ON ENTRY:    SI LOW = DASD TYPE FOR FORMAT           :
                        ;               DI     = DRIVE #                        :
                        ;                                                       :
                        ; ON EXIT:     @DSKETTE_STATUS REFLECTS STATUS         :
                        ;               AH = @DSKETTE_STATUS                    :
                        ;               CY = 1 IF ERROR                         :
                        ;--------------------------------------------------------
```

```
02D6                    FORMAT_SET      PROC    NEAR
02D6 E8 040F R                  CALL    XLAT_NEW                ; TRANSLATE STATE TO PRESENT ARCH.
02D9 56                         PUSH    SI                      ; SAVE DASD TYPE
02DA 8B C6                      MOV     AX,SI                   ; AH = ? , AL = DASD TYPE
02DC 32 E4                      XOR     AH,AH                   ; AH = 0 , AL = DASD TYPE
02DE 8B F0                      MOV     SI,AX                   ; SI = DASD TYPE
02E0 80 A5 0090 R 0F            AND     @DSK_STATE[DI],NOT MED_DET+DBL_STEP+RATE_MSK    ; CLEAR STATE
02E5 4E                         DEC     SI                      ; CHECK FOR 320/360K MEDIA & DRIVE
02E6 75 07                      JNZ     NOT_320                 ; BYPASS IF NOT
02E8 80 8D 0090 R 90            OR      @DSK_STATE[DI],MED_DET+RATE_250 ; SET TO 320/360
02ED EB 37                      JMP     SHORT S0
                        NOT_320:
02EF E8 05E9 R                  CALL    MED_CHANGE              ; CHECK FOR TIME_OUT
02F2 80 3E 0041 R 80            CMP     @DSKETTE_STATUS,TIME_OUT
02F7 74 2D                      JZ      S0                      ; IF TIME OUT TELL CALLER
02F9 4E                 S3:     DEC     SI                      ; CHECK FOR 320/360K IN 1.2M DRIVE
02FA 75 07                      JNZ     NOT_320_12              ; BYPASS IF NOT
02FC 80 8D 0090 R 70            OR      @DSK_STATE[DI],MED_DET+DBL_STEP+RATE_300 ; SET STATE
0301 EB 23                      JMP     SHORT S0
0303                    NOT_320_12:
0303 4E                         DEC     SI                      ; CHECK FOR 1.2M MEDIA IN 1.2M DRIVE
0304 75 07                      JNZ     NOT_12                  ; BYPASS IF NOT
0306 80 8D 0090 R 10            OR      @DSK_STATE[DI],MED_DET+RATE_500 ; SET STATE VARIABLE
030B EB 19                      JMP     SHORT S0                ; RETURN TO CALLER
030D                    NOT_12:
030D 4E                         DEC     SI                      ; CHECK FOR SET DASD TYPE 04
030E 75 20                      JNZ     FS_ERR                  ; BAD COMMAND EXIT IF NOT VALID TYPE
0310 F6 85 0090 R 04            TEST    @DSK_STATE[DI],DRV_DET  ; DRIVE DETERMINED ?
0315 74 09                      JZ      ASSUME                  ; IF STILL NOT DETERMINED ASSUME
0317 B0 50                      MOV     AL,MED_DET+RATE_300
0319 F6 85 0090 R 02            TEST    @DSK_STATE[DI],FMT_CAPA ; MULTIPLE FORMAT CAPABILITY ?
031E 75 02                      JNZ     OR_IT_IN                ; IF 1.2 M THEN DATA RATE 300
0320                    ASSUME:
0320 B0 90                      MOV     AL,MED_DET+RATE_250     ; SET UP
0322                    OR_IT_IN:
0322 08 85 0090 R               OR      @DSK_STATE[DI],AL       ; OR IN THE CORRECT STATE
0326 E8 0435 R          S0:     CALL    XLAT_OLD                ; TRANSLATE STATE TO COMPATIBLE MODE
0329 E8 0854 R                  CALL    SETUP_END               ; VARIOUS CLEANUPS
032C 5B                         POP     BX                      ; GET SAVED AL TO BL
032D 8A C3                      MOV     AL,BL                   ; PUT BACK FOR RETURN
032F C3                         RET
0330                    FS_ERR:
0330 C6 06 0041 R 01            MOV     @DSKETTE_STATUS,BAD_CMD ; UNKNOWN STATE,BAD COMMAND
0335 EB EF                      JMP     SHORT S0
0337                    FORMAT_SET      ENDP

                        ;--------------------------------------------------------------
                        ; SET_MEDIA  (AH = 18H)                                       :
                        ;       THIS ROUTINE SETS THE TYPE OF MEDIA AND DATA RATE     :
                        ;       TO BE USED FOR THE FOLLOWING FORMAT OPERATION.        :
                        ; ON ENTRY:                                                   :
                        ;       [BP]    = SECTOR PER TRACK                            :
                        ;       [BP+1]  = TRACK #                                     :
                        ;       DI      = DRIVE #                                     :
                        ; ON EXIT:                                                    :
                        ;       @DSKETTE_STATUS REFLECTS STATUS                       :
                        ;       IF NO ERROR:                                          :
                        ;           AH = 0                                            :
                        ;           CY = 0                                            :
                        ;           ES = SEGMENT OF MEDIA/DRIVE PARAMETER TABLE       :
                        ;           DI/[BP+6] = OFFSET OF MEDIA/DRIVE PARAMETER TABLE :
                        ;       IF ERROR:                                             :
                        ;           AH = @DSKETTE_STATUS                              :
                        ;           CY = 1                                            :
                        ;--------------------------------------------------------------
0337                    SET_MEDIA       PROC    NEAR
0337 E8 040F R                  CALL    XLAT_NEW                ; TRANSLATE STATE TO PRESENT ARCH.
033A F6 85 0090 R 01            TEST    @DSK_STATE[DI],TRK_CAPA ; CHECK FOR CHANGE LINE AVAILABLE
033F 74 0F                      JZ      SM_CMOS                 ; JUMP IF 40 TRACK DRIVE
0341 E8 05E9 R                  CALL    MED_CHANGE              ; RESET CHANGE LINE
0344 80 3E 0041 R 80            CMP     @DSKETTE_STATUS,TIME_OUT ; IF TIME OUT TELL CALLER
0349 74 66                      JE      SM_RTN
034B C6 06 0041 R 00            MOV     @DSKETTE_STATUS,0       ; CLEAR STATUS
0350                    SM_CMOS:
0350 E8 08EC R                  CALL    CMOS_TYPE               ; RETURN DRIVE TYPE IN (AL)
0353 72 38                      JC      MD_NOT_FND              ; ERROR IN CMOS
0355 0A C0                      OR      AL,AL                   ; TEST FOR NO DRIVE
0357 74 58                      JZ      SM_RTN                  ; RETURN IF SO
0359 E8 03B8 R                  CALL    DR_TYPE_CHECK           ; RTN CS:BX = MEDIA/DRIVE PARAM TBL
035C 72 2F                      JC      MD_NOT_FND              ; TYPE NOT IN TABLE (BAD CMOS)
035E 57                         PUSH    DI                      ; SAVE REG.
035F 33 DB                      XOR     BX,BX                   ; BX = INDEX TO DR_TYPE TABLE
0361 B9 0006                    MOV     CX,DR_CNT               ; CX = LOOP COUNT
0364                    DR_SEARCH:
0364 2E: 8A A7 0000 R           MOV     AH,CS:DR_TYPE[BX]       ; GET DRIVE TYPE
0369 80 E4 7F                   AND     AH,BIT7OFF              ; MASK OUT MSB
036C 3A C4                      CMP     AL,AH                   ; DRIVE TYPE MATCH ?
036E 75 17                      JNE     NXT_MD                  ; NO, CHECK NEXT DRIVE TYPE
0370 2E: 8B BF 0001 R           MOV     DI,CS:WORD PTR DR_TYPE[BX+1] ; DI = MEDIA/DRIVE PARAMETER TABLE
0375 2E: 8A 65 04               MOV     AH,CS:[DI].MD_SEC_TRK   ; GET SECTOR/TRACK
0379 38 66 00                   CMP     [BP],AH                 ; MATCH ?
037C 75 09                      JNE     NXT_MD                  ; NO, CHECK NEXT MEDIA
037E 2E: 8A 65 0B               MOV     AH,CS:[DI].MD_MAX_TRK   ; GET MAX. TRACK #
0382 38 66 01                   CMP     [BP+1],AH               ; MATCH ?
0385 74 0D                      JE      MD_FND                  ; YES, GO GET RATE
0387                    NXT_MD:
0387 83 C3 03                   ADD     BX,3                    ; CHECK NEXT DRIVE TYPE
038A E2 D8                      LOOP    DR_SEARCH
038C 5F                         POP     DI                      ; RESTORE REG.
038D                    MD_NOT_FND:
038D C6 06 0041 R 0C            MOV     @DSKETTE_STATUS,MED_NOT_FND     ; ERROR, MEDIA TYPE NOT FOUND
0392 EB 1D                      JMP     SHORT SM_RTN            ; RETURN
```

```
                        MD_FND:
03A4 2E: 8A 45 0C               MOV     AL,CS:[DI].MD_RATE      ; GET RATE
03A8 3C 40                      CMP     AL,RATE_300             ; DOUBLE STEP REQUIRED FOR RATE 300
03AA 75 02                      JNE     MD_SET
03AC 0C 20                      OR      AL,DBL_STEP
                        MD_SET:
03AE 89 7E 06                   MOV     [BP+6],DI               ; SAVE TABLE POINTER IN STACK
03A1 0C 10                      OR      AL,MED_DET              ; SET MESIA ESTABLISHED
03A3 5F                         POP     DI                      ; RESTORE REG.
03A4 80 A5 0090 R 0F            AND     @DSK_STATE[DI],NOT MED_DET+DBL_STEP+RATE_MSK    ; CLEAR STATE
03A9 08 85 0090 R               OR      @DSK_STATE[DI],AL       ; SET STATE
03AD 8C C8                      MOV     AX,CS                   ; SEGMENT MEDIA/DRIVE PARAMETER TABLE
03AF 8E C0                      MOV     ES,AX                   ; ES IS SEGMENT OF TABLE
03B1                    SM_RTN:
03B1 E8 0435 R                  CALL    XLAT_OLD                ; TRANSLATE STATE TO COMPATIBLE MODE
03B4 E8 0854 R                  CALL    SETUP_END               ; VARIOUS CLEANUPS
03B7 C3                         RET
03B8                    SET_MEDIA       ENDP

                        ;--------------------------------------------------------------------
                        ; DR_TYPE_CHECK                                                     :
                        ;       CHECK IF THE GIVEN DRIVE TYPE IN REGISTER (AL)              :
                        ;       IS SUPPORTED IN BIOS DRIVE TYPE TABLE                       :
                        ; ON ENTRY:                                                        :
                        ;       AL = DRIVE TYPE                                             :
                        ; ON EXIT:                                                         :
                        ;       CS = SEGMENT OF MEDIA/DRIVE PARAMETER TABLE (CODE)          :
                        ;       CY = 0   DRIVE TYPE SUPPORTED                               :
                        ;            BX = OFFSET TO MEDIA/DRIVE PARAMETER TABLE             :
                        ;       CY = 1   DRIVE TYPE NOT SUPPORTED                           :
                        ; REGISTERS ALTERED:  BX                                           :
                        ;--------------------------------------------------------------------
03B8                    DR_TYPE_CHECK   PROC    NEAR
03B8 50                         PUSH    AX
03B9 51                         PUSH    CX
03BA 33 DB                      XOR     BX,BX                   ; BX = INDEX TO DR_TYPE TABLE
03BC B9 0006                    MOV     CX,DR_CNT               ; CX = LOOP COUNT
03BF                    TYPE_CHK:
03BF 2E: 8A A7 0000 R           MOV     AH,CS:DR_TYPE[BX]       ; GET DRIVE TYPE
03C4 3A C4                      CMP     AL,AH                   ; DRIVE TYPE MATCH ?
03C6 74 08                      JE      DR_TYPE_VALID           ; YES, RETURN WITH CARRY RESET
03C8 83 C3 03                   ADD     BX,3                    ; CHECK NEXT DRIVE TYPE
03CB E2 F2                      LOOP    TYPE_CHK
03CD F9                         STC                             ; DRIVE TYPE NOT FOUND IN TABLE
03CE EB 05                      JMP     SHORT TYPE_RTN
03D0                    DR_TYPE_VALID:
03D0 2E: 8B 9F 0001 R           MOV     BX,CS:WORD PTR DR_TYPE[BX+1]    ; BX = MEDIA TABLE
03D5                    TYPE_RTN:
03D5 59                         POP     CX
03D6 58                         POP     AX
03D7 C3                         RET
03D8                    DR_TYPE_CHECK   ENDP

                        ;--------------------------------------------------------------------
                        ; SEND_SPEC                                                        :
                        ;       SEND THE SPECIFY COMMAND TO CONTROLLER USING DATA FROM      :
                        ;       THE DRIVE PARAMETER TABLE POINTED BY @DISK_POINTER          :
                        ; ON ENTRY:    @DISK_POINTER = DRIVE PARAMETER TABLE               :
                        ; ON EXIT :    NONE                                                :
                        ; REGISTERS ALTERED: CX, DX                                        :
                        ;--------------------------------------------------------------------
03D8                    SEND_SPEC       PROC    NEAR
03D8 50                         PUSH    AX                      ; SAVE AX
03D9 B8 03F3 R                  MOV     AX,OFFSET SPECBAC       ; LOAD ERROR ADDRESS
03DC 50                         PUSH    AX                      ; PUSH NEC_OUT ERROR RETURN
03DD B4 03                      MOV     AH,03H                  ; SPECIFY COMMAND
03DF E8 09F8 R                  CALL    NEC_OUTPUT              ; OUTPUT THE COMMAND
03E2 2A D2                      SUB     DL,DL                   ; FIRST SPECIFY BYTE
03E4 E8 0905 R                  CALL    GET_PARM                ; GET PARAMETER TO AH
03E7 E8 09F8 R                  CALL    NEC_OUTPUT              ; OUTPUT THE COMMAND
03EA B2 01                      MOV     DL,1                    ; SECOND SPECIFY BYTE
03EC E8 0905 R                  CALL    GET_PARM                ; GET PARAMETER TO AH
03EF E8 09F8 R                  CALL    NEC_OUTPUT              ; OUTPUT THE COMMAND
03F2 58                         POP     AX                      ; POP ERROR RETURN
03F3                    SPECBAC:
03F3 58                         POP     AX                      ; RESTORE ORIGINAL AX VALUE
03F4 C3                         RET
03F5                    SEND_SPEC       ENDP

                        ;--------------------------------------------------------------------
                        ; SEND_SPEC_MD                                                     :
                        ;       SEND THE SPECIFY COMMAND TO CONTROLLER USING DATA FROM      :
                        ;       THE MEDIA/DRIVE PARAMETER TABLE POINTED BY (CS:BX)          :
                        ; ON ENTRY:    CS:BX = MEDIA/DRIVE PARAMETER TABLE                 :
                        ; ON EXIT :    NONE                                                :
                        ; REGISTERS ALTERED: AX                                            :
                        ;--------------------------------------------------------------------
03F5                    SEND_SPEC_MD    PROC    NEAR
03F5 50                         PUSH    AX                      ; SAVE RATE DATA
03F6 B8 040D R                  MOV     AX,OFFSET SPEC_ESBAC    ; LOAD ERROR ADDRESS
03F9 50                         PUSH    AX                      ; PUSH NEC_OUT ERROR RETURN
03FA B4 03                      MOV     AH,03H                  ; SPECIFY COMMAND
03FC E8 09F8 R                  CALL    NEC_OUTPUT              ; OUTPUT THE COMMAND
03FF 2E: 8A 27                  MOV     AH,CS:[BX].MD_SPEC1     ; GET 1ST SPECIFY BYTE
0402 E8 09F8 R                  CALL    NEC_OUTPUT              ; OUTPUT THE COMMAND
0405 2E: 8A 67 01               MOV     AH,CS:[BX].MD_SPEC2     ; GET SECOND SPECIFY BYTE
0409 E8 09F8 R                  CALL    NEC_OUTPUT              ; OUTPUT THE COMMAND
040C 58                         POP     AX                      ; POP ERROR RETURN
040D                    SPEC_ESBAC:
040D 58                         POP     AX                      ; RESTORE RATE
040E C3                         RET
040F                    SEND_SPEC_MD    ENDP
```

```
                        PAGE
                        ;--------------------------------------------------------
                        ; XLAT_NEW                                              :
                        ;       TRANSLATES DISKETTE STATE LOCATIONS FROM COMPATIBLE :
                        ;       MODE TO NEW ARCHITECTURE.                       :
                        ;                                                       :
                        ; ON ENTRY:     DI : DRIVE                              :
                        ;--------------------------------------------------------
0000F                   XLAT_NEW        PROC    NEAR                    ;
0000F 83 FF 01                  CMP     DI,1                    ; VALID DRIVE ?
00012 77 1C                     JA      XN_OUT                  ; IF INVALID BACK
00014 80 BD 0090 R 00           CMP     @DSK_STATE[DI],0        ; NO DRIVE ?
00019 74 16                     JZ      DO_DET                  ; IF NO DRIVE ATTEMPT DETERMINE
0001B 8B CF                     MOV     CX,DI                   ; CX = DRIVE NUMBER
0001D C0 E1 02                  SHL     CL,2                    ; CL = SHIFT COUNT, A=0, B=4
00020 A0 008F R                 MOV     AL,@HF_CNTRL            ; DRIVE INFORMATION
00023 D2 C8                     ROR     AL,CL                   ; TO LOW NIBBLE
00025 24 07                     AND     AL,DRV_DET+FMT_CAPA+TRK_CAPA    ; KEEP DRIVE BITS
00027 80 A5 0090 R F8           AND     @DSK_STATE[DI],NOT DRV_DET+FMT_CAPA+TRK_CAPA
0002C 08 85 0090 R              OR      @DSK_STATE[DI],AL       ; UPDATE DRIVE STATE
00030                   XN_OUT:
00030 C3                        RET                             ;
00031                   DO_DET:
00031 E8 0B32 R                 CALL    DRIVE_DET               ; TRY TO DETERMINE
00034 C3                        RET
00035                   XLAT_NEW        ENDP
                        ;--------------------------------------------------------
                        ; XLAT_OLD                                              :
                        ;       TRANSLATES DISKETTE STATE LOCATIONS FROM NEW     :
                        ;       ARCHITECTURE TO COMPATIBLE MODE.                :
                        ;                                                       :
                        ; ON ENTRY:     DI : DRIVE                              :
                        ;--------------------------------------------------------
00035                   XLAT_OLD        PROC    NEAR
00035 83 FF 01                  CMP     DI,1                    ; VALID DRIVE ?
00038 77 73                     JA      XO_OUT                  ; IF INVALID BACK
0003A 80 BD 0090 R 00           CMP     @DSK_STATE[DI],0        ; NO DRIVE ?
0003F 74 6C                     JZ      XO_OUT                  ; IF NO DRIVE TRANSLATE DONE

                        ;-----  TEST FOR SAVED DRIVE INFORMATION ALREADY SET

00041 8B CF                     MOV     CX,DI                   ; CX = DRIVE NUMBER
00043 C0 E1 02                  SHL     CL,2                    ; CL = SHIFT COUNT, A=0, B=4
00046 B4 02                     MOV     AH,FMT_CAPA             ; LOAD MULTI DATA RATE BIT MASK
00048 D2 CC                     ROR     AH,CL                   ; ROTATE BY MASK
0004A 84 26 008F R              TEST    @HF_CNTRL,AH            ; MULTI-DATA RATE DETERMINED ?
0004E 75 16                     JNZ     SAVE_SET                ; IF SO, NO NEED TO RE-SAVE

                        ;-----  ERASE DRIVE BITS IN @HF_CNTRL FOR THIS DRIVE

00050 B4 07                     MOV     AH,DRV_DET+FMT_CAPA+TRK_CAPA    ; MASK TO KEEP
00052 D2 CC                     ROR     AH,CL                   ; FIX MASK TO KEEP
00054 F6 D4                     NOT     AH                      ; TRANSLATE MASK
00056 20 26 008F R              AND     @HF_CNTRL,AH            ; KEEP BITS FROM OTHER DRIVE INTACT

                        ;-----  ACCESS CURRENT DRIVE BITS AND STORE IN @HF_CNTRL

0005A 8A 85 0090 R              MOV     AL,@DSK_STATE[DI]       ; ACCESS STATE
0005E 24 07                     AND     AL,DRV_DET+FMT_CAPA+TRK_CAPA    ; KEEP DRIVE BITS
00060 D2 C8                     ROR     AL,CL                   ; FIX FOR THIS DRIVE
00062 08 06 008F R              OR      @HF_CNTRL,AL            ; UPDATE SAVED DRIVE STATE

                        ;-----  TRANSLATE TO COMPATIBILITY MODE

00066                   SAVE_SET:
00066 8A A5 0090 R              MOV     AH,@DSK_STATE[DI]       ; ACCESS STATE
0006A 8A FC                     MOV     BH,AH                   ; TO BH FOR LATER
0006C 80 E4 C0                  AND     AH,RATE_MSK             ; KEEP ONLY RATE
0006F 80 FC 00                  CMP     AH,RATE_500             ; RATE 500 ?
00072 74 10                     JZ      CHK_144                 ; YES, 1.2/1.2 OR 1.44/1.44
00074 B0 01                     MOV     AL,M3D1U                ; AL = 360 IN 1.2 UNESTABLISHED
00076 80 FC 40                  CMP     AH,RATE_300             ; RATE 300 ?
00079 75 16                     JNZ     CHK_250                 ; NO, 360/360 ,720/720 OR 720/1.44
0007B F6 C7 20                  TEST    BH,DBL_STEP             ; YES, DOUBLE STEP ?
0007E 75 1D                     JNZ     TST_DET                 ; YES, MUST BE 360 IN 1.2

00080                   UNKNO:
00080 B0 07                     MOV     AL,MED_UNK              ; 'NONE OF THE ABOVE'
00082 EB 20                     JMP     SHORT AL_SET            ; PROCESS COMPLETE

00084                   CHK_144:
00084 E8 08EC R                 CALL    CMOS_TYPE               ; RETURN DRIVE TYPE IN (AL)
00087 72 F7                     JC      UNKNO                   ; ERROR, SET 'NONE OF THE ABOVE'
00089 3C 02                     CMP     AL,02                   ; 1.2MB DRIVE ?
0008B 75 F3                     JNE     UNKNO                   ; NO, GO SET 'NONE OF THE ABOVE'
0008D B0 02                     MOV     AL,M1D1U                ; AL = 1.2 IN 1.2 UNESTABLISHED
0008F EB 0C                     JMP     SHORT TST_DET

00091                   CHK_250:
00091 B0 00                     MOV     AL,M3D3U                ; AL = 360 IN 360 UNESTABLISHED
00093 80 FC 80                  CMP     AH,RATE_250             ; RATE 250 ?
00096 75 E8                     JNZ     UNKNO                   ; IF SO FALL THRU
00098 F6 C7 01                  TEST    BH,TRK_CAPA             ; 80 TRACK CAPABILITY ?
0009B 75 E3                     JNZ     UNKNO                   ; IF SO JUMP, FALL THRU TEST DET

0009D                   TST_DET:
0009D F6 C7 10                  TEST    BH,MED_DET              ; DETERMINED ?
000A0 74 02                     JZ      AL_SET                  ; IF NOT THEN SET
000A2 04 03                     ADD     AL,3                    ; MAKE DETERMINED/ESTABLISHED

000A4                   AL_SET:
000A4 80 A5 0090 R F8           AND     @DSK_STATE[DI],NOT DRV_DET+FMT_CAPA+TRK_CAPA  ; CLEAR DRIVE
000A9 08 85 0090 R              OR      @DSK_STATE[DI],AL       ; REPLACE WITH COMPATIBLE MODE
000AD                   XO_OUT:
000AD C3                        RET
000AE                   XLAT_OLD        ENDP
```

```
                        PAGE
                        ;--------------------------------------------------------------------
                        ; RD_WR_VF                                                          :
                        ;       COMMON READ, WRITE AND VERIFY;                             :
                        ;       MAIN LOOP FOR STATE RETRIES.                               :
                        ;                                                                  :
                        ; ON ENTRY:     AH : READ/WRITE/VERIFY NEC PARAMETER               :
                        ;               AL : READ/WRITE/VERIFY DMA PARAMETER               :
                        ;                                                                  :
                        ; ON EXIT:      @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION    :
                        ;--------------------------------------------------------------------
04AE                    RD_WR_VF        PROC    NEAR
04AE 50                         PUSH    AX                      ; SAVE DMA, NEC PARAMETERS
04AF E8 040F R                  CALL    XLAT_NEW                ; TRANSLATE STATE TO PRESENT ARCH.
04B2 E8 056A R                  CALL    SETUP_STATE             ; INITIALIZE START AND END RATE
04B5 58                         POP     AX                      ; RESTORE READ/WRITE/VERIFY
                        
04B6                    DO_AGAIN:
04B6 50                         PUSH    AX                      ; SAVE READ/WRITE/VERIFY PARAMETER
04B7 E8 05E9 R                  CALL    MED_CHANGE              ; MEDIA CHANGE AND RESET IF CHANGED
04BA 58                         POP     AX                      ; RESTORE READ/WRITE/VERIFY
                        ;       JC      RWV_END                 ; MEDIA CHANGE ERROR OR TIME-OUT
04BB 73 03                      JNC     RWV
04BD E9 055B R                  JMP     RWV_END
04C0                    RWV:
04C0 50                         PUSH    AX                      ; SAVE READ/WRITE/VERIFY PARAMETER
                        
04C1 8A B5 0090 R               MOV     DH,@DSK_STATE[DI]       ; GET RATE STATE OF THIS DRIVE
04C5 80 E6 C0                   AND     DH,RATE_MSK             ; KEEP ONLY RATE
04C8 E8 08EC R                  CALL    CMOS_TYPE               ; RETURN DRIVE TYPE IN (AL)
04CB 72 46                      JC      RWV_ASSUME              ; ERROR IN CMOS
04CD 3C 01                      CMP     AL,1                    ; 40 TRACK DRIVE?
04CF 75 0B                      JNE     RWV_1                   ; NO, BYPASS CMOS VALIDITY CHECK
04D1 F6 85 0090 R 01            TEST    @DSK_STATE[DI],TRK_CAPA ; CHECK FOR 40 TRACK DRIVE
04D6 74 0F                      JZ      RWV_2                   ; YES, CMOS IS CORRECT
04D8 B0 02                      MOV     AL,2                    ; CHANGE TO 1.2 M
04DA EB 0B                      JMP     SHORT RWV_2             ; CONTINUE
04DC                    RWV_1:
04DC 72 09                      JB      RWV_2                   ; NO DRIVE SPECIFIED, CONTINUE
04DE F6 85 0090 R 01            TEST    @DSK_STATE[DI],TRK_CAPA ; IS IT REALLY 40 TRACK?
04E3 75 02                      JNZ     RWV_2                   ; NO, 80 TRACK
04E5 B0 01                      MOV     AL,1                    ; IT'S 40 TRACK, FIX CMOS VALUE
04E7                    RWV_2:
                        
04E7 0A C0                      OR      AL,AL                   ; TEST FOR NO DRIVE
04E9 74 28                      JZ      RWV_ASSUME              ; ASSUME TYPE, USE MAX TRACK
04EB E8 03B8 R                  CALL    DR_TYPE_CHECK           ; RTN CS:BX = MEDIA/DRIVE PARAM TBL
04EE 72 23                      JC      RWV_ASSUME              ; TYPE NOT IN TABLE (BAD CMOS)
                        
                        ;--- SEARCH FOR MEDIA/DRIVE PARAMETER TABLE
                        
04F0 57                         PUSH    DI                      ; SAVE DRIVE #
04F1 33 DB                      XOR     BX,BX                   ; BX = INDEX TO DR_TYPE TABLE
04F3 B9 0006                    MOV     CX,DR_CNT               ; CX = LOOP COUNT
04F6                    RWV_DR_SEARCH:
04F6 2E: 8A A7 0000 R           MOV     AH,CS:DR_TYPE[BX]       ; GET DRIVE TYPE
04FB 80 E4 7F                   AND     AH,BIT7OFF              ; MASK OUT MSB
04FE 3A C4                      CMP     AL,AH                   ; DRIVE TYPE MATCH ?
0500 75 0B                      JNE     RWV_NXT_MD              ; NO, CHECK NEXT DRIVE TYPE
0502                    RWV_DR_FND:
0502 2E: 8B BF 0001 R           MOV     DI,WORD PTR CS:DR_TYPE[BX+1]  ; DI = MEDIA/DRIVE PARAMETER TABLE
0507                    RWV_MD_SEARCH:
0507 2E: 3A 75 0C               CMP     DH,CS:[DI].MD_RATE      ; MATCH ?
050B 74 16                      JE      RWV_MD_FND              ; YES, GO GET 1ST SPECIFY BYTE
050D                    RWV_NXT_MD:
050D 83 C3 03                   ADD     BX,3                    ; CHECK NEXT DRIVE TYPE
0510 E2 E4                      LOOP    RWV_DR_SEARCH
0512 5F                         POP     DI                      ; RESTORE DRIVE #
                        
                        ;--- ASSUME PRIMARY DRIVE IS INSTALLED AS SHIPPED
                        
0513                    RWV_ASSUME:
0513 BB 0012 R                  MOV     BX,OFFSET MD_TBL1       ; POINT TO 40 TK 250 KBS
0516 F6 85 0090 R 01            TEST    @DSK_STATE[DI],TRK_CAPA ; TEST FOR 80 TRACK
051B 74 09                      JZ      RWV_MD_FND1             ; MUST BE 40 TRACK
051D BB 002C R                  MOV     BX,OFFSET MD_TBL3       ; POINT TO 80 TK 500 KBS
0520 EB 04 90                   JMP     RWV_MD_FND1             ; GO SET SPECIFY PARAMETERS
                        
                        ;--- CS:BX POINTS TO MEDIA/DRIVE PARAMETER TABLE
                        
0523                    RWV_MD_FND:
                        
0523 8B DF                      MOV     BX,DI                   ; BX = MEDIA/DRIVE PARAMETER TABLE
0525 5F                         POP     DI                      ; RESTORE DRIVE #
0526                    RWV_MD_FND1:
                        
                        ;--- SEND THE SPECIFY COMMAND TO THE CONTROLLER
                        
0526 E8 03F5 R                  CALL    SEND_SPEC_MD
0529 E8 063D R                  CALL    CHK_LASTRATE            ; ZF=1 ATTEMPT RATE IS SAME AS LAST RATE
052C 74 03                      JZ      RWV_DBL                 ; YES, SKIP SEND RATE COMMAND
052E E8 0624 R                  CALL    SEND_RATE               ; SEND DATA RATE TO NEC
                        
                        
0531                    RWV_DBL:
0531 53                         PUSH    BX                      ; SAVE MEDIA/DRIVE PARAM ADDRESS
0532 E8 086E R                  CALL    SETUP_DBL               ; CHECK FOR DOUBLE STEP
0535 5B                         POP     BX                      ; RESTORE ADDRESS
0536 72 1A                      JC      CHK_RET                 ; ERROR FROM READ ID, POSSIBLE RETRY
0538 58                         POP     AX                      ; RESTORE NEC,DMA COMMAND
0539 50                         PUSH    AX                      ; SAVE NEC COMMAND
053A 53                         PUSH    BX                      ; SAVE MEDIA/DRIVE PARAM ADDRESS
053B E8 064D R                  CALL    DMA_SETUP               ; SET UP THE DMA
053E 5B                         POP     BX                      ; RESTORE ADDRESS
053F 58                         POP     AX                      ; RESTORE NEC COMMAND
0540 72 1F                      JC      RWV_BAC                 ; CHECK FOR DMA BOUNDARY ERROR
0542 50                         PUSH    AX                      ; SAVE NEC COMMAND
0543 53                         PUSH    BX                      ; SAVE MEDIA/DRIVE PARAM ADDRESS
0544 E8 0700 R                  CALL    NEC_INIT                ; INITIALIZE NEC
0547 5B                         POP     BX                      ; RESTORE ADDRESS
0548 72 08                      JC      CHK_RET                 ; ERROR - EXIT
054A E8 0725 R                  CALL    RWV_COM                 ; OP CODE COMMON TO READ/WRITE/VERIFY
054D 72 03                      JC      CHK_RET                 ; ERROR - EXIT
054F E8 075B R                  CALL    NEC_TERM                ; TERMINATE, GET STATUS, ETC.
```

SECTION 5

```
                        CHK_RET:
0552 E8 07E5 R                  CALL    RETRY                   ; CHECK FOR, SETUP RETRY
0555 58                         POP     AX                      ; RESTORE READ/WRITE/VERIFY PARAMETER
0556 73 03                      JNC     RWV_END                 ; CY = 0 NO RETRY
0558 E9 04B6 R                  JMP     DO_AGAIN                ; CY = 1 MEANS RETRY

                        RWV_END:
055B E8 07AD R                  CALL    DSTATE                  ; ESTABLISH STATE IF SUCCESSFUL
055E E8 0827 R                  CALL    NUM_TRANS               ; AL = NUMBER TRANSFERRED

                        RWV_BAC:                                ; BAD DMA ERROR ENTRY
0561 50                         PUSH    AX                      ; SAVE NUMBER TRANSFERRED
0562 E8 0435 R                  CALL    XLAT_OLD                ; TRANSLATE STATE TO COMPATIBLE MODE
0565 58                         POP     AX                      ; RESTORE NUMBER TRANSFERRED
0566 E8 0854 R                  CALL    SETUP_END               ; VARIOUS CLEANUPS
0569 C3                         RET
056A                    RD_WR_VF        ENDP
                        ;--------------------------------------------------------------------
                        ; SETUP_STATE:  INITIALIZES START AND END RATES.                    :
                        ;--------------------------------------------------------------------
056A                    SETUP_STATE     PROC    NEAR            ;
056A F6 85 0090 R 10            TEST    @DSK_STATE[DI],MED_DET  ; MEDIA DETERMINED ?
056F 75 29                      JNZ     J1C                     ; NO STATES IF DETERMINED
0571 B8 0040                    MOV     AX,RATE_500*H+RATE_300  ; AH = START RATE,  AL = END RATE
0574 F6 85 0090 R 04            TEST    @DSK_STATE[DI],DRV_DET  ; DRIVE ?
0579 74 0A                      JZ      AX_SET                  ; DO NOT KNOW DRIVE
057B F6 85 0090 R 02            TEST    @DSK_STATE[DI],FMT_CAPA ; MULTI-RATE ?
0580 75 03                      JNZ     AX_SET                  ; JUMP IF YES
0582 B8 8080                    MOV     AX,RATE_250*X           ; START & END RATE = 250 FOR 360 DRIVE

0585                    AX_SET:
0585 80 A5 0090 R 1F            AND     @DSK_STATE[DI],NOT RATE_MSK+DBL_STEP ; TURN OFF THE RATE
058A 08 A5 0090 R               OR      @DSK_STATE[DI],AH       ; RATE FIRST TO TRY
058E 80 26 008B R F3            AND     @LASTRATE,NOT STRT_MSK  ; ERASE LAST TO TRY RATE BITS
0593 C0 C8 04                   ROR     AL,4                    ; TO OPERATION LAST RATE LOCATION
0596 08 06 008B R               OR      @LASTRATE,AL            ; LAST RATE
059A                    J1C:
059A C3                         RET
059B                    SETUP_STATE     ENDP
                        ;--------------------------------------------------------------------
                        ;  FMT_INIT: ESTABLISH STATE IF UNESTABLISHED AT FORMAT TIME.      :
                        ;--------------------------------------------------------------------
059B                    FMT_INIT        PROC    NEAR
059B F6 85 0090 R 10            TEST    @DSK_STATE[DI],MED_DET  ; IS MEDIA ESTABLISHED
05A0 75 42                      JNZ     F1_OUT                  ; IF SO RETURN
05A2 E8 08EC R                  CALL    CMOS_TYPE               ; RETURN DRIVE TYPE IN AL
05A5 72 3E                      JC      CL_DRV                  ; ERROR IN CMOS ASSUME NO DRIVE
05A7 FE C8                      DEC     AL                      ; MAKE ZERO ORIGIN
05A9 78 3A                      JS      CL_DRV                  ; NO DRIVE IF AL 0
05AB 8A A5 0090 R               MOV     AH,@DSK_STATE[DI]       ; AH = CURRENT STATE
05AF 80 E4 0F                   AND     AH,NOT MED_DET+DBL_STEP+RATE_MSK        ; CLEAR
05B2 0A C0                      OR      AL,AL                   ; CHECK FOR 360
05B4 75 05                      JNZ     N_360                   ; IF 360 WILL BE 0
05B6 80 CC 90                   OR      AH,MED_DET+RATE_250     ; ESTABLISH MEDIA
05B9 EB 25                      JMP     SHORT SKP_STATE         ; SKIP OTHER STATE PROCESSING

05BB                    N_360:
05BB FE C8                      DEC     AL                      ; 1.2 M DRIVE
05BD 75 05                      JNZ     N_12                    ; JUMP IF NOT
05BF 80 CC 10           F1_RATE:OR      AH,MED_DET+RATE_500     ; SET FORMAT RATE
05C2 EB 1C                      JMP     SHORT SKP_STATE         ; SKIP OTHER STATE PROCESSING

05C4                    N_12:
05C4 FE C8                      DEC     AL                      ; CHECK FOR TYPE 3
05C6 75 0F                      JNZ     N_720                   ; JUMP IF NOT
05C8 F6 C4 04                   TEST    AH,DRV_DET              ; IS DRIVE DETERMINED
05CB 74 10                      JZ      ISNT_12                 ; TREAT AS NON 1.2 DRIVE
05CD F6 C4 02                   TEST    AH,FMT_CAPA             ; IS 1.2M
05D0 74 0B                      JZ      ISNT_12                 ; JUMP IF NOT
05D2 80 CC 50                   OR      AH,MED_DET+RATE_300     ; RATE 300
05D5 EB 09                      JMP     SHORT SKP_STATE         ; CONTINUE

05D7                    N_720:
05D7 FE C8                      DEC     AL                      ; CHECK FOR TYPE 4
05D9 75 0A                      JNZ     CL_DRV                  ; NO DRIVE, CMOS BAD
05DB EB E2                      JMP     SHORT F1_RATE

05DD                    ISNT_12:
05DD 80 CC 90                   OR      AH,MED_DET+RATE_250     ; MUST BE RATE 250

05E0                    SKP_STATE:
05E0 88 A5 0090 R               MOV     @DSK_STATE[DI],AH       ; STORE AWAY

05E4                    F1_OUT:
05E4 C3                         RET

05E5                    CL_DRV:
05E5 32 E4                      XOR     AH,AH                   ; CLEAR STATE
05E7 EB F7                      JMP     SHORT SKP_STATE         ; SAVE IT
05E9                    FMT_INIT        ENDP
                        ;--------------------------------------------------------------------
                        ; MED_CHANGE                                                       :
                        ;       CHECKS FOR MEDIA CHANGE, RESETS MEDIA CHANGE,              :
                        ;       CHECKS MEDIA CHANGE AGAIN.                                 :
                        ;                                                                  :
                        ; ON EXIT:      CY = 1 MEANS MEDIA CHANGE OR TIMEOUT               :
                        ;               @DSKETTE_STATUS = ERROR CODE                       :
                        ;--------------------------------------------------------------------
05E9                    MED_CHANGE      PROC    NEAR
05E9 E8 0B28 R                  CALL    READ_DSKCHNG            ; READ DISK CHANGE LINE STATE
05EC 74 34                      JZ      MC_OUT                  ; BYPASS HANDLING DISK CHANGE LINE
05EE 80 A5 0090 R EF            AND     @DSK_STATE[DI],NOT MED_DET      ; CLEAR STATE FOR THIS DRIVE

                        ;       THIS SEQUENCE ENSURES WHENEVER A DISKETTE IS CHANGED THAT
                        ;       ON THE NEXT OPERATION THE REQUIRED MOTOR START UP TIME WILL
                        ;       BE WAITED. (DRIVE MOTOR MAY GO OFF UPON DOOR OPENING).

05F3 8B CF                      MOV     CX,DI                   ; CL = DRIVE #
05F5 B0 01                      MOV     AL,1                    ; MOTOR ON BIT MASK
05F7 D2 E0                      SHL     AL,CL                   ; TO APPROPRIATE POSITION
05F9 F6 D0                      NOT     AL                      ; KEEP ALL BUT MOTOR ON
05FB FA                         CLI                             ; NO INTERRUPTS
05FC 20 06 003F R               AND     @MOTOR_STATUS,AL        ; TURN MOTOR OFF INDICATOR
```

```
0600 FB                         STI                             ; INTERRUPTS ENABLED
0601 E8 091A R                  CALL    MOTOR_ON                ; TURN MOTOR ON

                        ;-----  THIS SEQUENCE OF SEEKS IS USED TO RESET DISKETTE CHANGE SIGNAL

0604 E8 00E7 R                  CALL    DISK_RESET              ; RESET NEC
0607 B5 01                      MOV     CH,0TH                  ; MOVE TO CYLINDER 1
0609 E8 0A24 R                  CALL    SEEK                    ; ISSUE SEEK
060C 32 ED                      XOR     CH,CH                   ; MOVE TO CYLINDER 0
060E E8 0A24 R                  CALL    SEEK                    ; ISSUE SEEK
0611 C6 06 0041 R 06            MOV     @DSKETTE_STATUS,MEDIA_CHANGE    ; STORE IN STATUS

0616 E8 0B28 R          OK1:    CALL    READ_DSKCHNG            ; CHECK MEDIA CHANGED AGAIN
0619 74 05                      JZ      OK2                     ; IF ACTIVE, NO DISKETTE, TIMEOUT

061B C6 06 0041 R 80    OK4:    MOV     @DSKETTE_STATUS,TIME_OUT; TIMEOUT IF DRIVE EMPTY

0620 F9                 OK2:    STC                             ; MEDIA CHANGED, SET CY
0621 C3                         RET
                        MC_OUT:
0622 F8                         CLC                             ; NO MEDIA CHANGED, CLEAR CY
0623 C3                         RET
0624                    MED_CHANGE      ENDP
                        ;--------------------------------------------------------------
                        ; SEND_RATE                                                   :
                        ;       SENDS DATA RATE COMMAND TO NEC                        :
                        ; ON ENTRY:     DI = DRIVE #                                  :
                        ; ON EXIT:      NONE                                          :
                        ; REGISTERS ALTERED: DX                                       :
                        ;--------------------------------------------------------------
0624                    SEND_RATE       PROC    NEAR            ;

0624 50                         PUSH    AX                      ; SAVE REG.
0625 80 26 008B R 3F            AND     @LASTRATE,NOT SEND_MSK  ; ELSE CLEAR LAST RATE ATTEMPTED
062A 8A 85 0090 R               MOV     AL,@DSK_STATE[DI]       ; GET RATE STATE OF THIS DRIVE
062E 24 C0                      AND     AL,SEND_MSK             ; KEEP ONLY RATE BITS
0630 08 06 008B R               OR      @LASTRATE,AL            ; SAVE NEW RATE FOR NEXT CHECK
0634 C0 C0 02                   ROL     AL,2                    ; MOVE TO BIT OUTPUT POSITIONS
0637 BA 03F7                    MOV     DX,03F7H                ; OUTPUT NEW DATA RATE
063A EE                         OUT     DX,AL

063B 58                         POP     AX                      ; RESTORE REG.
063C C3                         RET
063D                    SEND_RATE       ENDP


                        ;--------------------------------------------------------------
                        ; CHK_LASTRATE                                                :
                        ;       CHECK PREVIOUS DATA RATE SENT TO THE CONTROLLER.       :
                        ; ON ENTRY:                                                   :
                        ;       DI = DRIVE #                                          :
                        ; ON EXIT:                                                    :
                        ;       ZF = 1   DATA RATE IS THE SAME AS LAST RATE SENT TO NEC :
                        ;       ZF = 0   DATA RATE IS DIFFERENT FROM LAST RATE        :
                        ; REGISTERS ALTERED: NONE                                     :
                        ;--------------------------------------------------------------
063D                    CHK_LASTRATE    PROC    NEAR
063D 50                         PUSH    AX                      ; SAVE REG
063E 8A 26 008B R               MOV     AH,@LASTRATE            ; GET LAST DATA RATE SELECTED
0642 8A 85 0090 R               MOV     AL,@DSK_STATE[DI]       ; GET RATE STATE OF THIS DRIVE
0646 25 C0C0                    AND     AX,SEND_MSK*X           ; KEEP ONLY RATE BITS OF BOTH
0649 3A C4                      CMP     AL,AH                   ; COMPARE TO PREVIOUSLY TRIED
                                                                ; ZF = 1  RATE IS THE SAME
064B 58                         POP     AX                      ; RESTORE REG.
064C C3                         RET
064D                    CHK_LASTRATE    ENDP

                        SUBTTL (DSK3.ASM)
```

```
                        PAGE
                        ;--------------------------------------------------------------------
                        ; DMA_SETUP                                                         :
                        ;       THIS ROUTINE SETS UP THE DMA FOR READ/WRITE/VERIFY          :
                        ;       OPERATIONS.                                                :
                        ;                                                                  :
                        ; ON ENTRY:     AL = DMA COMMAND                                   :
                        ;                                                                  :
                        ; ON EXIT:      @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION    :
                        ;--------------------------------------------------------------------
064D                    DMA_SETUP       PROC    NEAR
064D FA                         CLI                             ; DISABLE INTERRUPTS DURING DMA SET-UP
064E E6 0C                      OUT     DMA+12,AL               ; SET THE FIRST/LAST F/F
0650 EB 00                      JMP     $+2                     ; WAIT FOR I/O
0652 E6 0B                      OUT     DMA+11,AL               ; OUTPUT THE MODE BYTE
0654 3C 42                      CMP     AL,42H                  ; DMA VERIFY COMMAND
0656 75 04                      JNE     NOT_VERF                ; NO
0658 33 C0                      XOR     AX,AX                   ; START ADDRESS
065A EB 10                      JMP     SHORT J33
065C                    NOT_VERF:
065C 8C C0                      MOV     AX,ES                   ; GET THE ES VALUE
065E C1 C0 04                   ROL     AX,4                    ; ROTATE LEFT
0661 8A E8                      MOV     CH,AL                   ; GET HIGHEST NIBBLE OF ES TO CH
0663 24 F0                      AND     AL,11110000B            ; ZERO THE LOW NIBBLE FROM SEGMENT
0665 03 46 02                   ADD     AX,[BP+2]               ; TEST FOR CARRY FROM ADDITION
0668 73 02                      JNC     J33
066A FE C5                      INC     CH                      ; CARRY MEANS HIGH 4 BITS MUST BE INC
066C                    J33:
066C 50                         PUSH    AX                      ; SAVE START ADDRESS
066D E6 04                      OUT     DMA+4,AL                ; OUTPUT LOW ADDRESS
066F EB 00                      JMP     $+2                     ; WAIT FOR I/O
0671 8A C4                      MOV     AL,AH
0673 E6 04                      OUT     DMA+4,AL                ; OUTPUT HIGH ADDRESS
0675 8A C5                      MOV     AL,CH                   ; GET HIGH 4 BITS
0677 EB 00                      JMP     $+2                     ; I/O WAIT STATE
0679 24 0F                      AND     AL,00001111B
067B E6 81                      OUT     081H,AL                 ; OUTPUT HIGH 4 BITS TO PAGE REGISTER
                        ;-----  DETERMINE COUNT
067D 8B C6                      MOV     AX,SI                   ; AL = # OF SECTORS
067F 86 C4                      XCHG    AL,AH                   ; AH = # OF SECTORS
0681 2A C0                      SUB     AL,AL                   ; AL = 0, AX = # OF SECTORS * 256
0683 D1 E8                      SHR     AX,1                    ; AX = # SECTORS * 128
0685 50                         PUSH    AX                      ; SAVE # OF SECTORS * 128
0686 B2 03                      MOV     DL,3                    ; GET BYTES/SECTOR PARAMETER
0688 E8 0905 R                  CALL    GET_PARM                ; "
068B 8A CC                      MOV     CL,AH                   ; SHIFT COUNT (0=128, 1=256 ETC)
068D 58                         POP     AX                      ; AX = # OF SECTORS * 128
068E D3 E0                      SHL     AX,CL                   ; SHIFT BY PARAMETER VALUE
0690 48                         DEC     AX                      ;-1 FOR DMA VALUE
0691 50                         PUSH    AX                      ; SAVE COUNT VALUE
0692 E6 05                      OUT     DMA+5,AL                ; LOW BYTE OF COUNT
0694 EB 00                      JMP     $+2                     ; WAIT FOR I/O
0696 8A C4                      MOV     AL,AH
0698 E6 05                      OUT     DMA+5,AL                ; HIGH BYTE OF COUNT
069A FB                         STI                             ; RE-ENABLE INTERRUPTS
069B 59                         POP     CX                      ; RECOVER COUNT VALUE
069C 58                         POP     AX                      ; RECOVER ADDRESS VALUE
069D 03 C1                      ADD     AX,CX                   ; ADD, TEST FOR 64K OVERFLOW
069F B0 02                      MOV     AL,2                    ; MODE FOR 8237
06A1 EB 00                      JMP     $+2                     ; WAIT FOR I/O
06A3 E6 0A                      OUT     DMA+10,AL               ; INITIALIZE THE DISKETTE CHANNEL

06A5 73 05                      JNC     NO_BAD                  ; CHECK FOR ERROR
06A7 C6 06 0041 R 09            MOV     @DSKETTE_STATUS,DMA_BOUNDARY    ; SET ERROR

06AC                    NO_BAD:
06AC C3                         RET                             ; CY SET BY ABOVE IF ERROR
06AD                    DMA_SETUP       ENDP
```

```
                        PAGE
                        ;--------------------------------------------------------
                        ; FMTDMA_SET                                            :
                        ;       THIS ROUTINE SETS UP THE DMA CONTROLLER FOR A FORMAT :
                        ;       OPERATION.                                      :
                        ;                                                       :
                        ; ON ENTRY:    NOTHING REQUIRED                        :
                        ;                                                       :
                        ; ON EXIT:     @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION :
                        ;--------------------------------------------------------
06AD                    FMTDMA_SET      PROC    NEAR
06AD B0 4A                      MOV     AL,04AH                 ; WILL WRITE TO THE DISKETTE
06AF FA                         CLI                             ; DISABLE INTERRUPTS DURING DMA SET-UP
06B0 E6 0C                      OUT     DMA+12,AL               ; SET THE FIRST/LAST F/F
06B2 EB 00                      JMP     $+2                     ; WAIT FOR I/O
06B4 E6 0B                      OUT     DMA+11,AL               ; OUTPUT THE MODE BYTE
                        ;
06B6 8C C0                      MOV     AX,ES                   ; GET THE ES VALUE
06B8 C1 C0 04                   ROL     AX,4                    ; ROTATE LEFT
06BB 8A E8                      MOV     CH,AL                   ; GET HIGHEST NIBBLE OF ES TO CH
06BD 24 F0                      AND     AL,11110000B            ; ZERO THE LOW NIBBLE FROM SEGMENT
06BF 03 46 02                   ADD     AX,[BP+2]               ; TEST FOR CARRY FROM ADDITION
06C2 73 02                      JNC     J33A
06C4 FE C5                      INC     CH                      ; CARRY MEANS HIGH 4 BITS MUST BE INC
06C6                    J33A:
06C6 50                         PUSH    AX                      ; SAVE START ADDRESS
06C7 E6 04                      OUT     DMA+4,AL                ; OUTPUT LOW ADDRESS
06C9 EB 00                      JMP     $+2                     ; WAIT FOR I/O
06CB 8A C4                      MOV     AL,AH
06CD E6 04                      OUT     DMA+4,AL                ; OUTPUT HIGH ADDRESS
06CF 8A C5                      MOV     AL,CH                   ; GET HIGH 4 BITS
06D1 EB 00                      JMP     $+2                     ; I/O WAIT STATE
06D3 24 0F                      AND     AL,00001111B
06D5 E6 81                      OUT     081H,AL                 ; OUTPUT HIGH 4 BITS TO PAGE REGISTER
                        ;
                        ;-----  DETERMINE COUNT
                        ;
06D7 B2 04                      MOV     DL,4                    ; SECTORS/TRACK VALUE IN PARM TABLE
06D9 E8 0905 R                  CALL    GET_PARM                ; '
06DC 86 C4                      XCHG    AL,AH
06DE 2A E4                      SUB     AH,AH                   ; AX = SECTORS/TRACK VALUE
06E0 C1 E0 02                   SHL     AX,2                    ; AX = SEC/TRK * 4 (OFFSET FOR C,H,R,N)
06E3 48                         DEC     AX                      ;-1 FOR DMA VALUE
06E4 50                         PUSH    AX                      ; SAVE # OF BYTES TO BE TRANSFERED
06E5 E6 05                      OUT     DMA+5,AL                ; LOW BYTE OF COUNT
06E7 EB 00                      JMP     $+2                     ; WAIT FOR I/O
06E9 8A C4                      MOV     AL,AH
06EB E6 05                      OUT     DMA+5,AL                ; HIGH BYTE OF COUNT
06ED FB                         STI                             ; RE-ENABLE INTERRUPTS
06EE 59                         POP     CX                      ; RECOVER COUNT VALUE
06EF 58                         POP     AX                      ; RECOVER ADDRESS VALUE
06F0 03 C1                      ADD     AX,CX                   ; ADD, TEST FOR 64K OVERFLOW
06F2 B0 02                      MOV     AL,2                    ; MODE FOR 8237
06F4 EB 00                      JMP     $+2                     ; WAIT FOR I/O
06F6 E6 0A                      OUT     DMA+10,AL               ; INITIALIZE THE DISKETTE CHANNEL
                        ;
06F8 73 05                      JNC     FMTDMA_OK               ; CHECK FOR ERROR
06FA C6 06 0041 R 09            MOV     @DSKETTE_STATUS,DMA_BOUNDARY    ; SET ERROR
                        ;
06FF                    FMTDMA_OK:
06FF C3                         RET                             ; CY SET BY ABOVE IF ERROR
0700                    FMTDMA_SET      ENDP
                        ;--------------------------------------------------------
                        ; NEC_INIT                                              :
                        ;       THIS ROUTINE SEEKS TO THE REQUESTED TRACK AND   :
                        ;       INITIALIZES THE NEC FOR THE READ/WRITE/VERIFY/FORMAT :
                        ;       OPERATION.                                      :
                        ;                                                       :
                        ; ON ENTRY:    AH : NEC COMMAND TO BE PERFORMED        :
                        ;                                                       :
                        ; ON EXIT:     @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION :
                        ;--------------------------------------------------------
0700                    NEC_INIT        PROC    NEAR
0700 50                         PUSH    AX                      ; SAVE NEC COMMAND
0701 E8 091A R                  CALL    MOTOR_ON                ; TURN MOTOR ON FOR SPECIFIC DRIVE
                        ;
                        ;-----  DO THE SEEK OPERATION
                        ;
0704 8A 6E 01                   MOV     CH,[BP+1]               ; CH = TRACK #
0707 E8 0A24 R                  CALL    SEEK                    ; MOVE TO CORRECT TRACK
070A 58                         POP     AX                      ; RECOVER COMMAND
070B 72 17                      JC      ER_1                    ; ERROR ON SEEK
070D BB 0724 R                  MOV     BX,OFFSET ER_1          ; LOAD ERROR ADDRESS
0710 53                         PUSH    BX                      ; PUSH NEC_OUT ERROR RETURN
                        ;
                        ;-----  SEND OUT THE PARAMETERS TO THE CONTROLLER
                        ;
0711 E8 09F8 R                  CALL    NEC_OUTPUT              ; OUTPUT THE OPERATION COMMAND
0714 8B C6                      MOV     AX,SI                   ; AH = HEAD #
0716 8B DF                      MOV     BX,DI                   ; BL = DRIVE #
0718 C0 E4 02                   SAL     AH,2                    ; MOVE IT TO BIT 2
071B 80 E4 04                   AND     AH,00000100B            ; ISOLATE THAT BIT
071E 0A E3                      OR      AH,BL                   ; OR IN THE DRIVE NUMBER
0720 E8 09F8 R                  CALL    NEC_OUTPUT              ; FALL THRU CY SET IF ERROR
0723 5B                         POP     BX                      ; THROW AWAY ERROR RETURN
0724                    ER_1:
0724 C3                         RET
0725                    NEC_INIT        ENDP
                        ;--------------------------------------------------------
                        ; RWV_COM                                               :
                        ;       THIS ROUTINE SENDS PARAMETERS TO THE NEC SPECIFIC :
                        ;       TO THE READ/WRITE/VERIFY OPERATIONS.            :
                        ;                                                       :
                        ; ON ENTRY:    CS:BX = ADDRESS OF MEDIA/DRIVE PARAMETER TABLE :
                        ; ON EXIT :    @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION :
                        ;--------------------------------------------------------
0725                    RWV_COM PROC    NEAR
0725 B8 075A R                  MOV     AX,OFFSET ER_2          ; LOAD ERROR ADDRESS
0728 50                         PUSH    AX                      ; PUSH NEC_OUT ERROR RETURN
0729 8A 66 01                   MOV     AH,[BP+1]               ; OUTPUT TRACK #
072C E8 09F8 R                  CALL    NEC_OUTPUT
072F 8B C6                      MOV     AX,SI                   ; OUTPUT HEAD #
0731 E8 09F8 R                  CALL    NEC_OUTPUT
0734 8A 66 00                   MOV     AH,[BP]                 ; OUTPUT SECTOR #
```

```
                        CALL    NEC_OUTPUT
                        MOV     DL,3                    ; BYTES/SECTOR PARAMETER FROM BLOCK
                        CALL    GET_PARM                ; . TO THE NEC
                        CALL    NEC_OUTPUT              ; OUTPUT TO CONTROLLER
                        MOV     DL,4                    ; EOT PARAMETER FROM BLOCK
                        CALL    GET_PARM                ; . TO THE NEC
                        CALL    NEC_OUTPUT              ; OUTPUT TO CONTROLLER

                        MOV     AH,CS:[BX].MD_GAP       ; GET GAP LENGTH
R15:                    CALL    NEC_OUTPUT
                        MOV     DL,6                    ; DTL PARAMETER FROM BLOCK
                        CALL    GET_PARM                ;  TO THE NEC
                        CALL    NEC_OUTPUT              ; OUTPUT TO CONTROLLER
                        POP     AX                      ; THROW AWAY ERROR EXIT
ER_2:
                        RET
RWV_COM ENDP
                        ;--------------------------------------------------------------------
                        ; NEC_TERM                                                          :
                        ;       THIS ROUTINE WAITS FOR THE OPERATION THEN ACCEPTS           :
                        ;       THE STATUS FROM THE NEC FOR THE READ/WRITE/VERIFY/          :
                        ;       FORMAT OPERATION.                                          :
                        ;                                                                  :
                        ; ON EXIT:     @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION    :
                        ;--------------------------------------------------------------------
NEC_TERM        PROC    NEAR

                        ;-----  LET THE OPERATION HAPPEN

                        PUSH    SI                      ; SAVE HEAD #, # OF SECTORS
                        CALL    WAIT_INT                ; WAIT FOR THE INTERRUPT
                        PUSHF
                        CALL    RESULTS                 ; GET THE NEC STATUS
                        JC      SET_END_POP
                        POPF
                        JC      SET_END                 ; LOOK FOR ERROR

                        ;-----  CHECK THE RESULTS RETURNED BY THE CONTROLLER

                        CLD                             ; SET THE CORRECT DIRECTION
                        MOV     SI,OFFSET @NEC_STATUS   ; POINT TO STATUS FIELD
                        LODS    @NEC_STATUS             ; GET ST0
                        AND     AL,11000000B            ; TEST FOR NORMAL TERMINATION
                        JZ      SET_END                 ;
                        CMP     AL,01000000B            ; TEST FOR ABNORMAL TERMINATION
                        JNZ     J18                     ; NOT ABNORMAL, BAD NEC

                        ;-----  ABNORMAL TERMINATION, FIND OUT WHY

                        LODS    @NEC_STATUS             ; GET ST1
                        SAL     AL,1                    ; TEST FOR EOT FOUND
                        MOV     AH,RECORD_NOT_FND
                        JC      J19
                        SAL     AL,2
                        MOV     AH,BAD_CRC
                        JC      J19
                        SAL     AL,1                    ; TEST FOR DMA OVERRUN
                        MOV     AH,BAD_DMA
                        JC      J19
                        SAL     AL,2                    ; TEST FOR RECORD NOT FOUND
                        MOV     AH,RECORD_NOT_FND
                        JC      J19
                        SAL     AL,1
                        MOV     AH,WRITE_PROTECT        ; TEST FOR WRITE_PROTECT
                        JC      J19
                        SAL     AL,1                    ; TEST MISSING ADDRESS MARK
                        MOV     AH,BAD_ADDR_MARK
                        JC      J19

                        ;-----  NEC MUST HAVE FAILED
J18:
                        MOV     AH,BAD_NEC
J19:
                        OR      @DSKETTE_STATUS,AH
SET_END:
                        CMP     @DSKETTE_STATUS,1       ; SET ERROR CONDITION
                        CMC                             ;
                        POP     SI                      ; RESTORE HEAD #, # OF SECTORS
                        RET

SET_END_POP:
                        POPF
                        JMP     SHORT SET_END
NEC_TERM        ENDP
                        ;--------------------------------------------------------------------
                        ; DSTATE:      ESTABLISH STATE UPON SUCCESSFUL OPERATION.          :
                        ;--------------------------------------------------------------------
DSTATE          PROC    NEAR
                        CMP     @DSKETTE_STATUS,0       ; CHECK FOR ERROR
                        JNZ     SETBAC                  ; IF ERROR JUMP
                        OR      @DSK_STATE[DI],MED_DET  ; NO ERROR, MARK MEDIA AS DETERMINED
                        TEST    @DSK_STATE[DI],DRV_DET  ; DRIVE DETERMINED ?
                        JNZ     SETBAC                  ; IF DETERMINED NO TRY TO DETERMINE
                        MOV     AL,@DSK_STATE[DI]       ; LOAD STATE
                        AND     AL,RATE_MSK             ; KEEP ONLY RATE
                        CMP     AL,RATE_250             ; RATE 250 ?
                        JNE     M_12                    ; NO, MUST BE 1.2M OR 1.44M DRV

                        ;--- CHECK IF IT IS 1.44M

                        CALL    CMOS_TYPE               ; RETURN DRIVE TYPE IN (AL)
                        JC      M_12                    ; CMOS BAD
                        CMP     AL,04                   ; 1.44MB DRIVE ?
                        JE      M_12                    ; YES
M_720:
                        AND     @DSK_STATE[DI],NOT FMT_CAPA ; TURN OFF FORMAT CAPA
                        OR      @DSK_STATE[DI],DRV_DET  ; MARK DRIVE DETERMINED
                        JMP     SHORT SETBAC            ; BACK

M_12:   OR      @DSK_STATE[DI],DRV_DET+FMT_CAPA  ; TURN ON DETERMINED & FMT CAPA

SETBAC:
                        RET
DSTATE          ENDP
```

```
                        ;--------------------------------------------------------------------
                        ; RETRY                                                             :
                        ;       DETERMINES WHETHER A RETRY IS NECESSARY. IF RETRY IS        :
                        ;       REQUIRED THEN STATE INFORMATION IS UPDATED FOR RETRY.       :
                        ;                                                                   :
                        ; ON EXIT:     CY = 1 FOR RETRY, CY = 0 FOR NO RETRY               :
                        ;--------------------------------------------------------------------
07E5                    RETRY   PROC    NEAR
07E5 80 3E 0041 R 00            CMP     @DSKETTE_STATUS,0       ; GET STATUS OF OPERATION
07EA 74 39                      JZ      NO_RETRY                ; SUCCESSFUL OPERATION
07EC 80 3E 0041 R 80            CMP     @DSKETTE_STATUS,TIME_OUT        ; IF TIME OUT NO RETRY
07F1 74 32                      JZ      NO_RETRY                ;
07F3 8A A5 0090 R               MOV     AH,@DSK_STATE[DI]       ; GET MEDIA STATE OF DRIVE
07F7 F6 C4 10                   TEST    AH,MED_DET              ; ESTABLISHED/DETERMINED ?
07FA 75 29                      JNZ     NO_RETRY                ; IF ESTABLISHED STATE THEN TRUE ERROR
07FC 80 E4 C0                   AND     AH,RATE_MSK             ; ISOLATE RATE
07FF 8A 2E 008B R               MOV     CH,@LASTRATE            ; GET START OPERATION STATE
0803 C0 C5 04                   ROL     CH,4                    ; TO CORRESPONDING BITS
0806 80 E5 C0                   AND     CH,RATE_MSK             ; ISOLATE RATE BITS
0809 3A EC                      CMP     CH,AH                   ; ALL RATES TRIED
080B 74 18                      JE      NO_RETRY                ; IF YES, THEN TRUE ERROR
                        ;
                        ;       SETUP STATE INDICATOR FOR RETRY ATTEMPT TO NEXT RATE
                        ;         00000000B (500) -> 10000000B (250)
                        ;         10000000B (250) -> 01000000B (300)
                        ;         01000000B (300) -> 00000000B (500)
080D 80 FC 01                   CMP     AH,RATE_500+1           ; SET CY FOR RATE 500
0810 D0 DC                      RCR     AH,1                    ; TO NEXT STATE
0812 80 E4 C0                   AND     AH,RATE_MSK             ; KEEP ONLY RATE BITS
0815 80 A5 0090 R 1F            AND     @DSK_STATE[DI],NOT RATE_MSK+DBL_STEP    ; RATE, DBL STEP OFF
081A 08 A5 0090 R               OR      @DSK_STATE[DI],AH       ; TURN ON NEW RATE
081E C6 06 0041 R 00            MOV     @DSKETTE_STATUS,0       ; RESET STATUS FOR RETRY
0823 F9                         STC                             ; SET CARRY FOR RETRY
0824 C3                         RET                             ; RETRY RETURN
                        ;
0825                    NO_RETRY:
0825 F8                         CLC                             ; CLEAR CARRY NO RETRY
0826 C3                         RET                             ; NO RETRY RETURN
0827                    RETRY   ENDP
                        ;--------------------------------------------------------------------
                        ; NUM_TRANS                                                         :
                        ;       THIS ROUTINE CALCULATES THE NUMBER OF SECTORS THAT          :
                        ;       WERE ACTUALLY TRANSFERRED TO/FROM THE DISKETTE.             :
                        ;                                                                   :
                        ; ON ENTRY:    [BP+1] = TRACK                                      :
                        ;              SI-HI  = HEAD                                       :
                        ;              [BP]   = START SECTOR                               :
                        ;                                                                   :
                        ; ON EXIT:     AL = NUMBER ACTUALLY TRANSFERRED                    :
                        ;--------------------------------------------------------------------
0827                    NUM_TRANS       PROC    NEAR
0827 32 C0                      XOR     AL,AL                   ; CLEAR FOR ERROR
0829 80 3E 0041 R 00            CMP     @DSKETTE_STATUS,0       ; CHECK FOR ERROR
082E 75 23                      JNZ     NT_OUT                  ; IF ERROR 0 TRANSFERRED
0830 B2 04                      MOV     DL,4                    ; SECTORS/TRACK OFFSET TO DL
0832 E8 0905 R                  CALL    GET_PARM                ; AH = SECTORS/TRACK
0835 8A 1E 0047 R               MOV     BL,@NEC_STATUS+5        ; GET ENDING SECTOR
0839 8B CE                      MOV     CX,SI                   ; CH = HEAD # STARTED
083B 3A 2E 0046 R               CMP     CH,@NEC_STATUS+4        ; GET HEAD ENDED UP ON
083F 75 0B                      JNZ     DIF_HD                  ; IF ON SAME HEAD, THEN NO ADJUST
                        ;
0841 8A 2E 0045 R               MOV     CH,@NEC_STATUS+3        ; GET TRACK ENDED UP ON
0845 3A 6E 01                   CMP     CH,[BP+1]               ; IS IT ASKED FOR TRACK
0848 74 04                      JZ      SAME_TRK                ; IF SAME TRACK NO INCREASE
                        ;
084A 02 DC                      ADD     BL,AH                   ; ADD SECTORS/TRACK
084C                    DIF_HD:
084C 02 DC                      ADD     BL,AH                   ; ADD SECTORS/TRACK
084E                    SAME_TRK:
084E 2A 5E 00                   SUB     BL,[BP]                 ; SUBTRACT START FROM END
0851 8A C3                      MOV     AL,BL                   ; TO AL
                        ;
0853                    NT_OUT:
0853 C3                         RET
0854                    NUM_TRANS       ENDP
                        ;--------------------------------------------------------------------
                        ; SETUP_END                                                         :
                        ;       RESTORES @MOTOR_COUNT TO PARAMETER PROVIDED IN TABLE        :
                        ;       AND LOADS @DSKETTE_STATUS TO AH, AND SETS CY.               :
                        ; ON EXIT:                                                          :
                        ;       AH, @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION         :
                        ;--------------------------------------------------------------------
0854                    SETUP_END       PROC    NEAR
0854 B2 02                      MOV     DL,2                    ; GET THE MOTOR WAIT PARAMETER
0856 50                         PUSH    AX                      ; SAVE NUMBER TRANSFERRED
0857 E8 0905 R                  CALL    GET_PARM
085A 88 26 0040 R               MOV     @MOTOR_COUNT,AH         ; STORE UPON RETURN
085E 58                         POP     AX                      ; RESTORE NUMBER TRANSFERRED
085F 8A 26 0041 R               MOV     AH,@DSKETTE_STATUS      ; GET STATUS OF OPERATION
0863 0A E4                      OR      AH,AH                   ; CHECK FOR ERROR
0865 74 02                      JZ      NUM_ERR                 ; NO ERROR
0867 32 C0                      XOR     AL,AL                   ; CLEAR NUMBER RETURNED
                        ;
                        NUM_ERR:
0869 80 FC 01                   CMP     AH,1                    ; SET THE CARRY FLAG TO INDICATE
086C F5                         CMC                             ; SUCCESS OR FAILURE
086D C3                         RET
086E                    SETUP_END       ENDP
```

```
                        PAGE
                        ;--------------------------------------------------------
                        ; SETUP_DBL                                             :
                        ;       CHECK DOUBLE STEP.                              :
                        ; ON ENTRY:                                             :
                        ;       DI = DRIVE                                      :
                        ; ON EXIT :   CY = 1 MEANS ERROR                        :
                        ;--------------------------------------------------------
0877                    SETUP_DBL       PROC    NEAR
086E 8A A5 0090 R               MOV     AH,@DSK_STATE[DI]       ; ACCESS STATE
0872 F6 C4 10                   TEST    AH,MED_DET              ; ESTABLISHED STATE ?
0875 75 5E                      JNZ     NO_DBL                  ; IF ESTABLISHED THEN DOUBLE DONE
                        ;-----  CHECK FOR TRACK 0 TO SPEED UP ACKNOWLEDGE OF UNFORMATTED DISKETTE
0877 C6 06 003E R 00            MOV     @SEEK_STATUS,0          ; SET RECALIBRATE REQUIRED ON ALL DRIVES
087C E8 091A R                  CALL    MOTOR_ON                ; ENSURE MOTOR STAY ON
087F B5 00                      MOV     CH,0                    ; LOAD TRACK 0
0881 E8 0A24 R                  CALL    SEEK                    ; SEEK TO TRACK 0
0884 E8 08D7 R                  CALL    READ_ID                 ; READ ID FUNCTION
0887 72 37                      JC      SD_ERR                  ; IF ERROR NO TRACK 0
                        ;-----  INITIALIZE START AND MAX TRACKS (TIMES 2 FOR BOTH HEADS)
0889 B9 0450                    MOV     CX,0450H                ; START, MAX TRACKS
088C F6 85 0090 R 01            TEST    @DSK_STATE[DI],TRK_CAPA ; TEST FOR 80 TRACK CAPABILITY
0891 74 02                      JZ      CNT_OK                  ; IF NOT COUNT IS SETUP
0893 B1 A0                      MOV     CL,0A0H                 ; MAXIMUM TRACK 1.2 MB
                        ;       ATTEMPT READ ID OF ALL TRACKS, ALL HEADS UNTIL SUCCESS; UPON SUCCESS,
                        ;       MUST SEE IF ASKED FOR TRACK IN SINGLE STEP MODE = TRACK ID READ; IF NOT
                        ;       THEN SET DOUBLE STEP ON.
0895 C6 06 0040 R FF    CNT_OK: MOV     @MOTOR_COUNT,0FFH       ; ENSURE MOTOR STAYS ON FOR OPERATION
089A 51                         PUSH    CX                      ; SAVE TRACK, COUNT
089B C6 06 0041 R 00            MOV     @DSKETTE_STATUS,0       ; CLEAR STATUS, EXPECT ERRORS
08A0 33 C0                      XOR     AX,AX                   ; CLEAR AX
08A2 D0 ED                      SHR     CH,1                    ; HALVE TRACK, CY = HEAD
08A4 C0 D0 03                   RCL     AL,3                    ; AX = HEAD IN CORRECT BIT
08A7 50                         PUSH    AX                      ; SAVE HEAD
08A8 E8 0A24 R                  CALL    SEEK                    ; SEEK TO TRACK
08AB 58                         POP     AX                      ; RESTORE HEAD
08AC 0B F8                      OR      DI,AX                   ; DI = HEAD OR'ED DRIVE
08AE E8 08D7 R                  CALL    READ_ID                 ; READ ID HEAD 0
08B1 9C                         PUSHF                           ; SAVE RETURN FROM READ_ID
08B2 81 E7 00FB                 AND     DI,11111011B            ; TURN OFF HEAD 1 BIT
08B6 9D                         POPF                            ; RESTORE ERROR RETURN
08B7 59                         POP     CX                      ; RESTORE COUNT
08B8 73 08                      JNC     DO_CHK                  ; IF OK, ASKED = RETURNED TRACK ?
08BA FE C5                      INC     CH                      ; INC FOR NEXT TRACK
08BC 3A E9                      CMP     CH,CL                   ; REACHED MAXIMUM YET
08BE 75 D5                      JNZ     CNT_OK                  ; CONTINUE TILL ALL TRIED
                        ;-----  FALL THRU, READ ID FAILED FOR ALL TRACKS
08C0 F9                 SD_ERR: STC                             ; SET CARRY FOR ERROR
08C1 C3                         RET                             ; SETUP_DBL ERROR EXIT
08C2 8A 0E 0045 R       DO_CHK: MOV     CL,@NEC_STATUS+3        ; LOAD RETURNED TRACK
08C6 88 8D 0094 R               MOV     @DSK_TRK[DI],CL         ; STORE TRACK NUMBER
08CA D0 ED                      SHR     CH,1                    ; HALVE TRACK
08CC 3A E9                      CMP     CH,CL                   ; IS IT THE SAME AS ASKED FOR TRACK
08CE 74 05                      JZ      NO_DBL                  ; IF SAME THEN NO DOUBLE STEP
08D0 80 8D 0090 R 20            OR      @DSK_STATE[DI],DBL_STEP ; TURN ON DOUBLE STEP REQUIRED
08D5 F8                 NO_DBL: CLC                             ; CLEAR ERROR FLAG
08D6 C3                         RET
08D7                    SETUP_DBL       ENDP
                        ;--------------------------------------------------------
                        ; READ_ID                                               :
                        ;       READ ID FUNCTION.                               :
                        ; ON ENTRY:   DI = BIT 2 = HEAD; BITS 1,0 = DRIVE       :
                        ;                                                       :
                        ; ON EXIT:    DI = BIT 2 IS RESET, BITS 1,0 = DRIVE     :
                        ;             @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION :
                        ;--------------------------------------------------------
08D7                    READ_ID PROC    NEAR
08D7 B8 08EB R                  MOV     AX,OFFSET ER_3          ; MOVE NEC OUTPUT ERROR ADDRESS
08DA 50                         PUSH    AX
08DB B4 4A                      MOV     AH,4AH                  ; READ ID COMMAND
08DD E8 09F8 R                  CALL    NEC_OUTPUT              ; TO CONTROLLER
08E0 8B C7                      MOV     AX,DI                   ; DRIVE # TO AH, HEAD 0
08E2 8A E0                      MOV     AH,AL
08E4 E8 09F8 R                  CALL    NEC_OUTPUT              ; TO CONTROLLER
08E7 E8 075B R                  CALL    NEC_TERM                ; WAIT FOR OPERATION, GET STATUS
08EA 58                         POP     AX                      ; THROW AWAY ERROR ADDRESS
08EB                    ER_3:
08EB C3                         RET
08EC                    READ_ID ENDP
                        ;--------------------------------------------------------
                        ; CMOS_TYPE                                             :
                        ;       RETURNS DISKETTE TYPE FROM CMOS                 :
                        ;                                                       :
                        ; ON ENTRY:   DI = DRIVE #                              :
                        ;                                                       :
                        ; ON EXIT:    AL = TYPE; CY REFLECTS STATUS             :
                        ;--------------------------------------------------------
08EC                    CMOS_TYPE       PROC    NEAR
08EC B0 0E                      MOV     AL,CMOS_DIAG            ; CMOS DIAGNOSTIC STATUS BYTE ADDRESS
08EE E8 0000 E                  CALL    CMOS_READ               ; GET CMOS STATUS
08F1 A8 C0                      TEST    AL,BAD_BAT+BAD_CKSUM    ; BATTERY GOOD AND CHECKSUM VALID ?
08F3 F9                         STC                             ; SET CY = 1 INDICATING ERROR FOR RETURN
08F4 75 0E                      JNZ     BAD_CM                  ; ERROR IF EITHER BIT ON
08F6 B0 10                      MOV     AL,CMOS_DISKETTE        ; ADDRESS OF DISKETTE BYTE IN CMOS
08F8 E8 0000 E                  CALL    CMOS_READ               ; GET DISKETTE BYTE
08FB 0B FF                      OR      DI,DI                   ; SEE WHICH DRIVE IN QUESTION
08FD 75 03                      JNZ     TB                      ; IF DRIVE 1, DATA IN LOW NIBBLE
08FF C0 C8 04                   ROR     AL,4                    ; EXCHANGE NIBBLES IF SECOND DRIVE
0902                    TB:
0902 24 0F                      AND     AL,00FH                 ; KEEP ONLY DRIVE DATA, RESET CY = 0
0904                    BAD_CM:
0904 C3                         RET
0905                    CMOS_TYPE       ENDP
```

```
                        ;--------------------------------------------------------------------
                        ; GET_PARM                                                          :
                        ;       THIS ROUTINE FETCHES THE INDEXED POINTER FROM THE          :
                        ;       DISK_BASE BLOCK POINTED TO BY THE DATA VARIABLE            :
                        ;       @DISK_POINTER. A BYTE FROM THAT TABLE IS THEN MOVED        :
                        ;       INTO AH, THE INDEX OF THAT BYTE BEING THE PARAMETER        :
                        ;       IN DL.                                                     :
                        ;                                                                  :
                        ; ON ENTRY:    DL = INDEX OF BYTE TO BE FETCHED                    :
                        ;                                                                  :
                        ; ON EXIT:     AH = THAT BYTE FROM BLOCK                           :
                        ;              AL,DH DESTROYED                                     :
                        ;--------------------------------------------------------------------
0052 0905               GET_PARM        PROC    NEAR
0053 0905 1E                    PUSH    DS
0054 0906 56                    PUSH    SI
0055 0907 2B C0                 SUB     AX,AX                   ; DS = 0 , BIOS DATA AREA
0056 0909 8E D8                 MOV     DS,AX
0057 090B 87 D3                 XCHG    DX,BX                   ; BL = INDEX
0058 090D 2A FF                 SUB     BH,BH                   ; BX = INDEX
0059                            ASSUME  DS:ABS0
0060 090F C5 36 0078 R          LDS     SI,@DISK_POINTER        ; POINT TO BLOCK
0061 0913 8A 20                 MOV     AH,[SI+BX]              ; GET THE WORD
0062 0915 87 D3                 XCHG    DX,BX                   ; RESTORE BX
0063 0917 5E                    POP     SI
0064 0918 1F                    POP     DS
0065 0919 C3                    RET
0066                            ASSUME  DS:DATA
0067 091A               GET_PARM        ENDP
0068                    ;--------------------------------------------------------------------
0069                    ; MOTOR_ON                                                         :
0070                    ;       TURN MOTOR ON AND WAIT FOR MOTOR START UP TIME. THE @MOTOR_COUNT:
0071                    ;       IS REPLACED WITH A SUFFICIENTLY HIGH NUMBER (0FFH) TO ENSURE:
0072                    ;       THAT THE MOTOR DOES NOT GO OFF DURING THE OPERATION. IF THE :
0073                    ;       MOTOR NEEDED TO BE TURNED ON, THE MULTITASKING HOOK FUNCTION:
0074                    ;       (AX=90FDH, INT 15H) IS CALLED TELLING THE OPERATING SYSTEM :
0075                    ;       THAT THE BIOS IS ABOUT TO WAIT FOR MOTOR START UP. IF THIS :
0076                    ;       FUNCTION RETURNS WITH CY = 1, IT MEANS THAT THE MINIMUM WAIT:
0077                    ;       HAS BEEN COMPLETED. AT THIS POINT A CHECK IS MADE TO ENSURE :
0078                    ;       THAT THE MOTOR WASN'T TURNED OFF BY THE TIMER. IF THE HOOK DID:
0079                    ;       NOT WAIT, THE WAIT FUNCTION (AH=086H) IS CALLED TO WAIT THE :
0080                    ;       PRESCRIBED AMOUNT OF TIME. IF THE CARRY FLAG IS SET ON RETURN,:
0081                    ;       IT MEANS THAT THE FUNCTION IS IN USE AND DID NOT PERFORM THE :
0082                    ;       WAIT. A TIMER 1 WAIT LOOP WILL THEN DO THE WAIT.           :
0083                    ;                                                                  :
0084                    ; ON ENTRY:    DI = DRIVE #                                        :
0085                    ;                                                                  :
0086                    ; ON EXIT:     AX,CX,DX DESTROYED                                  :
0087                    ;--------------------------------------------------------------------
0088 091A               MOTOR_ON        PROC    NEAR
0089 091A 53                    PUSH    BX                      ; SAVE REG.
0090 091B E8 0965 R             CALL    TURN_ON                 ; TURN ON MOTOR
0091 091E 72 43                 JC      MOT_TS_ON               ; IF CY=1 NO WAIT
0092 0920 E8 0435 R             CALL    XLAT_OLD                ; TRANSLATE STATE TO COMPATIBLE MODE
0093 0923 B8 90FD               MOV     AX,090FDH               ; LOAD WAIT CODE & TYPE
0094 0926 CD 15                 INT     15H                     ; TELL OPERATING SYSTEM ABOUT TO DO WAIT
0095 0928 9C                    PUSHF                           ; SAVE CY FOR TEST
0096 0929 E8 040F R             CALL    XLAT_NEW                ; TRANSLATE STATE TO PRESENT ARCH.
0097 092C 9D                    POPF                            ; RESTORE CY FOR TEST
0098 092D 73 05                 JNC     M_WAIT                  ; BYPASS LOOP IF OP SYSTEM HANDLED WAIT
0099 092F E8 0965 R             CALL    TURN_ON                 ; CHECK AGAIN IF MOTOR ON
0100 0932 72 2F                 JC      MOT_TS_ON               ; IF NO WAIT MEANS IT IS ON
0101
0102 0934               M_WAIT:
0103 0934 B2 0A                 MOV     DL,10                   ; GET THE MOTOR WAIT PARAMETER
0104 0936 E8 0905 R             CALL    GET_PARM
0105 0939 8A C4                 MOV     AL,AH                   ; AL = MOTOR WAIT PARAMETER
0106 093B 32 E4                 XOR     AH,AH                   ; AX = MOTOR WAIT PARAMETER
0107 093D 3C 08                 CMP     AL,8                    ; SEE IF AT LEAST A SECOND IS SPECIFIED
0108 093F 73 02                 JAE     GP2                     ; IF YES, CONTINUE
0109 0941 B0 08                 MOV     AL,8                    ; ONE SECOND WAIT FOR MOTOR START UP
0110
0111                    ;-----  AX CONTAINS NUMBER OF 1/8 SECONDS (125000 MICROSECONDS) TO WAIT
0112
0113 0943 50            GP2:    PUSH    AX                      ; SAVE WAIT PARAMETER
0114 0944 BA F424               MOV     DX,62500                ; LOAD LARGEST POSSIBLE MULTIPLIER
0115 0947 F7 E2                 MUL     DX                      ; MULTIPLY BY HALF OF WHAT'S NECESSARY
0116 0949 8B CA                 MOV     CX,DX                   ; CX = HIGH WORD
0117 094B 8B D0                 MOV     DX,AX                   ; CX,DX = 1/2 * (# OF MICROSECONDS)
0118 094D F8                    CLC                             ; CLEAR CARRY FOR ROTATE
0119 094E D1 D2                 RCL     DX,1                    ; DOUBLE LOW WORD, CY CONTAINS OVERFLOW
0120 0950 D1 D1                 RCL     CX,1                    ; DOUBLE HI, INCLUDING LOW WORD OVERFLOW
0121 0952 B4 86                 MOV     AH,86H                  ; LOAD WAIT CODE
0122 0954 CD 15                 INT     15H                     ; PERFORM WAIT
0123 0956 58                    POP     AX                      ; RESTORE WAIT PARAMETER
0124 0957 73 0A                 JNC     MOT_IS_ON               ; CY MEANS WAIT COULD NOT BE DONE
0125
0126                    ;-----  FOLLOWING LOOPS REQUIRED WHEN RTC WAIT FUNCTION IS ALREADY IN USE
0127
0128 0959               J13:                                    ;       WAIT FOR 1/8 SECOND PER (AL)
0129 0959 B9 205E               MOV     CX,8286                 ; COUNT FOR 1/8 SECOND AT 15.085737 US
0130 095C E8 0000 E             CALL    WAITF                   ; GO TO FIXED WAIT ROUTINE
0131 095F FE C8                 DEC     AL                      ; DECREMENT TIME VALUE
0132 0961 75 F6                 JNZ     J13                     ; ARE WE DONE YET
0133
0134 0963               MOT_IS_ON:
0135 0963 5B                    POP     BX                      ; RESTORE REG.
0136 0964 C3                    RET
0137 0965               MOTOR_ON        ENDP
0138                    ;--------------------------------------------------------------------
0139                    ; TURN_ON                                                          :
0140                    ;       TURN MOTOR ON AND RETURN WAIT STATE.                       :
0141                    ;                                                                  :
0142                    ; ON ENTRY:    DI = DRIVE #                                        :
0143                    ;                                                                  :
0144                    ; ON EXIT:     CY = 0 MEANS WAIT REQUIRED                          :
0145                    ;              CY = 1 MEANS NO WAIT REQUIRED                       :
0146                    ;              AX,BX,CX,DX DESTROYED                               :
0147                    ;--------------------------------------------------------------------
0148 0965               TURN_ON PROC    NEAR
0149 0965 8B DF                 MOV     BX,DI                   ; BX = DRIVE #
0150 0967 8A CB                 MOV     CL,BL                   ; CL = DRIVE #
0151 0969 C0 C3 04              ROL     BL,4                    ; BL = DRIVE SELECT
0152 096C FA                    CLI                             ; NO INTERRUPTS WHILE DETERMINING STATUS
```

```
                        MOV     @MOTOR_COUNT,0FFH       ; ENSURE MOTOR STAYS ON FOR OPERATION
                        MOV     AL,@MOTOR_STATUS        ; GET DIGITAL OUTPUT REGISTER REFLECTION
                        AND     AL,00110000B            ; KEEP ONLY DRIVE SELECT BITS
                        MOV     AH,1                    ; MASK FOR DETERMINING MOTOR BIT
                        SHL     AH,CL                   ; AH = MOTOR ON, A=00000001, B=00000010

                ;  AL = DRIVE SELECT FROM @MOTOR_STATUS
                ;  BL = DRIVE SELECT DESIRED
                ;  AH = MOTOR ON MASK DESIRED

                        CMP     AL,BL                   ; REQUESTED DRIVE ALREADY SELECTED ?
                        JNZ     TURN_IT_ON              ; IF NOT SELECTED JUMP
                        TEST    AH,@MOTOR_STATUS        ; TEST MOTOR ON BIT
                        JNZ     NO_MOT_WAIT             ; JUMP IF MOTOR ON AND SELECTED

                TURN_IT_ON:
                        OR      AH,BL                   ; AH = DRIVE SELECT AND MOTOR ON
                        MOV     BH,@MOTOR_STATUS        ; SAVE COPY OF @MOTOR_STATUS BEFORE
                        AND     BH,00001111B            ; KEEP ONLY MOTOR BITS
                        AND     @MOTOR_STATUS,11001111B ; CLEAR OUT DRIVE SELECT
                        OR      @MOTOR_STATUS,AH        ; OR IN DRIVE SELECTED AND MOTOR ON
                        MOV     AL,@MOTOR_STATUS        ; GET DIGITAL OUTPUT REGISTER REFLECTION
                        MOV     BL,AL                   ; BL=@MOTOR_STATUS AFTER, BH=BEFORE
                        AND     BL,00001111B            ; KEEP ONLY MOTOR BITS
                        STI                             ; ENABLE INTERRUPTS AGAIN
                        AND     AL,00111111B            ; STRIP AWAY UNWANTED BITS
                        ROL     AL,4                    ; PUT BITS IN DESIRED POSITIONS
                        OR      AL,00001100B            ; NO RESET, ENABLE DMA/INTERRUPT
                        MOV     DX,03F2H                ; SELECT DRIVE AND TURN ON MOTOR
                        OUT     DX,AL                   ; "
                        CMP     BL,BH                   ; NEW MOTOR TURNED ON ?
                        JZ      NO_MOT_WAIT             ; NO WAIT REQUIRED IF JUST SELECT
                        CLC                             ; SET CARRY MEANING WAIT
                        RET

                NO_MOT_WAIT:
                        STC                             ; SET NO WAIT REQUIRED
                        STI                             ; INTERRUPTS BACK ON
                        RET
                TURN_ON ENDP
                ;--------------------------------------------------------------------
                ; HD_WAIT                                                           :
                ;       WAIT FOR HEAD SETTLE TIME.                                  :
                ;                                                                   :
                ; ON ENTRY:     DI : DRIVE #                                        :
                ;                                                                   :
                ; ON EXIT:      AX,BX,CX,DX DESTROYED                               :
                ;--------------------------------------------------------------------
                HD_WAIT         PROC    NEAR
                        MOV     DL,9                    ; GET HEAD SETTLE PARAMETER
                        CALL    GET_PARM                ; "
                        TEST    @MOTOR_STATUS,10000000B ; SEE IF A WRITE OPERATION
                        JZ      ISNT_WRITE              ; IF NOT, DO NOT ENFORCE ANY VALUES
                        OR      AH,AH                   ; CHECK FOR ANY WAIT?
                        JNZ     DO_WAT                  ; IF THERE DO NOT ENFORCE
                        MOV     AH,HD12_SETTLE          ; LOAD 1.2M HEAD SETTLE MINIMUM
                        MOV     AL,@DSK_STATE[DI]       ; LOAD STATE
                        AND     AL,RATE_MSK             ; KEEP ONLY RATE
                        CMP     AL,RATE_250             ; 1.2 M DRIVE ?
                        JNZ     DO_WAT                  ; DEFAULT HEAD SETTLE LOADED

                GP3:    MOV     AH,HD320_SETTLE         ; USE 320/360 HEAD SETTLE
                        JMP     SHORT DO_WAT            ; "

                ISNT_WRITE:
                        OR      AH,AH                   ; CHECK FOR NO WAIT
                        JZ      HW_DONE                 ; IF NOT WRITE AND 0 ITS OK

                ;-----  AH CONTAINS NUMBER OF MILLISECONDS TO WAIT

                DO_WAT: MOV     AL,AH                   ; AL = # MILLISECONDS
                        XOR     AH,AH                   ; AX = # MILLISECONDS
                        PUSH    AX                      ; SAVE HEAD SETTLE PARAMETER
                        MOV     DX,1000                 ; SET UP FOR MULTIPLY TO MICROSECONDS
                        MUL     DX                      ; DX,AX = # MICROSECONDS
                        MOV     CX,DX                   ; CX,AX = # MICROSECONDS
                        MOV     DX,AX                   ; CX,DX = # MICROSECONDS
                        MOV     AH,86H                  ; LOAD WAIT CODE
                        INT     15H                     ; PERFORM WAIT
                        POP     AX                      ; RESTORE HEAD SETTLE PARAMETER
                        JNC     HW_DONE                 ; CHECK FOR EVENT WAIT ACTIVE

                J29:                                    ;       1 MILLISECOND LOOP
                        MOV     CX,66                   ; COUNT AT 15.085737 US PER COUNT
                        CALL    WAITF                   ; DELAY FOR 1 MILLISECOND
                        DEC     AL                      ; DECREMENT THE COUNT
                        JNZ     J29                     ; DO AL MILLISECOND # OF TIMES
                HW_DONE:
                        RET
                HD_WAIT ENDP
                ;--------------------------------------------------------------------
                ; NEC_OUTPUT                                                        :
                ;       THIS ROUTINE SENDS A BYTE TO THE NEC CONTROLLER AFTER       :
                ;       TESTING FOR CORRECT DIRECTION AND CONTROLLER READY THIS     :
                ;       ROUTINE WILL TIME OUT IF THE BYTE IS NOT ACCEPTED WITHIN    :
                ;       A REASONABLE AMOUNT OF TIME, SETTING THE DISKETTE STATUS    :
                ;       ON COMPLETION.                                              :
                ;                                                                   :
                ; ON ENTRY:                                                         :
                ;       AH = BYTE TO BE OUTPUT                                      :
                ; ON EXIT:                                                         :
                ;       CY = 0  SUCCESS                                             :
                ;       CY = 1  FAILURE -- DISKETTE STATUS UPDATED                  :
                ;               IF A FAILURE HAS OCCURRED, THE RETURN IS MADE       :
                ;               ONE LEVEL HIGHER THAN THE CALLER OF NEC_OUTPUT.     :
                ;               THIS REMOVES THE REQUIREMENT OF TESTING AFTER       :
                ;               EVERY CALL OF NEC_OUTPUT.                           :
                ;       AX,CX,DX DESTROYED                                          :
                ;--------------------------------------------------------------------
                NEC_OUTPUT      PROC    NEAR
                        PUSH    BX                      ; SAVE REG.
                        MOV     DX,03F4H                ; STATUS PORT
                        MOV     BL,2                    ; HIGH ORDER COUNTER
                        XOR     CX,CX                   ; COUNT FOR TIME OUT
```

```
                        ;-----  FALL THRU TO ERROR RETURN

                        ;-----  DIRECTION AND STATUS OK; OUTPUT BYTE

                        ;--------------------------------------------------------
                        ; SEEK                                                  :
                        ;       THIS ROUTINE WILL MOVE THE HEAD ON THE NAMED DRIVE :
                        ;       TO THE NAMED TRACK.  IF THE DRIVE HAS NOT BEEN ACCESSED :
                        ;       SINCE THE DRIVE RESET COMMAND WAS ISSUED, THE DRIVE :
                        ;       WILL BE RECALIBRATED.                           :
                        ;                                                       :
                        ; ON ENTRY:     DI = DRIVE #                           :
                        ;               CH = TRACK #                           :
                        ;                                                       :
                        ; ON EXIT:      @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION. :
                        ;               AX,BX,CX,DX DESTROYED                  :
                        ;--------------------------------------------------------

                        ;-----  ISSUE RECALIBRATE FOR 80 TRACK DISKETTES

                        ;-----  DRIVE IS IN SYNCHRONIZATION WITH CONTROLLER, SEEK TO TRACK

                        ;-----  WAIT FOR HEAD SETTLE

                        ;--------------------------------------------------------
                        ; RECAL                                                  :
                        ;       RECALIBRATE DRIVE                              :
                        ;                                                       :
                        ; ON ENTRY      DI = DRIVE #                           :
                        ;                                                       :
                        ; ON EXIT:      CY  REFLECTS STATUS OF OPERATION.      :
                        ;--------------------------------------------------------
```

```
0A8D 8B DF                      MOV     BX,DI                   ; BX = DRIVE #
0A8F 8A E3                      MOV     AH,BL
0A91 E8 09F8 R                  CALL    NEC_OUTPUT              ; OUTPUT THE DRIVE NUMBER
0A94 E8 0A9A R                  CALL    CHK_STAT_2              ; GET THE INTERRUPT AND SENSE INT STATUS
0A97 58                         POP     AX                      ; THROW AWAY ERROR
0A98                    RC_BACK:
0A98 59                         POP     CX
0A99 C3                         RET
0A9A                    RECAL   ENDP
                        ;--------------------------------------------------------------
                        ; CHK_STAT_2                                                  :
                        ;       THIS ROUTINE HANDLES THE INTERRUPT RECEIVED AFTER      :
                        ;       RECALIBRATE OR SEEK TO THE ADAPTER. THE               :
                        ;       INTERRUPT IS WAITED FOR, THE INTERRUPT STATUS SENSED, :
                        ;       AND THE RESULT RETURNED TO THE CALLER.                :
                        ;                                                             :
                        ; ON EXIT:     @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION.:
                        ;--------------------------------------------------------------
0A9A                    CHK_STAT_2      PROC    NEAR
0A9A B8 0AB8 R                  MOV     AX,OFFSET CS_BACK       ; LOAD NEC_OUTPUT ERROR ADDRESS
0A9D 50                         PUSH    AX
0A9E E8 0AC1 R                  CALL    WAIT_INT                ; WAIT FOR THE INTERRUPT
0AA1 72 14                      JC      J34                     ; IF ERROR, RETURN IT
0AA3 B4 08                      MOV     AH,08H                  ; SENSE INTERRUPT STATUS COMMAND
0AA5 E8 09F8 R                  CALL    NEC_OUTPUT
0AA8 E8 0AE9 R                  CALL    RESULTS                 ; READ IN THE RESULTS
0AAB 72 0A                      JC      J34
0AAD A0 0042 R                  MOV     AL,@NEC_STATUS          ; GET THE FIRST STATUS BYTE
0AB0 24 60                      AND     AL,01100000B            ; ISOLATE THE BITS
0AB2 3C 60                      CMP     AL,01100000B            ; TEST FOR CORRECT VALUE
0AB4 74 03                      JZ      J35                     ; IF ERROR, GO MARK IT
0AB6 F8                         CLC                             ; GOOD RETURN
0AB7                    J34:
0AB7 58                         POP     AX                      ; THROW AWAY ERROR RETURN
0AB8                    CS_BACK:
0AB8 C3                         RET

0AB9                    J35:
0AB9 80 0E 0041 R 40            OR      @DSKETTE_STATUS,BAD_SEEK
0ABE F9                         STC                             ; ERROR RETURN CODE
0ABF EB F6                      JMP     SHORT J34
0AC1                    CHK_STAT_2      ENDP
                        ;--------------------------------------------------------------
                        ; WAIT_INT                                                    :
                        ;       THIS ROUTINE WAITS FOR AN INTERRUPT TO OCCUR A TIME OUT :
                        ;       ROUTINE TAKES PLACE DURING THE WAIT, SO THAT AN ERROR  :
                        ;       MAY BE RETURNED IF THE DRIVE IS NOT READY.             :
                        ;                                                              :
                        ; ON EXIT:     @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION.:
                        ;--------------------------------------------------------------
0AC1                    WAIT_INT        PROC    NEAR
0AC1 FB                         STI                             ; TURN ON INTERRUPTS, JUST IN CASE
0AC2 F8                         CLC                             ; CLEAR TIMEOUT INDICATOR
0AC3 B8 9001                    MOV     AX,09001H               ; LOAD WAIT CODE AND TYPE
0AC6 CD 15                      INT     15H                     ; PERFORM OTHER FUNCTION
0AC8 72 11                      JC      J36A                    ; BYPASS TIMING LOOP IF TIMEOUT DONE
0ACA B3 0A                      MOV     BL,10                   ; CLEAR THE COUNTERS
0ACC 33 C9                      XOR     CX,CX                   ; FOR 2 SECOND WAIT
0ACE                    J36:
0ACE F6 06 003E R 80            TEST    @SEEK_STATUS,INT_FLAG   ; TEST FOR INTERRUPT OCCURRING
0AD3 75 0C                      JNZ     J37
0AD5 E2 F7                      LOOP    J36                     ; COUNT DOWN WHILE WAITING
0AD7 FE CB                      DEC     BL                      ; SECOND LEVEL COUNTER
0AD9 75 F3                      JNZ     J36

0ADB 80 0E 0041 R 80    J36A:   OR      @DSKETTE_STATUS,TIME_OUT        ; NOTHING HAPPENED
0AE0 F9                         STC                             ; ERROR RETURN
0AE1                    J37:
0AE1 9C                         PUSHF                           ; SAVE CURRENT CARRY
0AE2 80 26 003E R 7F            AND     @SEEK_STATUS,NOT INT_FLAG       ; TURN OFF INTERRUPT FLAG
0AE7 9D                         POPF                            ; RECOVER CARRY
0AE8 C3                         RET                             ; GOOD RETURN CODE
0AE9                    WAIT_INT        ENDP
                        ;--------------------------------------------------------------
                        ; RESULTS                                                     :
                        ;       THIS ROUTINE WILL READ ANYTHING THAT THE NEC CONTROLLER :
                        ;       RETURNS FOLLOWING AN INTERRUPT.                        :
                        ;                                                              :
                        ; ON EXIT:     @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION.:
                        ;       AX,BX,CX,DX DESTROYED                                  :
                        ;--------------------------------------------------------------
0AE9                    RESULTS PROC    NEAR
0AE9 57                         PUSH    DI
0AEA BF 0042 R                  MOV     DI,OFFSET @NEC_STATUS   ; POINTER TO DATA AREA
0AED B3 07                      MOV     BL,7                    ; MAX STATUS BYTES
0AEF BA 03F4                    MOV     DX,03F4H                ; STATUS PORT

                        ;-----  WAIT FOR REQUEST FOR MASTER

0AF2 B7 02              R10:    MOV     BH,2                    ; HIGH ORDER COUNTER
0AF4 33 C9                      XOR     CX,CX                   ; COUNTER
0AF6                    J39:                                    ; WAIT FOR MASTER
0AF6 EC                         IN      AL,DX                   ; GET STATUS
0AF7 24 C0                      AND     AL,11000000B            ; KEEP ONLY STATUS AND DIRECTION
0AF9 3C C0                      CMP     AL,11000000B            ; STATUS 1 AND DIRECTION 1 ?
0AFB 74 0E                      JZ      J42                     ; STATUS AND DIRECTION OK
0AFD E2 F7                      LOOP    J39                     ; LOOP TILL TIMEOUT

0AFF FE CF                      DEC     BH                      ; DECREMENT HIGH ORDER COUNTER
0B01 75 F3                      JNZ     J39                     ; REPEAT TILL DELAY DONE

0B03 80 0E 0041 R 80            OR      @DSKETTE_STATUS,TIME_OUT
0B08 F9                         STC                             ; SET ERROR RETURN
0B09 EB 1B                      JMP     SHORT POPRES            ; POP REGISTERS AND RETURN

                        ;-----  READ IN THE STATUS

0B0B                    J42:
                        ;       JMP     $+2                     ; I/O DELAY
0B0B 42                         INC     DX                      ; POINT AT DATA PORT
0B0C EC                         IN      AL,DX                   ; GET THE DATA
0B0D 88 05                      MOV     [DI],AL                 ; STORE THE BYTE
0B0F 47                         INC     DI                      ; INCREMENT THE POINTER
```

```
0B10 B9 0003                    MOV     CX,3                    ; MINIMUM 24 MICROSECONDS FOR NEC
0B13 E8 0000 E                  CALL    WAITF                   ; WAIT 30 TO 45 MICROSECONDS
0B16 4A                         DEC     DX                      ; POINT AT STATUS PORT
0B17 EC                         IN      AL,DX                   ; GET STATUS
0B18 A8 10                      TEST    AL,00010000B            ; TEST FOR NEC STILL BUSY
0B1A 74 0A                      JZ      POPRES                  ; RESULTS DONE ?

0B1C FE CB                      DEC     BL                      ; DECREMENT THE STATUS COUNTER
0B1E 75 D2                      JNZ     R10                     ; GO BACK FOR MORE
0B20 80 0E 0041 R 20            OR      @DSKETTE_STATUS,BAD_NEC ; TOO MANY STATUS BYTES
0B25 F9                         STC                             ; SET ERROR FLAG

                        ;-----  RESULT OPERATION IS DONE

                        POPRES:
0B26                            POP     DI
0B26 5F                         RET                             ; RETURN WITH CARRY SET
0B27 C3                 RESULTS ENDP
0B28                    ;--------------------------------------------------------------------
                        ; READ_DSKCHNG                                                      :
                        ;       READS THE STATE OF THE DISK CHANGE LINE.                    :
                        ;                                                                   :
                        ; ON ENTRY:     DI = DRIVE #                                       :
                        ;                                                                   :
                        ; ON EXIT:      DI = DRIVE #                                       :
                        ;               ZF = 0 ; DISK CHANGE LINE INACTIVE                 :
                        ;               ZF = 1 ; DISK CHANGE LINE ACTIVE                   :
                        ;               AX,CX,DX DESTROYED                                 :
                        ;--------------------------------------------------------------------
0B28                    READ_DSKCHNG    PROC    NEAR
0B28 E8 091A R                  CALL    MOTOR_ON                ; TURN ON THE MOTOR IF OFF
0B2B BA 03F7                    MOV     DX,03F7H                ; ADDRESS DIGITAL INPUT REGISTER
0B2E EC                         IN      AL,DX                   ; INPUT DIGITAL INPUT REGISTER
0B2F A8 80                      TEST    AL,DSK_CHG              ; CHECK FOR DISK CHANGE LINE ACTIVE
0B31 C3                         RET                             ; RETURN TO CALLER WITH ZERO FLAG SET
0B32                    READ_DSKCHNG    ENDP
                        ;--------------------------------------------------------------------
                        ; DRIVE_DET                                                         :
                        ;       DETERMINES WHETHER DRIVE IS 80 OR 40 TRACKS AND             :
                        ;       UPDATES STATE INFORMATION ACCORDINGLY.                      :
                        ;                                                                   :
                        ; ON ENTRY:     DI = DRIVE #                                       :
                        ;--------------------------------------------------------------------
0B32                    DRIVE_DET       PROC    NEAR
0B32 E8 091A R                  CALL    MOTOR_ON                ; TURN ON MOTOR IF NOT ALREADY ON
0B35 E8 0A83 R                  CALL    RECAL                   ; RECALIBRATE DRIVE
0B38 72 3C                      JC      DD_BAC                  ; ASSUME NO DRIVE PRESENT
0B3A B5 30                      MOV     CH,TRK_SLAP             ; SEEK TO TRACK 48
0B3C E8 0A24 R                  CALL    SEEK                    ; "
0B3F 72 35                      JC      DD_BAC                  ; ERROR NO DRIVE
0B41 B5 0B                      MOV     CH,QUIET_SEEK+1         ; SEEK TO TRACK 10
0B43                    SK_GIN:
0B43 FE CD                      DEC     CH                      ; DECREMENT TO NEXT TRACK
0B45 51                         PUSH    CX                      ; SAVE TRACK
0B46 E8 0A24 R                  CALL    SEEK                    ; "
0B49 72 2C                      JC      POP_BAC                 ; POP AND RETURN
0B4B B8 0B77 R                  MOV     AX,OFFSET POP_BAC       ; LOAD NEC OUTPUT ERROR ADDRESS
0B4E 50                         PUSH    AX                      ; "
0B4F B4 04                      MOV     AH,SENSE_DRV_ST         ; SENSE DRIVE STATUS COMMAND BYTE
0B51 E8 09F8 R                  CALL    NEC_OUTPUT              ; OUTPUT TO NEC
0B54 8B C7                      MOV     AX,DI                   ; AL = DRIVE
0B56 8A E0                      MOV     AH,AL                   ; AH = DRIVE
0B58 E8 09F8 R                  CALL    NEC_OUTPUT              ; OUTPUT TO NEC
0B5B E8 0AE9 R                  CALL    RESULTS                 ; GO GET STATUS
0B5E 58                         POP     AX                      ; THROW AWAY ERROR ADDRESS
0B5F 59                         POP     CX                      ; RESTORE TRACK
0B60 F6 06 0042 R 10            TEST    @NEC_STATUS,HOME        ; TRACK 0 ?
0B65 74 DC                      JZ      SK_GIN                  ; GO TILL TRACK 0
0B67 0A ED                      OR      CH,CH                   ; IS HOME AT TRACK 0 ?
0B69 74 06                      JZ      IS_80                   ; MUST BE 80 TRACK DRIVE

                        ;       DRIVE IS A 360; SET DRIVE TO DETERMINED;
                        ;       SET MEDIA TO DETERMINED AT RATE 250.

0B6B 80 8D 0090 R 94            OR      @DSK_STATE[DI],DRV_DET+MED_DET+RATE_250
0B70 C3                         RET                             ; ALL INFORMATION SET

0B71                    IS_80:
0B71 80 8D 0090 R 01            OR      @DSK_STATE[DI],TRK_CAPA ; SETUP 80 TRACK CAPABILITY
0B76                    DD_BAC:
0B76 C3                         RET

0B77                    POP_BAC:
0B77 59                         POP     CX                      ; THROW AWAY
0B78 C3                         RET

0B79                    DRIVE_DET       ENDP
                        ;--------------------------------------------------------------------
                        ; DISK_INT                                                          :
                        ;       THIS ROUTINE HANDLES THE DISKETTE INTERRUPT.                :
                        ;                                                                   :
                        ; ON EXIT:      THE INTERRUPT FLAG IS SET IN @SEEK_STATUS.          :
                        ;--------------------------------------------------------------------
0B79                    DISK_INT_1      PROC    FAR             ; ENTRY POINT FOR ORG 0EF57H
0B79 50                         PUSH    AX                      ; SAVE WORK REGISTER
0B7A 1E                         PUSH    DS                      ; SAVE REGISTERS
0B7B E8 0000 E                  CALL    DDS                     ; SETUP DATA ADDRESSING
0B7E 80 0E 003E R 80            OR      @SEEK_STATUS,INT_FLAG   ; TURN ON INTERRUPT OCCURRED
0B83 1F                         POP     DS                      ; RESTORE USER (DS)
0B84 B0 20                      MOV     AL,EOI                  ; END OF INTERRUPT MARKER
0B86 E6 20                      OUT     INTA00,AL               ; INTERRUPT CONTROL PORT
0B88 FB                         STI                             ; RE-ENABLE INTERRUPTS
0B89 B8 9101                    MOV     AX,09101H               ; INTERRUPT POST CODE AND TYPE
0B8C CD 15                      INT     15H                     ; GO PERFORM OTHER TASK
0B8E 58                         POP     AX                      ; RECOVER REGISTER
0B8F CF                         IRET                            ; RETURN FROM INTERRUPT
0B90                    DISK_INT_1      ENDP
```

```
                        PAGE
                        ;--------------------------------------------------------------
                        ; DSKETTE_SETUP                                               :
                        ;       THIS ROUTINE DOES A PRELIMINARY CHECK TO SEE WHAT TYPE  :
                        ;       OF DISKETTE DRIVES ARE ATTACH TO THE SYSTEM.           :
                        ;--------------------------------------------------------------
0B90                    DSKETTE_SETUP PROC      NEAR
0B90 50                         PUSH    AX                      ; SAVE REGISTERS
0B91 53                         PUSH    BX
0B92 51                         PUSH    CX
0B93 52                         PUSH    DX
0B94 57                         PUSH    DI
0B95 1E                         PUSH    DS
0B96 E8 0000 E                  CALL    DDS                     ; POINT DATA SEGMENT TO BIOS DATA AREA
0B99 80 0E 00A0 R 01            OR      @RTC_WAIT_FLAG,01       ; NO RTC WAIT, FORCE USE OF LOOP
0B9E 33 FF                      XOR     DI,DI                   ; INITIALIZE DRIVE POINTER
0BA0 C7 06 0090 R 0000          MOV     WORD PTR @DSK_STATE,0   ; INITIALIZE STATES
0BA6 80 26 008B R 33            AND     @LASTRATE,NOT STRT_MSK+SEND_MSK  ; CLEAR START & SEND
0BAB 80 0E 008B R C0            OR      @LASTRATE,SEND_MSK      ; INITIALIZE SENT TO IMPOSSIBLE
0BB0 C6 06 003E R 00            MOV     @SEEK_STATUS,0          ; INDICATE RECALIBRATE NEEDED
0BB5 C6 06 0040 R 00            MOV     @MOTOR_COUNT,0          ; INITIALIZE MOTOR COUNT
0BBA C6 06 003F R 00            MOV     @MOTOR_STATUS,0         ; INITIALIZE DRIVES TO OFF STATE
0BBF C6 06 0041 R 00            MOV     @DSKETTE_STATUS,0       ; NO ERRORS
                        
0BC4                    SUP0:
0BC4 E8 0B32 R                  CALL    DRIVE_DET               ; DETERMINE DRIVE
0BC7 E8 0435 R                  CALL    XLAT_OLD                ; TRANSLATE STATE TO COMPATIBLE MODE
0BCA 47                         INC     DI                      ; POINT TO NEXT DRIVE
0BCB 83 FF 02                   CMP     DI,MAX_DRV              ; SEE IF DONE
0BCE 75 F4                      JNZ     SUP0                    ; REPEAT FOR EACH DRIVE
0BD0 C6 06 003E R 00            MOV     @SEEK_STATUS,0          ; FORCE RECALIBRATE
0BD5 80 26 00A0 R FE            AND     @RTC_WAIT_FLAG,0FEH     ; ALLOW FOR RTC WAIT
0BDA E8 0854 R                  CALL    SETUP_END               ; VARIOUS CLEANUPS
0BDD 1F                         POP     DS                      ; RESTORE CALLERS RESISTERS
0BDE 5F                         POP     DI
0BDF 5A                         POP     DX
0BE0 59                         POP     CX
0BE1 5B                         POP     BX
0BE2 58                         POP     AX
0BE3 C3                         RET
0BE4                    DSKETTE_SETUP ENDP
0BE4                    CODE    ENDS
                                END
```

```
                        PAGE 118,121
                        TITLE DISK ----- 09/25/85  FIXED DISK BIOS
                        .286C
                        .LIST
0000                    CODE    SEGMENT BYTE PUBLIC

                                PUBLIC  DISK_IO
                                PUBLIC  DISK_SETUP
                                PUBLIC  HD_INT

                                EXTRN   CMOS_READ:NEAR
                                EXTRN   CMOS_WRITE:NEAR
                                EXTRN   DDS:NEAR
                                EXTRN   E_MSG:NEAR
                                EXTRN   F1780:NEAR
                                EXTRN   F1781:NEAR
                                EXTRN   F1782:NEAR
                                EXTRN   F1790:NEAR
                                EXTRN   F1791:NEAR
                                EXTRN   FD_TBL:NEAR

                        ;--- INT 13H ------------------------------------------------------------------------
                        ;                                                                                   :
                        ; FIXED DISK I/O INTERFACE                                                          :
                        ;                                                                                   :
                        ;       THIS INTERFACE PROVIDES ACCESS TO 5 1/4" FIXED DISKS THROUGH                :
                        ;       THE IBM FIXED DISK CONTROLLER.                                              :
                        ;                                                                                   :
                        ;       THE  BIOS  ROUTINES  ARE  MEANT  TO  BE  ACCESSED  THROUGH                  :
                        ;       SOFTWARE  INTERRUPTS  ONLY.    ANY  ADDRESSES  PRESENT  IN                 :
                        ;       THESE  LISTINGS  ARE  INCLUDED  ONLY  FOR  COMPLETENESS,                    :
                        ;       NOT  FOR  REFERENCE.   APPLICATIONS  WHICH  REFERENCE  ANY                  :
                        ;       ABSOLUTE  ADDRESSES  WITHIN  THE  CODE  SEGMENTS  OF  BIOS                  :
                        ;       VIOLATE  THE  STRUCTURE  AND  DESIGN  OF  BIOS.                             :
                        ;                                                                                   :
                        ;-----------------------------------------------------------------------------------:
                        ;                                                                                   :
                        ; INPUT  (AH)= HEX COMMAND VALUE                                                    :
                        ;                                                                                   :
                        ;       (AH)= 00H  RESET DISK (DL = 80H,81H) / DISKETTE                             :
                        ;       (AH)= 01H  READ THE STATUS OF THE LAST DISK OPERATION INTO (AL)             :
                        ;                    NOTE: DL < 80H - DISKETTE                                      :
                        ;                          DL > 80H - DISK                                          :
                        ;       (AH)= 02H  READ THE DESIRED SECTORS INTO MEMORY                             :
                        ;       (AH)= 03H  WRITE THE DESIRED SECTORS FROM MEMORY                            :
                        ;       (AH)= 04H  VERIFY THE DESIRED SECTORS                                       :
                        ;       (AH)= 05H  FORMAT THE DESIRED TRACK                                         :
                        ;       (AH)= 06H  UNUSED                                                           :
                        ;       (AH)= 07H  UNUSED                                                           :
                        ;       (AH)= 08H  RETURN THE CURRENT DRIVE PARAMETERS                              :
                        ;       (AH)= 09H  INITIALIZE DRIVE CHARACTERISTICS                                 :
                        ;                    INTERRUPT 41 POINTS TO DATA BLOCK FOR DRIVE 0                  :
                        ;                    INTERRUPT 46 POINTS TO DATA BLOCK FOR DRIVE 1                  :
                        ;       (AH)= 0AH  READ LONG                                                        :
                        ;       (AH)= 0BH  WRITE LONG  (READ & WRITE LONG ENCOMPASS 512 + 4 BYTES ECC)      :
                        ;       (AH)= 0CH  SEEK                                                             :
                        ;       (AH)= 0DH  ALTERNATE DISK RESET (SEE DL)                                    :
                        ;       (AH)= 0EH  UNUSED                                                           :
                        ;       (AH)= 0FH  UNUSED                                                           :
                        ;       (AH)= 10H  TEST DRIVE READY                                                 :
                        ;       (AH)= 11H  RECALIBRATE                                                      :
                        ;       (AH)= 12H  UNUSED                                                           :
                        ;       (AH)= 13H  UNUSED                                                           :
                        ;       (AH)= 14H  CONTROLLER INTERNAL DIAGNOSTIC                                   :
                        ;       (AH)= 15H  READ DASD TYPE                                                   :
                        ;                                                                                   :
                        ;-----------------------------------------------------------------------------------:
                        ;                                                                                   :
                        ;       REGISTERS USED FOR FIXED DISK OPERATIONS                                    :
                        ;                                                                                   :
                        ;               (DL)    -  DRIVE NUMBER      (80H-81H FOR DISK, VALUE CHECKED)      :
                        ;               (DH)    -  HEAD NUMBER       (0-15 ALLOWED, NOT VALUE CHECKED)      :
                        ;               (CH)    -  CYLINDER NUMBER   (0-1023, NOT VALUE CHECKED)(SEE CL)    :
                        ;               (CL)    -  SECTOR NUMBER     (1-17, NOT VALUE CHECKED)              :
                        ;                                                                                   :
                        ;                  NOTE: HIGH 2 BITS OF CYLINDER NUMBER ARE PLACED                  :
                        ;                        IN THE HIGH 2 BITS OF THE CL REGISTER                      :
                        ;                        (10 BITS TOTAL)                                            :
                        ;                                                                                   :
                        ;               (AL)    -  NUMBER OF SECTORS (MAXIMUM POSSIBLE RANGE 1-80H,         :
                        ;                                             FOR READ/WRITE LONG 1-79H)            :
                        ;                                                                                   :
                        ;               (ES:BX) -  ADDRESS OF BUFFER FOR READS AND WRITES,                  :
                        ;                          (NOT REQUIRED FOR VERIFY)                                :
                        ;                                                                                   :
                        ;               FORMAT (AH=5) ES:BX POINTS TO A 512 BYTE BUFFER.  THE FIRST         :
                        ;                          2*(SECTORS/TRACK) BYTES CONTAIN F,N FOR EACH SECTOR.     :
                        ;                          F = 00H FOR A GOOD SECTOR                                :
                        ;                              80H FOR A BAD SECTOR                                 :
                        ;                          N = SECTOR NUMBER                                        :
                        ;                          FOR AN INTERLEAVE OF 2 AND 17 SECTORS/TRACK              :
                        ;                          THE TABLE SHOULD BE:                                     :
                        ;                                                                                   :
                        ;                  DB      00H,01H,00H,0AH,00H,02H,00H,0BH,00H,03H,00H,0CH          :
                        ;                  DB      00H,04H,00H,0DH,00H,05H,00H,0EH,00H,06H,00H,0FH          :
                        ;                  DB      00H,07H,00H,10H,00H,08H,00H,11H,00H,09H                  :
                        ;                                                                                   :
                        ;-----------------------------------------------------------------------------------:
```

```
                        PAGE
                        ;--------------------------------------------------------------------------------
                        ; OUTPUT                                                                        :
                        ;       AH = STATUS OF CURRENT OPERATION                                        :
                        ;            STATUS BITS ARE DEFINED IN THE EQUATES BELOW                       :
                        ;       CY = 0  SUCCESSFUL OPERATION (AH=0 ON RETURN)                           :
                        ;       CY = 1  FAILED OPERATION (AH HAS ERROR REASON)                          :
                        ;                                                                               :
                        ;       NOTE:   ERROR 11H  INDICATES THAT THE DATA READ HAD A RECOVERABLE       :
                        ;               ERROR WHICH WAS CORRECTED BY THE ECC ALGORITHM.  THE DATA       :
                        ;               IS PROBABLY GOOD.  HOWEVER THE BIOS ROUTINE INDICATES AN        :
                        ;               ERROR TO ALLOW THE CONTROLLING PROGRAM A CHANCE TO DECIDE       :
                        ;               FOR ITSELF.  THE ERROR MAY NOT RECUR IF THE DATA IS            :
                        ;               REWRITTEN.                                                      :
                        ;                                                                               :
                        ;       IF DRIVE PARAMETERS WERE REQUESTED (DL >= 80H),                         :
                        ;          INPUT:                                                               :
                        ;            (DL) = DRIVE NUMBER                                                :
                        ;          OUTPUT:                                                              :
                        ;            (DL) = NUMBER OF CONSECUTIVE ACKNOWLEDGING DRIVES ATTACHED (1-2)   :
                        ;                   (CONTROLLER CARD ZERO TALLY ONLY)                           :
                        ;            (DH) = MAXIMUM USEABLE VALUE FOR HEAD NUMBER                       :
                        ;            (CH) = MAXIMUM USEABLE VALUE FOR CYLINDER NUMBER                   :
                        ;            (CL) = MAXIMUM USEABLE VALUE FOR SECTOR NUMBER                     :
                        ;                   AND CYLINDER NUMBER HIGH BITS                               :
                        ;                                                                               :
                        ;       IF READ DASD TYPE WAS REQUESTED,                                        :
                        ;                                                                               :
                        ;       AH = 0 - NOT PRESENT                                                    :
                        ;            1 - DISKETTE - NO CHANGE LINE AVAILABLE                            :
                        ;            2 - DISKETTE - CHANGE LINE AVAILABLE                               :
                        ;            3 - FIXED DISK                                                     :
                        ;       CX,DX = NUMBER OF 512 BYTE BLOCKS WHEN AH = 3                           :
                        ;                                                                               :
                        ;       REGISTERS WILL BE PRESERVED EXCEPT WHEN THEY ARE USED TO RETURN         :
                        ;       INFORMATION.                                                            :
                        ;                                                                               :
                        ;       NOTE: IF AN ERROR IS REPORTED BY THE DISK CODE, THE APPROPRIATE         :
                        ;             ACTION IS TO RESET THE DISK, THEN RETRY THE OPERATION.            :
                        ;                                                                               :
                        ;--------------------------------------------------------------------------------

= 00FF                  SENSE_FAIL      EQU     0FFH            ; NOT IMPLEMENTED
= 00E0                  NO_ERR          EQU     0E0H            ; STATUS ERROR/ERROR REGISTER=0
= 00CC                  WRITE_FAULT     EQU     0CCH            ; WRITE FAULT ON SELECTED DRIVE
= 00BB                  UNDEF_ERR       EQU     0BBH            ; UNDEFINED ERROR OCCURRED
= 00AA                  NOT_RDY         EQU     0AAH            ; DRIVE NOT READY
= 0080                  TIME_OUT        EQU     80H             ; ATTACHMENT FAILED TO RESPOND
= 0040                  BAD_SEEK        EQU     40H             ; SEEK OPERATION FAILED
= 0020                  BAD_CNTLR       EQU     20H             ; CONTROLLER HAS FAILED
= 0011                  DATA_CORRECTED  EQU     11H             ; ECC CORRECTED DATA ERROR
= 0010                  BAD_ECC         EQU     10H             ; BAD ECC ON DISK READ
= 000B                  BAD_TRACK       EQU     0BH             ; NOT IMPLEMENTED
= 000A                  BAD_SECTOR      EQU     0AH             ; BAD SECTOR FLAG DETECTED
= 0009                  DMA_BOUNDARY    EQU     09H             ; DATA EXTENDS TOO FAR
= 0007                  INIT_FAIL       EQU     07H             ; DRIVE PARAMETER ACTIVITY FAILED
= 0005                  BAD_RESET       EQU     05H             ; RESET FAILED
= 0004                  RECORD_NOT_FND  EQU     04H             ; REQUESTED SECTOR NOT FOUND
= 0002                  BAD_ADDR_MARK   EQU     02H             ; ADDRESS MARK NOT FOUND
= 0001                  BAD_CMD         EQU     01H             ; BAD COMMAND PASSED TO DISK I/O

                        ;--------------------------------------------------------
                        ;                                                       :
                        ; FIXED DISK PARAMETER TABLE                            :
                        ;                                                       :
                        ;  -  THE TABLE IS COMPOSED OF A BLOCK DEFINED AS:      :
                        ;                                                       :
                        ;  +0    (1 WORD) - MAXIMUM NUMBER OF CYLINDERS         :
                        ;  +2    (1 BYTE) - MAXIMUM NUMBER OF HEADS             :
                        ;  +3    (1 WORD) - NOT USED/SEE PC-XT                  :
                        ;  +5    (1 WORD) - STARTING WRITE PRECOMPENSATION CYL   :
                        ;  +7    (1 BYTE) - MAXIMUM ECC DATA BURST LENGTH       :
                        ;  +8    (1 BYTE) - CONTROL BYTE                        :
                        ;                   BIT    7 DISABLE RETRIES -OR-      :
                        ;                   BIT    6 DISABLE RETRIES           :
                        ;                   BIT    3 MORE THAN 8 HEADS         :
                        ;  +9    (3 BYTES)- NOT USED/SEE PC-XT                  :
                        ; +12    (1 WORD) - LANDING ZONE                        :
                        ; +14    (1 BYTE) - NUMBER OF SECTORS/TRACK             :
                        ; +15    (1 BYTE) - RESERVED FOR FUTURE USE             :
                        ;                                                       :
                        ;        - TO DYNAMICALLY DEFINE A SET OF PARAMETERS    :
                        ;          BUILD A TABLE FOR UP TO 15 TYPES AND PLACE   :
                        ;          THE CORRESPONDING VECTOR INTO INTERRUPT 41    :
                        ;          FOR DRIVE 0 AND INTERRUPT 46 FOR DRIVE 1.     :
                        ;                                                       :
                        ;--------------------------------------------------------
```

```
                        PAGE
                        ;--------------------------------------------------------
                        ;                                                       :
                        ; HARDWARE SPECIFIC VALUES                              :
                        ;                                                       :
                        ;  -  CONTROLLER I/O PORT                               :
                        ;                                                       :
                        ;     > WHEN READ FROM:                                 :
                        ;       HF_PORT+0 - READ DATA (FROM CONTROLLER TO CPU)   :
                        ;       HF_PORT+1 - GET ERROR REGISTER                  :
                        ;       HF_PORT+2 - GET SECTOR COUNT                    :
                        ;       HF_PORT+3 - GET SECTOR NUMBER                   :
                        ;       HF_PORT+4 - GET CYLINDER LOW                    :
                        ;       HF_PORT+5 - GET CYLINDER HIGH (2 BITS)          :
                        ;       HF_PORT+6 - GET SIZE/DRIVE/HEAD                 :
                        ;       HF_PORT+7 - GET STATUS REGISTER                 :
                        ;                                                       :
                        ;     > WHEN WRITTEN TO:                                :
                        ;       HF_PORT+0 - WRITE DATA (FROM CPU TO CONTROLLER) :
                        ;       HF_PORT+1 - SET PRECOMPENSATION CYLINDER        :
                        ;       HF_PORT+2 - SET SECTOR COUNT                    :
                        ;       HF_PORT+3 - SET SECTOR NUMBER                   :
                        ;       HF_PORT+4 - SET CYLINDER LOW                    :
                        ;       HF_PORT+5 - SET CYLINDER HIGH (2 BITS)          :
                        ;       HF_PORT+6 - SET SIZE/DRIVE/HEAD                 :
                        ;       HF_PORT+7 - SET COMMAND REGISTER                :
                        ;                                                       :
                        ;--------------------------------------------------------

= 01F0                  HF_PORT         EQU     01F0H           ; DISK PORT
= 03F6                  HF_REG_PORT     EQU     03F6H

                        ;-----          STATUS REGISTER

= 0001                  ST_ERROR        EQU     00000001B       ;
= 0002                  ST_INDEX        EQU     00000010B       ;
= 0004                  ST_CORRCTD      EQU     00000100B       ; ECC CORRECTION SUCCESSFUL
= 0008                  ST_DRQ          EQU     00001000B       ;
= 0010                  ST_SEEK_COMPL   EQU     00010000B       ; SEEK COMPLETE
= 0020                  ST_WRT_FLT      EQU     00100000B       ; WRITE FAULT
= 0040                  ST_READY        EQU     01000000B       ;
= 0080                  ST_BUSY         EQU     10000000B       ;

                        ;-----          ERROR REGISTER

= 0001                  ERR_DAM         EQU     00000001B       ; DATA ADDRESS MARK NOT FOUND
= 0002                  ERR_TRK_0       EQU     00000010B       ; TRACK 0 NOT FOUND ON RECAL
= 0004                  ERR_ABORT       EQU     00000100B       ; ABORTED COMMAND
                        ;               EQU     00001000B       ; NOT USED
= 0010                  ERR_ID          EQU     00010000B       ; ID NOT FOUND
                        ;               EQU     00100000B       ; NOT USED
= 0040                  ERR_DATA_ECC    EQU     01000000B
= 0080                  ERR_BAD_BLOCK   EQU     10000000B


= 0010                  RECAL_CMD       EQU     00010000B       ; DRIVE RECAL   (10H)
= 0020                  READ_CMD        EQU     00100000B       ;       READ    (20H)
= 0030                  WRITE_CMD       EQU     00110000B       ;       WRITE   (30H)
= 0040                  VERIFY_CMD      EQU     01000000B       ;       VERIFY  (40H)
= 0050                  FMTTRK_CMD      EQU     01010000B       ; FORMAT TRACK  (50H)
= 0060                  INIT_CMD        EQU     01100000B       ;   INITIALIZE  (60H)
= 0070                  SEEK_CMD        EQU     01110000B       ;       SEEK    (70H)
= 0090                  DIAG_CMD        EQU     10010000B       ; DIAGNOSTIC    (90H)
= 0091                  SET_PARM_CMD    EQU     10010001B       ; DRIVE PARMS   (91H)
= 0001                  NO_RETRIES      EQU     00000001B       ; CMD MODIFIER  (01H)
= 0002                  ECC_MODE        EQU     00000010B       ; CMD MODIFIER  (02H)
= 0008                  BUFFER_MODE     EQU     00001000B       ; CMD MODIFIER  (08H)

= 0002                  MAX_FILE        EQU     2
= 0002                  S_MAX_FILE      EQU     2

= 0025                  DELAY_1         EQU     25H             ; DELAY FOR OPERATION COMPLETE
= 0600                  DELAY_2         EQU     0600H           ; DELAY FOR READY
= 0100                  DELAY_3         EQU     0100H           ; DELAY FOR DATA REQUEST

= 0008                  HF_FAIL         EQU     08H             ; CMOS FLAG IN BYTE 0EH

                        ;-----          COMMAND BLOCK REFERENCE

=                       @CMD_BLOCK      EQU     BYTE PTR [BP]-8 ; @CMD_BLOCK REFERENCES BLOCK HEAD IN SS
                                                                ;  (BP) POINTS TO COMMAND BLOCK TAIL
                                                                ;       AS DEFINED BY THE "ENTER" PARMS
```

SECTION 5

```
                        PAGE
                        ;--------------------------------------------------------
                        ; FIXED DISK I/O SETUP                                  :
                        ;                                                      :
                        ;  -  ESTABLISH TRANSFER VECTORS FOR THE FIXED DISK    :
                        ;  -  PERFORM POWER ON DIAGNOSTICS                     :
                        ;     SHOULD AN ERROR OCCUR A "1701" MESSAGE IS DISPLAYED :
                        ;                                                      :
                        ;--------------------------------------------------------
                                ASSUME  CS:CODE,DS:ABS0                 ; WORK OFF DS REGISTER

0000                    DISK_SETUP      PROC    NEAR
0000 FA                         CLI
0001 B8 ---- R                  MOV     AX,ABS0                         ; GET ABSOLUTE SEGMENT
0004 8E D8                      MOV     DS,AX                           ; SET SEGMENT REGISTER
0006 A1 004C R                  MOV     AX,WORD PTR @ORG_VECTOR         ; GET DISKETTE VECTOR
0009 A3 0100 R                  MOV     WORD PTR @DISK_VECTOR,AX        ;  INTO INT 40H
000C A1 004E R                  MOV     AX,WORD PTR @ORG_VECTOR+2
000F A3 0102 R                  MOV     WORD PTR @DISK_VECTOR+2,AX
0012 C7 06 004C R 01A9 R        MOV     WORD PTR @ORG_VECTOR,OFFSET DISK_IO     ; FIXED DISK HANDLER
0018 8C 0E 004E R               MOV     WORD PTR @ORG_VECTOR+2,CS
001C C7 06 01D8 R 06DA R        MOV     WORD PTR @HDISK_INT,OFFSET HD_INT       ; FIXED DISK INTERRUPT
0022 8C 0E 01DA R               MOV     WORD PTR @HDISK_INT+2,CS
0026 C7 06 0104 R 0000 E        MOV     WORD PTR @HF_TBL_VEC,OFFSET FD_TBL      ; PARM TABLE DRIVE 80
002C 8C 0E 0106 R               MOV     WORD PTR @HF_TBL_VEC+2,CS
0030 C7 06 0118 R 0000 E        MOV     WORD PTR @HFT_TBL_VEC,OFFSET FD_TBL     ; PARM TABLE DRIVE 81
0036 8C 0E 011A R               MOV     WORD PTR @HF1_TBL_VEC+2,CS
003A E4 A1                      IN      AL,INTB01                       ; TURN ON SECOND INTERRUPT CHIP
003C 24 BF                      AND     AL,0BFH
003E EB 00                      JMP     $+2
0040 E6 A1                      OUT     INTB01,AL
0042 E4 21                      IN      AL,INTA01               ; LET INTERRUPTS PASS THRU TO
0044 24 FB                      AND     AL,0FBH                 ;  SECOND CHIP
0046 EB 00                      JMP     $+2
0048 E6 21                      OUT     INTA01,AL

004A FB                         STI
                                ASSUME  DS:DATA,ES:ABS0
004B 1E                         PUSH    DS                      ; MOVE ABS0 POINTER TO
004C 07                         POP     ES                      ; EXTRA SEGMENT POINTER
004D E8 0000 E                  CALL    DDS                     ; ESTABLISH DATA SEGMENT
0050 C6 06 0074 R 00            MOV     @DISK_STATUS1,0         ; RESET THE STATUS INDICATOR
0055 C6 06 0075 R 00            MOV     @HF_NUM,0               ; ZERO NUMBER OF FIXED DISKS
005A C6 06 0076 R 00            MOV     @CONTROL_BYTE,0
005F B0 8E                      MOV     AL,CMOS_DIAG+NMI
0061 E8 0000 E                  CALL    CMOS_READ               ; CHECK CMOS VALIDITY
0064 8A E0                      MOV     AH,AL                   ; SAVE CMOS FLAG
0066 24 C0                      AND     AL,BAD_BAT+BAD_CKSUM    ; CHECK FOR VALID CMOS
0068 74 03                      JZ      L1
006A E9 00F8 R                  JMP     POD_DONE                ; CMOS NOT VALID -- NO FIXED DISKS
006D                    L1:
006D 80 E4 F7                   AND     AH,NOT HF_FAIL          ; ALLOW FIXED DISK IPL
0070 8A C4                      MOV     AL,CMOS_DIAG+NMI        ; WRITE IT BACK
0072 E8 0000 E                  CALL    CMOS_WRITE
0075 B0 92                      MOV     AL,CMOS_DISK+NMI
0077 E8 0000 E                  CALL    CMOS_READ
007A C6 06 0077 R 00            MOV     @PORT_OFF,0             ; ZERO CARD OFFSET
007F 8A D8                      MOV     BL,AL                   ; SAVE FIXED DISK BYTE
0081 25 00F0                    AND     AX,00F0H                ; GET FIRST DRIVE TYPE AS OFFSET
0084 74 72                      JZ      POD_DONE                ; NO FIXED DISKS

0086 3C F0                      CMP     AL,0F0H                 ; CHECK FOR EXTENDED DRIVE TYPE BYTE USE
0088 75 10                      JNE     L2                      ; USE DRIVE TYPE 1 --> 14 IF NOT IN USE

008A B0 99                      MOV     AL,CMOS_DISK_1+NMI      ; GET EXTENDED TYPE FOR DRIVE C:
008C E8 0000 E                  CALL    CMOS_READ               ; FROM CMOS
008F 3C 00                      CMP     AL,0                    ; IS TYPE SET TO ZERO
0091 74 65                      JE      POD_DONE                ; EXIT IF NOT VALID AND NO FIXED DISKS
0093 3C 2F                      CMP     AL,47                   ; IS TYPE WITHIN VALID RANGE
0095 77 61                      JA      POD_DONE                ; EXIT WITH NO FIXED DISKS IF NOT VALID
0097 C1 E0 04                   SHL     AX,4                    ; ADJUST TYPE TO HIGH NIBBLE
009A                    L2:
009A 05 FFF0 E                  ADD     AX,OFFSET FD_TBL-16D    ; COMPUTE OFFSET OF FIRST DRIVE TABLE
009D 26: A3 0104 R              MOV     WORD PTR @HF_TBL_VEC,AX ; SAVE IN VECTOR POINTER
00A1 C6 06 0075 R 01            MOV     @HF_NUM,1               ; AT LEAST ONE DRIVE
00A6 8A C3                      MOV     AL,BL
00A8 C0 E0 04                   SHL     AL,4                    ; GET SECOND DRIVE TYPE
00AB 74 2A                      JZ      SHORT L4                ; ONLY ONE DRIVE
00AD B4 00                      MOV     AH,0

00AF 3C F0                      CMP     AL,0F0H                 ; CHECK FOR EXTENDED DRIVE TYPE BYTE USE
00B1 75 10                      JNE     L3                      ; USE DRIVE TYPE 1 --> 14 IF NOT IN USE

00B3 B0 9A                      MOV     AL,CMOS_DISK_2+NMI      ; GET EXTENDED TYPE FOR DRIVE D:
00B5 E8 0000 E                  CALL    CMOS_READ               ; FROM CMOS
00B8 3C 00                      CMP     AL,0                    ; IS TYPE SET TO ZERO
00BA 74 1B                      JE      L4                      ; SKIP IF SECOND FIXED DISK NOT VALID
00BC 3C 2F                      CMP     AL,47                   ; IS TYPE WITHIN VALID RANGE
00BE 77 17                      JA      L4                      ; SKIP IF NOT VALID
00C0 C1 E0 04                   SHL     AX,4                    ; ADJUST TYPE TO HIGH NIBBLE
00C3                    L3:
00C3 05 FFF0 E                  ADD     AX,OFFSET FD_TBL-16D    ; COMPUTE OFFSET FOR SECOND FIXED DISK
00C6 8B D8                      MOV     BX,AX
00C8 2E: 83 3F 00               CMP     WORD PTR CS:[BX],0      ; CHECK FOR ZERO CYLINDERS IN TABLE
00CC 74 09                      JE      L4                      ; SKIP DRIVE IF NOT A VALID TABLE ENTRY
00CE 26: A3 0118 R              MOV     WORD PTR @HF1_TBL_VEC,AX
00D2 C6 06 0075 R 02            MOV     @HF_NUM,2               ; TWO DRIVES
00D7                    L4:
00D7 B2 80                      MOV     DL,80H                  ; CHECK THE CONTROLLER
00D9 B4 14                      MOV     AH,14H                  ; USE CONTROLLER DIAGNOSTIC COMMAND
00DB CD 13                      INT     13H                     ; CALL BIOS WITH DIAGNOSTIC COMMAND
00DD 72 1A                      JC      CTL_ERRX                ;  DISPLAY ERROR MESSAGE IF BAD RETURN
00DF A1 006C R                  MOV     AX,@TIMER_LOW           ; GET START TIMER COUNTS
00E2 8B D8                      MOV     BX,AX
00E4 05 0444                    ADD     AX,6*182                ; 60 SECONDS* 18.2
00E7 8B C8                      MOV     CX,AX
00E9 E8 0104 R                  CALL    HD_RESET_1              ; SET UP DRIVE 0
00EC 80 3E 0075 R 01            CMP     @HF_NUM,1               ; WERE THERE TWO DRIVES?
00F1 76 05                      JBE     POD_DONE                ; NO-ALL DONE
00F3 B2 81                      MOV     DL,81H                  ; SET UP DRIVE 1
00F5 E8 0104 R                  CALL    HD_RESET_1
00F8                    POD_DONE:
00F8 C3                         RET
```

```
                        ;----- POD ERROR

                        CTL_ERRX:
00F9 BE 0000 E                  MOV     SI,OFFSET F1782         ; CONTROLLER ERROR
00FC E8 017C R                  CALL    SET_FAIL                ; DO NOT IPL FROM DISK
00FF E8 0000 E                  CALL    E_MSG                   ; DISPLAY ERROR AND SET (BP) ERROR FLAG
0102 EB F4                      JMP     POD_DONE

                        HD_RESET_1      PROC    NEAR
0104 53                         PUSH    BX                      ; SAVE TIMER LIMITS
0105 51                         PUSH    CX
0106 B4 09              RES_1:  MOV     AH,09H                  ; SET DRIVE PARAMETERS
0108 CD 13                      INT     13H
010A 72 06                      JC      RES_2
010C B4 11                      MOV     AH,11H                  ; RECALIBRATE DRIVE
010E CD 13                      INT     13H
0110 73 19                      JNC     RES_CK                  ; DRIVE OK
0112 E8 018A R          RES_2:  CALL    POD_TCHK                ; CHECK TIME OUT
0115 73 EF                      JNC     RES_1
0117 BE 0000 E          RES_FL: MOV     SI,OFFSET F1781         ; INDICATE DISK 1 FAILURE
011A F6 C2 01                   TEST    DL,1
011D 75 57                      JNZ     RES_E1
011F BE 0000 E                  MOV     SI,OFFSET F1780         ; INDICATE DISK 0 FAILURE
0122 E8 017C R                  CALL    SET_FAIL                ; DO NOT TRY TO IPL DISK 0
0125 EB 4F                      JMP     SHORT RES_E1
0127 B4 00              RES_RS: MOV     AH,00H                  ; RESET THE DRIVE
0129 CD 13                      INT     13H
012B B4 08              RES_CK: MOV     AH,08H                  ; GET MAX CYLINDER,HEAD,SECTOR
012D 8A DA                      MOV     BL,DL                   ; SAVE DRIVE CODE
012F CD 13                      INT     13H
0131 72 38                      JC      RES_ER
0133 89 0E 0042 R               MOV     WORD PTR @NEC_STATUS,CX ; SAVE MAX CYLINDER, SECTOR
0137 8A D3                      MOV     DL,BL                   ; RESTORE DRIVE CODE
0139 B8 0401            RES_3:  MOV     AX,0401H                ; VERIFY THE LAST SECTOR
013C CD 13                      INT     13H
013E 73 39                      JNC     RES_OK                  ; VERIFY OK
0140 80 FC 0A                   CMP     AH,BAD_SECTOR           ; OK ALSO IF JUST ID READ
0143 74 34                      JE      RES_OK
0145 80 FC 11                   CMP     AH,DATA_CORRECTED
0148 74 2F                      JE      RES_OK
014A 80 FC 10                   CMP     AH,BAD_ECC
014D 74 2A                      JE      RES_OK
014F E8 018A R                  CALL    POD_TCHK                ; CHECK FOR TIME OUT
0152 72 17                      JC      RES_ER                  ; FAILED
0154 8B 0E 0042 R               MOV     CX,WORD PTR @NEC_STATUS ; GET SECTOR ADDRESS, AND CYLINDER
0158 8A C1                      MOV     AL,CL                   ; SEPARATE OUT SECTOR NUMBER
015A 24 3F                      AND     AL,3FH
015C FE C8                      DEC     AL                      ; TRY PREVIOUS ONE
015E 74 C7                      JZ      RES_RS                  ; WE'VE TRIED ALL SECTORS ON TRACK
0160 80 E1 C0                   AND     CL,0C0H                 ; KEEP CYLINDER BITS
0163 0A C8                      OR      CL,AL                   ; MERGE SECTOR WITH CYLINDER BITS
0165 89 0E 0042 R               MOV     WORD PTR @NEC_STATUS,CX ; SAVE CYLINDER, NEW SECTOR NUMBER
0169 EB CE                      JMP     RES_3                   ; TRY AGAIN
016B BE 0000 E          RES_ER: MOV     SI,OFFSET F1791         ; INDICATE DISK 1 ERROR
016E F6 C2 01                   TEST    DL,1
0171 75 03                      JNZ     RES_E1
0173 BE 0000 E                  MOV     SI,OFFSET F1790         ; INDICATE DISK 0 ERROR
0176                    RES_E1:
0176 E8 0000 E                  CALL    E_MSG                   ; DISPLAY ERROR AND SET (BP) ERROR FLAG
0179                    RES_OK:
0179 59                         POP     CX                      ; RESTORE TIMER LIMITS
017A 5B                         POP     BX
017B C3                         RET
017C                    HD_RESET_1      ENDP

017C                    SET_FAIL        PROC    NEAR
017C B8 8E8E                    MOV     AX,X*(CMOS_DIAG+NMI)    ; GET CMOS ERROR BYTE
017F E8 0000 E                  CALL    CMOS_READ
0182 0C 08                      OR      AL,HF_FAIL              ; SET DO NOT IPL FROM DISK FLAG
0184 86 E0                      XCHG    AH,AL                   ; SAVE IT
0186 E8 0000 E                  CALL    CMOS_WRITE              ; PUT IT OUT
0189 C3                         RET
018A                    SET_FAIL        ENDP

018A                    POD_TCHK        PROC NEAR               ; CHECK FOR 30 SECOND TIME OUT
018A 58                         POP     AX                      ; SAVE RETURN
018B 59                         POP     CX                      ; GET TIME OUT LIMITS
018C 5B                         POP     BX
018D 53                         PUSH    BX                      ; AND SAVE THEM AGAIN
018E 51                         PUSH    CX
018F 50                         PUSH    AX                      ; RESTORE RETURN
0190 A1 006C R                  MOV     AX,@TIMER_LOW           ; AX = CURRENT TIME
                                                                ; BX = START TIME
                                                                ; CX = END TIME
0193 3B D9                      CMP     BX,CX
0195 72 06                      JB      TCHK1                   ; START < END
0197 3B D8                      CMP     BX,AX
0199 72 0C                      JB      TCHKG                   ; END < START < CURRENT
019B EB 04                      JMP     SHORT TCHK2             ; END, CURRENT < START
019D 3B C3              TCHK1:  CMP     AX,BX
019F 72 04                      JB      TCHKNG                  ; CURRENT < START < END
01A1 3B C1              TCHK2:  CMP     AX,CX
01A3 72 02                      JB      TCHKG                   ; START < CURRENT < END
                                                                ; OR CURRENT < END < START
01A5 F9                 TCHKNG: STC                             ; CARRY SET INDICATES TIME OUT
01A6 C3                         RET
01A7 F8                 TCHKG:  CLC                             ; INDICATE STILL TIME
01A8 C3                         RET
01A9                    POD_TCHK        ENDP

01A9                    DISK_SETUP      ENDP
```

```
                        PAGE
                        ;--------------------------------------
                        ;       FIXED DISK BIOS ENTRY POINT    :
                        ;--------------------------------------

01A9                    DISK_IO PROC    FAR
                                ASSUME  DS:DATA,ES:NOTHING
01A9 80 FA 80                   CMP     DL,80H                  ; TEST FOR FIXED DISK DRIVE
01AC 73 05                      JAE     A1                      ; YES, HANDLE HERE
01AE CD 40                      INT     40H                     ; DISKETTE HANDLER
01B0                    RET_2:
01B0 CA 0002                    RET     2                       ; BACK TO CALLER

01B3                    A1:
01B3 FB                         STI                             ; ENABLE INTERRUPTS
01B4 0A E4                      OR      AH,AH
01B6 75 09                      JNZ     A2
01B8 CD 40                      INT     40H                     ; RESET NEC WHEN AH=0
01BA 2A E4                      SUB     AH,AH
01BC 80 FA 81                   CMP     DL,(80H + S_MAX_FILE - 1)
01BF 77 EF                      JA      RET_2
01C1                    A2:
01C1 80 FC 08                   CMP     AH,08H                  ; GET PARAMETERS IS A SPECIAL CASE
01C4 75 03                      JNZ     A3
01C6 E9 0393 R                  JMP     GET_PARM_N
01C9 80 FC 15           A3:     CMP     AH,15H                  ; READ DASD TYPE IS ALSO
01CC 75 03                      JNZ     A4
01CE E9 0353 R                  JMP     READ_DASD_TYPE

01D1                    A4:                                     ;       SAVE REGISTERS DURING OPERATION
01D1 C8 0008 00                 ENTER   8,0                     ; SAVE (BP) AND MAKE ROOM FOR @CMD_BLOCK
01D5 53                         PUSH    BX                      ;  IN THE STACK. THE COMMAND BLOCK IS:
01D6 51                         PUSH    CX                      ;   @CMD_BLOCK == BYTE PTR [BP]-8
01D7 52                         PUSH    DX
01D8 1E                         PUSH    DS
01D9 06                         PUSH    ES
01DA 56                         PUSH    SI
01DB 57                         PUSH    DI
01DC 0A E4                      OR      AH,AH                   ; CHECK FOR RESET
01DE 75 02                      JNZ     A5
01E0 B2 80                      MOV     DL,80H                  ; FORCE DRIVE 80 FOR RESET
01E2 E8 0225 R          A5:     CALL    DISK_IO_CONT            ; PERFORM THE OPERATION
01E5 E8 0000 E                  CALL    DDS                     ; ESTABLISH SEGMENT
01E8 8A 26 0074 R               MOV     AH,@DISK_STATUS1        ; GET STATUS FROM OPERATION
01EC 80 FC 01                   CMP     AH,1                    ; SET THE CARRY FLAG TO INDICATE
01EF F5                         CMC                             ;  SUCCESS OR FAILURE
01F0 5F                         POP     DI                      ; RESTORE REGISTERS
01F1 5E                         POP     SI
01F2 07                         POP     ES
01F3 1F                         POP     DS
01F4 5A                         POP     DX
01F5 59                         POP     CX
01F6 5B                         POP     BX
01F7 C9                         LEAVE                           ; ADJUST (SP) AND RESTORE (BP)
01F8 CA 0002                    RET     2                       ; THROW AWAY SAVED FLAGS
01FB                    DISK_IO ENDP

01FB                    M1      LABEL   WORD                    ; FUNCTION TRANSFER TABLE
01FB 02C1 R                     DW      DISK_RESET              ; 000H
01FD 0315 R                     DW      RETURN_STATUS           ; 001H
01FF 031E R                     DW      DISK_READ               ; 002H
0201 0325 R                     DW      DISK_WRITE              ; 003H
0203 032C R                     DW      DISK_VERF               ; 004H
0205 033E R                     DW      FMT_TRK                 ; 005H
0207 02B9 R                     DW      BAD_COMMAND             ; 006H  FORMAT BAD SECTORS
0209 02B9 R                     DW      BAD_COMMAND             ; 007H  FORMAT DRIVE
020B 02B9 R                     DW      BAD_COMMAND             ; 008H  RETURN PARAMETERS
020D 03F1 R                     DW      INIT_DRV                ; 009H
020F 0423 R                     DW      RD_LONG                 ; 00AH
0211 042A R                     DW      WR_LONG                 ; 00BH
0213 0431 R                     DW      DISK_SEEK               ; 00CH
0215 02C1 R                     DW      DISK_RESET              ; 00DH
0217 02B9 R                     DW      BAD_COMMAND             ; 00EH  READ BUFFER
0219 02B9 R                     DW      BAD_COMMAND             ; 00FH  WRITE BUFFER
021B 044F R                     DW      TST_RDY                 ; 010H
021D 0466 R                     DW      HDISK_RECAL             ; 011H
021F 02B9 R                     DW      BAD_COMMAND             ; 012H  MEMORY DIAGNOSTIC
0221 02B9 R                     DW      BAD_COMMAND             ; 013H  DRIVE DIAGNOSTIC
0223 048E R                     DW      CTLR_DIAGNOSTIC         ; 014H  CONTROLLER DIAGNOSTIC
= 002A                  MIL     EQU     $-M1
                        ;
0225                    DISK_IO_CONT    PROC    NEAR
0225 E8 0000 E                  CALL    DDS                     ; ESTABLISH SEGMENT
0228 80 FC 01                   CMP     AH,01H                  ; RETURN STATUS
022B 75 03                      JNZ     SU0
022D E9 0315 R                  JMP     RETURN_STATUS
0230                    SU0:
0230 C6 06 0074 R 00            MOV     @DISK_STATUS1,0         ; RESET THE STATUS INDICATOR
0235 53                         PUSH    BX                      ; SAVE DATA ADDRESS
0236 8A 1E 0075 R               MOV     BL,@HF_NUM              ; GET NUMBER OF DRIVES
023A 50                         PUSH    AX
023B 80 E2 7F                   AND     DL,7FH                  ; GET DRIVE AS 0 OR 1
023E 3A DA                      CMP     BL,DL
0240 76 75                      JBE     BAD_COMMAND_POP         ; INVALID DRIVE
0242 06                         PUSH    ES
0243 E8 06C4 R                  CALL    GET_VEC                 ; GET DISK PARAMETERS
0246 26: 8B 47 05               MOV     AX,WORD PTR ES:[BX][5]  ; GET WRITE PRE-COMPENSATION CYLINDER
024A C1 E8 02                   SHR     AX,2
024D 88 46 F8                   MOV     @CMD_BLOCK,AL
0250 26: 8A 47 08               MOV     AL,BYTE PTR ES:[BX][8]  ; GET CONTROL BYTE MODIFIER
0254 52                         PUSH    DX
0255 BA 03F6                    MOV     DX,HF_REG_PORT
0258 EE                         OUT     DX,AL                   ; SET EXTRA HEAD OPTION
0259 5A                         POP     DX
025A 07                         POP     ES
025B 8A 26 0076 R               MOV     AH,@CONTROL_BYTE        ; SET EXTRA HEAD OPTION IN
025F 80 E4 C0                   AND     AH,0C0H                 ; CONTROL BYTE
0262 0A E0                      OR      AH,AL
0264 88 26 0076 R               MOV     @CONTROL_BYTE,AH
0268 58                         POP     AX
0269 88 46 F9                   MOV     @CMD_BLOCK+1,AL         ; SECTOR COUNT
026C 50                         PUSH    AX
026D 8A C1                      MOV     AL,CL                   ; GET SECTOR NUMBER
026F 24 3F                      AND     AL,3FH
```

```
0271 88 46 FA                   MOV     @CMD_BLOCK+2,AL
0274 88 6E FB                   MOV     @CMD_BLOCK+3,CH         ; GET CYLINDER NUMBER
0277 8A C1                      MOV     AL,CL
0279 C0 E8 06                   SHR     AL,6
027C 88 46 FC                   MOV     @CMD_BLOCK+4,AL         ; CYLINDER HIGH ORDER 2 BITS
027F 8A C2                      MOV     AL,DL                   ; DRIVE NUMBER
0281 C0 E0 04                   SHL     AL,4
0284 80 E6 0F                   AND     DH,0FH                  ; HEAD NUMBER
0287 0A C6                      OR      AL,DH
0289 0C A0                      OR      AL,80H OR 20H           ; ECC AND 512 BYTE SECTORS
028B 88 46 FD                   MOV     @CMD_BLOCK+5,AL         ; ECC/SIZE/DRIVE/HEAD
028E 58                         POP     AX
028F 50                         PUSH    AX
0290 8A C4                      MOV     AL,AH                   ; GET INTO LOW BYTE
0292 32 E4                      XOR     AH,AH                   ; ZERO HIGH BYTE
0294 D1 E0                      SAL     AX,1                    ; *2 FOR TABLE LOOKUP
0296 8B F0                      MOV     SI,AX                   ; PUT INTO SI FOR BRANCH
0298 3D 002A                    CMP     AX,MIL                  ; TEST WITHIN RANGE
029B 73 1A                      JNB     BAD_COMMAND_POP
029D 58                         POP     AX                      ; RESTORE AX
029E 5B                         POP     BX                      ; AND DATA ADDRESS
029F 51                         PUSH    CX
02A0 50                         PUSH    AX                      ; ADJUST ES:BX
02A1 8B CB                      MOV     CX,BX                   ; GET 3 HIGH ORDER NIBBLES OF BX
02A3 C1 E9 04                   SHR     CX,4
02A6 8C C0                      MOV     AX,ES
02A8 03 C1                      ADD     AX,CX
02AA 8E C0                      MOV     ES,AX
02AC 81 E3 000F                 AND     BX,000FH                ; ES:BX CHANGED TO ES:000X
02B0 58                         POP     AX
02B1 59                         POP     CX
02B2 2E: FF A4 01FB R           JMP     WORD PTR CS:[SI + OFFSET MI]
02B7                    BAD_COMMAND_POP:
02B7 58                         POP     AX
02B8 5B                         POP     BX
02B9                    BAD_COMMAND:
02B9 C6 06 0074 R 01            MOV     @DISK_STATUS1,BAD_CMD   ; COMMAND ERROR
02BE B0 00                      MOV     AL,0
02C0 C3                         RET
02C1                    DISK_IO_CONT    ENDP

                        ;--------------------------------------
                        ;       RESET THE DISK SYSTEM  (AH=00H) :
                        ;--------------------------------------

02C1                    DISK_RESET      PROC    NEAR
02C1 FA                         CLI
02C2 E4 A1                      IN      AL,INTB01               ; GET THE MASK REGISTER
02C4 EB 00                      JMP     $+2
02C6 24 BF                      AND     AL,0BFH                 ; ENABLE FIXED DISK INTERRUPT
02C8 E6 A1                      OUT     INTB01,AL
02CA FB                         STI                             ; START INTERRUPTS
02CB B0 04                      MOV     AL,04H
02CD BA 03F6                    MOV     DX,HF_REG_PORT
02D0 EE                         OUT     DX,AL                   ; RESET
02D1 B9 000A                    MOV     CX,10                   ; DELAY COUNT
02D4 49                 DRD:    DEC     CX
02D5 75 FD                      JNZ     DRD                     ; WAIT 4.8 MICRO-SEC
02D7 A0 0076 R                  MOV     AL,@CONTROL_BYTE
02DA 24 0F                      AND     AL,0FH                  ; SET HEAD OPTION
02DC EE                         OUT     DX,AL                   ; TURN RESET OFF
02DD E8 05F3 R                  CALL    NOT_BUSY
02E0 75 2D                      JNZ     DRERR                   ; TIME OUT ON RESET
02E2 BA 01F1                    MOV     DX,HF_PORT+1
02E5 EC                         IN      AL,DX                   ; GET RESET STATUS
02E6 3C 01                      CMP     AL,1
02E8 75 25                      JNZ     DRERR                   ; BAD RESET STATUS
02EA 80 66 FD EF                AND     @CMD_BLOCK+5,0EFH       ; SET TO DRIVE 0
02EE 2A D2                      SUB     DL,DL
02F0 E8 03F1 R                  CALL    INIT_DRV                ; SET MAX HEADS
02F3 E8 0466 R                  CALL    HDISK_RECAL             ; RECAL TO RESET SEEK SPEED
02F6 80 3E 0075 R 01            CMP     @HF_NUM,1               ; CHECK FOR DRIVE 1
02FB 76 0C                      JBE     DRE
02FD 80 4E FD 10                OR      @CMD_BLOCK+5,010H       ; SET TO DRIVE 1
0301 B2 01                      MOV     DL,1
0303 E8 03F1 R                  CALL    INIT_DRV                ; SET MAX HEADS
0306 E8 0466 R                  CALL    HDISK_RECAL             ; RECAL TO RESET SEEK SPEED
0309 C6 06 0074 R 00    DRE:    MOV     @DISK_STATUS1,0         ; IGNORE ANY SET UP ERRORS
030E C3                         RET
030F C6 06 0074 R 05    DRERR:  MOV     @DISK_STATUS1,BAD_RESET ; CARD FAILED
0314 C3                         RET
0315                    DISK_RESET      ENDP

                        ;--------------------------------------
                        ;       DISK STATUS ROUTINE  (AH = 01H) :
                        ;--------------------------------------

0315                    RETURN_STATUS   PROC    NEAR
0315 A0 0074 R                  MOV     AL,@DISK_STATUS1        ; OBTAIN PREVIOUS STATUS
0318 C6 06 0074 R 00            MOV     @DISK_STATUS1,0         ; RESET STATUS
031D C3                         RET
031E                    RETURN_STATUS   ENDP
```

```
                        PAGE
                        ;--------------------------------------
                        ;       DISK READ ROUTINE     (AH = 02H) :
                        ;--------------------------------------

031E                    DISK_READ       PROC    NEAR
031E C6 46 FE 20                MOV     @CMD_BLOCK+6,READ_CMD
0322 E9 04C6 R                  JMP     COMMAND1
0325                    DISK_READ       ENDP

                        ;--------------------------------------
                        ;       DISK WRITE ROUTINE    (AH = 03H) :
                        ;--------------------------------------

0325                    DISK_WRITE      PROC    NEAR
0325 C6 46 FE 30                MOV     @CMD_BLOCK+6,WRITE_CMD
0329 E9 0505 R                  JMP     COMMAND0
032C                    DISK_WRITE      ENDP

                        ;--------------------------------------
                        ;       DISK VERIFY           (AH = 04H) :
                        ;--------------------------------------

032C                    DISK_VERF       PROC    NEAR
032C C6 46 FE 40                MOV     @CMD_BLOCK+6,VERIFY_CMD
0330 E8 055C R                  CALL    COMMAND
0333 75 08                      JNZ     VERF_EXIT               ; CONTROLLER STILL BUSY
0335 E8 05C2 R                  CALL    WAIT
0338 75 03                      JNZ     VERF_EXIT               ; TIME OUT
033A E8 0630 R                  CALL    CHECK_STATUS
033D                    VERF_EXIT:
033D C3                         RET
033E                    DISK_VERF       ENDP

                        ;--------------------------------------
                        ;       FORMATTING            (AH = 05H) :
                        ;--------------------------------------

033E                    FMT_TRK PROC    NEAR                    ; FORMAT TRACK  (AH = 005H)
033E C6 46 FE 50                MOV     @CMD_BLOCK+6,FMTTRK_CMD
0342 06                         PUSH    ES
0343 53                         PUSH    BX
0344 E8 06C4 R                  CALL    GET_VEC                 ; GET DISK PARAMETERS ADDRESS
0347 26: 8A 47 0E               MOV     AL,ES:[BX][14]          ; GET SECTORS/TRACK
034B 88 46 F9                   MOV     @CMD_BLOCK+1,AL         ; SET SECTOR COUNT IN COMMAND
034E 5B                         POP     BX
034F 07                         POP     ES
0350 E9 050A R                  JMP     CMD_OF                  ; GO EXECUTE THE COMMAND
0353                    FMT_TRK ENDP


                        ;--------------------------------------
                        ;       READ DASD TYPE        (AH = 15H) :
                        ;--------------------------------------

0353                    READ_DASD_TYPE  LABEL   NEAR
0353                    READ_D_T        PROC    FAR             ; GET DRIVE PARAMETERS
0353 1E                         PUSH    DS                      ; SAVE REGISTERS
0354 06                         PUSH    ES
0355 53                         PUSH    BX
                                ASSUME  DS:DATA
0356 E8 0000 E                  CALL    DDS                     ; ESTABLISH ADDRESSING
0359 C6 06 0074 R 00            MOV     @DISK_STATUS1,0
035E 8A 1E 0075 R               MOV     BL,@HF_NUM              ; GET NUMBER OF DRIVES
0362 80 E2 7F                   AND     DL,7FH                  ; GET DRIVE NUMBER
0365 3A DA                      CMP     BL,DL
0367 76 22                      JBE     RDT_NOT_PRESENT         ; RETURN DRIVE NOT PRESENT
0369 E8 06C4 R                  CALL    GET_VEC                 ; GET DISK PARAMETER ADDRESS
036C 26: 8A 47 02               MOV     AL,ES:[BX][2]           ; HEADS
0370 26: 8A 4F 0E               MOV     CL,ES:[BX][14]
0374 F6 E9                      IMUL    CL                      ; * NUMBER OF SECTORS
0376 26: 8B 0F                  MOV     CX,ES:[BX]              ; MAX NUMBER OF CYLINDERS
0379 49                         DEC     CX                      ; LEAVE ONE FOR DIAGNOSTICS
037A F7 E9                      IMUL    CX                      ; NUMBER OF SECTORS
037C 8B CA                      MOV     CX,DX                   ; HIGH ORDER HALF
037E 8B D0                      MOV     DX,AX                   ; LOW ORDER HALF
0380 2B C0                      SUB     AX,AX
0382 B4 03                      MOV     AH,03H                  ; INDICATE FIXED DISK
0384 5B                 RDT2:   POP     BX                      ; RESTORE REGISTERS
0385 07                         POP     ES
0386 1F                         POP     DS
0387 F8                         CLC                             ; CLEAR CARRY
0388 CA 0002                    RET     2
038B                    RDT_NOT_PRESENT:
038B 2B C0                      SUB     AX,AX                   ; DRIVE NOT PRESENT RETURN
038D 8B C8                      MOV     CX,AX                   ; ZERO BLOCK COUNT
038F 8B D0                      MOV     DX,AX
0391 EB F1                      JMP     RDT2
0393                    READ_D_T        ENDP
```

```
                        PAGE
                        ;--------------------------------------
                        ;       GET PARAMETERS          (AH = 08H) :
                        ;--------------------------------------

0393                    GET_PARM_N      LABEL   NEAR
0393                    GET_PARM        PROC    FAR                     ; GET DRIVE PARAMETERS
0393 1E                         PUSH    DS                              ; SAVE REGISTERS
0394 06                         PUSH    ES
0395 53                         PUSH    BX
                                ASSUME  DS:ABS0
0396 B8 ---- R                  MOV     AX,ABS0                         ; ESTABLISH ADDRESSING
0399 8E D8                      MOV     DS,AX
039B F6 C2 01                   TEST    DL,1                            ; CHECK FOR DRIVE 1
039E 74 06                      JZ      G0
03A0 C4 1E 0118 R               LES     BX,@HF1_TBL_VEC
03A4 EB 04                      JMP     SHORT G1
03A6 C4 1E 0104 R       G0:     LES     BX,@HF_TBL_VEC
                                ASSUME  DS:DATA
03AA                    G1:
03AA E8 0000 E                  CALL    DDS                             ; ESTABLISH SEGMENT
03AD 80 EA 80                   SUB     DL,80H
03B0 80 FA 02                   CMP     DL,MAX_FILE                     ; TEST WITHIN RANGE
03B3 73 2C                      JAE     G4
03B5 C6 06 0074 R 00            MOV     @DISK_STATUS1,0
03BA 26: 8B 07                  MOV     AX,ES:[BX]                      ; MAX NUMBER OF CYLINDERS
03BD 2D 0002                    SUB     AX,2                            ; ADJUST FOR 0-N
03C0 8A E8                      MOV     CH,AL
03C2 25 0300                    AND     AX,0300H                        ; HIGH TWO BITS OF CYLINDER
03C5 D1 E8                      SHR     AX,1
03C7 D1 E8                      SHR     AX,1
03C9 26: 0A 47 0E               OR      AL,ES:[BX][14]                  ; SECTORS
03CD 8A C8                      MOV     CL,AL
03CF 26: 8A 77 02               MOV     DH,ES:[BX][2]                   ; HEADS
03D3 FE CE                      DEC     DH                              ; 0-N RANGE
03D5 8A 16 0075 R               MOV     DL,@HF_NUM                      ; DRIVE COUNT
03D9 2B C0                      SUB     AX,AX
03DB                    G5:
03DB 5B                         POP     BX                              ; RESTORE REGISTERS
03DC 07                         POP     ES
03DD 1F                         POP     DS
03DE CA 0002                    RET     2
03E1                    G4:
03E1 C6 06 0074 R 07            MOV     @DISK_STATUS1,INIT_FAIL ; OPERATION FAILED
03E6 B4 07                      MOV     AH,INIT_FAIL
03E8 2A C0                      SUB     AL,AL
03EA 2B D2                      SUB     DX,DX
03EC 2B C9                      SUB     CX,CX
03EE F9                         STC                                     ; SET ERROR FLAG
03EF EB EA                      JMP     G5
03F1                    GET_PARM        ENDP

                        ;--------------------------------------
                        ;       INITIALIZE DRIVE        (AH = 09H) :
                        ;--------------------------------------

03F1                    INIT_DRV        PROC    NEAR
03F1 C6 46 FE 91                MOV     @CMD_BLOCK+6,SET_PARM_CMD
03F5 E8 06C4 R                  CALL    GET_VEC                         ; ES:BX -> PARAMETER BLOCK
03F8 26: 8A 47 02               MOV     AL,ES:[BX][2]                   ; GET NUMBER OF HEADS
03FC FE C8                      DEC     AL                              ; CONVERT TO 0-INDEX
03FE 8A 66 FD                   MOV     AH,@CMD_BLOCK+5                 ; GET SDH REGISTER
0401 80 E4 F0                   AND     AH,0F0H                         ; CHANGE HEAD NUMBER
0404 0A E0                      OR      AH,AL                           ;  TO MAX HEAD
0406 88 66 FD                   MOV     @CMD_BLOCK+5,AH
0409 26: 8A 47 0E               MOV     AL,ES:[BX][14]                  ; MAX SECTOR NUMBER
040D 88 46 F9                   MOV     @CMD_BLOCK+1,AL
0410 2B C0                      SUB     AX,AX
0412 88 46 FB                   MOV     @CMD_BLOCK+3,AL                 ; ZERO FLAGS
0415 E8 055C R                  CALL    COMMAND                         ; TELL CONTROLLER
0418 75 08                      JNZ     INIT_EXIT                       ; CONTROLLER BUSY ERROR
041A E8 05F3 R                  CALL    NOT_BUSY                        ; WAIT FOR IT TO BE DONE
041D 75 03                      JNZ     INIT_EXIT                       ; TIME OUT
041F E8 0630 R                  CALL    CHECK_STATUS
0422                    INIT_EXIT:
0422 C3                         RET
0423                    INIT_DRV        ENDP

                        ;--------------------------------------
                        ;       READ LONG               (AH = 0AH) :
                        ;--------------------------------------

0423                    RD_LONG         PROC    NEAR
0423 C6 46 FE 22                MOV     @CMD_BLOCK+6,READ_CMD OR ECC_MODE
0427 E9 04C6 R                  JMP     COMMAND1
042A                    RD_LONG         ENDP

                        ;--------------------------------------
                        ;       WRITE LONG              (AH = 0BH) :
                        ;--------------------------------------

042A                    WR_LONG         PROC    NEAR
042A C6 46 FE 32                MOV     @CMD_BLOCK+6,WRITE_CMD OR ECC_MODE
042E E9 0505 R                  JMP     COMMAND0
0431                    WR_LONG         ENDP

                        ;--------------------------------------
                        ;       SEEK                    (AH = 0CH) :
                        ;--------------------------------------

0431                    DISK_SEEK       PROC    NEAR
0431 C6 46 FE 70                MOV     @CMD_BLOCK+6,SEEK_CMD
0435 E8 055C R                  CALL    COMMAND
0438 75 14                      JNZ     DS_EXIT                         ; CONTROLLER BUSY ERROR
043A E8 05C2 R                  CALL    WAIT
043D 75 0F                      JNZ     DS_EXIT                         ; TIME OUT ON SEEK
043F E8 0630 R                  CALL    CHECK_STATUS
0442 80 3E 0074 R 40            CMP     @DISK_STATUS1,BAD_SEEK
0447 75 05                      JNE     DS_EXIT
0449 C6 06 0074 R 00            MOV     @DISK_STATUS1,0
044E                    DS_EXIT:
044E C3                         RET

044F                    DISK_SEEK       ENDP
```

SECTION 5

```
                        PAGE
                        ;--------------------------------------
                        ;       TEST DISK READY      (AH = 10H) :
                        ;--------------------------------------

044F                    TST_RDY PROC    NEAR
044F E8 05F3 R                  CALL    NOT_BUSY                ; WAIT FOR CONTROLLER
0452 75 11                      JNZ     TR_EX
0454 8A 46 FD                   MOV     AL,@CMD_BLOCK+5         ; SELECT DRIVE
0457 BA 01F6                    MOV     DX,HF_PORT+6
045A EE                         OUT     DX,AL
045B E8 0642 R                  CALL    CHECK_ST                ; CHECK STATUS ONLY
045E 75 05                      JNZ     TR_EX
0460 C6 06 0074 R 00            MOV     @DISK_STATUS1,0         ; WIPE OUT DATA CORRECTED ERROR
0465 C3                 TR_EX:  RET
0466                    TST_RDY ENDP

                        ;--------------------------------------
                        ;       RECALIBRATE          (AH = 11H) :
                        ;--------------------------------------

0466                    HDISK_RECAL     PROC    NEAR
0466 C6 46 FE 10                MOV     @CMD_BLOCK+6,RECAL_CMD
046A E8 055C R                  CALL    COMMAND                 ; START THE OPERATION
046D 75 19                      JNZ     RECAL_EXIT              ; ERROR
046F E8 05C2 R                  CALL    WAIT                    ; WAIT FOR COMPLETION
0472 74 05                      JZ      RECAL_X                 ; TIME OUT ONE OK ?
0474 E8 05C2 R                  CALL    WAIT                    ; WAIT FOR COMPLETION LONGER
0477 75 0F                      JNZ     RECAL_EXIT              ; TIME OUT TWO TIMES IS ERROR
0479                    RECAL_X:
0479 E8 0630 R                  CALL    CHECK_STATUS
047C 80 3E 0074 R 40            CMP     @DISK_STATUS1,BAD_SEEK  ; SEEK NOT COMPLETE
0481 75 05                      JNE     RECAL_EXIT              ; IS OK
0483 C6 06 0074 R 00            MOV     @DISK_STATUS1,0
0488                    RECAL_EXIT:
0488 80 3E 0074 R 00            CMP     @DISK_STATUS1,0
048D C3                         RET
048E                    HDISK_RECAL     ENDP

                        ;--------------------------------------
                        ;       CONTROLLER DIAGNOSTIC (AH = 14H) :
                        ;--------------------------------------

048E                    CTLR_DIAGNOSTIC PROC    NEAR
048E FA                         CLI                             ; DISABLE INTERRUPTS WHILE CHANGING MASK
048F E4 A1                      IN      AL,INTB01               ; TURN ON SECOND INTERRUPT CHIP
0491 24 BF                      AND     AL,0BFH
0493 EB 00                      JMP     $+2
0495 E6 A1                      OUT     INTB01,AL
0497 E4 21                      IN      AL,INTA01               ; LET INTERRUPTS PASS THRU TO
0499 24 FB                      AND     AL,0FBH                 ;  SECOND CHIP
049B EB 00                      JMP     $+2
049D E6 21                      OUT     INTA01,AL
049F FB                         STI
04A0 E8 05F3 R                  CALL    NOT_BUSY                ; WAIT FOR CARD
04A3 75 1A                      JNZ     CD_ERR                  ; BAD CARD
04A5 BA 01F7                    MOV     DX,HF_PORT+7
04A8 B0 90                      MOV     AL,DIAG_CMD             ; START DIAGNOSE
04AA EE                         OUT     DX,AL
04AB E8 05F3 R                  CALL    NOT_BUSY                ; WAIT FOR IT TO COMPLETE
04AE B4 80                      MOV     AH,TIME_OUT
04B0 75 0F                      JNZ     CD_EXIT                 ; TIME OUT ON DIAGNOSTIC
04B2 BA 01F1                    MOV     DX,HF_PORT+1            ; GET ERROR REGISTER
04B5 EC                         IN      AL,DX
04B6 A2 008D R                  MOV     @HF_ERROR,AL            ; SAVE IT
04B9 B4 00                      MOV     AH,0
04BB 3C 01                      CMP     AL,1                    ; CHECK FOR ALL OK
04BD 74 02                      JE      SHORT CD_EXIT
04BF B4 20              CD_ERR: MOV     AH,BAD_CNTLR
04C1                    CD_EXIT:
04C1 88 26 0074 R               MOV     @DISK_STATUS1,AH
04C5 C3                         RET
04C6                    CTLR_DIAGNOSTIC ENDP

                        ;--------------------------------------
                        ; COMMAND1                            :
                        ;       REPEATEDLY INPUTS DATA TILL   :
                        ;       NSECTOR RETURNS ZERO          :
                        ;--------------------------------------
04C6                    COMMAND1:
04C6 E8 06A1 R                  CALL    CHECK_DMA               ; CHECK 64K BOUNDARY ERROR
04C9 72 39                      JC      CMD_ABORT
04CB 8B FB                      MOV     DI,BX
04CD E8 055C R                  CALL    COMMAND                 ; OUTPUT COMMAND
04D0 75 32                      JNZ     CMD_ABORT
04D2                    CMD_11:
04D2 E8 05C2 R                  CALL    WAIT                    ; WAIT FOR DATA REQUEST INTERRUPT
04D5 75 2D                      JNZ     TM_OUT                  ; TIME OUT
04D7 B9 0100                    MOV     CX,256D                 ; SECTOR SIZE IN WORDS
04DA BA 01F0                    MOV     DX,HF_PORT
04DD FA                         CLI
04DE FC                         CLD
04DF F3/ 6D                     REP     INSW                    ; GET THE SECTOR
04E1 FB                         STI
04E2 F6 46 FE 02                TEST    @CMD_BLOCK+6,ECC_MODE   ; CHECK FOR NORMAL INPUT
04E6 74 12                      JZ      CMD_T3
04E8 E8 061A R                  CALL    WAIT_DRQ                ; WAIT FOR DATA REQUEST
04EB 72 17                      JC      TM_OUT
04ED BA 01F0                    MOV     DX,HF_PORT
04F0 B9 0004                    MOV     CX,4                    ; GET ECC BYTES
04F3 EC                 CMD_12: IN      AL,DX
04F4 26: 88 05                  MOV     ES:BYTE PTR [DI],AL     ; GO SLOW FOR BOARD
04F7 47                         INC     DI
04F8 E2 F9                      LOOP    CMD_12
04FA E8 0630 R          CMD_13: CALL    CHECK_STATUS
04FD 75 05                      JNZ     CMD_ABORT               ; ERROR RETURNED
04FF FE 4E F9                   DEC     @CMD_BLOCK+1            ; CHECK FOR MORE
0502 75 CE                      JNZ     SHORT CMD_11
0504                    CMD_ABORT:
0504                    TM_OUT:
0504 C3                         RET
```

```
                                PAGE
                                ;--------------------------------------
                                ; COMMANDO                            :
                                ;       REPEATEDLY OUTPUTS DATA TILL   :
                                ;       NSECTOR RETURNS ZERO           :
                                ;--------------------------------------
0505                            COMMANDO:
0505 E8 06A1 R                          CALL    CHECK_DMA               ; CHECK 64K BOUNDARY ERROR
0508 72 FA                              JC      CMD_ABORT
050A 8B F3                      CMD_OF: MOV     SI,BX
050C E8 055C R                          CALL    COMMAND                 ; OUTPUT COMMAND
050F 75 F3                              JNZ     CMD_ABORT
0511 E8 061A R                          CALL    WAIT_DRQ                ; WAIT FOR DATA REQUEST
0514 72 EE                              JC      TM_OUT                  ; TOO LONG
0516 1E                         CMD_01: PUSH    DS
0517 06                                 PUSH    ES                      ; MOVE ES TO DS
0518 1F                                 POP     DS
0519 B9 0100                            MOV     CX,256D                 ; PUT THE DATA OUT TO THE CARD
051C BA 01F0                            MOV     DX,HF_PORT
051F FA                                 CLI
0520 FC                                 CLD
0521 F3/ 6F                             REP     OUTSW
0523 FB                                 STI
0524 1F                                 POP     DS                      ; RESTORE DS
0525 F6 46 FE 02                        TEST    @CMD_BLOCK+6,ECC_MODE   ; CHECK FOR NORMAL OUTPUT
0529 74 12                              JZ      CMD_03
052B E8 061A R                          CALL    WAIT_DRQ                ; WAIT FOR DATA REQUEST
052E 72 D4                              JC      TM_OUT
0530 BA 01F0                            MOV     DX,HF_PORT
0533 B9 0004                            MOV     CX,4                    ; OUTPUT THE ECC BYTES
0536 26: 8A 04                  CMD_02: MOV     AL,ES:BYTE PTR [SI]
0539 EE                                 OUT     DX,AL
053A 46                                 INC     SI
053B E2 F9                              LOOP    CMD_02
053D                            CMD_03:
053D E8 05C2 R                          CALL    WAIT                    ; WAIT FOR SECTOR COMPLETE INTERRUPT
0540 75 C2                              JNZ     TM_OUT                  ; ERROR RETURNED
0542 E8 0630 R                          CALL    CHECK_STATUS
0545 75 BD                              JNZ     CMD_ABORT
0547 F6 06 008C R 08                    TEST    @HF_STATUS,ST_DRQ       ; CHECK FOR MORE
054C 75 C8                              JNZ     SHORT CMD_01
054E BA 01F2                            MOV     DX,HF_PORT+2            ; CHECK RESIDUAL SECTOR COUNT
0551 EC                                 IN      AL,DX                   ;
0552 A8 FF                              TEST    AL,0FFH                 ;
0554 74 05                              JZ      CMD_04                  ; COUNT = 0  OK
0556 C6 06 0074 R BB                    MOV     @DISK_STATUS1,UNDEF_ERR ; OPERATION ABORTED - PARTIAL TRANSFER
055B                            CMD_04:
055B C3                                 RET

                                ;--------------------------------------------------
                                ; COMMAND                                         :
                                ;       THIS ROUTINE OUTPUTS THE COMMAND BLOCK    :
                                ; OUTPUT                                          :
                                ;       BL = STATUS                               :
                                ;       BH = ERROR REGISTER                       :
                                ;--------------------------------------------------

055C                            COMMAND PROC    NEAR
055C 53                                 PUSH    BX                      ; WAIT FOR SEEK COMPLETE AND READY
055D B9 0600                            MOV     CX,DELAY_2              ; SET INITIAL DELAY BEFORE TEST
0560                            COMMAND1:
0560 51                                 PUSH    CX                      ; SAVE LOOP COUNT
0561 E8 044F R                          CALL    TST_RDY                 ; CHECK DRIVE READY
0564 59                                 POP     CX
0565 74 0B                              JZ      COMMAND2                ; DRIVE IS READY
0567 80 3E 0074 R 80                    CMP     @DISK_STATUS1,TIME_OUT  ; TST_RDY TIMED OUT--GIVE UP
056C 74 48                              JZ      CMD_TIMEOUT
056E E2 F0                              LOOP    COMMAND1                ; KEEP TRYING FOR A WHILE
0570 EB 49                              JMP     SHORT COMMAND4          ; ITS NOT GOING TO GET READY
0572                            COMMAND2:
0572 5B                                 POP     BX
0573 57                                 PUSH    DI
0574 C6 06 008E R 00                    MOV     @HF_INT_FLAG,0          ; RESET INTERRUPT FLAG
0579 FA                                 CLI                             ; INHIBIT INTERRUPTS WHILE CHANGING MASK
057A E4 A1                              IN      AL,INTB01               ; TURN ON SECOND INTERRUPT CHIP
057C 24 BF                              AND     AL,0BFH
057E EB 00                              JMP     $+2
0580 E6 A1                              OUT     INTB01,AL
0582 E4 21                              IN      AL,INTA01               ; LET INTERRUPTS PASS THRU TO
0584 24 FB                              AND     AL,0FBH                 ;  SECOND CHIP
0586 EB 00                              JMP     $+2
0588 E6 21                              OUT     INTA01,AL
058A FB                                 STI
058B 33 FF                              XOR     DI,DI                   ; INDEX THE COMMAND TABLE
058D BA 01F1                            MOV     DX,HF_PORT+1            ; DISK ADDRESS
0590 F6 06 0076 R C0                    TEST    @CONTROL_BYTE,0C0H      ; CHECK FOR RETRY SUPPRESSION
0595 74 11                              JZ      COMMAND3
0597 8A 46 FE                           MOV     AL,@CMD_BLOCK+6         ; YES-GET OPERATION CODE
059A 24 F0                              AND     AL,0F0H                 ; GET RID OF MODIFIERS
059C 3C 20                              CMP     AL,20H                  ; 20H-40H IS READ, WRITE, VERIFY
059E 72 08                              JB      COMMAND3
05A0 3C 40                              CMP     AL,40H
05A2 77 04                              JA      COMMAND3
05A4 80 4E FE 01                        OR      @CMD_BLOCK+6,NO_RETRIES ; VALID OPERATION FOR RETRY SUPPRESS
05A8                            COMMAND3:
05A8 8A 43 F8                           MOV     AL,[@CMD_BLOCK+DI]      ; GET THE COMMAND STRING BYTE
05AB EE                                 OUT     DX,AL                   ; GIVE IT TO CONTROLLER
05AC 47                                 INC     DI                      ; NEXT BYTE IN COMMAND BLOCK
05AD 42                                 INC     DX                      ; NEXT DISK ADAPTER REGISTER
05AE 81 FA 01F8                         CMP     DX,HF_PORT+8            ; ALL DONE?
05B2 75 F4                              JNZ     COMMAND3                ; NO--GO DO NEXT ONE
05B4 5F                                 POP     DI
05B5 C3                                 RET                             ; ZERO FLAG IS SET
05B6                            CMD_TIMEOUT:
05B6 C6 06 0074 R 20                    MOV     @DISK_STATUS1,BAD_CNTLR
05BB                            COMMAND4:
05BB 5B                                 POP     BX
05BC 80 3E 0074 R 00                    CMP     @DISK_STATUS1,0         ; SET CONDITION CODE FOR CALLER
05C1 C3                                 RET
05C2                            COMMAND ENDP
```

```
                        PAGE
                        ;--------------------------------------
                        ;       WAIT FOR INTERRUPT             :
                        ;--------------------------------------
05C2                    WAIT    PROC    NEAR
05C2 FB                         STI                             ; MAKE SURE INTERRUPTS ARE ON
05C3 2B C9                      SUB     CX,CX                   ; SET INITIAL DELAY BEFORE TEST
05C5 F8                         CLC
05C6 B8 9000                    MOV     AX,9000H                ; DEVICE WAIT INTERRUPT
05C9 CD 15                      INT     15H
05CB 72 0F                      JC      WT2                     ; DEVICE TIMED OUT

05CD B3 25                      MOV     BL,DELAY_1              ; SET DELAY COUNT

                        ;-----  WAIT LOOP

05CF F6 06 008E R 80    WT1:    TEST    @HF_INT_FLAG,80H        ; TEST FOR INTERRUPT
05D4 E1 F9                      LOOPZ   WT1
05D6 75 0B                      JNZ     WT3                     ; INTERRUPT--LETS GO
05D8 FE CB                      DEC     BL
05DA 75 F3                      JNZ     WT1                     ; KEEP TRYING FOR A WHILE

05DC C6 06 0074 R 80    WT2:    MOV     @DISK_STATUS1,TIME_OUT  ; REPORT TIME OUT ERROR
05E1 EB 0A                      JMP     SHORT WT4
05E3 C6 06 0074 R 00    WT3:    MOV     @DISK_STATUS1,0
05E8 C6 06 008E R 00            MOV     @HF_INT_FLAG,0
05ED 80 3E 0074 R 00    WT4:    CMP     @DISK_STATUS1,0         ; SET CONDITION CODE FOR CALLER
05F2 C3                         RET
05F3                    WAIT    ENDP

                        ;--------------------------------------
                        ;       WAIT FOR CONTROLLER NOT BUSY   :
                        ;--------------------------------------
05F3                    NOT_BUSY        PROC    NEAR
05F3 FB                         STI                             ; MAKE SURE INTERRUPTS ARE ON
05F4 53                         PUSH    BX
05F5 2B C9                      SUB     CX,CX                   ; SET INITIAL DELAY BEFORE TEST
05F7 BA 01F7                    MOV     DX,HF_PORT+7
05FA B3 25                      MOV     BL,DELAY_1
05FC EC                 NB1:    IN      AL,DX                   ; CHECK STATUS
05FD A8 80                      TEST    AL,ST_BUSY
05FF E0 FB                      LOOPNZ  NB1
0601 74 0B                      JZ      NB2                     ; NOT BUSY--LETS GO
0603 FE CB                      DEC     BL
0605 75 F5                      JNZ     NB1                     ; KEEP TRYING FOR A WHILE
0607 C6 06 0074 R 80            MOV     @DISK_STATUS1,TIME_OUT  ; REPORT TIME OUT ERROR
060C EB 05                      JMP     SHORT NB3
060E C6 06 0074 R 00    NB2:    MOV     @DISK_STATUS1,0
0613 5B                 NB3:    POP     BX
0614 80 3E 0074 R 00            CMP     @DISK_STATUS1,0         ; SET CONDITION CODE FOR CALLER
0619 C3                         RET
061A                    NOT_BUSY        ENDP

                        ;--------------------------------------
                        ;       WAIT FOR DATA REQUEST          :
                        ;--------------------------------------
061A                    WAIT_DRQ        PROC    NEAR
061A B9 0100                    MOV     CX,DELAY_3
061D BA 01F7                    MOV     DX,HF_PORT+7
0620 EC                 WQ_1:   IN      AL,DX                   ; GET STATUS
0621 A8 08                      TEST    AL,ST_DRQ               ; WAIT FOR DRQ
0623 75 09                      JNZ     WQ_OK
0625 E2 F9                      LOOP    WQ_1                    ; KEEP TRYING FOR A SHORT WHILE
0627 C6 06 0074 R 80            MOV     @DISK_STATUS1,TIME_OUT  ; ERROR
062C F9                         STC
062D C3                         RET
062E F8                 WQ_OK:  CLC
062F C3                         RET
0630                    WAIT_DRQ        ENDP
                        ;--------------------------------------
                        ;       CHECK FIXED DISK STATUS        :
                        ;--------------------------------------
0630                    CHECK_STATUS    PROC    NEAR
0630 E8 0642 R                  CALL    CHECK_ST                ; CHECK THE STATUS BYTE
0633 75 07                      JNZ     CHECK_S1                ; AN ERROR WAS FOUND
0635 A8 01                      TEST    AL,ST_ERROR             ; WERE THERE ANY OTHER ERRORS
0637 74 03                      JZ      CHECK_S1                ; NO ERROR REPORTED
0639 E8 0676 R                  CALL    CHECK_ER                ; ERROR REPORTED
063C                    CHECK_S1:
063C 80 3E 0074 R 00            CMP     @DISK_STATUS1,0         ; SET STATUS FOR CALLER
0641 C3                         RET
0642                    CHECK_STATUS    ENDP
                        ;--------------------------------------
                        ;       CHECK FIXED DISK STATUS BYTE   :
                        ;--------------------------------------
0642                    CHECK_ST        PROC    NEAR
0642 BA 01F7                    MOV     DX,HF_PORT+7            ; GET THE STATUS
0645 EC                         IN      AL,DX
0646 A2 008C R                  MOV     @HF_STATUS,AL
0649 B4 00                      MOV     AH,0
064B A8 80                      TEST    AL,ST_BUSY              ; IF STILL BUSY
064D 75 1A                      JNZ     CKST_EXIT               ;  REPORT OK
064F B4 CC                      MOV     AH,WRITE_FAULT
0651 A8 20                      TEST    AL,ST_WRT_FLT           ; CHECK FOR WRITE FAULT
0653 75 14                      JNZ     CKST_EXIT
0655 B4 AA                      MOV     AH,NOT_RDY
0657 A8 40                      TEST    AL,ST_READY             ; CHECK FOR NOT READY
0659 74 0E                      JZ      CKST_EXIT
065B B4 40                      MOV     AH,BAD_SEEK
065D A8 10                      TEST    AL,ST_SEEK_COMPL        ; CHECK FOR SEEK NOT COMPLETE
065F 74 08                      JZ      CKST_EXIT
0661 B4 11                      MOV     AH,DATA_CORRECTED
0663 A8 04                      TEST    AL,ST_CORRCTD           ; CHECK FOR CORRECTED ECC
0665 75 02                      JNZ     CKST_EXIT
0667 B4 00                      MOV     AH,0
0669                    CKST_EXIT:
0669 88 26 0074 R               MOV     @DISK_STATUS1,AH        ; SET ERROR FLAG
066D 80 FC 11                   CMP     AH,DATA_CORRECTED       ; KEEP GOING WITH DATA CORRECTED
0670 74 03                      JZ      CKST_EXT
0672 80 FC 00                   CMP     AH,0
0675                    CKST_EXI:
0675 C3                         RET
0676                    CHECK_ST        ENDP
```

```
                        PAGE
                        ;--------------------------------------
                        ;       CHECK FIXED DISK ERROR REGISTER :
                        ;--------------------------------------
0676                    CHECK_ER        PROC    NEAR
0676 BA 01F1                    MOV     DX,HF_PORT+1            ; GET THE ERROR REGISTER
0679 EC                         IN      AL,DX
067A A2 008D R                  MOV     @HF_ERROR,AL
067D 53                         PUSH    BX
067E B9 0008                    MOV     CX,8                    ; TEST ALL 8 BITS
0681 D0 E0              CK1:    SHL     AL,1                    ; MOVE NEXT ERROR BIT TO CARRY
0683 72 02                      JC      CK2                     ; FOUND THE ERROR
0685 E2 FA                      LOOP    CK1                     ; KEEP TRYING
0687 BB 0698 R          CK2:    MOV     BX,OFFSET ERR_TBL       ; COMPUTE ADDRESS OF
068A 03 D9                      ADD     BX,CX                   ; ERROR CODE'
068C 2E: 8A 27                  MOV     AH,BYTE PTR CS:[BX]     ; GET ERROR CODE
068F 88 26 0074 R       CKEX:   MOV     @DISK_STATUS1,AH        ; SAVE ERROR CODE
0693 5B                         POP     BX
0694 80 FC 00                   CMP     AH,0
0697 C3                         RET
0698 E0                 ERR_TBL DB      NO_ERR
0699 02 40 01 BB                DB      BAD_ADDR_MARK,BAD_SEEK,BAD_CMD,UNDEF_ERR
069D 04 BB 10 0A                DB      RECORD_NOT_FND,UNDEF_ERR,BAD_ECC,BAD_SECTOR
06A1                    CHECK_ER        ENDP

                        ;--------------------------------------------------------
                        ; CHECK_DMA                                             :
                        ;  -CHECK ES:BX AND # SECTORS TO MAKE SURE THAT IT WILL :
                        ;    FIT WITHOUT SEGMENT OVERFLOW.                      :
                        ;  -ES:BX HAS BEEN REVISED TO THE FORMAT SSSS:000X      :
                        ;  -OK IF # SECTORS < 80H (7FH IF LONG READ OR WRITE)   :
                        ;  -OK IF # SECTORS = 80H (7FH) AND BX <= 00H (04H)     :
                        ;  -ERROR OTHERWISE                                     :
                        ;--------------------------------------------------------
06A1                    CHECK_DMA       PROC    NEAR
06A1 50                         PUSH    AX                      ; SAVE REGISTERS
06A2 B8 8000                    MOV     AX,8000H                ; AH = MAX # SECTORS   AL = MAX OFFSET
06A5 F6 46 FE 02                TEST    @CMD_BLOCK+6,ECC_MODE
06A9 74 03                      JZ      CKD1
06AB B8 7F04                    MOV     AX,7F04H                ; ECC IS 4 MORE BYTES
06AE 3A 66 F9           CKD1:   CMP     AH,@CMD_BLOCK+1         ; NUMBER OF SECTORS
06B1 77 06                      JA      CKDOK                   ; IT WILL FIT
06B3 72 07                      JB      CKDERR                  ; TOO MANY
06B5 3A C3                      CMP     AL,BL                   ; CHECK OFFSET ON MAX SECTORS
06B7 72 03                      JB      CKDERR                  ; ERROR
06B9 F8                 CKDOK:  CLC                             ; CLEAR CARRY
06BA 58                         POP     AX
06BB C3                         RET                             ; NORMAL RETURN
06BC F9                 CKDERR: STC                             ; INDICATE ERROR
06BD C6 06 0074 R 09            MOV     @DISK_STATUS1,DMA_BOUNDARY
06C2 58                         POP     AX
06C3 C3                         RET
06C4                    CHECK_DMA       ENDP

                        ;--------------------------------------
                        ;       SET UP  ES:BX-> DISK PARMS      :
                        ;--------------------------------------
06C4                    GET_VEC PROC    NEAR
06C4 2B C0                      SUB     AX,AX                   ; GET DISK PARAMETER ADDRESS
06C6 8E C0                      MOV     ES,AX
                                ASSUME  ES:ABS0
06C8 F6 C2 01                   TEST    DL,1
06CB 74 07                      JZ      GV_0
06CD 26: C4 1E 0118 R           LES     BX,@HF1_TBL_VEC         ; ES:BX -> DRIVE PARAMETERS
06D2 EB 05                      JMP     SHORT GV_EXIT
06D4                    GV_0:
06D4 26: C4 1E 0104 R           LES     BX,@HF_TBL_VEC          ; ES:BX -> DRIVE PARAMETERS
06D9                    GV_EXIT:
06D9 C3                         RET
06DA                    GET_VEC ENDP

                        ;--- HARDWARE INT  76H -- ( IRQ LEVEL  14 ) ---------------------------------
                        ;                                                                            :
                        ;       FIXED DISK INTERRUPT ROUTINE                                         :
                        ;                                                                            :
                        ;--------------------------------------------------------------------------------

06DA                    HD_INT  PROC    NEAR
06DA 50                         PUSH    AX
06DB 1E                         PUSH    DS
06DC E8 0000 E                  CALL    DDS
06DF C6 06 008E R FF            MOV     @HF_INT_FLAG,0FFH       ; ALL DONE
06E4 B0 20                      MOV     AL,EOI                  ; NON-SPECIFIC END OF INTERRUPT
06E6 E6 A0                      OUT     INTB00,AL               ; FOR CONTROLLER #2
06E8 EB 00                      JMP     $+2                     ; WAIT
06EA E6 20                      OUT     INTA00,AL               ; FOR CONTROLLER #1
06EC 1F                         POP     DS
06ED FB                         STI                             ; RE-ENABLE INTERRUPTS
06EE B8 9100                    MOV     AX,9100H                ; DEVICE POST
06F1 CD 15                      INT     15H                     ;  INTERRUPT
06F3 58                         POP     AX
06F4 CF                         IRET                            ; RETURN FROM INTERRUPT
06F5                    HD_INT  ENDP

06F5 31 31 2F 31 35 2F          DB      '11/15/85'              ; RELEASE MARKER
     38 35
06FD                    CODE    ENDS
                                END
```