

```

0002 ; FILE: [MASTER TRIPLE DENSITY]CBIOSX.Z80
0003 ; ASMB CBIOSX.AAA HEX OPCODE DATE=082684 TIME=133000 XREF WIDTH=132
0004 ; 5/16/84: ADDED TALLY PRINTER DRIVER TO PUN:
0005 ; 5/26/84: CHANGED TALLY PRINTER TO I/O PORT B
0006 ; 8/26/84: ADDED IOBYTE TEST IN LST: DRIVER TO JP TO PUN:
0007 ; 11/12/85: CHANGED FROM 60K TO 59K SYSGEM TO ACCOMODATE RAMDISK
0008 ; 11/30/88: ADDED XON/XOFF PROTOCOL FOR DIABLO IO-4 PORT-A
0009 ; 03/22/96: ADDED DUPLICATE STATUS REQUEST SERIO I/O (PORT B, TALYST:)
0010 ;
0011 ; THIS LISTING FOR THE TRIPLE DENSE BIOS FOR CP/M. IT SUPPORTS 3 DISKETTE
0012 ; FORMATS: SINGLE DENSITY, DOUBLE DENSITY (26 256 BYTE SECTORS), AND MAXIMUM
0013 ; DENSITY (9 1024 BYTE SECTORS)
0014 ;
(FFFF) 0015 TRUE EQU 0FFFFH ; SET UP VALUE OF LOGICAL TRUE
(0000) 0016 FALSE EQU NOT TRUE ; SET UP VALUE OF LOGICAL FALSE
0017 ;
(003B) 0018 MSIZE EQU 59 ; SYSTEM MEMORY SIZE
0019 ;
(FFFF) 0020 MOVCPM EQU TRUE ; DIGITAL RESEARCH'S STANDARD MOVCPM WAS USED
0021 ; TO SET UP BDOS AND CCP IF TRUE
(0000) 0022 MCPMDD EQU FALSE ; MOVCPMDD WAS USED TO SET UP BDOS AND CCP IF
0023 ; TRUE
(0000) 0024 MVCPMO EQU FALSE ; THE OLD MOVCPM WHICH SET UP BDOS AND CCP FOR
0025 ; THE ORIGINAL TRIPLE DENSE BIOS WAS USED IF
0026 ; TRUE
(FFFF) 0027 IF MOVCPM
(9C00) 0028 CBASE EQU (MSIZE-20)*1024
0029 ENDIF
(0000) 0030 IF MCPMDD
0031 CBASE EQU ((MSIZE-20)*1024)-0100H
0032 ENDIF
(0000) 0033 IF MVCPMO
0034 CBASE EQU ((MSIZE-20)*1024)+0100H
0035 ENDIF
(D000) 0036 CPMB EQU CBASE+03400H ; START OF CP/M (CCP)
(D806) 0037 BDOS EQU CPMB+0806H ; BDOS CALL ENTRY POINT IN CP/M
(E600) 0038 CBIOS EQU CPMB+01600H ; START OF BIOS IN CP/M
0039 ;
(E600) 0040 ORG CBIOS ; CP/M BIOS BASE ADDRESS
0041 ;
(000D) 0042 CR EQU 0DH ; EQUATE FOR CARRIAGE RETURN
(000A) 0043 LF EQU 0AH ; EQUATE FOR LINE FEED
(000C) 0044 FF EQU 0CH ; EQUATE FOR FORM FEED
0045 ;
(0060) 0046 X EQU 060H ; DISK CONTROLLER PORTS BASE ADDRESS
(0063) 0047 SELCTP EQU X+03H ; DRIVE SELECT PORT
(0064) 0048 STATSP EQU X+04H ; CONTROLLER STATUS PORT
(0065) 0049 TRACKP EQU X+05H ; TRACK PORT
(0066) 0050 SECTRP EQU X+06H ; SECTOR PORT
(0067) 0051 DATAP EQU X+07H ; DATA PORT
(0064) 0052 CMDP EQU X+04H ; CONTROLLER COMMAND PORT
0053 ;
(000A) 0054 RWRTRY EQU 0AH ; READ/WRITE RETRY COUNT FOR ERROR RECOVERY
(0005) 0055 SRTRY EQU 05H ; SEEK RETRY COUNT FOR ERROR RECOVERY
(0A05) 0056 RETRYC EQU RWRTRY*256+SRTRY ; READ/WRITE OPERATION RECOVERY COUNTS
0057 ;
(0000) 0058 STEPR EQU 00H ; HEAD STEP RATE BITS FOR RESTORE, SEEK %
0059 ; STEP IN, AND STEP OUT COMMANDS (3 MSEC) %
0060 ;STEPR EQU 01H ; HEAD STEP RATE BITS FOR RESTORE, SEEK ^
0061 ; STEP IN, AND STEP OUT COMMANDS (6 MSEC) ^
(0008) 0062 RESTOR EQU 08H+STEPR ; CONTROLLER COMMAND TO RESTORE (HOME) HEAD %
(001C) 0063 SKHDLV EQU 01CH+STEPR ; CONTROLLER COMMAND TO SEEK SPECIFIED TRACK
(005C) 0064 STEPI EQU 05CH+STEPR ; CONTROLLER COMMAND TO STEP HEAD IN 1 TRACK
(007C) 0065 STEPO EQU 07CH+STEPR ; CONTROLLER COMMAND TO STEP HEAD OUT 1 TRACK
(00D0) 0066 FORINT EQU 0D0H ; CONTROLLER COMMAND TO FORCE INTERRUPT
(0088) 0067 RDSEC EQU 088H ; CONTROLLER COMMAND TO READ SECTOR
(00A8) 0068 WTSEC EQU 0A8H ; CONTROLLER COMMAND TO WRITE SECTOR
(00C0) 0069 RDAD EQU 0C0H ; CONTROLLER COMMAND TO READ SECTOR ADDRESS
(00FF) 0070 DESEL EQU 0FFH ; SELECT COMMAND TO DESELECT ALL DRIVES
0071 ;
(0091) 0072 STERM EQU 091H ; SEEK/STEP STATUS MASK TO CHECK NOT READY,
0073 ; SEEK ERROR, AND BUSY CONTROLLER STATUS BITS
(00FE) 0074 RDEM EQU 0FEH ; READ SECTOR STATUS MASK TO CHECK NOT READY,
0075 ; RECORD TYPE, RECORD NOT FOUND,CRC ERROR,
0076 ; LOST DATA, AND DRQ
    
```

```

(00FC) 0077 WTEM EQU 0FCH ; WRITE SECTOR STATUS MASK TO CHECK NOT READY,
0078 ; WRITE PROTECT,WRITE FAULT,RECORD NOT FOUND,
0079 ; CRC ERROR, LOST DATA, AND DRQ
(00F8) 0080 RAEM EQU 0F8H ; READ ADDRESS STATUS MASK TO CHECK NOT READY,
0081 ; RECORD NOT FOUND, AND CRC ERROR (IGNORE LOST
0082 ; DATA AND DRQ)
(003F) 0083 WCLDBM EQU 03FH ; MASK TO CHECK FOR WRITE FAULT, CRC ERROR,
0084 ; LOST DATA, DRQ, AND BUSY
(007F) 0085 NRDYM EQU 07FH ; MASK TO CHECK ALL STATUS BITS EXCEPT NOT
0086 ; READY
(FF80) 0087 DNRDYM EQU NOT(NRDYM) ; MASK TO CHECK DRIVE NOT READY
(0010) 0088 SEEKEM EQU 010H ; MASK TO CHECK SEEK ERROR STATUS BIT
(0020) 0089 HDNTLM EQU 020H ; MASK TO CHECK HEAD NOT LOADED STATUS BIT
(0001) 0090 BUSYM EQU 01H ; MASK TO CHECK CONTROLLER BUSY STATUS BIT
0091 ;
(0020) 0092 STEPDM EQU 020H ; MASK TO CHANGE STEP DIRECTION IN STEP COMMAND
(0004) 0093 EBITM EQU 04H ; MASK TO SET E BIT IN READ AND WRITE COMMANDS
0094 ; TO DELAY 15 MSEC IF HEAD NOT LOADED
(0010) 0095 SIDEM EQU 010H ; MASK TO CHANGE SIDE IN DRIVE SELECT COMMAND
(007F) 0096 EWAITM EQU 07FH ; MASK TO ENABLE WAIT IN DRIVE SELECT COMMAND
(FF80) 0097 DWAITM EQU NOT(EWAITM) ; MASK TO DISABLE WAIT IN DRIVE SELECT COMMAND
(000F) 0098 DRIVM EQU 0FH ; MASK TO OBTAIN DRIVE BITS FROM SELECT STATUS
(0003) 0099 DRIVNM EQU 03H ; MASK TO ISOLATE DRIVE NUMBER
0100 ;
(0006) 0101 DENSB EQU 06H ; BIT NUMBER OF DENSITY SELECT IN DRIVE SELECT
0102 ; COMMAND
(0001) 0103 F1B EQU 01H ; BIT NUMBER OF F1 BIT IN CONTROLLER COMMAND
0104 ; TO CHECK THAT THE SELECTED SIDE FLAG (F2)
0105 ; AND THE SIDE (HEAD) ADDRESS BYTE AGREE
(0004) 0106 SIDEB EQU 4 ; BIT NUMBER OF SIDE SELECT BIT
0107 ;
(0002) 0108 NDRIVS EQU 2 ; EQUATE FOR NUMBER OF DRIVES SUPPORTED
0109 ;
(0100) 0110 BSTACK EQU 0100H ; TOP OF STACK AREA USED BY BOOT
0111 ;
(0003) 0112 MONOC EQU 003H ; COMMAND TO SWITCH SD MONITOR PROM OUT AND %
0113 ; SHADOW RAM IN, BUT DON'T STROBE PRINTER %
(007F) 0114 MONOP EQU 07FH ; PORT ADDRESS TO SWITCH SD MONITOR ON AND OFF%
0115 ;
(007E) 0116 PDATA EQU 07EH ; PARALLEL DATA PORT ADDRESS %
(0001) 0117 PMASK EQU 001H ; PARALLEL PORT BUSY MASK %
0118 ;
(007D) 0119 SERST EQU 07DH ; SERIAL I/O STATUS PORT ADDRESS %
(007C) 0120 SERIN EQU 07CH ; SERIAL INPUT PORT ADDRESS %
(007C) 0121 SEROUT EQU 07CH ; SERIAL OUTPUT PORT ADDRESS %
(0002) 0122 ICHRDY EQU 002H ; INPUT CHARACTER READY STATUS MASK %
(0001) 0123 OREADY EQU 001H ; READY FOR OUTPUT CHARACTER STATUS MASK %
(0004) 0124 SIOAS EQU 004H ; IO-4 SERIAL PORT A (STATUS)
(0005) 0125 SIOAD EQU 005H ; IO-4 SERIAL PORT A (DATA)
(0040) 0126 SIOAIR EQU 040H ; IO-4 SERIAL PORT A (INPUT STATUS MASK)
(0080) 0127 SIOAOR EQU 080H ; IO-4 SERIAL PORT A (OUTPUT STATUS MASK)
(0006) 0128 SIOBS EQU 006H ; IO-4 SERIAL PORT B (STATUS) %
(0007) 0129 SIOBD EQU 007H ; IO-4 SERIAL PORT B (DATA) %
(0040) 0130 SIOBIR EQU 040H ; IO-4 SERIAL PORT B (INPUT STATUS MASK) %
(0080) 0131 SIOBOR EQU 080H ; IO-4 SERIAL PORT B (OUTPUT STATUS MASK) %
0132 ;
0133 ; THE FOLLOWING ARE EQUATES FOR PAGE 0 CELLS USED BY BIOS
0134 ;
(0003) 0135 IOB EQU 003H ; PAGE 0 I/O BYTE %
(0040) 0136 PODMA EQU 040H ; PAGE 0 DMA ADDRESS STORAGE
(0042) 0137 PODRV EQU 042H ; PAGE 0 ACTIVE (SELECTED) DRIVE STORAGE
(0043) 0138 POTRK EQU 043H ; PAGE 0 CURRENT TRACK STORAGE
(0045) 0139 POSEC EQU 045H ; PAGE 0 CURRENT SECTOR STORAGE
(0047) 0140 POLSEL EQU 047H ; PAGE 0 LAST SELECT COMMAND STORAGE
(0048) 0141 POSAB EQU 048H ; PAGE 0 SECTOR ADDRESS BYTES BUFFER
(004B) 0142 POSLB EQU 04BH ; PAGE 0 SECTOR LENGTH ADDRESS BYTE
(004C) 0143 PODDFL EQU 04CH ; PAGE 0 DRIVE DENSITY FLAG BYTES
(0050) 0144 POSSDS EQU 050H ; PAGE 0 DRIVE SINGLE/DOUBLE SIDED FLAG BYTES
(0054) 0145 POCMD EQU 054H ; PAGE 0 LAST COMMAND SENT STORAGE
(0055) 0146 POCST EQU 055H ; PAGE 0 LAST CONTROLLER STATUS BYTE STORAGE
(0056) 0147 PORDEC EQU 056H ; PAGE 0 READ ERROR COUNTER
(0058) 0148 POWTEC EQU 058H ; PAGE 0 WRITE ERROR COUNTER
0149 ;
0150 ; THE FOLLOWING ARE EQUATES FOR OFFSETS OF THE ELEMENTS OF THE READ (READCL)
0151 ; AND WRITE (WRITCL) CONTROL TABLES
0152 ;
(0000) 0153 HSTDSK EQU 00H ; HOST DRIVE NUMBER
(0001) 0154 HSTTRK EQU 01H ; HOST TRACK NUMBER

```

```

(0003) 0155 HSTSEC EQU 03H ; HOST SECTOR NUMBER
(0004) 0156 HSTPSC EQU 04H ; DESIRED PHYSICAL SECTOR ON TRACK
(0005) 0157 HSTACT EQU 05H ; HOST ACTIVE FLAG
(0006) 0158 HSTWRT EQU 06H ; WRITE TO HOST PENDING FLAG
(0007) 0159 UNACNT EQU 07H ; UNALLOCATED RECORD COUNT IN BLOCK
(0008) 0160 UNADSK EQU 08H ; LAST DRIVE NUMBER OF BLOCK CONTAINING
0161 ; UNALLOCATED RECORD
(0009) 0162 UNATRK EQU 09H ; LAST TRACK NUMBER OF BLOCK CONTAINING
0163 ; UNALLOCATED RECORD
(000B) 0164 UNASEC EQU 0BH ; LAST UNALLOCATED SECTOR NUMBER
0165 ;
(0000) 0166 WBOOTV EQU 0000H ; ADDRESS OF PAGE 0 VECTOR TO WARM BOOT
(0005) 0167 CBDOS EQU 0005H ; ADDRESS OF PAGE 0 VECTOR TO CALL BDOS
0168 ;
E600 C333E8 0169 JP BOOT ; ARRIVE HERE FROM COLD START LOAD
E603 C34DE8 0170 WBOOTA: JP WBOOT ; ARRIVE HERE FOR WARM START
E606 C39CEE 0171 JP CONST ; CHECK FOR CONSOLE CHAR READY
E609 C386EC 0172 JP DCONI ; DESELECT DRIVE THEN READ CONSOLE CHARACTER IN
E60C C3B0EE 0173 JP CONOUT ; WRITE CONSOLE CHARACTER OUT
E60F C3BBEE 0174 JP LIST ; WRITE LISTING CHARACTER OUT
E612 C35FEF 0175 JP PUNCH ; WRITE CHARACTER TO PUNCH DEVICE
E615 C35CEF 0176 JP READER ; READ READER DEVICE
E618 C395E9 0177 JP HOME ; MOVE TO TRACK 00 ON SELECTED DISK
E61B C336E9 0178 JP SELDSK ; SELECT DISK DRIVE
E61E C398E9 0179 JP SETTRK ; SET TRACK NUMBER
E621 C39DE9 0180 JP SETSEC ; SET SECTOR NUMBER
E624 C3A2E9 0181 JP SETDMA ; SET DMA ADDRESS
E627 C3B3EC 0182 JP READ ; READ SELECTED SECTOR
E62A C31DED 0183 JP WRITE ; WRITE SELECTED SECTOR
E62D C3F0EE 0184 JP LISTST ; RETURN LIST STATUS
E630 C3A7E9 0185 JP SECTRN ; SECTOR TRANSLATE SUBROUTINE
0186 ;
0187 ; THE FOLLOWING SIGN ON MESSAGE IS DISPLAYED BY BIOS AFTER A COLD BOOT
0188 ;
E633 0C0D0A35 0189 SIGNON: DB FF,CR,LF,'59K CPM 2.2 ZCPR 11/30/88',CR,LF
E652 54414C4C 0190 DB 'TALLY LST: DIABLO UL1: ',CR,LF
0191 ;
0192 ; THE FOLLOWING ARE DISK PARAMETER HEADERS FOR THE 4 CORRESPONDING DISK DRIVES
0193 ; SUPPORTED BY THIS BIOS. THE FORMAT OF THE HEADERS IS AS FOLLOWS:
0194 ;
0195 ; DRIV: DW XLT,SCRATCH,SCRATCH,SCRATCH,DIRBUF,DPB,CSV,ALV
0196 ;
0197 ; XLT = ADDRESS OF LOGICAL TO PHYSICAL TRANSLATION VECTOR (SKEW TABLE)
0198 ; SCRATCH = 3 BDOS SCRATCH CELLS, 6 BYTES (INITIALIZATION NOT NEEDED)
0199 ; DIRBUF = ADDRESS OF 128 BYTE SCRATCH PAD FOR DIRECTORY OPERATIONS
0200 ; DPB = ADDRESS OF DISK PARAMETER BLOCK FOR DRIVE
0201 ; CSV = ADDRESS OF SCRATCH PAD FOR CHECK FOR CHANGED DISKS IN DRIVE.
0202 ; EACH DIRECTORY ENTRY OCCUPIES 32 BYTES. THUS THERE ARE 4
0203 ; DIRECTORY ENTRIES PER 128 BYTE CP/M DIRECTORY RECORD. SINCE
0204 ; CSV MUST RESERVE A BYTE FOR THE RECORD CHECKSUM OF EACH
0205 ; DIRECTORY RECORD, THE SPACE RESERVED IS THE MAXIMUM NUMBER OF
0206 ; ENTRIES/4 (DRM+1/4). MAXIMUM DENSITY WITH 256 ENTRIES IS THE
0207 ; WORST CASE SO 64 BYTES ARE RESERVED FOR EACH CSV AREA.
0208 ; ALV = ADDRESS OF BDOS SCRATCH PAD FOR STORAGE ALLOCATION INFORMATION
0209 ; FOR DRIVE. NOTE: THE SIZE OF THE ALV AREA MUST BE SUCH THAT
0210 ; THERE IS A BIT FOR EACH BLOCK ON THE DISK. IN DOUBLE SIDED
0211 ; MAXIMUM DENSITY (WORST CASE), THERE ARE (1024*9*154)/4096 =
0212 ; 346.5 BLOCKS. 346.5/8 BITS PER BYTE ROUNDED UP = 44 BYTES.
0213 ; THUS 44 BYTES ARE RESERVED FOR EACH ALV AREA IN THIS BIOS.
0214 ;
0215 ;
0216 ; DRIVE A
0217 ;
E66D E1E6 0218 DPARMH: DW DDKTb,0000H,0000H,0000H,DIRBUF,PSSDD,CSV0,ALV0
0000
0000
0000
E7F7
E8E7
93F8
67F8
0219 ;
0220 ; DRIVE B
0221 ;
E67D E1E6 0222 DW DDKTb,0000H,0000H,0000H,DIRBUF,PSSDD,CSV1,ALV1
0000
0000
0000

```

```

E7F7
E8E7
FFF8
D3F8

0223 ;
0224 ; DRIVE C
0225 ;
E68D E1E6 0226 DW DDSTKB,0000H,0000H,0000H,DIRBUF,PSSDD,CSV2,ALV2
0000
0000
0000
E7F7
E8E7
6BF9
3FF9

0227 ;
0228 ; DRIVE D
0229 ;
E69D E1E6 0230 DW DDSTKB,0000H,0000H,0000H,DIRBUF,PSSDD,CSV3,ALV3
0000
0000
0000
E7F7
E8E7
D7F9
ABF9

0231 ;
0232 ; THE FOLLOWING IS THE SINGLE DENSITY SECTOR SKEW TABLE USED BY BIOS
0233 ;
E6AD 01070D13 0234 SDSKTB: DB 001H,007H,00DH,013H,019H,005H,00BH,011H,017H,003H,009H,00FH
E6B9 1502080E 0235 DB 015H,002H,008H,00EH,014H,01AH,006H,00CH,012H,018H,004H,00AH
E6C5 1016 0236 DB 010H,016H
E6C7 1B21272D 0237 DB 01BH,021H,027H,02DH,033H,01FH,025H,02BH,031H,01DH,023H,029H
E6D3 2F1C2228 0238 DB 02FH,01CH,022H,028H,02EH,034H,020H,026H,02CH,032H,01EH,024H
E6DF 2A30 0239 DB 02AH,030H
0240 ;
0241 ; THE FOLLOWING IS THE DOUBLE DENSITY SECTOR SKEW TABLE USED BY BIOS
0242 ;
0243 ;DDSKTB: DB 001H,002H,013H,014H,025H,026H,003H,004H,015H,016H,027H,028H ; ^
0244 ; DB 005H,006H,017H,018H,029H,02AH,007H,008H,019H,01AH,02BH,02CH ; ^
0245 ; DB 009H,00AH,01BH,01CH,02DH,02EH,00BH,00CH,01DH,01EH,02FH,030H ; ^
0246 ; DB 00DH,00EH,01FH,020H,031H,032H,00FH,010H,021H,022H,033H,034H ; ^
0247 ; DB 011H,012H,023H,024H ; ^
0248 ; DB 035H,036H,047H,048H,059H,05AH,037H,038H,049H,04AH,05BH,05CH ; ^
0249 ; DB 039H,03AH,04BH,04CH,05DH,05EH,03BH,03CH,04DH,04EH,05FH,060H ; ^
0250 ; DB 03DH,03EH,04FH,050H,061H,062H,03FH,040H,051H,052H,063H,064H ; ^
0251 ; DB 041H,042H,053H,054H,065H,066H,043H,044H,055H,056H,067H,068H ; ^
0252 ; DB 045H,046H,057H,058H ; ^
0253 ;
E6E1 01020D0E 0254 DDSKTB: DB 001H,002H,00DH,00EH,019H,01AH,025H,026H,031H,032H,009H,00AH ; %
E6ED 15162122 0255 DB 015H,016H,021H,022H,02DH,02EH,005H,006H,011H,012H,01DH,01EH ; %
E6F9 292A0304 0256 DB 029H,02AH,003H,004H,00FH,010H,01BH,01CH,027H,028H,033H,034H ; %
E705 0B0C1718 0257 DB 00BH,00CH,017H,018H,023H,024H,02FH,030H,007H,008H,013H,014H ; %
E711 1F202B2C 0258 DB 01FH,020H,02BH,02CH ; %
E715 35364142 0259 DB 035H,036H,041H,042H,04DH,04EH,059H,05AH,065H,066H,03DH,03EH ; %
E721 494A5556 0260 DB 049H,04AH,055H,056H,061H,062H,039H,03AH,045H,046H,051H,052H ; %
E72D 5D5E3738 0261 DB 05DH,05EH,037H,038H,043H,044H,04FH,050H,05BH,05CH,067H,068H ; %
E739 3F404B4C 0262 DB 03FH,040H,04BH,04CH,057H,058H,063H,064H,03BH,03CH,047H,048H ; %
E745 53545F60 0263 DB 053H,054H,05FH,060H ; %
0264 ;
0265 ; THE FOLLOWING IS THE MAXIMUM DENSITY SECTOR SKEW TABLE USED BY BIOS
0266 ;
0267 ;MDSKTB: DB 001H,002H,003H,004H,005H,006H,007H,008H,021H,022H,023H,024H ; ^
0268 ; DB 025H,026H,027H,028H,041H,042H,043H,044H,045H,046H,047H,048H ; ^
0269 ; DB 019H,01AH,01BH,01CH,01DH,01EH,01FH,020H,039H,03AH,03BH,03CH ; ^
0270 ; DB 03DH,03EH,03FH,040H,011H,012H,013H,014H,015H,016H,017H,018H ; ^
0271 ; DB 031H,032H,033H,034H,035H,036H,037H,038H,009H,00AH,00BH,00CH ; ^
0272 ; DB 00DH,00EH,00FH,010H,029H,02AH,02BH,02CH,02DH,02EH,02FH,030H ; ^
0273 ; DB 049H,04AH,04BH,04CH,04DH,04EH,04FH,050H,069H,06AH,06BH,06CH ; ^
0274 ; DB 06DH,06EH,06FH,070H,089H,08AH,08BH,08CH,08DH,08EH,08FH,090H ; ^
0275 ; DB 061H,062H,063H,064H,065H,066H,067H,068H,081H,082H,083H,084H ; ^
0276 ; DB 085H,086H,087H,088H,059H,05AH,05BH,05CH,05DH,05EH,05FH,060H ; ^
0277 ; DB 079H,07AH,07BH,07CH,07DH,07EH,07FH,080H,051H,052H,053H,054H ; ^
0278 ; DB 055H,056H,057H,058H,071H,072H,073H,074H,075H,076H,077H,078H ; ^
0279 ;
E749 01020304 0280 MDSKTB: DB 001H,002H,003H,004H,005H,006H,007H,008H,019H,01AH,01BH,01CH ; %
E755 1D1E1F20 0281 DB 01DH,01EH,01FH,020H,031H,032H,033H,034H,035H,036H,037H,038H ; %
E761 090A0B0C 0282 DB 009H,00AH,00BH,00CH,00DH,00EH,00FH,010H,021H,022H,023H,024H ; %

```

```

E76D 25262728 0283 DB 025H,026H,027H,028H,039H,03AH,03BH,03CH,03DH,03EH,03FH,040H ; %
E779 11121314 0284 DB 011H,012H,013H,014H,015H,016H,017H,018H,029H,02AH,02BH,02CH ; %
E785 2D2E2F30 0285 DB 02DH,02EH,02FH,030H,041H,042H,043H,044H,045H,046H,047H,048H ; %
E791 494A4B4C 0286 DB 049H,04AH,04BH,04CH,04DH,04EH,04FH,050H,061H,062H,063H,064H ; %
E79D 65666768 0287 DB 065H,066H,067H,068H,079H,07AH,07BH,07CH,07DH,07EH,07FH,080H ; %
E7A9 51525354 0288 DB 051H,052H,053H,054H,055H,056H,057H,058H,069H,06AH,06BH,06CH ; %
E7B5 6D6E6F70 0289 DB 06DH,06EH,06FH,070H,081H,082H,083H,084H,085H,086H,087H,088H ; %
E7C1 595A5B5C 0290 DB 059H,05AH,05BH,05CH,05DH,05EH,05FH,060H,071H,072H,073H,074H ; %
E7CD 75767778 0291 DB 075H,076H,077H,078H,089H,08AH,08BH,08CH,08DH,08EH,08FH,090H ; %

```

```

0292 ;
0293 ; THE FOLLOWING ARE DISK PARAMETER BLOCKS FOR THE DISK FORMATS SUPPORTED BY
0294 ; THIS BIOS. THE FORMAT OF EACH BLOCK IS AS FOLLOWS:

```

```

0295 ;
0296 ; DKLB: DW SPT
0297 ; DB BSH,BLM,EXM
0298 ; DW DSM,DRM
0299 ; DB AL0,AL1
0300 ; DW CKS,OFF
0301 ;
0302 ; SPT = NUMBER OF SECTORS PER TRACK
0303 ; BSH = DATA ALLOCATION BLOCK SHIFT FACTOR I. E. IS IS THE NUMBER OF
0304 ; TIMES 128 IS SHIFTED LEFT TO COMPUTE THE BLOCK SIZE IN BYTES
0305 ; BLM = BLOCK MULTIPLE I. E. NUMBER OF 128 BYTE SECTORS PER BLOCK-1
0306 ; EXM = THE EXTENT MASK SPECIFIES THE NUMBER OF LOGICAL 16K BYTE FILE
0307 ; EXTENTS IN A PHYSICAL FILE EXTENT. THE BITS IN THE MASK ARE
0308 ; SET FROM RIGHT TO LEFT. EACH BIT SET MULTIPLIES THE BASIC
0309 ; NUMBER OF LOGICAL EXTENTS (1) BY 2. THE NUMBER OF LOGICAL
0310 ; EXTENTS IN A PHYSICAL EXTENT IS (1 SHL(NUMBER OF BITS SET IN
0311 ; EXM)). HOWEVER IF THERE ARE MORE THAN 256 BLOCKS ON THE DISK
0312 ; (DSM>255), THE EXTENT MASK MUST BE SHIFTED RIGHT BY 1. WHEN
0313 ; DSM>255, EACH BLOCK NUMBER IN A DIRECTORY ENTRY REQUIRES 2
0314 ; BYTES INSTEAD OF 1. THUS EACH DIRECTORY ENTRY (PHYSICAL
0315 ; EXTENT) CAN CONTAIN ONLY 8 BLOCKS INSTEAD OF 16 WHEN DSM<255.
0316 ; DSM = MAXIMUM BLOCK NUMBER FOR THIS DISK FORMAT, NOT COUNTING THE
0317 ; RESERVED SYSTEM TRACKS.
0318 ; NOTE: (DSM+1)*(BLOCK SIZE)=DISK STORAGE CAPACITY IN BYTES
0319 ; DRM = NUMBER OF DIRECTORY ENTRIES WHICH CAN BE STORED ON THIS DISK-1
0320 ; AL0,AL1 = EACH BIT SET IN AL0,AL1 REPRESENTS A BLOCK RESERVED FOR
0321 ; DIRECTORY ENTRIES. ALLOCATION STARTS AT THE HIGH ORDER BIT
0322 ; OF AL0. EACH DIRECTORY ENTRY OCCUPIES 32 BYTES SO IF THE
0323 ; BLOCK SIZE IS 1024, THERE ARE 32 DIRECTORY ENTRIES PER BLOCK.
0324 ; THUS IF DRM=127 (128 ENTRIES) IN THIS CASE, 4 BLOCKS WOULD BE
0325 ; REQUIRED TO HOLD THE DIRECTORY. SO AL0=0F0H, AND AL1 = 0.
0326 ; CKS = SIZE OF THE DIRECTORY CHECK AREA I. E. SIZE OF AREA ADDRESSED
0327 ; BY THE CHECK VECTOR (CSV) IN THE PARAMETER HEADER. THE SIZE
0328 ; IS COMPUTED BY CKS = (DRM+1)/4. IT IS NECESSARY ONLY IF THE
0329 ; MEDIA IS REMOVABLE
0330 ; OFF = RESERVED TRACKS AT BEGINNING OF (LOGICAL) DISK
0331 ;
0332 ;

```

```

0333 ; THE DISK PARAMETER BLOCKS CAN ALSO BE GENERATED BY THE DISKDEF MACRO. THE
0334 ; FORMAT OF THE DISKDEF STATEMENT IS AS FOLLOWS:

```

```

0335 ;
0336 ; DISKDEF DN,FSC,LSC,[SKF],BLS,DKS,DIR,CKS,OFS,[0]
0337 ;
0338 ;
0339 ; DN = LOGICAL DISK NUMBER, 0 TO N-1
0340 ; FSC = FIRST PHYSICAL SECTOR NUMBER ON A TRACK
0341 ; LSC = LAST PHYSICAL SECTOR NUMBER ON A TRACK
0342 ; SKF = OPTIONAL SECTOR SKEW FACTOR USED TO CREATE SKEW TABLE IF
0343 ; WANTED
0344 ; BLS = DATA ALLOCATION BLOCK SIZE IN BYTES
0345 ; DKS = DISK SIZE IN BLS UNITS
0346 ; DIR = TOTAL NUMBER OF DIRECTORY ENTRIES DESIRED
0347 ; CKS = NUMBER OF "CHECKED" DIRECTORY ENTRIES ON A DIRECTORY SCAN.
0348 ; USED TO DETECT CHANGED DISKS.
0349 ; OFS = NUMBER OF TRACKS TO SKIP WHEN DRIVE IS ADDRESSED I. E. NUMBER
0350 ; OF RESERVED SYSTEM TRACKS
0351 ; [0] = OPTIONAL CP/M 1.4 COMPATIBILITY FLAG
0352 ;
0353 ;

```

```

0354 ; THE FOLLOWING PARAMETER BLOCK IS FOR A SINGLE SIDED, SINGLE DENSITY DRIVE
0355 ;

```

```

E7D9 1A00 0357 PSSSD: DISKDEF 0,1,26,,1024,243,64,64,2
E7DB 030700 0358 DB 3,7,0 ; SECTORS PER TRACK ; BLOCK SHIFT, SECTORS PER BLOCK-1, EXM
E7DE F200 0359 DW 242,63 ; DISK SIZE-1 IN BLOCKS, DIRECTORY ENTRIES-1
3F00

```

```

E7E2 C000      0360      DB      0C0H,000H      ; MASK TO RESERVE BLOCKS FOR DIRECTORY
E7E4 1000      0361      DW      16,2              ; CHECK SIZE, TRACKS RESERVED
0200
0362 ;
0363 ; THE FOLLOWING PARAMETER BLOCK IS FOR A SINGLE SIDED DOUBLE DENSITY DRIVE
0364 ;
0365 ;PSSDD:      DISKDEF 1,1,52,,2048,243,128,128,2
E7E8 3400      0366      PSSDD:      DW      52              ; SECTORS PER TRACK
E7EA 040F01    0367      DB      4,15,001H      ; BLOCK SHIFT, SECTORS PER BLOCK-1, EXM      %
0368 ;          DB      4,15,0              ; BLOCK SHIFT, SECTORS PER BLOCK-1, EXM      ^
E7ED F200      0369      DW      242,127      ; DISK SIZE-1 IN BLOCKS, DIRECTORY ENTRIES-1
7F00
E7F1 C000      0370      DB      0C0H,000H      ; MASK TO RESERVE BLOCKS FOR DIRECTORY
E7F3 2000      0371      DW      32,2              ; CHECK SIZE, TRACKS RESERVED
0200
0372 ;
0373 ; THE FOLLOWING PARAMETER BLOCK IS FOR A DOUBLE SIDED DOUBLE DENSITY DRIVE
0374 ; (26 256 BYTE SECTORS PER TRACK)
0375 ;
0376 ;PDSDD:      DISKDEF 2,1,104,,4096,243,256,256,2
E7F7 6800      0377      PDSDD:      DW      104             ; SECTORS PER TRACK
E7F9 051F03    0378      DB      5,31,003H      ; BLOCK SHIFT, SECTORS PER BLOCK-1, EXM
E7FC F200      0379      DW      242,255      ; DISK SIZE-1 IN BLOCKS, DIRECTORY ENTRIES-1
FF00
E800 C000      0380      DB      0C0H,000H      ; MASK TO RESERVE BLOCKS FOR DIRECTORY
E802 4000      0381      DW      64,2              ; CHECK SIZE, TRACKS RESERVED
0200
0382 ;
0383 ; THE FOLLOWING PARAMETER BLOCK IS FOR A DOUBLE SIDED SINGLE DENSITY DRIVE
0384 ; (26 256 BYTE SECTORS PER TRACK)
0385 ;
0386 ;PSSDD:      DISKDEF 3,1,52,,2048,243,128,128,2
E806 3400      0387      PSSDD:      DW      52              ; SECTORS PER TRACK
E808 040F01    0388      DB      4,15,001H      ; BLOCK SHIFT, SECTORS PER BLOCK-1, EXM      %
0389 ;          DB      4,15,0              ; BLOCK SHIFT, SECTORS PER BLOCK-1, EXM      ^
E80B F200      0390      DW      242,127      ; DISK SIZE-1 IN BLOCKS, DIRECTORY ENTRIES-1
7F00
E80F C000      0391      DB      0C0H,000H      ; MASK TO RESERVE BLOCKS FOR DIRECTORY
E811 2000      0392      DW      32,2              ; CHECK SIZE, TRACKS RESERVED      %
0200
0393 ;          DW      32,1              ; CHECK SIZE, TRACKS RESERVED      ^
0394 ;
0395 ; THE FOLLOWING PARAMETER BLOCK IS FOR A SINGLE SIDED MAXIMUM DENSITY DRIVE
0396 ; (9 1024 BYTE SECTORS PER TRACK)
0397 ;
0398 ;PSSMD:      DISKDEF 4,1,72,,4096,168,128,128,2
E815 4800      0399      PSSMD:      DW      72              ; SECTORS PER TRACK
E817 051F03    0400      DB      5,31,003H      ; BLOCK SHIFT, SECTORS PER BLOCK-1, EXM
E81A A700      0401      DW      167,127      ; DISK SIZE-1 IN BLOCKS, DIRECTORY ENTRIES-1
7F00
E81E 8000      0402      DB      080H,000H      ; MASK TO RESERVE BLOCKS FOR DIRECTORY
E820 2000      0403      DW      32,2              ; CHECK SIZE, TRACKS RESERVED
0200
0404 ;
0405 ; THE FOLLOWING PARAMETER BLOCK IS FOR A DOUBLE SIDED MAXIMUM DENSITY DRIVE
0406 ; (9 1024 BYTE SECTORS PER TRACK)
0407 ;
0408 ;PDSMD:      DISKDEF 5,1,144,,4096,335,256,256,2
E824 9000      0409      PDSMD:      DW      144             ; SECTORS PER TRACK
E826 051F01    0410      DB      5,31,001H      ; BLOCK SHIFT, SECTORS PER BLOCK-1, EXM
E829 4F01      0411      DW      335,255      ; DISK SIZE-1 IN BLOCKS, DIRECTORY ENTRIES-1
FF00
E82D C000      0412      DB      0C0H,000H      ; MASK TO RESERVE BLOCKS FOR DIRECTORY
E82F 4000      0413      DW      64,2              ; CHECK SIZE, TRACKS RESERVED
0200
0414 ;
0415 ; THIS IS THE COLD BOOT SUBROUTINE. IT LOADS BDOS AND CCP AND THEN GOES TO
0416 ; CCP
0417 ;
E833 2133E6    0418      BOOT: LD      HL,SIGNON      ; GET ADDRESS OF SIGN ON MESSAGE
E836 063A      0419      LD      B,03AH      ; GET NUMBER OF CHARACTERS IN MESSAGE (58)
E838 4E         0420      BNXTCH: LD     C,(HL)      ; GET NEXT CHARACTER TO OUTPUT TO CONSOLE
E839 CDB0EE      0421      CALL    CONOUT      ; DISPLAY CHARACTER ON CONSOLE
E83C 23         0422      INC     HL           ; INCREMENT TO NEXT CHARACTER
E83D 10F9      0423      DJNZ   BNXTCH      ; GO GET NEXT CHARACTER IF MORE IN MESSAGE
E83F 210000     0424      LD      HL,00000H      ; GET ADDRESS OF BLOCK TO ZERO
E842 220000     0425      LD      (00000H),HL    ; ZERO FIRST 2 WORDS OF BLOCK
E845 110100     0426      LD      DE,00001H      ; SET DESTINATION ADDRESS 1 BEYOND SOURCE

```

```

E848 017F00 0427 LD BC,0007FH ; SET UP TO ZERO 127 MORE BYTES
E84B EDB0 0428 LDIR ; INSURE BYTES 0 THROUGH 0FFH ARE ZERO
0429 ; I. E. ZERO PAGE 0
0430 ;
0431 ; THIS IS THE WARM BOOT ENTRY POINT. IT GETS CONTROL WHEN THE USER JUMPS TO
0432 ; 0000H OR WHEN THE OPERATOR ENTERS ^C AT THE CONSOLE TO A CP/M ROOT PROGRAM
0433 ;
E84D 310001 0434 WBOOT: LD SP,BSTACK ; SET STACK POINTER TO TOP OF BOOT STACK
E850 218080 0435 LD HL,08080H ; GET DOUBLE BYTE OF UNDEFINED STATUS FLAGS
E853 224C00 0436 LD (PODDFL),HL ; SET DRIVE DENSITY FLAGS
E856 224E00 0437 LD (PODDFL+2),HL ; TO UNDEFINED STATUS
E859 225000 0438 LD (POSSDS),HL ; SET SINGLE/DOUBLE SIDED FLAGS FOR DRIVES
E85C 225200 0439 LD (POSSDS+2),HL ; TO UNDEFINED STATUS
E85F 0E00 0440 LD C,000H ; SET DRIVE TO SELECT TO 0
E861 CD36E9 0441 CALL SELDSK ; SELECT DRIVE 0
E864 CD68EB 0442 CALL SGTDEN ; GET DENSITY FLAG FOR DRIVE 0 AND CHECK IF
0443 ; SINGLE DENSITY
E867 CAF2E8 0444 JP Z,WBGCPM ; GO STORE JUMPS IN PAGE 0 AND START CP/M
0445 ; IF SINGLE DENSITY
E86A FE40 0446 CP 040H ; CHECK IF MAXIMUM DENSITY
E86C 280C 0447 JR Z,WBMDL ; GO SET UP TO LOAD BDOS AND CCP FROM MAXIMUM
0448 ; DENSITY DISKETTE
E86E 0616 0449 LD B,016H ; GET NUMBER OF 256 BYTE SECTORS TO LOAD
E870 110002 0450 LD DE,00200H ; SET UP OFFSET TO LOAD NEXT SECTOR, LOADING
0451 ; ALTERNATE SECTORS
E873 0E15 0452 LD C,015H ; GET NUMBER OF LAST SECTOR ON 1ST PASS
0453 ; (ROTATION)
E875 2100D1 0454 LD HL,CPMB+0100H ; GET OFFSET TO LOAD SECOND SECTOR ON SECOND
0455 ; PASS (ROTATION)
E878 180A 0456 JR WBMODI ; GO STORE INFORMATION INTO INSTRUCTIONS
E87A 0605 0457 WBMDL: LD B,005H ; GET NUMBER OF 1024 BYTE SECTORS TO LOAD
E87C 110008 0458 LD DE,00800H ; GET OFFSET TO LOAD NEXT SECTOR, LOADING
0459 ; ALTERNATE SECTORS
E87F 0E05 0460 LD C,005H ; GET NUMBER OF LAST SECTOR ON 1ST PASS
0461 ; (ROTATION)
E881 2100D4 0462 LD HL,CPMB+0400H ; GET ADDRESS TO LOAD SECOND SECTOR ON SECOND
0463 ; PASS (ROTATION)
E884 22C4E8 0464 WBMODI: LD (WBLD2A+1),HL ; STORE ADDRESS TO LOAD SECOND SECTOR ON SECOND
0465 ; PASS IN LD HL, INSTRUCTION
E887 ED53B5E8 0466 LD (WBLDNO+1),DE ; PUT OFFSET TO LOAD NEXT SECTOR IN LD DE,
0467 ; INSTRUCTION
E88B 78 0468 LD A,B ; GET TOTAL NUMBER OF SECTORS TO LOAD
E88C 32AAE8 0469 LD (WBLNSC+1),A ; STORE IN LD B, INSTRUCTION
E88F 79 0470 LD A,C ; GET SECTOR NUMBER OF LAST SECTOR ON 1ST PASS
E890 32BFE8 0471 LD (WBCPL1+1),A ; STORE IN CP INSTRUCTION IN LOAD LOOP
E893 0E03 0472 LD C,003H ; SET BOOT RETRY COUNT TO 3
E895 C5 0473 PUSH BC ; SAVE NUMBER OF SECTORS TO LOAD AND BOOT RETRY
0474 ; COUNT ON THE STACK
E896 2100D0 0475 WBRTRY: LD HL,CPMB ; GET ADDRESS TO START LOADING BDOS AND CCP
E899 224000 0476 LD (PODMA),HL ; SET IT AS THE CURRENT DMA ADDRESS
E89C 3E01 0477 LD A,001H ; SET CURRENT
E89E 324500 0478 LD (POSEC),A ; SECTOR AND
E8A1 324300 0479 LD (POTRK),A ; TRACK TO 1
E8A4 AF 0480 XOR A ; SET DRIVE
E8A5 324200 0481 LD (PODRV),A ; TO A (0)
E8A8 C1 0482 POP BC ; GET NUMBER OF SECTORS TO LOAD AND RETRY COUNT
E8A9 0600 0483 WBLNSC: LD B,000H ; GET NUMBER OF SECTORS TO LOAD
E8AB C5 0484 WBLDLP: PUSH BC ; SAVE NUMBER OF SECTORS LEFT TO LOAD AND RETRY
0485 ; COUNT
E8AC CD36EC 0486 CALL RDBOOT ; READ NEXT BDOS/CCP SECTOR
E8AF 207B 0487 JR NZ,WBCRTY ; GO CHECK IF ANOTHER RETRY O K IF ERROR
E8B1 2A4000 0488 LD HL,(PODMA) ; GET CURRENT DMA ADDRESS
E8B4 110002 0489 WBLDNO: LD DE,00200H ; GET OFFSET TO LOAD NEXT SECTOR
E8B7 19 0490 ADD HL,DE ; COMPUTE DMA ADDRESS OF NEXT SECTOR
E8B8 224000 0491 LD (PODMA),HL ; SET NEW DMA ADDRESS
E8BB 3A4500 0492 LD A,(POSEC) ; GET CURRENT SECTOR NUMBER
E8BE FE15 0493 WBCPL1: CP 015H ; CHECK IF LAST SECTOR OF PASS 1
E8C0 3807 0494 JR C,WBGNSC ; GO SET UP TO LOAD NEXT SECTOR IF NOT
E8C2 AF 0495 XOR A ; ZERO NEXT SECTOR NUMBER TO LOAD SINCE
0496 ; FOLLOWING INC'S SET IT TO 2 (FIRST SECTOR OF
0497 ; THE SECOND PASS
E8C3 2100D1 0498 WBLD2A: LD HL,CPMB+0100H ; GET ADDRESS TO LOAD FIRST SECTOR (2) ON
0499 ; SECOND ROTATION
E8C6 224000 0500 LD (PODMA),HL ; MAKE IT THE CURRENT DMA ADDRESS
E8C9 3C 0501 WBGNSC: INC A ; BUMP TO NEXT SECTOR NUMBER, EVERY OTHER
E8CA 3C 0502 INC A ; SECTOR LOADED ON EACH OF 2 PASSES
E8CB 324500 0503 LD (POSEC),A ; SET SECTOR NUMBER TO LOAD
E8CE C1 0504 POP BC ; GET NUMBER OF SECTORS LEFT TO LOAD AND BOOT

```

```

0505 ; RETRY COUNT
E8CF 10DA 0506 DJNZ WBLDLP ; GO LOAD NEXT BDOS/CCP SECTOR IF MORE
E8D1 CD68EB 0507 CALL SGTDEN ; GET DENSITY FLAG OF THE CURRENTLY SELECTED
0508 ; DRIVE (A)
E8D4 FE40 0509 CP 040H ; CHECK IF MAXIMUM DENSITY
E8D6 2017 0510 JR NZ,WBINIU ; GO INIT PAGE 0 JUMPS IF NOT MAXIMUM DENSITY
E8D8 C5 0511 PUSH BC ; SAVE RETRY COUNT ON THE STACK
E8D9 E5 0512 PUSH HL ; PUT ADDRESS TO LOAD NEXT SECTOR ON THE STACK
E8DA 21E7EF 0513 LD HL,READBF ; GET ADDRESS OF BIOS SECTOR READ BUFFER
E8DD 224000 0514 LD (PODMA),HL ; MAKE IT CURRENT DMA ADDRESS
E8E0 CD36EC 0515 CALL RDBOOT ; READ NEXT SECTOR
E8E3 D1 0516 POP DE ; MAKE ADDRESS TO LOAD NEXT SECTOR MOVE
0517 ; DESTINATION ADDRESS
E8E4 2046 0518 JR NZ,WBCRTY ; GO CHECK IF ANOTHER RETRY OK IF ERROR
E8E6 C1 0519 POP BC ; GET RETRY COUNT OFF STACK SO STACK POSITION
0520 ; IS CORRECT
E8E7 010002 0521 LD BC,00200H ; SET UP TO MOVE 2 PAGES (1/2 OF 1024 BYTE
0522 ; SECTOR)
E8EA 21E7EF 0523 LD HL,READBF ; SET SOURCE ADDRESS TO READ BUFFER
E8ED EDB0 0524 LDIR ; MOVE LAST 2 PAGES OF BDOS TO EXECUTION
0525 ; LOCATION
E8EF CD9AEE 0526 WBINIUI: CALL INITUD ; CALL USER DEVICE INITIALIZER (CURRENTLY NOP)
E8F2 3EC3 0527 WBGCPM: LD A,0C3H ; GET JUMP INSTRUCTION OP CODE
E8F4 320000 0528 LD (WBOOTV),A ; STORE AT ADDRESS 0
E8F7 2103E6 0529 LD HL,WBOOTA ; GET ADDRESS OF WBOOT VECTOR
E8FA 220100 0530 LD (WBOOTV+1),HL ; STORE IN JUMP INSTRUCTION AT ADDRESS 1 AND 2
E8FD 320500 0531 LD (CBDOS),A ; STORE JUMP INSTRUCTION OP CODE AT ADDRESS 5
E900 2106D8 0532 LD HL,BDOS ; GET ADDRESS OF USER ENTRY POINT TO BDOS
E903 220600 0533 LD (CBDOS+1),HL ; STORE IN JUMP INSTRUCTION AT ADDRESS 6 AND 7
E906 018000 0534 LD BC,00080H ; GET DEFAULT BUFFER ADDRESS
E909 CDA2E9 0535 CALL SETDMA ; MAKE IT THE CURRENT DMA ADDRESS
E90C AF 0536 XOR A
E90D DD21C3EF 0537 LD IX,READCL ; GET ADDRESS OF READ CONTROL TABLE
E911 DD7707 0538 LD (IX+UNACNT),A ; ZERO UNALLOCATED RECORD COUNT IN BLOCK
E914 DD7705 0539 LD (IX+HSTACT),A ; ZERO ACTIVE HOST FLAG
E917 DD21D1EF 0540 LD IX,WRITCL ; GET ADDRESS OF WRITE CONTROL TABLE
E91B DD7707 0541 LD (IX+UNACNT),A ; ZERO UNALLOCATED RECORD COUNT IN BLOCK
E91E DD7705 0542 LD (IX+HSTACT),A ; ZERO ACTIVE HOST FLAG
E921 3D 0543 DEC A ; SET A TO FF
E922 DD7703 0544 LD (IX+HSTSEC),A ; SET SECTOR IN WRITCL TO FF
E925 3A0400 0545 LD A,(00004H) ; GET LAST LOGGED DISK NUMBER
E928 4F 0546 LD C,A ; SEND TO CCP IN C FOR LOG IN
E929 C300D0 0547 JP CPMB ; GO TO CCP
E92C C1 0548 WBCRTY: POP BC ; GET NUMBER OF SECTORS LEFT TO LOAD AND BOOT
0549 ; RETRY COUNT
E92D 0D 0550 DEC C ; CHECK IF ANOTHER RETRY OK
E92E CA35E9 R 0551 JP Z,WBHALT ; GO HALT IF RETRY COUNT EXHAUSTED
E931 C5 0552 PUSH BC ; SAVE NUMBER OF SECTORS LEFT TO LOAD AND BOOT
0553 ; RETRY COUNT ON THE STACK
E932 C396E8 0554 JP WBRTRY ; GO RETRY BOOT LOAD OF CCP AND BDOS
E935 76 0555 WBHALT: HALT
0556 ;
0557 ; THIS BIOS SUBROUTINE SELECTS THE DISK DRIVE SPECIFIED BY C
0558 ;
E936 210000 0559 SELDSK: LD HL,00000H ; ZERO HL FOR PARAMETER BLOCK OFFSET
0560 ; COMPUTATION
E939 79 0561 LD A,C ; DRIVE NUMBER TO SELECT TO A
E93A FE02 0562 CP NDRIVS ; CHECK IF GREATER THAN LAST DRIVE SUPPORTED
E93C D0 0563 RET NC ; RETURN IF DRIVE NUMBER INVALID
E93D 32BFEF 0564 LD (DRIVE),A ; SAVE DRIVE NUMBER TO SELECT
E940 324200 0565 LD (PODRV),A ; MAKE IT THE CURRENTLY SELECTED DRIVE
E943 6F 0566 LD L,A ; DRIVE NUMBER TO L
E944 29 0567 ADD HL,HL ; SHIFT HL LEFT 4 I. E. MULTIPLY BY 16 TO
E945 29 0568 ADD HL,HL ; COMPUTE THE OFFSET OF THE DISK PARAMETER
E946 29 0569 ADD HL,HL ; HEADER FOR THE SELECTED DRIVE SINCE EACH
E947 29 0570 ADD HL,HL ; PARAMETER HEADER IS 16 BYTES
E948 116DE6 0571 LD DE,DPARMH ; GET THE BASE ADDRESS OF THE DISK PARAMETER
0572 ; HEADERS FOR THE DRIVES
E94B 19 0573 ADD HL,DE ; COMPUTE THE ADDRESS OF THE DISK PARAMETER
0574 ; HEADER FOR THE SELECTED DRIVE
E94C E5 0575 PUSH HL ; SAVE SELECTED DISK PARAMETER HEADER ADDRESS
0576 ; ON THE STACK
E94D CD42EB 0577 CALL SDNSID ; SET SINGLE/DOUBLE SIDED FLAG AND DENSITY FLAG
0578 ; FOR DRIVE AND RETURN DENSITY FLAG
E950 FEFF 0579 CP 0FFH ; CHECK IF SINGLE DENSITY SET
E952 2010 0580 JR NZ,SLCDD ; GO CHECK DOUBLE DENSITY IF NOT
E954 11ADE6 0581 LD DE,SDSKTB ; GET ADDRESS OF SINGLE DENSITY SECTOR SKEW
0582 ; TABLE

```



```

E957 01D9E7 0583 LD BC,PSSD ; GET ADDRESS OF SSSD PARAMETER BLOCK
E95A CDFAEB 0584 CALL GETSID ; CHECK IF DRIVE IS SINGLE OR DOUBLE SIDED
E95D 2827 0585 JR Z,SLSDPH ; GO SET UP SDSKTB AND PSSD IN DISK PARAMETER
0586 ; HEADER IF SINGLE SIDED
E95F 0106E8 0587 LD BC,PDSSD ; GET ADDRESS OF DSSD PARAMETER BLOCK
E962 1822 0588 JR SLSDPH ; GO SET SDSKTB AND PDSSD IN DISK PARAMETER
0589 ; HEADER OF DOUBLE SIDED
E964 11E1E6 0590 SLCDD: LD DE,DDSKTB ; GET ADDRESS OF DOUBLE DENSITY (26 256 BYTE
0591 ; SECTORS PER TRACK) SECTOR SKEW TABLE
E967 01E8E7 0592 LD BC,PSSDD ; GET ADDRESS OF SSDD PARAMETER BLOCK
E96A FE40 0593 CP 040H ; CHECK IF DRIVE IS MAXIMUM DENSITY
E96C 2010 0594 JR NZ,SLCDS ; GO CHECK IF DRIVE IS DOUBLE SIDED IF NOT
E96E 1149E7 0595 LD DE,MDSKTB ; GET ADDRESS OF MAXIMUM DENSITY (9 1024 BYTE
0596 ; SECTORS PER TRACK) SECTOR SKEW TABLE
E971 0115E8 0597 LD BC,PSSMD ; GET ADDRESS OF SSMD PARAMETER BLOCK
E974 CDFAEB 0598 CALL GETSID ; CHECK IF DRIVE IS DOUBLE SIDED
E977 280D 0599 JR Z,SLSDPH ; GO STORE MDSKTB AND PSSMD IN DISK PARAMETER
0600 ; HEADER IF NOT DOUBLE SIDED
E979 0124E8 0601 LD BC,PDSMD ; GET ADDRESS OF DSMD PARAMETER BLOCK
E97C 1808 0602 JR SLSDPH ; GO STORE MDSKTB AND PDSMD IN DISK PARAMETER
0603 ; HEADER
E97E CDFAEB 0604 SLCDS: CALL GETSID ; CHECK IF DRIVE IS DOUBLE SIDED
E981 2803 0605 JR Z,SLSDPH ; GO STORE DDSKTB AND PSSDD IN DISK PARAMETER
0606 ; HEADER IF NOT DOUBLE SIDED
E983 01F7E7 0607 LD BC,PDSDD ; GET ADDRESS OF DSDD PARAMETER BLOCK
E986 E1 0608 SLSDPH: POP HL ; GET DISK PARAMETER HEADER ADDRESS
E987 E5 0609 PUSH HL ; PUT DISK PARAMETER HEADER ADDRESS BACK ON
0610 ; STACK
E988 73 0611 LD (HL),E ; PUT LSB OF SECTOR SKEW TABLE ADDRESS IN DISK
0612 ; PARAMETER HEADER
E989 23 0613 INC HL ; BUMP TO MSB OF SECTOR SKEW TABLE IN DISK
0614 ; PARAMETER HEADER ADDRESS
E98A 72 0615 LD (HL),D ; PUT MSB OF SECTOR SKEW TABLE ADDRESS IN DISK
0616 ; PARAMETER HEADER
E98B 23 0617 INC HL ; INCREMENT TO NEXT ADDRESS IN DISK PARAMETER
0618 ; HEADER
E98C 110800 0619 LD DE,00008H ; GET OFFSET TO STORE DISK PARAMETER BLOCK
0620 ; ADDRESS IN DISK PARAMETER HEADER
E98F 19 0621 ADD HL,DE ; COMPUTE ADDRESS OF DISK PARAMETER BLOCK IN
0622 ; IN DISK PARAMETER HEADER
E990 71 0623 LD (HL),C ; PUT LSB OF DISK PARAMETER BLOCK ADDRESS IN
0624 ; DISK PARAMETER HEADER
E991 23 0625 INC HL ; BUMP TO MSB OF DISK PARAMETER BLOCK ADDRESS
0626 ; IN THE DISK PARAMETER HEADER
E992 70 0627 LD (HL),B ; PUT MSB OF DISK PARAMETER BLOCK ADDRESS IN
0628 ; THE DISK PARAMETER HEADER
E993 E1 0629 POP HL ; RESTORE ADDRESS OF DISK PARAMETER HEADER FOR
0630 ; DRIVE
E994 C9 0631 RET
0632 ;
0633 ; THIS BIOS ENTRY POINT SETS THE CURRENT TRACK ON SELECTED DRIVE TO 0 (HOME)
0634 ;
E995 AF 0635 HOME: XOR A ; ZERO TRACK NUMBER
E996 4F 0636 LD C,A ; PUT 0 IN BIOS SETTRK REGISTER
E997 47 0637 LD B,A
0638 ;
0639 ; THIS BIOS SUBROUTINE SETS THE CURRENT TRACK TO THE CONTENTS OF C
0640 ;
E998 79 0641 SETTRK: LD A,C ; NEW TRACK NUMBER TO A
E999 32C0EF 0642 LD (TRACK),A ; STORE NEW CURRENT TRACK
E99C C9 0643 RET
0644 ;
0645 ; THIS BIOS SUBROUTINE SETS THE CURRENT SECTOR ON THE CURRENT TRACK TO THE
0646 ; CONTENTS OF C
0647 ;
E99D 79 0648 SETSEC: LD A,C ; NEW SECTOR NUMBER TO A
E99E 32C2EF 0649 LD (SECTOR),A ; SAVE NEW CURRENT SECTOR NUMBER TO READ/WRITE
E9A1 C9 0650 RET
0651 ;
0652 ; THIS BIOS SUBROUTINE SETS THE CURRENT DMA ADDRESS TO THE CONTENTS OF BC
0653 ;
E9A2 ED43E5EF 0654 SETDMA: LD (DMADDR),BC ; SAVE NEW DMA ADDRESS TO READ TO/WRITE FROM
E9A6 C9 0655 RET
0656 ;
0657 ; THIS SUBROUTINE PERFORMS THE LOGICAL TO PHYSICAL SECTOR TRANSLATION
0658 ; BASED ON THE CURRENT SECTOR SKEW TABLE
0659 ;
E9A7 0600 0660 SECTRN: LD B,000H ; ZERO B BEFORE DOUBLE ADD OF BC AND HL SINCE C

```

```

0661                                     ; CONTAINS THE SECTOR NUMBER
E9A9 EB 0662 EX DE,HL ; SECTOR TRANSLATION TABLE ADDRESS TO HL
E9AA 09 0663 ADD HL,BC ; COMPUTE ADDRESS OF SECTOR NUMBER IN TABLE
E9AB 6E 0664 LD L,(HL) ; GET PHYSICAL SECTOR NUMBER TO READ
E9AC 2600 0665 LD H,000H ; ZERO H SINCE SECTOR NUMBER IS IN HL
E9AE C9 0666 RET
0667 ;
0668 ; THIS SUBROUTINE PERFORMS THE PHYSICAL SECTOR READS
0669 ;
E9AF CD4BEA 0670 PREAD: CALL DRINIT ; INSURE DRIVE, TRACK, SECTOR, SIDE, AND
0671 ; DENSITY ARE SET UP BEFORE READ. RETURNS
0672 ; WITH B SET TO BUFFER SIZE AND E SET TO THE
0673 ; NUMBER OF BUFFERS TO INPUT/OUTPUT PER SECTOR
E9B2 C0 0674 RET NZ ; RETURN IF ERROR INITIALIZING DRIVE
E9B3 3E88 0675 LD A,RDSEC ; GET READ SECTOR COMMAND
E9B5 325400 0676 LD (POCMD),A ; SET READ SECTOR AS LAST COMMAND SENT
E9B8 CDCDE9 0677 CALL SF1SD2 ; SET BIT 1 (F1) IN COMMAND IF SIDE 2
E9BB CD25EB 0678 CALL CKHDLD ; SET 15 MSEC DELAY BIT IN COMMAND IF HEAD
0679 ; NOT LOADED
E9BE D364 0680 OUT (CMDP),A ; SEND READ SECTOR COMMAND
E9C0 0E67 0681 LD C,DATAP ; GET DATA PORT NUMBER FOR VERSAFLOPPY II
E9C2 50 0682 LD D,B ; SAVE BUFFER BLOCK SIZE IN D
E9C3 EDB2 0683 PRDLP: INIR ; INPUT A BUFFER OF DATA FROM THE SECTOR
E9C5 42 0684 LD B,D ; RESTORE BUFFER BLOCK SIZE
E9C6 1D 0685 DEC E ; CHECK IF ALL BLOCKS OF BUFFER ARE IN
E9C7 20FA 0686 JR NZ,PRDLP ; GO INPUT ANOTHER IF NOT
E9C9 1EFE 0687 LD E,RDEM ; SET STATUS MASK TO CHECK FOR NOT READY, WRITE
0688 ; PROTECT, RECORD TYPE, RECORD NOT FOUND, CRC
0689 ; ERROR, LOST DATA, AND DRQ
E9CB 1845 0690 JR CBUSY1 ; GO GET AND CHECK CONTROLLER STATUS
0691 ;
0692 ; THIS SUBROUTINE SETS BIT 1 IN THE CONTORLLER COMMAND IF SIDE 2 OF THE DISK
0693 ; IS SELECTED.
0694 ;
E9CD 3A4700 0695 SF1SD2: LD A,(POLSEL) ; GET LAST SELECT COMMAND
E9D0 E610 0696 AND SIDEM ; CHECK IF SIDE 2 IS SELECTED
E9D2 3A5400 0697 LD A,(POCMD) ; GET COMMAND TO SEND
E9D5 C0 0698 RET NZ ; RETURN IF NOT SIDE 2
E9D6 CBCF 0699 SET F1B,A ; SET BIT 1 (F1) IN COMMAND IF SIDE 2
E9D8 325400 0700 LD (POCMD),A ; SAVE AS LAST COMMAND SENT
E9DB C9 0701 RET
0702 ;
0703 ; THIS SUBROUTINE PERFORMS THE PHYSICAL SECTOR WRITES
0704 ;
E9DC CD4BEA 0705 PWRITE: CALL DRINIT ; INSURE DRIVE, TRACK, SECTOR DENSITY AND SIDE
0706 ; ARE SET UP BEFORE WRITING. RETURNS WITH B
0707 ; SET TO THE BUFFER SIZE, AND E SET TO THE
0708 ; NUMBER OF BUFFERS TO INPUT/OUTPUT PER SECTOR
E9DF C0 0709 RET NZ ; RETURN IF ERROR INITIALIZING DRIVE
E9E0 3EA8 0710 LD A,WTSEC ; GET SECTOR WRITE COMMAND
E9E2 325400 0711 LD (POCMD),A ; SET AS LAST COMMAND SENT
E9E5 CDCDE9 0712 CALL SF1SD2 ; SET BIT 1 IN COMMAND IF SIDE 2 SELECTED
E9E8 CD25EB 0713 CALL CKHDLD ; SET 15 MSEC DELAY BIT IN COMMAND IF HEAD NOT
0714 ; LOADED
E9EB D364 0715 OUT (CMDP),A ; SEND WRITE COMMAND TO CONTROLLER
E9ED 0E67 0716 LD C,DATAP ; GET DATA PORT ADDRESS FOR VERSAFLOPPY II
E9EF 50 0717 LD D,B ; SAVE BUFFER BLOCK SIZE IN D
E9F0 EDB3 0718 PWRTOL: OTIR ; OUTPUT BUFFER BLOCK TO CONTROLLER TO WRITE
E9F2 42 0719 LD B,D ; RESTORE BUFFER BLOCK SIZE FOR OTIR
E9F3 1D 0720 DEC E ; CHECK IF ALL BLOCKS OF THE SECTOR ARE WRITTEN
E9F4 20FA 0721 JR NZ,PWRTOL ; GO OUTPUT NEXT BLOCK IF NOT
E9F6 1EFC 0722 LD E,WTEM ; SET UP STATUS MASK TO CHECK FOR NOT READY,
0723 ; WRITE PROTECT, RECORD TYPE, RECORD NOT FOUND,
0724 ; CRC ERROR, LOST DATA, AND DRQ
E9F8 1818 0725 JR CBUSY1 ; GO GET AND CHECK CONTROLLER STATUS AND RETURN
0726 ;
0727 ; THIS SUBROUTINE SENDS THE COMMAND IN A AND SETS THE STATUS ERROR MASK IN E
0728 ; TO 0 TO IGNORE ALL ERRORS
0729 ;
E9FA 1E00 0730 SCOMIE: LD E,000H ; SET CONTROLLER STATUS MASK TO IGNORE ERRORS
0731 ;
0732 ; THIS SUBROUTINE ENTRY POINT SENDS THE COMMAND IN A AND CHECKS THE CONTROLLER
0733 ; STATUS USING THE ERROR MASK IN E
0734 ;
E9FC 57 0735 SCOMME: LD D,A ; SAVE CONTROLLER COMMAND IN D
E9FD DB64 0736 CKBSYL: IN A,(STATSP) ; GET CONTROLLER STATUS
E9FF E601 0737 AND BUSYM ; CHECK IF CONTROLLER IS BUSY
EA01 20FA 0738 JR NZ,CKBSYL ; GO CHECK STATUS AGAIN IF BUSY

```

```

EA03 DB63      0739      IN      A,(SELCTP)      ; GET DRIVE SELECT STATUS
                0740      ; AND      EWAITM      ; SET WAIT ENABLE (BUSY)
EA05 F680      0741      OR      DWAITM      ; SET WAIT DISABLE (NOT BUSY)
EA07 D363      0742      OUT      (SELCTP),A      ; SET WAIT DISABLE
EA09 7A        0743      LD      A,D          ; PUT CONTROLLER COMMAND BACK IN A
EA0A D364      0744      OUT      (CMDP),A      ; SEND COMMAND
EA0C F5        0745      PUSH   AF          ; DELAY BEFORE CHECKING
EA0D F1        0746      POP    AF          ;
EA0E F5        0747      PUSH   AF          ;
EA0F F1        0748      POP    AF          ;
EA10 F5        0749      PUSH   AF          ;
EA11 F1        0750      POP    AF          ;
EA12 DB64      0751      CBUSY1: IN      A,(STATSP)      ; GET CONTROLLER STATUS
EA14 E601      0752      AND    BUSYM      ; CHECK IF CONTROLLER BUSY
EA16 20FA      0753      JR     NZ,CBUSY1   ; GO CHECK STATUS AGAIN IF BUSY
EA18 DB63      0754      IN      A,(SELCTP)      ; GET DRIVE SELECT STATUS
EA1A F680      0755      OR      DWAITM      ; SET WAIT DISABLE
EA1C D363      0756      OUT      (SELCTP),A      ; DISABLE WAIT ON DRIVE SELECT PORT
EA1E DB64      0757      IN      A,(STATSP)      ; GET CONTROLLER STATUS
EA20 A3        0758      AND    E          ; CHECK IF ANY UNMASKED ERRORS OCCURRED
EA21 C8        0759      RET    Z          ; RETURN IF NO ERRORS
                0760      ;
                0761      ; THIS ENTRY POINT SETS ERROR STATUS AND RETURNS
                0762      ;
EA22 5F        0763      STERST: LD      E,A          ; SAVE STATUS IN E
EA23 325500    0764      LD      (POCST),A      ; SAVE LAST CONTROLLER ERROR STATUS BYTE
EA26 3EFF      0765      LD      A,OFFH      ; SET ERROR OCCURRED ON LAST DISK CONTROLLER
EA28 32E4EF    0766      LD      (ERFLAG),A      ; OPERATION FLAG
EA2B B7        0767      OR      A          ; INSURE NZ FLAG IS SET (ERROR)
EA2C C9        0768      RET
                0769      ;
                0770      ; THIS SUBROUTINE SELECTS THE DRIVE SPECIFIED FOR THE NEXT OPERATION WITH THE
                0771      ; PROPER DENSITY.
                0772      ;
EA2D 3A4200    0773      SELDDN: LD      A,(P0DRV)      ; GET CURRENTLY SELECTED DRIVE
EA30 CD35EB    0774      CALL   DNTODS      ; CONVERT DRIVE NUMBER TO DRIVE SELECT COMMAND
EA33 CBB7      0775      RES    DENSB,A      ; ZERO DENSITY TO SELECT DOUBLE DENSITY
EA35 5F        0776      LD      E,A          ; SAVE SELECT COMMAND IN E
EA36 CD6EEB    0777      CALL   GETDEN      ; CHECK IF SINGLE OR DOUBLE DENSITY
EA39 7B        0778      LD      A,E          ; PUT SELECT COMMAND BACK IN A
EA3A 2002      0779      JR     NZ,SLDCT0    ; GO CHECK IF TRACK 0 IF DOUBLE DENSITY
EA3C CBF7      0780      SET    DENSB,A      ; SET SINGLE DENSITY SELECT
EA3E 5F        0781      SLDCT0: LD      E,A          ; SAVE SELECT COMMAND IN E
EA3F 3A4300    0782      LD      A,(P0TRK)     ; GET TRACK NUMBER
EA42 B7        0783      OR      A          ; CHECK IF TRACK 0
EA43 7B        0784      LD      A,E          ; PUT SELECT COMMAND BACK IN A
EA44 2002      0785      JR     NZ,SLDSC     ; GO SEND COMMAND IF NOT TRACK 0
EA46 CBF7      0786      SET    DENSB,A      ; SET SINGLE DENSITY SELECT IF TRACK 0
EA48 D363      0787      SLDSC: OUT      (SELCTP),A      ; SEND SELECT COMMAND
EA4A C9        0788      RET
                0789      ;
                0790      ; THIS SUBROUTINE INSURES THAT THE DRIVE, TRACK, SECTOR, DENSITY, AND SIDE ARE
                0791      ; SET UP BEFORE WRITING OR READING.
                0792      ;
EA4B AF        0793      DRINIT: XOR    A          ; ZERO LAST ERROR
EA4C 325500    0794      LD      (POCST),A      ;
EA4F DB63      0795      IN      A,(SELCTP)      ; GET CURRENT SELECT STATUS
EA51 E60F      0796      AND    DRIVM      ; ISOLATE DRIVE FIELD
EA53 0EFF      0797      LD      C,OFFH      ; INITIALIZE DRIVE NUMBER
EA55 0C        0798      DRIDL P: INC    C          ; BUMP NUMBER OF DRIVE SELECTED
EA56 1F        0799      RRA          ; CHECK IF THIS CORRESPONDS TO DRIVE SELECT CMD
EA57 38FC      0800      JR     C,DRIDL P    ; GO BUMP AGAIN IF NOT
EA59 3A4200    0801      LD      A,(P0DRV)     ; GET DRIVE NUMBER OF NEXT OPERATION
EA5C E603      0802      AND    DRIVNM      ; ISOLATE DRIVE NUMBER BITS
EA5E B9        0803      CP     C          ; CHECK IF SAME AS SELECTED DRIVE
EA5F 2829      0804      JR     Z,SELDRD     ; GO SELECT DRIVE AND DENSITY IF SAME AS
                0805      ; CURRENTLY SELECTED DRIVE
EA61 CD2DEA    0806      CALL   SELDDN      ; SELECT DRIVE AND DENSITY
EA64 CD80EB    0807      CALL   DSELCR      ; SEND DRIVE SELECT COMMAND, CHECK FOR
                0808      ; CONTROLLER READY
EA67 2809      0809      JR     Z,DRICRA     ; GO READ ADDRESS BYTES IF CONTROLLER READY
EA69 1E80      0810      LD      E,DNRDYM     ; SET NOT READY STATUS
                0811      ;
                0812      ; THIS ENTRY POINT DESELECTS THE DRIVE AND SETS UP ERROR STATUS FOR RETURN
                0813      ;
EA6B 3EFF      0814      DSELDR: LD      A,DESEL      ; GET DESELECT, CLEAR WAIT COMMAND
EA6D D363      0815      OUT      (SELCTP),A      ; DESELECT DRIVE AND CLEAR WAIT
EA6F 7B        0816      LD      A,E          ; PUT STATUS BACK IN A

```

EA70	18B0	0817	JR	STERST	; GO SAVE ERROR STATUS, SET UP ERROR RETURN
		0818			; STATUS, AND RETURN
EA72	CDBEB	0819	DRICRA:	CALL RDADDR	; READ ADDRESS BYTES INTO 48-4B
EA75	280E	0820	JR	Z,SELSTK	; GO OUTPUT TRACK ADDRESS IF READ OK
EA77	3E08	0821	LD	A,RESTOR	; GET RESTORE COMMAND
EA79	CDFAE9	0822	CALL	SCOMIE	; RESTORE TO TRACK 0 IF ERROR READING TRACK
		0823			; ADDRESS
EA7C	3E5C	0824	LD	A,STEPI	; GET STEP IN COMMAND
EA7E	32E3EF	0825	LD	(STEP),A	; SET LAST STEP COMMAND TO STEP IN
EA81	3E00	0826	LD	A,000H	; SET TRACK NUMBER POSITION TO 0
EA83	1803	0827	JR	SELOTK	; GO OUTPUT TRACK 0 TO TRACK PORT
EA85	3A4800	0828	SELSTK:	LD A,(POSAB)	; GET TRACK NUMBER FROM ADDRESS BYTES JUST READ
EA88	D365	0829	SELOTK:	OUT (TRACKP),A	; SET TRACK NUMBER IN CONTROLLER
EAB8A	CD2DEA	0830	SELDRD:	CALL SELDDN	; SELECT DRIVE AND DENSITY
EAB8D	1E91	0831	LD	E,STERM	; SET STATUS ERROR MASK TO CHECK NOT READY,
		0832			; SEEK ERROR AND BUSY
EAB8F	3A4300	0833	LD	A,(POTRK)	; GET DESIRED TRACK NUMBER
EA92	4F	0834	LD	C,A	; DESIRED TRACK NUMBER TO C
EA93	DB65	0835	IN	A,(TRACKP)	; GET TRACK NUMBER OF CURRENT POSITION
EA95	B9	0836	CP	C	; CHECK IF TRACK CORRECT
EA96	282A	0837	JR	Z,SELSEC	; GO SET UP SECTOR NUMBER IF TRACK CORRECT
EA98	3E7C	0838	LD	A,STEPO	; GET STEP OUT COMMAND
EA9A	3002	0839	JR	NC,SELSTP	; GO STEP OUT IF TRACK # < CURRENT POSITION
EA9C	3E5C	0840	LD	A,STEPI	; GET STEP IN COMMAND
EA9E	32E3EF	0841	SELSTP:	LD (STEP),A	; SAVE LAST STEP COMMAND
EAA1	0603	0842	LD	B,003H	; SET RETRY COUNT TO 3
EAA3	79	0843	SELGTK:	LD A,C	; GET DESIRED TRACK TO SEEK
EAA4	D367	0844	OUT	(DATAP),A	; SET TRACK # IN CONTROLLER TO DESIRED TRACK
EAA6	3E1C	0845	LD	A,SKHDLV	; GET SEEK, HEAD LOAD, VERIFY COMMAND
EAA8	325400	0846	LD	(POCMD),A	; SET THE ABOVE AS LAST COMMAND SENT
EABAB	C5	0847	PUSH	BC	; SAVE DESIRED TRACK AND RETRY COUNT ON STACK
EABAC	CDFCE9	0848	CALL	SCOMME	; SEND SEEK COMMAND %
		0849	; CALL	SCOMIE	; SEND SEEK COMMAND (IGNORE ERRORS) ^
EAAF	C1	0850	POP	BC	; RESTORE TRACK AND RETRY COUNT FROM STACK
EAB0	2810	0851	JR	Z,SELSEC	; GO SET UP SECTOR IF NO ERROR SEEKING TRACK
EAB2	3E08	0852	LD	A,RESTOR	; GET RESTORE, HEAD LOAD COMMAND
EAB4	C5	0853	PUSH	BC	; SAVE DESIRED TRACK AND RETRY COUNT ON STACK
EAB5	CDFCE9	0854	CALL	SCOMME	; SEND RESTORE COMMAND
EAB8	C1	0855	POP	BC	; GET TRACK AND RETRY COUNT FROM THE STACK
EAB9	1E91	0856	LD	E,STERM	; RELOAD SEEK TRACK STATUS ERROR MASK %
EABB	10E6	0857	DUNZ	SELGTK	; GO SEND SEEK AGAIN IF RETRY NOT EXHAUSTED
EABD	3E10	0858	LD	A,SEEKEM	; SET STATUS TO SEEK ERROR
EABF	C322EA	0859	JP	STERST	; GO SET ERROR STATUS AND RETURN IF
		0860			; UNRECOVERABLE
EAC2	0610	0861	SELSEC:	LD B,SIDEM	; SET DESIRED SIDE SELECT TO SIDE 1
EAC4	3A4500	0862	LD	A,(POSEC)	; GET DESIRED SECTOR NUMBER
EAC7	57	0863	LD	D,A	; SAVE SECTOR NUMBER IN D
EAC8	0E1A	0864	LD	C,01AH	; GET HIGHEST SECTOR NUMBER ON SIDE 1, SD, DD
EACA	3A4300	0865	LD	A,(POTRK)	; GET TRACK NUMBER
EACD	B7	0866	OR	A	; CHECK IF TRACK 0 (SINGLE DENSITY)
EACE	7A	0867	LD	A,D	; DESIRED SECTOR BACK TO A
EACF	280A	0868	JR	Z,SELCS	; GO CHECK IF DESIRED SECTOR IS ON SIDE 2 IF
		0869			; TRACK IS 0 (SINGLE DENSITY)
EAD1	CD6EEB	0870	CALL	GETDEN	; GET DENSITY FLAG FOR SELECTED DRIVE
EAD4	FE40	0871	CP	040H	; CHECK IF MAXIMUM DENSITY
EAD6	7A	0872	LD	A,D	; DESIRED SECTOR BACK TO A
EAD7	2002	0873	JR	NZ,SELCS	; GO CHECK IF DESIRED SECTOR MAY BE ON SIDE 2
		0874			; IF SINGLE OR DOUBLE DENSITY (26 SECTORS PER
		0875			; TRACK
EAD9	0E09	0876	LD	C,009H	; GET HIGHEST SECTOR NUMBER ON SIDE 1, MD
EADB	B9	0877	SELCS:	CP C	; CHECK IF DESIRED SECTOR ON SIDE 1
EADC	2806	0878	JR	Z,SELCSL	; GO SELECT IF
EADE	3804	0879	JR	C,SELCSL	; SECTOR ON SIDE 1
EAE0	0600	0880	LD	B,000H	; SET DESIRED SIDE SELECT STATUS TO SIDE 2
EAE2	91	0881	SUB	C	; COMPUTE SECTOR NUMBER ON SIDE 2
EAE3	57	0882	LD	D,A	; SAVE SECTOR NUMBER ON SIDE 2 IN D
EAE4	DB63	0883	SELCSL:	IN A,(SELCTP)	; GET DRIVE SELECT STATUS
EAE6	324700	0884	LD	(POLSEL),A	; SAVE LAST SELECT COMMAND
EAE9	E610	0885	AND	SIDEM	; ISOLATE SIDE BIT IN SELECT STATUS
EABE	B8	0886	CP	B	; CHECK IF SAME AS DESIRED SIDE
EABEC	CAF9EA	R 0887	JP	Z,SELOSC	; GO OUTPUT SECTOR IF CORRECT SIDE SELECTED
EAEF	3A4700	0888	LD	A,(POLSEL)	; GET SELECT STATUS
EAF2	EE10	0889	XOR	SIDEM	; SWITCH TO OTHER SIDE (2)
EAF4	D363	0890	OUT	(SELCTP),A	; SEND SELECT COMMAND
EAF6	324700	0891	LD	(POLSEL),A	; SAVE LAST SELECT COMMAND SENT
EAF9	7A	0892	SELOSC:	LD A,D	; GET DESIRED SECTOR NUMBER
EAF8A	D366	0893	OUT	(SECTRP),A	; SEND TO SECTOR NUMBER PORT IN CONTROLLER
EAF8C	2A4000	0894	LD	HL,(PODMA)	; GET DMA ADDRESS

```

EAFF 0600      0895      LD      B,000H      ; SET BUFFER BYTE COUNT TO 256 BYTE BLOCKS
EB01 1E01      0896      LD      E,001H      ; SET NUMBER OF BLOCKS TO 1
EB03 CD6EEB     0897      CALL   GETDEN      ; GET DENSITY FLAG FOR SELECTED DRIVE
EB06 2811      0898      JR      Z,SELSD     ; GO SET 128 BYTE BLOCKS IF SINGLE DENSITY
EB08 3A4300    0899      LD      A,(P0TRK)   ; GET SELECTED TRACK NUMBER
EB0B B7          0900      OR      A          ; CHECK IF TRACK 0 (SINGLE DENSITY)
EB0C 280B      0901      JR      Z,SELSD     ; GO SET UP 128 BYTE BLOCKS IF TRACK 0, SD
EB0E CD6EEB     0902      CALL   GETDEN      ; GET DENSITY FLAG FOR SELECTED DRIVE
EB11 FE40      0903      CP      040H       ; CHECK IF MAXIMUM DENSITY
EB13 2008      0904      JR      NZ,SELSW    ; GO SET WAIT ENABLE (BUSY) IF DOUBLE DENSITY
EB15 1E04      0905      LD      E,004H      ; CHANGE BLOCK COUNT TO 4 IF MAXIMUM DENSITY
EB17 1804      0906      JR      SELSW       ; GO SET WAIT ENABLE
EB19 0680      0907      SELSD: LD      B,080H      ; SET BUFFER BLOCK BYTE COUNT TO 128
EB1B 1E01      0908      LD      E,001H      ; SET NUMBER OF BLOCKS PER BUFFER TO 1 ????????
0909      ;
0910      ; THIS ENTRY POINT SETS THE SELECT WAIT ENABLE BIT TO 0 (INVERSE LOGIC).
0911      ;
EB1D DB63      0912      SELSW: IN      A,(SELCTP) ; GET SELECT STATUS
EB1F E67F      0913      AND     EWAITM      ; SET WAIT ENABLE (INVERSE LOGIC)
EB21 D363      0914      OUT    (SELCTP),A   ; SEND SELECT COMMAND WITH WAIT ENABLE (BUSY)
EB23 AF        0915      XOR    A          ; SET STATUS TO NO SELECT ERRORS
EB24 C9        0916      RET
0917      ;
0918      ; THIS SUBROUTINE CHECKS IF THE HEAD IS LOADED, IF IT IS NOT, IT SETS THE 15
0919      ; MSEC DELAY BIT (2) IN THE CONTROLLER COMMAND IN A
0920      ;
EB25 57        0921      CKHDL: LD      D,A          ; SAVE COMMAND TO SEND IN D
EB26 3ED0      0922      LD      A,FORINT    ; GET FORCE INTERRUPT COMMAND
EB28 D364      0923      OUT    (CMDP),A     ; SEND FORCE INTERRUPT TO TERMINATE ANY ACTIVE
0924      ; COMMAND
EB2A F5        0925      PUSH   AF          ; DELAY BEFORE CHECKING
EB2B F1        0926      POP    AF          ; CONTROLLER STATUS
EB2C DB64      0927      IN      A,(STATSP) ; GET CONTROLLER STATUS
EB2E E620      0928      AND     HDNTLM      ; CHECK IF HEAD IS LOADED
EB30 7A        0929      LD      A,D         ; PUT COMMAND TO SEND IN A
EB31 C0        0930      RET     NZ         ; RETURN IF HEAD LOADED
EB32 F604      0931      OR     EBITM       ; SET 15 MSEC DELAY (E BIT) IF HEAD NOT LOADED
EB34 C9        0932      RET
0933      ;
0934      ; THIS SUBROUTINE SETS UP THE DRIVE SELECT BITS IN THE DRIVE SELECT COMMAND
0935      ; FOR THE DRIVE SPECIFIED IN A
0936      ;
EB35 E603      0937      DNTODS: AND     DRIVNM      ; ISOLATE DRIVE NUMBER BITS
EB37 B7        0938      OR     A          ; CHECK IF DRIVE 0 SELECTED
EB38 47        0939      LD      B,A        ; DRIVE NUMBER TO SELECT LOOP COUNTER
EB39 3E01      0940      LD      A,001H     ; START WITH DRIVE 0 SELECT BIT
EB3B 2803      0941      JR      Z,DSCPL    ; GO COMPLEMENT IF DRIVE 0 SPECIFIED
EB3D 07        0942      DSLP1: RLC        ; PUT DRIVE SELECT BIT IN NEXT DRIVE POSITION
EB3E 10FD      0943      DJNZ   DSLP1      ; GO SHIFT TO NEXT DRIVE IF NOT CORRECT YET
EB40 2F        0944      DSCPL: CPL        ; COMPLEMENT DRIVE SELECT FIELD BECAUSE THE
0945      ; VERSAFLOPPY SELECT PORT LOGIC IS REVERSE
EB41 C9        0946      RET
0947      ;
0948      ; THIS SUBROUTINE CHECKS TO SEE IF THE DENSITY OF THE DRIVE HAS BEEN SET UP
0949      ; I. E. INITIALIZED FOR THE MEDIA IN THE DRIVE. IF NOT, IT SELECTS THE DRIVE
0950      ; WITH DOUBLE DENSITY SET AND CHECKS IF THE DRIVE IS READY. IF READY, IT GOES
0951      ; TO READ ADDRESS AND RETURN. OTHERWISE IT SETS ERROR STATUS AND RETURNS
0952      ; CAUSING A DEFAULT TO SINGLE SIDED, SINGLE DENSITY FORMAT.
0953      ;
EB42 CD68EB     0954      SDNSID: CALL   SGTDEN      ; GET DENSITY FLAG FOR THE SELECTED DRIVE
EB45 FE80      0955      CP      080H       ; CHECK IF DENSITY HAS ALREADY BEEN SPECIFIED
EB47 C0        0956      RET     NZ         ; RETURN IF PREVIOUSLY SPECIFIED
EB48 3ABFEF     0957      LD      A,(DRIVE)   ; GET CURRENTLY SELECTED DRIVE NUMBER
EB4B CD35EB     0958      CALL   DNTODS      ; SET UP DRIVE FIELD IN DRIVE SELECT COMMAND
0959      ; FOR THE DRIVE SPECIFIED
EB4E CBB7      0960      RES    DENSB,A     ; ZERO DENSITY TO SELECT DOUBLE DENSITY
EB50 CBA7      0961      RES    SIDEB,A     ; ZERO SIDE BIT TO SELECT SIDE 2 %
EB52 CD80EB     0962      SDNSSEL: CALL  DSELCR    ; SEND DRIVE SELECT COMMAND, CHECK IF DRIVE %
0963      ; READY %
0964      ; CALL   DSELCR    ; SEND DRIVE SELECT COMMAND, CHECK IF DRIVE ^
0965      ; READY ^
EB55 283B      0966      JR      Z,RD2AD     ; GO READ ADDRESS ON TRACK 2 IF DRIVE READY
EB57 DB63      0967      IN      A,(SELCTP) ; GET DRIVE SELECT STATUS %
EB59 E610      0968      AND     A,SIDEM     ; CHECK IF SIDE 1 ALREADY SELECTED %
EB5B 2006      0969      JR      NZ,SDNNRD   ; GO SET NOT READY STATUS IF SIDE 1 %
EB5D DB63      0970      IN      A,(SELCTP) ; GET DRIVE SELECT STATUS %
EB5F F610      0971      OR     SIDEM       ; SET BIT TO SELECT SIDE 1 %
EB61 18EF      0972      JR      SDNSSEL    ; GO RESELECT WITH SIDE 1 SELECT SET %

```

```

EB63 1E80      0973 SDNNRD: LD   E,DNRDYM      ; SET STATUS TO RETURN TO NOT READY  %
          0974 ;     LD   E,DNRDYM      ; SET STATUS TO RETURN TO NOT READY  ^
EB65 C36BEA    0975 ;     JP   DSELDR        ; GO Deselect AND SET UP STATUS TO RETURN
          0976 ;
          0977 ; THIS SUBROUTINE GETS THE DENSITY FLAGS FOR THE DRIVE SPECIFIED IN A AND SETS
          0978 ; IT AS THE CURRENTLY SELECTED DRIVE. ?????????????????????????????????????????
          0979 ;
EB68 3ABFEF    0980 SGTDEN:  LD   A,(DRIVE)      ; GET CURRENTLY SELECTED DRIVE NUMBER
EB6B 324200    0981 ;     LD   (PODRV),A        ; SAVE SELECTED DRIVE IN SELECTED DRIVE IN
          0982 ;     ; PAGE 0
          0983 ;
          0984 ; THIS SUBROUTINE ENTRY POINT GETS THE DENSITY FLAG FOR THE SELECTED DRIVE.
          0985 ;
EB6E E5        0986 GETDEN:  PUSH  HL           ; SAVE HL AND
EB6F C5        0987 ;     PUSH  BC           ;     BC ON THE STACK
EB70 3A4200    0988 LD   A,(PODRV)      ; GET CURRENTLY SELECTED DRIVE NUMBER
EB73 214C00    0989 LD   HL,PODDFL      ; GET BASE ADDRESS OF DRIVE DENSITY FLAGS
EB76 4F        0990 LD   C,A           ; DRIVE NUMBER TO C
EB77 0600     0991 LD   B,000H        ; ZERO B SINCE ADDING BC TO HL
EB79 09        0992 ADD  HL,BC         ; COMPUTE ADDRESS OF DENSITY FLAG FOR DRIVE
EB7A 7E        0993 LD   A,(HL)        ; GET DENSITY FLAG FOR DRIVE
EB7B C1        0994 POP  BC           ; RESTORE BC
EB7C E1        0995 POP  HL           ;     AND HL FROM STACK
EB7D FEFF     0996 CP   OFFH         ; CHECK IF DENSITY FLAG IS SINGLE DENSITY TO
          0997 ;     ; RETURN THE CONDITION CODE
EB7F C9        0998 RET
          0999 ;
          1000 ; THIS SUBROUTINE SENDS THE DRIVE SELECT COMMAND IN A, DELAYS 41.541 MSEC, AND
          1001 ; THEN CHECKS CONTROLLER STATUS FOR NOT READY.
          1002 ;
EB80 D363     1003 DSELCR:  OUT   (SELCTP),A    ; SEND DRIVE SELECT COMMAND
EB82 0631     1004 ;     LD   B,031H        ; GET OUTER DELAY LOOP COUNT (49)
EB84 3EF0     1005 DELAYO:  LD   A,0F0H        ; GET INNER DELAY LOOP COUNT (240)
EB86 3D        1006 DELAYI:  DEC  A           ; CHECK IF INNER DELAY EXPIRED
EB87 C286EB   R 1007 ;     JP   NZ,DELAYI      ; GO CHECK AGAIN IF NOT
EB8A 05        1008 ;     DEC  B           ; CHECK IF OUTER DELAY COUNT EXPIRED
EB8B 20F7     1009 ;     JR   NZ,DELAYO      ; GO START INNER DELAY LOOP AGAIN IF
          1010 ;     ; 41.541 MSEC HAVE NOT PASSED
EB8D DB64     1011 ;     IN   A,(STATSP)     ; GET DISK CONTROLLER STATUS
EB8F E680     1012 ;     AND  DNRDYM        ; CHECK AND ISOLATE NOT READY STATUS
EB91 C9        1013 ;     RET
          1014 ;
          1015 ; THIS ENTRY POINT HOMES THE HEAD, STEPS IT TO TRACK 2 AND READS A SET OF
          1016 ; SECTOR ADDRESS BYTES ON TRACK 2
          1017 ;
EB92 3E08     1018 RD2AD:  LD   A,RESTOR      ; GET RESTORE COMMAND, LOAD HEAD
EB94 0602     1019 ;     LD   B,002H        ; GET NUMBER OF TRACKS TO STEP IN
EB96 CDFAE9   1020 ;     CALL SCOMIE        ; SEND RESTORE COMMAND (IGNORE ERRORS)
EB99 3E5C     1021 RD2STP: LD   A,STPEI      ; GET STEP IN, VERIFY HEAD LOAD COMMAND
EB9B CDFAE9   1022 ;     CALL SCOMIE        ; SEND STEP COMMAND (IGNORE ERRORS)
EB9E 10F9     1023 ;     DJNZ RD2STP        ; GO SEND STEP COMMAND AGAIN IF NOT AT TRACK 2
EBA0 0603     1024 ;     LD   B,3           ; SET READ ADDRESS RETRY COUNT TO 3  %
EBA2 C5        1025 RD2ELP: PUSH BC         ; SAVE READ ADDRESS RETRY COUNT ON STACK  %
EBA3 CDDBE8   1026 ;     CALL RDADDR        ; READ 4 ADDRESS BYTES INTO MEMORY STARTING AT
          1027 ;     ; ADDRESS 0048H
EBA6 C1        1028 ;     POP  BC           ; GET READ ADDRESS RETRY COUNT FROM STACK  %
EBA7 280B     1029 ;     JR   Z,RD2256      ; GO SET 256 BYTES PER SECTOR IF READ OK  %
EBA9 3E04     1030 ;     LD   A,04H        ; GET CRC ERROR STATUS BIT  %
EBAB A3        1031 ;     AND  E           ; CHECK IF CRC ERROR CAUSED ERROR STATUS  %
EBAC 2802     1032 ;     JR   Z,RD128      ; GO ASSUME 128 BYTE SINGLE DENSITY IF NOT CRC%
EBAE 10F2     1033 ;     DJNZ RD2ELP        ; GO RETRY READ ADDRESS IF RETRY NOT EXHAUSTED%
EBB0 1EFF     1034 RD128:  LD   E,OFFH        ; SET UP 128 BYTE SECTOR STATUS (SINGLE  %
          1035 ;     ; DENSITY) STATUS  %
          1036 ;     ; LD   E,OFFH        ; SET UP 128 BYTE SECTOR STATUS (SINGLE  ^
          1037 ;     ; DENSITY) STATUS  ^
EBB2 180B     1038 ;     JR   RD2SSZ        ; GO SET 128 BYTES PER SECTOR IF ERROR READING%
          1039 ;     ; ADDRESS BYTES  %
EBB4 1E00     1040 RD2256: LD   E,000H        ; SET UP 256 BYTE PER SECTOR (DOUBLE DENSITY) %
          1041 ;     ; STATUS  %
          1042 ;     ; JR   NZ,RD2SSZ      ; GO SET 128 BYTES PER SECTOR IF ERROR READING^
          1043 ;     ; ADDRESS BYTES  ^
          1044 ;     ; LD   E,000H        ; SET UP 256 BYTE PER SECTOR (DOUBLE DENSITY) ^
          1045 ;     ; STATUS  ^
EBB6 3A4B00   1046 ;     LD   A,(POSLB)     ; GET SECTOR LENGTH ADDRESS BYTE JUST READ
EBB9 FE03     1047 ;     CP   003H         ; CHECK IF 1024 BYTE SECTORS
EBBB 2002     1048 ;     JR   NZ,RD2SSZ      ; GO SET 256 BYTE SECTORS IF NOT
EBBD 1E40     1049 ;     LD   E,040H        ; SET UP MAXIMUM DENSITY STATUS
EBBF 214C00   1050 RD2SSZ:  LD   HL,PODDFL      ; GET BASE ADDRESS OF DENSITY FLAGS FOR DRIVES

```

```

EBC2 3ABFEF 1051 LD A,(DRIVE) ; GET CURRENTLY SELECTED DRIVE NUMBER
EBC5 4F 1052 LD C,A ; DRIVE NUMBER TO C
EBC6 0600 1053 LD B,000H ; ZERO B SINCE BC IS ADDED TO HL
EBC8 09 1054 ADD HL,BC ; COMPUTE ADDRESS OF DENSITY FLAG FOR DRIVE
EBC9 73 1055 LD (HL),E ; STORE DENSITY FLAG FOR DRIVE
EBCA 1600 1056 LD D,000H ; SET SINGLE/DOUBLE SIDED FLAG TO SINGLE
EBCC DB63 1057 IN A,(SELCTP) ; GET DRIVE SELECT STATUS
1058 ; AND 020H ; CHECK IF 5/8" BIT SET ??????? (010H ?) ^
1059 ; ; TO CHECK IF SIDE 2 IS SELECTED ??????????????^
EBCE E610 1060 AND SIDEM ; CHECK IF SIDE 2 IS SELECTED %
EBD0 2002 1061 JR NZ,RD2SSD ; GO SET SINGLE SIDED FLAG IF SINGLE
EBD2 1601 1062 LD D,001H ; SET DOUBLE SIDED STATUS
EBD4 215000 1063 RD2SSD: LD HL,POSSDS ; GET BASE ADDRESS OF SINGLE/DOUBLE SIDED FLAGS
1064 ; FOR DRIVES
EBD7 09 1065 ADD HL,BC ; COMPUTE ADDRESS OF SINGLE/DOUBLE SIDED FLAG
1066 ; FOR SELECTED DRIVE
EBD8 72 1067 LD (HL),D ; SAVE SINGLE/DOUBLE SIDED FLAG FOR DRIVE
EBD9 7B 1068 LD A,E ; DENSITY STATUS BACK TO A
EBDA C9 1069 RET
1070 ;
1071 ; THIS SUBROUTINE PERFORMS A READ ADDRESS ON THE CURRENTLY SELECTED TRACK OF
1072 ; THE CURRENTLY SELECTED DRIVE.
1073 ;
EBDB 214800 1074 RDADDR: LD HL,POSAB ; GET ADDRESS TO READ ADDRESS BYTES INTO
EBDE 0604 1075 LD B,004H ; SET UP TO READ 4 ADDRESS BYTES
EBE0 1EF8 1076 LD E,RAEM ; SET UP ERROR MASK TO CHECK NOT READY, RECORD
1077 ; NOT FOUND, AND CRC ERROR (IGNORE LOST DATA,
1078 ; DRQ)
EBE2 CD1DEB 1079 CALL SELSW ; SET WAIT ENABLE
EBE5 3EC0 1080 LD A,RDAD ; GET READ ADDRESS COMMAND
EBE7 325400 1081 LD (POCMD),A ; SAVE LAST COMMAND SENT (READ ADDRESS)
EBEA CDCDE9 1082 CALL SF1SD2 ; SET BIT 1 IN COMMAND IF SIDE 2 SELECTED
EBED D364 1083 OUT (CMDP),A ; SEND READ ADDRESS COMMAND
EBEF 0E67 1084 LD C,DATAP ; SET UP DISK CONTROLLER DATA PORT ADDRESS
EBF1 EDB2 1085 INIR ; INPUT 4 ADDRESS BYTES INTO 48, 49, 4A, AND 4B
EBF3 DB67 1086 IN A,(DATAP) ; BLEED OFF 2 %
EBF5 DB67 1087 IN A,(DATAP) ; ADDRESS CRC BYTES %
EBF7 C312EA 1088 JP CBUSY1 ; GO GET CONTROLLER STATUS, CHECK IF ERROR,
1089 ; AND RETURN
1090 ;
1091 ; THIS SUBROUTINE CHECKS IF THE CURRENTLY SELECTED DRIVE IS SINGLE OR DOUBLE
1092 ; SIDED.
1093 ;
EBFA E5 1094 GETSID: PUSH HL ; SAVE HL
EBFB C5 1095 PUSH BC ; AND BC ON THE STACK
EBFC 215000 1096 LD HL,POSSDS ; GET BASE ADDRESS OF DRIVE SINGLE/DOUBLE SIDED
1097 ; FLAGS
EBFF 3A4200 1098 LD A,(PODRV) ; GET DRIVE NUMBER
EC02 4F 1099 LD C,A ; DRIVE NUMBER TO C
EC03 0600 1100 LD B,000H ; ZERO MSB OF DRIVE (16 BIT ADD)
EC05 09 1101 ADD HL,BC ; COMPUTE ADDRESS OF SINGLE/DOUBLE SIDED FLAG
1102 ; FOR DRIVE
EC06 7E 1103 LD A,(HL) ; GET SINGLE/DOUBLE SIDED FLAG FOR DRIVE
EC07 C1 1104 POP BC ; RESTORE BC
EC08 E1 1105 POP HL ; AND HL FROM THE STACK
EC09 B7 1106 OR A ; CHECK IF DRIVE IS SINGLE SIDED
EC0A C9 1107 RET
1108 ;
1109 ; THIS ENTRY POINT AND SUBROUTINE WRITES THE PHYSICAL SECTOR AND RETRIEES THE
1110 ; WRITE IN CASE OF AN ERROR.
1111 ;
EC0B CD8DEC 1112 PWRITR: CALL SPODTS ; MOVE DMA ADDRESS, DRIVE, TRACK, AND SECTOR TO
1113 ; THEIR CORRESPONDING CELLS IN PAGE 0
EC0E 01050A 1114 LD BC,RETRYC ; INITIALIZE WRITE ERROR AND RESEEK RETRY
1115 ; COUNTS
EC11 C5 1116 PWRRT: PUSH BC ; SAVE WRITE RETRY AND RESEEK RETRY COUNTS ON
1117 ; THE STACK
EC12 3E00 1118 LD A,000H ; ZERO ERROR ON LAST
EC14 32E4EF 1119 LD (ERFLAG),A ; OPERATION FLAG
EC17 F3 1120 DI ; DISABLE INTERRUPTS DURING WRITE
EC18 CDDCE9 1121 CALL PWRITE ; WRITE SPECIFIED SECTOR
EC1B FB 1122 EI ; RE-ENABLE INTERRUPTS
EC1C C1 1123 POP BC ; RESTORE WRITE RETRY AND RESEEK ERROR COUNTERS
EC1D C8 1124 RET Z ; RETURN IF NO ERRORS
EC1E 2A5800 1125 LD HL,(POWTEC) ; GET COUNT OF WRITE ERRORS SINCE LAST WARM OR
1126 ; COLD BOOT
EC21 23 1127 INC HL ; INCREMENT COUNT OF WRITE ERRORS
EC22 225800 1128 LD (POWTEC),HL ; SAVE NEW WRITE ERROR COUNT

```

```

EC25 7B      1129      LD      A,E          ; CONTROLLER ERROR STATUS BYTE TO A
EC26 E63F    1130      AND     WCLDBM       ; CHECK IF ERROR WRITE PROTECT OR NOT READY
EC28 282F    1131      JR      Z,PUNREC    ; GO SET UNRECOVERABLE ERROR STATUS IN WRITE
                    1132      ; PROTECT OR NOT READY
EC2A 10E5    1133      DJNZ   PWRRTR       ; GO RETRY WRITE IF RETRY COUNT NOT EXHAUSTED
EC2C CD5FEC  1134      CALL   STEPOI       ; REPOSITION HEAD ON TRACK VIA STEP OUT/IN OR
                    1135      ; IN/OUT
EC2F 20E0    1136      JR      NZ,PWRRTR   ; GO RETRY WRITE IF RESEEK RETRY COUNT NOT
                    1137      ; EXHAUSTED
EC31 1826    1138      JR      PUNREC      ; GO SET UNRECOVERABLE ERROR IF RESEEK RETRY
                    1139      ; COUNT EXHAUSTED
                    1140      ;
                    1141      ; THIS ENTRY POINT AND SUBROUTINE READS A PHYSICAL SECTOR AND RETRIES THE READ
                    1142      ; IN CASE OF AN ERROR.
                    1143      ;
EC33 CD8DEC  1144      PREADR: CALL   SPODTS ; MOVE DMA ADDRESS, DRIVE, TRACK AND SECTOR TO
                    1145      ; THEIR CORRESPONDING PAGE 0 CELLS
                    1146      ;
                    1147      ; THIS SUBROUTINE ENTRY POINT IS USED BY BOOT TO READ PHYSICAL SECTORS OF BDOS
                    1148      ; AND CCP.
                    1149      ;
EC36 01050A 1150      RDBOOT: LD      BC,RETRYC ; INITIALIZE READ ERROR AND RESEEK RETRY COUNTS
EC39 C5      1151      PRDRTY: PUSH   BC      ; SAVE RETRY COUNTS ON THE STACK
EC3A 3E00    1152      LD      A,000H      ; ZERO ERROR STATUS
EC3C 32E4EF  1153      LD      (ERFLAG),A ; ON LAST OPERATION
EC3F F3      1154      DI          ; DISABLE INTERRUPTS DURING READ
EC40 CDAFE9  1155      CALL   PREAD       ; READ SPECIFIED SECTOR
EC43 FB      1156      EI          ; RE-ENABLE INTERRUPTS
EC44 C1      1157      POP     BC         ; GET READ AND SEEK RETRY COUNTS
EC45 C8      1158      RET      Z         ; RETURN IF NO ERRORS ON THE READ
EC46 2A5600  1159      LD      HL,(PORDEC) ; GET COUNT OF READ ERRORS SINCE LAST BOOT
                    1160      ; INC      A      ; INCREMENT COUNT OF READ ERRORS ??????????????^
EC49 23      1161      INC     HL         ; INCREMENT COUNT OF READ ERRORS %
EC4A 225600  1162      LD      (PORDEC),HL ; RESTORE COUNT OF READ ERRORS SINCE BOOT OR ^C
EC4D 7B      1163      LD      A,E        ; READ OPERATION STATUS TO A
EC4E E67F    1164      AND     NRDYM       ; CHECK IF NOT READY CAUSED ERROR
EC50 2807    1165      JR      Z,PUNREC    ; GO SET UNRECOVERABLE ERROR STATUS IF NOT
                    1166      ; READY CAUSED ERROR
EC52 10E5    1167      DJNZ   PRDRTY      ; GO RETRY READ IF RETRY ERROR COUNT NOT
                    1168      ; EXHAUSTED
EC54 CD5FEC  1169      CALL   STEPOI       ; RE-SEEK VIA STEP IN/OUT AND CHECK IF SEEK
                    1170      ; RETRY OK
EC57 20E0    1171      JR      NZ,PRDRTY  ; GO RETRY READ IF SEEK RETRY COUNT NOT
                    1172      ; EXHAUSTED
                    1173      ;
                    1174      ; THIS ENTRY POINT SET UNRECOVERABLE ERROR STATUS AND SETS THE OPERATION ERROR
                    1175      ; FLAG IN CASE OF AN UNRECOVERABLE ERROR WRITING OR READING.
                    1176      ;
EC59 AF      1177      PUNREC: XOR     A      ; SET UNRECOVERABLE ERROR FLAG IN CASE OF AN
EC5A 3C      1178      INC     A          ; UNRECOVERABLE ERROR WRITING OR READING
EC5B 32E4EF  1179      LD      (ERFLAG),A ; SET OPERATION ERROR FLAG
EC5E C9      1180      RET
                    1181      ;
                    1182      ; THIS SUBROUTINE REPOSITIONS THE HEAD ON THE SAME TRACK VIA A PAIR OF STEP
                    1183      ; OUT AND STEP IN COMMANDS. IT FIRST SENDS THE LAST STEP COMMAND SENT AND
                    1184      ; THEN SENDS A STEP COMMAND IN THE OPPOSITE DIRECTION.
                    1185      ;
EC5F 3AC0EF  1186      STEPOI: LD      A,(TRACK) ; GET CURRENT TRACK NUMBER %
EC62 B7      1187      OR      A          ; CHECK IF TRACK 0 %
EC63 3E5C    1188      LD      A,STEPI    ; SET COMMAND TO STEP IN IN CASE TRACK 0 %
EC65 280C    1189      JR      Z,RSEEK1   ; GO SEND STEP IN IF TRACK 0 %
EC67 3AC0EF  1190      LD      A,(TRACK)  ; GET CURRENT TRACK %
EC6A FE4C    1191      CP      76        ; CHECK IF INNER MOST TRACK (76) %
EC6C 3E7C    1192      LD      A,STEPO    ; SET COMMAND TO STEP OUT IN CASE TRACK 76 %
EC6E 2803    1193      JR      Z,RSEEK1   ; GO SEND STEP OUT IF TRACK 76 %
EC70 3AE3EF  1194      LD      A,(STEP)   ; GET LAST STEP COMMAND SENT %
                    1195      ;STEPOI: LD      A,(STEP) ; GET LAST STEP COMMAND SENT ^
                    1196      ; PUSH   BC      ; SAVE READ/WRITE AND SEEK RETRY ERROR COUNTS ^
EC73 C5      1197      RSEEK1: PUSH  BC     ; SAVE READ/WRITE AND SEEK RETRY ERROR COUNTS %
EC74 CDFCE9  1198      CALL   SCOMME      ; SEND LAST STEP COMMAND NOTE: COMMAND IS ALSO
                    1199      ; PUT INTO D
EC77 7A      1200      LD      A,D        ; LAST STEP COMMAND BACK TO A
EC78 EE20    1201      XOR     STEPDM     ; CHANGE STEP OUT/IN
EC7A CDFCE9  1202      CALL   SCOMME      ; SEND OPPOSITE STEP COMMAND
EC7D 7A      1203      LD      A,D        ; LAST STEP COMMAND BACK TO A
EC7E 32E3EF  1204      LD      (STEP),A   ; SAVE LAST STEP COMMAND
EC81 C1      1205      POP     BC         ; RESTORE READ/WRITE AND SEEK RETRY ERROR COUNT
EC82 0D      1206      DEC     C          ; DECREMENT RE-SEEK RETRY COUNT

```



```

EC83 060A      1207      LD      B,RWRTRY      ; RESET READ/WRITE RETRY ERROR COUNT TO 10
EC85 C9        1208      RET
                1209      ;
                1210      ; THIS CODE IS PERFORMED PRIOR TO GOING TO CONIN
                1211      ;
EC86 3EFF      1212      DCONI:   LD      A,DESEL      ; GET DRIVE DESELECT COMMAND
EC88 D363      1213      OUT     (SELCTP),A      ; SEND DRIVE DESELECT COMMAND
EC8A C3A4EE    1214      JP      CONIN          ; GO GET CONSOLE CHARACTER
                1215      ;
                1216      ; THIS SUBROUTINE MOVES THE DESIRED DMA ADDRESS, DRIVE, TRACK, AND SECTOR TO
                1217      ; THEIR CORRESPONDING PAGE 0 CELLS.
                1218      ;
EC8D 3AC0EF    1219      SPODTS:  LD      A,(TRACK)      ; GET CURRENTLY SELECTED TRACK NUMBER
EC90 B7         1220      OR      A              ; CHECK IF TRACK 0
EC91 2805      1221      JR      Z,SPODMA      ; GO SET DMA ADDRESS INTO PAGE 0 IF TRACK 0
                1222      ; SINCE THE READ/WRITE WILL BE DIRECTLY TO THE
                1223      ; DMA ADDRESS AND NOT BE DEBLOCKED FROM A
                1224      ; BUFFER SINCE SECTOR LENGTH IS 128 BYTES
EC93 CD6EEB    1225      CALL   GETDEN         ; GET THE DENSITY FLAG FOR THE SELECTED DRIVE
EC96 2012      1226      JR      NZ,SPOIYA     ; GO SET THE CONTENTS OF IY INTO THE PAGE 0 DMA
                1227      ; ADDRESS SINCE IY CONTAINS THE ADDRESS OF THE
                1228      ; HOST BUFFER FOR THE PHYSICAL WRITE/READ
                1229      ; FROM/TO WHICH THE DESIRED SECTOR WILL BE
                1230      ; BLOCKED/DEBLOCKED
EC98 2AE5EF    1231      SPODMA:  LD      HL,(DMADDR)     ; GET CURRENT DMA ADDRESS
EC9B 224000    1232      LD      (PODMA),HL    ; STORE IN PAGE 0 CURRENT DMA ADDRESS
EC9E 21BFEF    1233      LD      HL,DRIVE      ; GET ADDRESS OF CURRENTLY SELECTED DRIVE I. E.
                1234      ; BASE ADDRESS OF DRIVE, TRACK, AND SECTOR
ECA1 114200    1235      SPODRV:  LD      DE,P0DRV      ; GET BASE ADDRESS OF PAGE 0 DRIVE, TRACK,
                1236      ; AND SECTOR
ECA4 010400    1237      LD      BC,00004H     ; SET UP TO MOVE DRIVE, TRACK, AND SECTOR
                1238      ; TO THEIR CORRESPONDING PAGE 0 LOCATIONS
ECA7 EDB0      1239      LDIR
ECA9 C9        1240      RET
ECAA FD224000  1241      SPOIYA:  LD      (PODMA),IY     ; SET PAGE 0 DMA ADDRESS TO THE ADDRESS OF THE
                1242      ; HOST BUFFER BEING WRITTEN/READ
ECAE DDE5      1243      PUSH   IX             ; MOVE THE ADDRESS OF THE READ/WRITE HOST
ECB0 E1        1244      POP    HL             ; BUFFER CONTROL TABLE (WHICH CONTAINS DRIVE,
                1245      ; TRACK, AND SECTOR) TO HL
ECB1 18EE      1246      JR      SPODRV       ; GO MOVE DRIVE, TRACK, AND SECTOR FROM THE
                1247      ; HOST BUFFER CONTROL TABLE TO PAGE 0
                1248      ;
                1249      ; THIS BIOS SUBROUTINE READS THE PREVIOUSLY SPECIFIED SECTOR ON THE PREVIOUSLY
                1250      ; SPECIFIED TRACK
                1251      ;
ECB3 3AC0EF    1252      READ: LD      A,(TRACK) ; GET CURRENT TRACK NUMBER
ECB6 B7         1253      OR      A              ; CHECK IF TRACK, MUST READ IF SINGLE DENSITY
ECB7 CA33EC    1254      JP      Z,PREADR      ; GO READ SECTOR IF TRACK 0
ECBA CD68EB    1255      CALL   SGTDEN         ; CHECK DRIVE DENSITY FLAG
ECBD CA33EC    1256      JP      Z,PREADR      ; GO READ SECTOR IF SINGLE DENSITY
ECC0 FE40      1257      CP      040H          ; CHECK IF MAXIMUM DENSITY
ECC2 2804      1258      JR      Z,REDS8       ; GO SET UP SHIFT COUNT TO COMPUTE SECTOR
                1259      ; NUMBER CONTAINING THE DESIRED SECTOR FOR
                1260      ; MAXIMUM DENSITY (1024 BYTE SECTORS WHICH
                1261      ; CONTAIN 8 128 WORD RECORDS)
ECC4 0601      1262      LD      B,001H        ; SET UP SHIFT COUNT FOR 2 RECORDS PER SECTOR
                1263      ; DOUBLE DENSITY (256 BYTE SECTORS)
                1264      ; LD      D,B          ; SAVE SHIFT COUNT IN D ??? RESET IN RWOPER ??^
                1265      ; LD      C,007H      ; SET UP SHIFT COUNT TO RIGHT JUSTIFY RECORD ^
                1266      ; ; NUMBER IN MULTIRECORD SECTOR ??????????????????^
ECC6 1802      1267      JR      REDPSC        ; GO COMPUTE PHYSICAL SECTOR CONTAINING DESIRED
                1268      ; SECTOR
ECC8 0603      1269      REDS8:  LD      B,003H        ; SET UP SHIFT COUNT FOR 8 RECORDS PER SECTOR,
                1270      ; MAXIMUM DENSITY
                1271      ; LD      D,B          ; SAVE SHIFT COUNT IN D ??? RESET IN RWOPER ??^
                1272      ; LD      C,005H      ; SET UP SHIFT COUNT TO RIGHT JUSTIFY RECORD ^
                1273      ; ; NUMBER IN MULTIRECORD SECTOR ^
ECCA 3AC2EF    1274      REDPSC:  LD      A,(SECTOR)     ; GET CURRENT SECTOR NUMBER
ECCD 3D         1275      DEC     A              ; CORRECT FOR SECTOR STARTING AT 1, NOT 0
ECCE B7         1276      REDCPS:  OR      A              ; ZERO CARRY FLAG SINCE IT IS SHIFTED INTO A
ECCF 1F         1277      RRA
                1278      ; SHIFT SECTOR, I. E. DIVIDE TO COMPUTE
                1279      ; PHYSICAL SECTOR NUMBER-1
ECD0 10FC      1279      DJNZ   REDCPS         ; GO SHIFT AGAIN IF DIVISION IS NOT FINISHED
ECD2 3C         1280      INC     A              ; COMPUTE SECTOR NUMBER OF DESIRED 128 BYTE
                1281      ; RECORD
ECD3 DD21D1EF  1282      LD      IX,WRITCL     ; GET ADDRESS OF THE WRITE HOST CONTROL TABLE
ECD7 DDBE03    1283      CP      (IX+HSTSEC)   ; CHECK IF DESIRED SECTOR NUMBER IS THE SAME AS
                1284      ; THE SECTOR JUST WRITTEN

```

```

ECDA 2026      1285      JR      NZ,REDSR      ; GO SET UP READ CONTROL REGISTERS IF NOT
ECDC 3AC0EF    1286      LD      A,(TRACK)    ; GET CURRENT TRACK NUMBER
ECDF DDBE01    1287      CP      (IX+HSTTRK)  ; CHECK IF SAME AS THE TRACK JUST WRITTEN
ECE2 201E      1288      JR      NZ,REDSR      ; GO SET UP READ CONTROL REGISTERS IF NOT
ECE4 3ABFEF    1289      LD      A,(DRIVE)    ; GET CURRENTLY SELECTED DRIVE NUMBER
ECE7 DDBE00    1290      CP      (IX+HSTDISK) ; CHECK IF SAME AS DRIVE LAST WRITTEN TO
ECEA 2016      1291      JR      NZ,REDSR      ; GO SET UP READ CONTROL REGISTERS IF NOT
ECEC 21D1EF    1292      LD      HL,WRITCL    ; SET MOVE SOURCE TO ADDRESS OF WRITE CONTROL
                    1293      ; TABLE
ECEEF 11C3EF    1294      LD      DE,READCL    ; SET MOVE DESTINATION TO ADDRESS OF READ
                    1295      ; CONTROL TABLE
ECF2 010E00    1296      LD      BC,0000EH    ; SET NUMBER OF BYTES TO MOVE TO 14
ECF5 EDB0      1297      LDIR     ; MOVE WRITE HOST CONTROL TABLE TO READ HOST
                    1298      ; CONTROL TABLE
ECF7 21E7F3    1299      LD      HL,WRITBF    ; SET MOVE SOURCE ADDRESS TO HOST WRITE BUFFER
ECFA 11E7EF    1300      LD      DE,READBF    ; SET MOVE DESTINATION ADDRESS TO HOST READ
                    1301      ; BUFFER
ECFD 010004    1302      LD      BC,00400H    ; SET NUMBER OF BYTES TO MOVE TO 1024
ED00 EDB0      1303      LDIR     ; MOVE HOST WRITE BUFFER INTO HOST READ BUFFER
ED02 DD21C3EF    1304      REDSCR: LD      IX,READCL ; SET HOST CONTROL TABLE POINTER TO ADDRESS OF
                    1305      ; READ HOST CONTROL TABLE
ED06 FD21E7EF    1306      LD      IY,READBF    ; SET POINTER TO HOST BUFFER TO READ HOST
                    1307      ; BUFFER ADDRESS
ED0A AF        1308      XOR     A            ; SET NUMBER OF UNALLOCATED RECORDS IN
ED0B DD7707    1309      LD      (IX+UNACNT),A ; BLOCK TO 0
ED0E 3C        1310      INC    A
ED0F 32E1EF    1311      LD      (READOP),A   ; SET READ OPERATION TRUE
ED12 32E0EF    1312      LD      (RSFLAG),A   ; SET READ SECTOR FLAG
ED15 3E02      1313      LD      A,002H       ; SET WRTYPE
ED17 32E2EF    1314      LD      (WRTYPE),A   ; TO NEW BLOCK
ED1A C3C8ED    1315      JP      RWOPER       ; GO TO READ/WRITE COMMON CODE TO PROCESS
                    1316      ; THE READ/DEBLOCK OPERATION
                    1317      ;
                    1318      ; THIS BIOS SUBROUTINE WRITES THE PREVIOUSLY SPECIFIED SECTOR ON THE
                    1319      ; PREVIOUSLY SPECIFIED TRACK
                    1320      ;
ED1D 3AC0EF    1321      WRITE: LD      A,(TRACK) ; GET CURRENT TRACK NUMBER
ED20 B7        1322      OR     A            ; CHECK IF TRACK 0 (SINGLE DENSITY 128 BYTE
                    1323      ; SECTORS)
ED21 CA0BEC    1324      JP      Z,PWRITR     ; GO WRITE IF SINGLE DENSITY, SECTOR BLOCKING
                    1325      ; NOT REQUIRED
ED24 CD68EB    1326      CALL   SGTDEN        ; MOVE SELECTED DRIVE TO PAGE 0 AND GET DENSITY
ED27 CA0BEC    1327      JP      Z,PWRITR     ; GO WRITE IF SINGLE DENSITY, SECTOR BLOCKING
                    1328      ; NOT REQUIRED
ED2A DD21D1EF    1329      LD      IX,WRITCL    ; SET HOST CONTROL TABLE POINTER TO THE ADDRESS
                    1330      ; OF THE WRITE CONTROL TABLE
ED2E FD21E7F3    1331      LD      IY,WRITBF    ; SET POINTER TO HOST BUFFER TO THE ADDRESS OF
                    1332      ; THE WRITE HOST BUFFER
ED32 AF        1333      XOR     A            ; ZERO READ OPERATION STATUS,
ED33 32E1EF    1334      LD      (READOP),A   ; NOT A READ
ED36 79        1335      LD      A,C          ; GET TYPE OF WRITE, 0 = NORMAL, 1 = DIRECTORY,
                    1336      ; 2 = NEW BLOCK
ED37 32E2EF    1337      LD      (WRTYPE),A   ; SAVE TYPE OF WRITE
ED3A FE02      1338      CP      002H         ; CHECK IF WRITE TO FIRST SECTOR OF NEW BLOCK
ED3C 2026      1339      JR      NZ,CKUNA     ; GO CHECK IF UNALLOCATED SECTORS REMAINING IN
                    1340      ; IN BLOCK IF NOT NEW BLOCK
ED3E 3E10      1341      LD      A,010H       ; SET NUMBER OF SECTORS REMAINING IN BLOCK TO
ED40 DD7707    1342      LD      (IX+UNACNT),A ; 16 I. E. DOUBLE DENSITY HAS 16 CP/M SECTORS
                    1343      ; IN EACH BLOCK ALLOCATED TO A FILE
ED43 CD68EB    1344      CALL   SGTDEN        ; GET DENSITY FLAGS FOR DRIVE
ED46 FE40      1345      CP      040H         ; CHECK IF MAXIMUM DENSITY (BLOCK SIZE 32)
ED48 2805      1346      JR      Z,BS32       ; GO SET COUNT OF UNALLOCATED SECTORS IN BLOCK
                    1347      ; TO 32 IF MAXIMUM DENSITY
ED4A CDFAEB    1348      CALL   GETSID        ; CHECK IF DRIVE IS DOUBLE SIDED
ED4D 2805      1349      JR      Z,NOT32      ; DON'T CHANGE UNALLOCATED SECTOR COUNT TO 32
                    1350      ; IF DRIVE IS SINGLE SIDED
ED4F 3E20      1351      BS32: LD      A,020H  ; SET NUMBER OF SECTORS REMAINING IN BLOCK TO
ED51 DD7707    1352      LD      (IX+UNACNT),A ; 32 I. E. MAXIMUM DENSITY AND DOUBLE SIDED
                    1353      ; DOUBLE DENSITY HAVE 32 CP/M SECTORS IN EACH
                    1354      ; BLOCK ALLOCATED TO A FILE
ED54 DDE5      1355      NOT32: PUSH   IX       ; ADDRESS OF HOST WRITE
ED56 E1        1356      POP    HL            ; CONTROL TABLE TO HL
ED57 110800    1357      LD      DE,00008H    ; GET OFFSET OF NEW DRIVE, TRACK, AND SECTOR IN
                    1358      ; THE HOST WRITE CONTROL TABLE
ED5A 19        1359      ADD    HL,DE         ; COMPUTE ADDRESS OF NEW DRIVE, ETC IN HOST
                    1360      ; WRITE CONTROL TABLE
ED5B EB        1361      EX     DE,HL         ; PUT IN MOVE DESTINATION REGISTER
ED5C 21BFEF    1362      LD      HL,DRIVE     ; GET ADDRESS OF SELECTED DRIVE I. E. BASE

```

```

1363 ; ADDRESS OF DRIVE, TRACK, AND SECTOR
ED5F 010400 1364 LD BC,00004H ; MOVE 4 BYTES
ED62 EDB0 1365 LDIR ; MOVE DRIVE, TRACK, AND SECTOR INTO HOST WRITE
1366 ; CONTROL TABLE
ED64 DD3507 1367 CKUNA: DEC (IX+UNACNT) ; DECREMENT COUNT OF UNALLOCATED SECTORS
1368 ; REMAINING IN BLOCK
ED67 FAC0ED 1369 JP M,ALLOC ; GO SET COUNT TO 0 AND SET PRE-READ REQUIRED
1370 ; FLAG SINCE NOT WRITE TO UNALLOCATED BLOCK
ED6A 3ABFEF 1371 LD A,(DRIVE) ; GET CURRENT DRIVE NUMBER
ED6D DDBE08 1372 CP (IX+UNADSK) ; CHECK IF SAME AS UNALLOCATED BLOCK DRIVE
ED70 204E 1373 JR NZ,ALLOC ; GO SET COUNT TO 0 AND SET PRE-READ REQUIRED
1374 ; FLAG SINCE NOT WRITE TO UNALLOCATED BLOCK
ED72 3AC0EF 1375 LD A,(TRACK) ; GET CURRENT TRACK NUMBER
ED75 DDBE09 1376 CP (IX+UNATRK) ; CHECK IF SAME AS UNALLOCATED BLOCK TRACK
ED78 2046 1377 JR NZ,ALLOC ; GO SET COUNT OT 0 AND SET PRE-READ REQUIRED
1378 ; FLAG SINCE NOT WRITE TO UNALLOCATED BLOCK
ED7A 3AC2EF 1379 LD A,(SECTOR) ; GET CURRENT SECTOR NUMBER
ED7D DDBE0B 1380 CP (IX+UNASEC) ; CHECK IF SAME AS UNALLOCATED BLOCK SECTOR
ED80 203E 1381 JR NZ,ALLOC ; GO SET COUNT TO 0 AND SET PRE-READ REQUIRED
1382 ; FLAG SINCE NOT WRITE TO UNALLOCATED BLOCK
ED82 CD68EB 1383 CALL SGTDEN ; GET DENSITY FLAG FOR DRIVE
ED85 FE40 1384 CP 040H ; CHECK IF MAXIMUM DENSITY
ED87 280D 1385 JR Z,WRTSMD ; GO SET UP MAXIMUM DENSITY SECTORS PER TRACK
1386 ; IF MAXIMUM DENSITY
ED89 013400 1387 LD BC,00034H ; SET NUMBER OF CP/M SECTORS PER TRACK TO 52
1388 ; ASSUMING SSDD
ED8C CDFAEB 1389 CALL GETSID ; CHECK IF DRIVE IS DOUBLE SIDED
ED8F 2815 1390 JR Z,WRTDSK ; GO GET ADDRESS OF DOUBLE DENSITY SKEW TABLE
1391 ; IF NOT DOUBLE SIDED
ED91 016800 1392 LD BC,00068H ; SET NUMBER OF CP/M SECTORS PER TRACK TO 104
1393 ; IF DSDD
ED94 1810 1394 JR WRTDSK ; GO GET ADDRESS OF DOUBLE DENSITY SKEW TABLE
ED96 014800 1395 WRTSMD: LD BC,00048H ; SET NUMBER OF CP/M SECTORS PER TRACK TO 72
1396 ; ASSUMING SSMD
ED99 CDFAEB 1397 CALL GETSID ; CHECK IF DRIVE IS DOUBLE SIDED
ED9C 2803 1398 JR Z,WRTMSK ; GO GET ADDRESS OF MAXIMUM DENSITY SKEW TABLE
1399 ; IF SINGLE SIDED
ED9E 019000 1400 LD BC,00090H ; SET NUMBER OF CP/M SECTORS PER TRACK TO 144
1401 ; IF DSMD
EDA1 2149E7 1402 WRTMSK: LD HL,MDSKTB ; GET ADDRESS OF MAXIMUM DENSITY (9 1024 BYTE
1403 ; SECTORS PER TRACK) SECTOR SKEW TABLE
EDA4 1803 1404 JR WRTSSK ; GO SEARCH SKEW TABLE FOR RECORD (CP/M SECTOR)
EDA6 21E1E6 1405 WRTDSK: LD HL,DDSKTB ; GET ADDRESS OF DOUBLE DENSITY (26 256 BYTE
1406 ; SECTORS PER TRACK) SECTOR SKEW TABLE
EDA9 DD7E0B 1407 WRTSSK: LD A,(IX+UNASEC) ; GET RECORD NUMBER (CP/M SECTOR NUMBER)
EDAC EDB1 1408 CPIR ; SEARCH SKEW TABLE FOR RECORD TO SEE IF
1409 ; IT MAY BE THE LAST IN THE TRACK
EDAE 7E 1410 LD A,(HL) ; GET NEXT RECORD NUMBER
EDAF EAB7ED 1411 JP PE,WRTSRN ; GO SAVE NEXT RECORD NUMBER IF CURRENT IS NOT
1412 ; THE LAST IN THE TRACK
EDB2 3E01 1413 LD A,001H ; IF LAST IN THE TABLE, SET TO THE FIRST RECORD
1414 ; IF THE NEXT TRACK
EDB4 DD3409 1415 INC (IX+UNATRK) ; INCREMENT TRACK (UNALLOCATED BLOCK SPANS A
1416 ; TRACK)
EDB7 DD770B 1417 WRTSRN: LD (IX+UNASEC),A ; SAVE NUMBER OF NEXT SECTOR
EDBA AF 1418 XOR A ; ZERO SECTOR
EDBB 32E0EF 1419 LD (RSFLAG),A ; READ REQUIRED FLAG
EDBE 1808 1420 JR RWOPER ; GO TO READ/WRITE COMMON LOGIC TO PROCESS
1421 ; WRITE/BLOCK
EDC0 AF 1422 ALLOC: XOR A ; ZERO COUNT OF UNALLOCATED RECORDS IN BLOCK
EDC1 DD7707 1423 LD (IX+UNACNT),A ; SINCE WRITE IS NOT TO AN UNALLOCATED AREA
EDC4 3C 1424 INC A ; SET SECTOR PRE-READ REQUIRED FLAG SINCE
EDC5 32E0EF 1425 LD (RSFLAG),A ; RECORD IS IN AN ALLOCATED BLOCK
1426 ;
1427 ; THE FOLLOWING LOGIC IS COMMON TO BOTH READ AND WRITE
1428 ;
EDC8 AF 1429 RWOPER: XOR A ; SET LAST OPERATION
EDC9 32E4EF 1430 LD (ERFLAG),A ; ERROR STATUS TO 0
EDCC CD68EB 1431 CALL SGTDEN ; GET DRIVE DENSITY FLAG
EDCF FE40 1432 CP 040H ; CHECK IF MAXIMUM DENSITY
EDD1 2807 1433 JR Z,RWSMD ; GO SET UP A SHIFT FACTOR OF 3 IF MAXIMUM
1434 ; DENSITY (8 CP/M RECORDS PER SECTOR) TO
1435 ; COMPUTE SECTOR NUMBER OF RECORD
EDD3 0601 1436 LD B,001H ; SET SHIFT TO 1 (2 CP/M RECORDS PER SECTOR
1437 ; DOUBLE DENSITY) TO COMPUTE SECTOR NUMBER OF
1438 ; RECORD
EDD5 50 1439 LD D,B ; SAVE SHIFT COUNT IN D
EDD6 0E07 1440 LD C,007H ; SET UP SHIFT COUNT TO RIGHT JUSTIFY RELATIVE

```

```

1441 ; RECORD NUMBER IN SECTOR
EDD8 1805 1442 JR RWCRN ; GO COMPUTE SECTOR NUMBER CONTAINING DESIRED
1443 ; RECORD, AND THE RELATIVE RECORD NUMBER IN THE
1444 ; SECTOR
EDDA 0603 1445 RWSMD: LD B,003H ; SET SHIFT TO 3 (8 CP/M RECORDS PER SECTOR
1446 ; MAXIMUM DENSITY) TO COMPUTE SECTOR NUMBER
1447 ; CONTAINING RECORD
EDDC 50 1448 LD D,B ; SAVE SHIFT COUNT IN D
EDDD 0E05 1449 LD C,005H ; SET UP SHIFT COUNT TO RIGHT JUSTIFY RELATIVE
1450 ; RECORD NUMBER IN SECTOR
EDDF 3AC2EF 1451 RWCRN: LD A,(SECTOR) ; GET CURRENT SECTOR NUMBER
EDE2 3D 1452 DEC A ; CORRECT FOR SECTOR STARTING AT 1, NOT 0
EDE3 B7 1453 RWCSEC: OR A ; ZERO CARRY SINCE IT IS SHIFTED INTO A
EDE4 1F 1454 RRA ; SHIFT RECORD NUMBER I. E. DIVIDE TO COMPUTE
1455 ; SECTOR NUMBER-1
EDE5 10FC 1456 DJNZ RWCSEC ; GO DIVIDE BY 2 AGAIN IF SECTOR NUMBER-1 NOT
1457 ; YET COMPUTED
EDE7 3C 1458 INC A ; COMPUTE SECTOR NUMBER OF DESIRED 128 BYTE
1459 ; RECORD
EDE8 DD7704 1460 LD (IX+HSTPSC),A ; SAVE SECTOR NUMBER CONTAINING DESIRED RECORD
EDEB 42 1461 LD B,D ; GET SHIFT COUNT TO PUT RELATIVE RECORD NUMBER
1462 ; INTO UPPER OF A
EDEC 3AC2EF 1463 LD A,(SECTOR) ; GET CURRENT SECTOR NUMBER
EDEF 3D 1464 DEC A ; CORRECT FOR SECTOR STARTING AT 1 NOT 0
1465 ;RWRREC: OR A ; ZERO CARRY ?????????????????????????????????????????^
1466 ; RRCA ; PUT THE RELATIVE RECORD NUMBER WITHIN THE ^
1467 ; ; SECTOR OF THE DESIRED RECORD INTO THE UPPER ^
1468 ; ; OF A ^
EDF0 0F 1469 RWRREC: RRCA ; PUT THE RELATIVE RECORD NUMBER WITHIN THE %
1470 ; SECTOR OF THE DESIRED RECORD INTO THE UPPER %
1471 ; OF A %
EDF1 10FD 1472 DJNZ RWRREC ; GO SHIFT AGAIN IF RELATIVE RECORD NOT YET IN
1473 ; THE UPPER OF A
EDF3 B7 1474 RWRJRC: OR A ; ZERO CARRY SINCE IT IS SHIFTED INTO A FOR
1475 ; ZERO FILL
EDF4 1F 1476 RRA ; RIGHT JUSTIFY RELATIVE RECORD NUMBER IN THE
1477 ; SECTOR
EDF5 0D 1478 DEC C ; CHECK IF RELATIVE RECORD NUMBER IS RIGHT
1479 ; JUSTIFIED
EDF6 20FB 1480 JR NZ,RWRJRC ; GO SHIFT AGAIN IF NOT
EDF8 32DFEF 1481 LD (SEKHST),A ; SAVE RELATIVE RECORD NUMBER OF DESIRED RECORD
EDFB AF 1482 XOR A ; ZERO BEFORE CHECKING HOST ACTIVE %
EDFC DDB605 1483 OR (IX+HSTACT) ; CHECK HOST SECTOR ACTIVE FLAG NOT SET
1484 ; ; AND RELATIVE RECORD NUMBER=0 ????????????????????????????????????????? ^
EDFF DD360501 1485 LD (IX+HSTACT),01H ; SET HOST SECTOR ACTIVE FLAG
EE03 2837 1486 JR Z,FILHST ; GO FILL HOST BUFFER AS REQUIRED IF NOT ACTIVE
EE05 DD7E00 1487 LD A,(IX+HSTDISK) ; GET HOST DRIVE NUMBER
EE08 324200 1488 LD (PODRV),A ; SAVE DRIVE NUMBER IN PAGE 0 CELL
EE0B 21BFEF 1489 LD HL,DRIVE ; GET ADDRESS OF CURRENTLY SELECTED DRIVE
EE0E BE 1490 CP (HL) ; CHECK IF SAME AS HOST DRIVE
EE0F 2024 1491 JR NZ,NMATCH ; GO CHECK IF WRITE REQUIRED IF NOT THE SAME
EE11 3AC0EF 1492 LD A,(TRACK) ; GET CURRENT TRACK NUMBER
EE14 DDBE01 1493 CP (IX+HSTTRK) ; CHECK IF SAME AS HOST TRACK
EE17 201C 1494 JR NZ,NMATCH ; GO CHECK IF WRITE REQUIRED IF NOT
EE19 DD7E04 1495 LD A,(IX+HSTPSC) ; GET DESIRED PHYSICAL SECTOR
EE1C DDBE03 1496 CP (IX+HSTSEC) ; CHECK IF SAME AS HOST SECTOR
EE1F 2014 1497 JR NZ,NMATCH ; GO CHECK IF WRITE REQUIRED IF NOT
EE21 3AC0EF 1498 LD A,(TRACK) ; GET CURRENT TRACK NUMBER
EE24 FE02 1499 CP 002H ; CHECK IF DIRECTORY TRACK
EE26 2034 1500 JR NZ,MATCH ; GO COPY DATA TO/FROM HOST BUFFER IF NOT TK 2
EE28 DD7E04 1501 LD A,(IX+HSTPSC) ; GET DESIRED PHYSICAL SECTOR
EE2B FE01 1502 CP 001H ; CHECK IF FIRST SECTOR
EE2D 202D 1503 JR NZ,MATCH ; GO COPY DATA TO/FROM HOST BUFFER IF NOT
1504 ; SECTOR 1
EE2F 3ADFEF 1505 LD A,(SEKHST) ; GET OFFSET OF DESIRED RECORD IN MULTI-RECORD
1506 ; SECTOR
EE32 B7 1507 OR A ; CHECK IF RELATIVE RECORD 0 (FIRST RECORD)
EE33 2027 1508 JR NZ,MATCH ; SKIP PRE-WRITE/PRE-READ UNLESS FIRST RECORD
1509 ; OF DIRECTORY TRACK
EE35 DD7E06 1510 NMATCH: LD A,(IX+HSTWRT) ; GET WRITE PENDING FLAG
EE38 B7 1511 OR A ; CHECK IF WRITE IS PENDING I. E. DATA HAS BEEN
1512 ; MOVED INTO THE HOST SECTOR BUFFER BUT NOT YET
1513 ; WRITTEN TO THE DISK
EE39 C40BEC 1514 CALL NZ,PWRITR ; WRITE SECTOR FROM BUFFER BEFORE WORKING WITH
1515 ; THE DESIRED SECTOR
EE3C 3ABFEF 1516 FILHST: LD A,(DRIVE) ; GET CURRENTLY SELECTED DRIVE NUMBER
EE3F DD7700 1517 LD (IX+HSTDISK),A ; SET HOST DRIVE NUMBER IN TABLE
EE42 324200 1518 LD (PODRV),A ; STORE DRIVE NUMBER IN PAGE 0 DRIVE CELL

```

```

EE45 3AC0EF      1519      LD      A,(TRACK)      ; GET CURRENT TRACK NUMBER
EE48 DD7701      1520      LD      (IX+HSTTRK),A ; SET HOST TRACK NUMBER IN TABLE
EE4B DD7E04      1521      LD      A,(IX+HSTPSC) ; GET DESIRED PHYSICAL SECTOR
EE4E DD7703      1522      LD      (IX+HSTSEC),A ; SET AS HOST SECTOR NUMBER
EE51 3AE0EF      1523      LD      A,(RSFLAG)    ; CHECK IF SECTOR
EE54 B7            1524      OR      A              ; PRE-READ IS REQUIRED
EE55 C433EC      1525      CALL   NZ,PREADR     ; READ SPECIFIED SECTOR IF REQUIRED
EE58 AF          1526      XOR     A              ; ZERO HOST WRITE FLAG SINCE A SECTOR WAS JUST
EE59 DD7706      1527      LD      (IX+HSTWRT),A ; READ AND NO RECORDS HAVE BEEN WRITTEN TO IT
EE5C 210000      1528      MATCH: LD      HL,0000H   ; INITIALIZE OFFSET OF DESIRED RECORD IN HOST
                                1529      ; SECTOR
EE5F 118000      1530      LD      DE,00080H    ; GET SIZE OF RECORDS IN SECTOR
EE62 3ADFEF      1531      LD      A,(SEKHST)   ; GET RELATIVE RECORD NUMBER IN MULTI-RECORD
                                1532      ; SECTOR
EE65 B7            1533      OR      A              ; CHECK IF FIRST RECORD IS DESIRED RECORD
EE66 2804        1534      JR      Z,RWIMOV     ; GO SET UP TO MOVE RECORD IF FIRST RECORD
EE68 19          1535      RWCSO: ADD     HL,DE         ; BUMP TO NEXT RECORD OFFSET
EE69 3D          1536      DEC     A              ; CHECK IF OFFSET OF DESIRED RECORD COMPUTED
EE6A 20FC        1537      JR      NZ,RWCSO     ; GO BUMP RECORD OFFSET AGAIN IF NOT
EE6C FDE5        1538      RWIMOV: PUSH   IY        ; PUT SECTOR BUFFER
EE6E D1          1539      POP     DE             ; ADDRESS IN DE
EE6F 19          1540      ADD     HL,DE         ; COMPUTE ADDRESS OF DESIRED RECORD IN
                                1541      ; MULTI-RECORD SECTOR
EE70 ED5BE5EF    1542      LD      DE,(DMADDR)  ; GET CURRENT DMA ADDRESS FOR RECORD READ/WRITE
EE74 018000      1543      LD      BC,0080H    ; GET COUNT OF BYTES IN RECORD TO MOVE
EE77 3AE1EF      1544      LD      A,(READOP)   ; CHECK IF
EE7A B7            1545      OR      A              ; READING
EE7B 2006        1546      JR      NZ,RWMOVE    ; GO MOVE RECORD TO DMA ADDRESS IF READING
EE7D 3E01        1547      LD      A,001H       ; SET HOST WRITE FLAG TO INDICATE THAT A WRITE
EE7F DD7706      1548      LD      (IX+HSTWRT),A ; IS PENDING I. E. DATA HAS BEEN PUT IN THE
                                1549      ; HOST WRITE SECTOR BUFFER BUT NOT WRITTEN TO
                                1550      ; THE DISK
EE82 EB          1551      EX      DE,HL        ; SWITCH SOURCE AND DESTINATION SINCE WRITING
EE83 EDB0        1552      RWMOVE: LDIR          ; MOVE DESIRED RECORD AS REQUIRED
EE85 3AE2EF      1553      LD      A,(WRTYPE)   ; GET WRITE TYPE FLAG
EE88 FE01        1554      CP      001H         ; CHECK IF DIRECTORY WRITE
EE8A 3AE4EF      1555      LD      A,(ERFLAG)   ; GET LAST OPERATION STATUS
EE8D C0           1556      RET     NZ            ; RETURN IF NOT WRITING DIRECTORY
EE8E B7           1557      OR      A              ; CHECK IF ERROR ON PREVIOUS OPERATION
EE8F C0           1558      RET     NZ            ; RETURN IF ERROR ON PREVIOUS OPERATION
EE90 DD7706      1559      LD      (IX+HSTWRT),A ; ZERO HOST WRITE FLAG SINCE NO WRITE WILL BE
                                1560      ; BE PENDING AFTER THE FOLLOWING WRITE
EE93 CD0BEC      1561      CALL   PWRITR        ; WRITE SPECIFIED SECTOR
EE96 3AE4EF      1562      LD      A,(ERFLAG)   ; GET LAST OPERATION ERROR FLAG
EE99 C9           1563      RET
                                1564      ;
                                1565      ;
                                1566      ;
                                1567      ; DB      0,0,0,0,0,0,0,0 ; ^
                                1568      ;
                                1569      ; THIS SPACE IS RESERVED FOR A USER DEVICE INITIALIZATION SUBROUTINE. IT IS
                                1570      ; CALLED BY WBOOT BUT IS CURRENTLY NOT USED.
                                1571      ;
EE9A 00           1572      INITUD: NOP
                                1573      ; DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 ; ^
EE9B C9           1574      RET
                                1575      ;
                                1576      ; THIS BIOS SUBROUTINE GETS THE CONSOLE STATUS
                                1577      ;
EE9C DB7D        1578      CONST: IN      A,(SERST) ; READ UART STATUS PORT
EE9E E602        1579      AND     ICHRDY        ; CHECK IF INPUT CHARACTER READY
EEA0 C8          1580      RET     Z              ; RETURN IF CHARACTER IS NOT READY
EEA1 F6FF        1581      OR      OFFH          ; SET CHARACTER READY STATUS
EEA3 C9          1582      RET
                                1583      ; DB      0,0,0,0,0,0,0,0,0,0 ; ^
                                1584      ;
                                1585      ; THIS BIOS SUBROUTINE GETS A CHARACTER FROM THE CONSOLE
                                1586      ;
EEA4 DB7D        1587      CONIN: IN      A,(SERST) ; READ UART STATUS PORT
EEA6 E602        1588      AND     ICHRDY        ; CHECK IF INPUT CHARACTER IS READY
EEA8 CAA4EE      R 1589      JP      Z,CONIN      ; GO CHECK AGAIN IF NO CHARACTER READY
EEAB DB7C        1590      IN      A,(SERIN)    ; READ SERIAL INPUT PORT ON UART TO GET
                                1591      ; CHARACTER
EEAD E67F        1592      AND     07FH         ; ZERO PARITY BIT ON INPUT
EEAF C9          1593      RET
                                1594      ; DB      0,0,0,0,0,0,0,0,0,0 ; ^
                                1595      ;
                                1596      ; THIS BIOS SUBROUTINE OUTPUTS A CHARACTER TO THE CONSOLE

```

```

1597 ;
EEB0 DB7D 1598 CONOUT: IN A,(SERST) ; READ UART STATUS PORT
EEB2 E601 1599 AND OREADY ; CHECK IF UART IS READY TO ACCEPT ANOTHER
1600 ; CHARACTER TO TRANSMIT
EEB4 CAB0EE R 1601 JP Z,CONOUT ; GO CHECK STATUS AGAIN IF UART NOT READY
EEB7 79 1602 LD A,C ; CHARACTER TO TRANSMIT TO A
EEB8 D37C 1603 OUT (SEROUT),A ; OUTPUT CHARACTER TO UART FOR TRANSMISSION
EEBA C9 1604 RET
1605 ; DB 0,0,0,0,0,0,0,0,0,0 ; ^
1606 ;
1607 ; THIS BIOS SUBROUTINE OUTPUTS A CHARACTER TO THE LIST DEVICE
1608 ;
1609 ;LIST: IN A,(003H) ; ^
1610 ; AND 004H ; ^
1611 ; JP Z,LIST ; ^
1612 ; LD A,C ; ^
1613 ; OUT (002H),A ; ^
1614 ; RET ; ^
1615 ; DB 0,0,0,0,0,0,0,0,0,0 ; ^
1616 ;
EEBB 59 1617 LIST: LD E,C ; SAVE PRINTER CHARACTER IN E %
EEBC 3A0300 1618 LD A,(IOB) ; GET I/O BYTE IN CASE PRINTER REASSIGNED %
EEBF E6C0 1619 AND 0C0H ; ISOLATE LIST DEVICE ASSIGNMENT BITS %
EEC1 FEC0 1620 CP 0C0H ; CHECK IF ASSIGNED TO DIABLO
EEC3 CA60EF 1621 JP Z,DIABLO ; YES, OUTPUT TO DIABLO (XON/XOFF)
EEC6 FE80 1622 CP 080H ; CHECK IF ASSIGNED TO LPT:
EEC8 CAA4EF 1623 JP Z,TALLY ; YES,OUTPUT TO (TALLY 2000) PRINTER
EECB FE40 1624 CP 040H ; CHECK IF ASSIGNED TO CRT %
EECD 2006 1625 JR NZ,PARAL ; GO SET UP FOR PARALLEL PORT IF NOT %
EECF 0E7C 1626 LD C,SERIN ; GET SERIAL CRT INPUT PORT ADDRESS %
EED1 1601 1627 LD D,OREADY ; GET CRT OUTPUT READY STATUS MASK %
EED3 1812 1628 JR SERIAL ; GO CHECK STATUS AND SEND TO CRT %
EED5 0602 1629 PARAL: LD B,02H ; GET STROBE BYTE (BIT 0 = 0) WITH BIT 1 SET %
1630 ; SO SBC-200 MEMORY IS NOT SWITCHED IN %
EED7 CD09EF 1631 PLST: CALL LSTS1 ; CHECK IF PRINTER READY FOR CHARACTER AND %
1632 ; SET UP PARALLEL PORT ADDRESS AND STATUS MASK%
EEDA 28FB 1633 JR Z,PLST ; GO CHECK STATUS AGAIN IF BUSY %
EEDC 7B 1634 LD A,E ; PUT PRINTER CHARACTER BACK IN A %
EEDD ED79 1635 OUT (C),A ; SEND CHARACTER TO LATCHED PRINTER PORT %
EEDF 0C 1636 INC C ; CHANGE TO STATUS/STROBE PORT %
EEE0 78 1637 LD A,B ; GET STROBE BYTE (BIT 0 = 0) %
EEE1 ED79 1638 OUT (C),A ; STROBE ON TO PRINTER %
EEE3 3C 1639 INC A ; SET STROBE OFF (BIT 0 = 1) %
EEE4 ED79 1640 OUT (C),A ; STROBE OFF TO PRINTER %
EEE6 C9 1641 RET ; %
EEE7 CD46EF 1642 SERIAL: CALL LSTSB ; CHECK IF CRT READY FOR CHARACTER %
EEEA 28FB 1643 JR Z,SERIAL ; GO CHECK STATUS AGAIN IF NOT %
EEEC 7B 1644 LD A,E ; CHARACTER BACK TO A %
EEDD ED79 1645 OUT (C),A ; SEND CHARACTER TO CRT %
EEEF C9 1646 RET ; %
1647 ;
1648 ; THIS BIOS SUBROUTINE GETS THE STATUS OF THE LIST DEVICE
1649 ;
1650 ;LISTST: IN A,(003H) ; ^
1651 ; AND 004H ; ^
1652 ; RET Z ; ^
1653 ; OR 0FFH ; ^
1654 ; RET ; ^
1655 ; DB 0,0,0,0,0,0,0,0,0,0 ; ^
1656 ;
EEF0 3A0300 1657 LISTST: LD A,(IOB) ; GET I/O BYTE %
EEF3 E6C0 1658 AND 0C0H ; ISOLATE LIST DEVICE ASSIGNMENT BITS %
EEF5 FEC0 1659 CP 0C0H ; CHECK IF ASSIGNED TO DIABLO
EEF7 CA7EEF 1660 JP Z,DABLOST ; YES, CHECK ITS STATUS
EEFA FE80 1661 CP 080H ; CHECK IF ASSIGNED TO TALLY
EEFC CAAEEF 1662 JP Z,TALYST ; YES, CHECK ITS STATUS
EEFF FE40 1663 CP 040H ; CHECK IF ASSIGNED TO CRT %
EF01 2006 1664 JR NZ,LSTS1 ; GO PREPARE FOR PARALLEL PRINTER IF NOT %
EF03 0E7C 1665 LD C,SERIN ; SET PORT TO CRT INPUT PORT %
EF05 1601 1666 LD D,OREADY ; GET OUTPUT READY MASK FOR CRT PORT %
EF07 183D 1667 JR LSTSB ; GO CHECK IF READY FOR CHARACTER %
EF09 0E7E 1668 LSTS1: LD C,PDATA ; SET PORT ADDRESS TO PRINTER DATA PORT %
EF0B 1601 1669 LD D,PMASK ; GET PRINTER BUSY STATUS MASK %
EF0D D9 1670 EXX ; SAVE PORT AND STROBE BYTE %
1671 ; SAVE STATUS MASK AND PRINTER CHARACTER %
EF0E 115BEF 1672 LD DE,LSTSE ; GET LAST ADDRESS USED BY STATUS SUBROUTINE %
EF11 21FFDF 1673 LD HL,0DFFFH ; GET LAST RAM ADDRESS BEFORE SHADOW RAM %
EF14 B7 1674 OR A ; ZERO CARRY FLAG %

```



```

1753 ;
EF7E DB04 1754 DABLOST: IN A,(SIOAS) ; GET STATUS BYTE
EF80 E680 1755 AND SIOAOR ; TEST OUTPUT BIT
EF82 281F 1756 JR Z,DABLOST1 ; NOT READY...EXIT WITH A=0
EF84 3ABEEF 1757 LD A,(DBLOACT) ; LINE IS READY...ARE WE WAITING FOR XON?
EF87 2F 1758 CPL ; ONE'S COMPLEMENT
EF88 FEFF 1759 CP OFFH ; ARE WE WAITING FOR XON?
EF8A 2817 1760 JR Z,DABLOST1 ; NOT WAITING...SAY READY
EF8C DB04 1761 IN A,(SIOAS) ; WAITING, CHECK FOR HANDSHAKE.....
EF8E E640 1762 AND SIOAIR ; .....RECEIVED
EF90 2811 1763 JR Z,DABLOST1 ; NOT YET...SAY STILL BUSY
EF92 DB05 1764 IN A,(SIOAD) ; GOT SOMETHING
EF94 E67F 1765 AND 07FH ; STRIP PARITY
EF96 FE11 1766 CP 'Q'-40H ; XON?
EF98 3E00 1767 LD A,000H ; DEFAULT PRINTER NOT READY
EF9A 2007 1768 JR NZ,DABLOST1 ; NOT XON, RETURN FALSE
EF9C 3E00 1769 LD A,000H
EF9E 32BEEF 1770 LD (DBLOACT),A ; CLEAR DIABLO ACTIVE FLAG
EFA1 3EFF 1771 LD A,OFFH ; PRINTER IS READY
EFA3 C9 1772 DABLOST1: RET
1773 ;
1774 ;
1775 ; THIS SUBROUTINE OUTPUTS CHARACTERS TO THE RS232 PROTOCOL DEVICE
1776 ; ON IO-4 SERIAL PORT B.
1777 ;
EFA4 CDAEEF 1778 TALLY: CALL TALLYST ; GET IO-4 STATUS
EFA7 CAA4EF R 1779 JP Z,TALLY ; CHECK STATUS AGAIN IF IO-4 NOT READY
EFAA 79 1780 LD A,C ; CHARACTER TO TRANSMIT TO A
1781 ; CP 'a' ; CHECK IF CHARACTER < 'a'
1782 ; JR C,OUTLP ; LESS THAN 'a',NO CONVERSION REQUIRED
1783 ; CP 'z'+1 ; CHECK IF CHARACTER > 'z'
1784 ; JR NC,OUTLP ; GREATER THAN 'z', NO CONV. REQD.
1785 ; AND 05FH ; CHANGE FROM LOWER TO UPPER CASE
EFAB D307 1786 OUTLP: OUT (SIOBD),A ; OUTPUT CHARACTER TO IO-4 FOR XMIT.
EFAD C9 1787 RET
1788 ;
EFAE DB06 1789 TALLYST: IN A,(SIOBS) ; GET IO-4 STATUS
EFB0 DB06 1790 IN A,(SIOBS) ; GET IT AGAIN (DELAY)
EFB2 DB06 1791 IN A,(SIOBS) ; GET IT AGAIN (DELAY)
EFB4 E680 1792 AND SIOBOR ; CHK IF IO-4 RDY TO ACCEPT CHAR. TO XMT
EFB6 EE80 1793 XOR SIOBOR
EFB8 2002 1794 JR NZ,NOTRDY ; RETURN 0 IN A IF PRINTER BUSY
EFBA 3D 1795 DEC A ; SET A TO OFFH IF PRINTER READY
EFBE C9 1796 RET
EFBC AF 1797 NOTRDY: XOR A ; SET PRINTER BUSY STATUS
EFBD C9 1798 RET
1799 ;
EFBE 00 1800 DBLOACT: DB 0 ; SET IF DIABLO HANDSHAKE ACTIVE
1801 ;
EFBF (0001) 1802 DRIVE: DS 1 ; CURRENT DRIVE NUMBER SELECTED
EFC0 (0002) 1803 TRACK: DS 2 ; CURRENT TRACK NUMBER
EFC2 (0001) 1804 SECTOR: DS 1 ; CURRENT SECTOR NUMBER TO READ/WRITE
1805 ;
1806 ; THE FOLLOWING IS THE READ HOST CONTROL TABLE.
1807 ;
EFC3 (0001) 1808 READCL: DS 1 ; HSTDSK - HOST DRIVE NUMBER
EFC4 (0002) 1809 DS 2 ; HSTTRK - HOST TRACK NUMBER
EFC6 (0001) 1810 DS 1 ; HSTSEC - HOST SECTOR NUMBER
EFC7 (0001) 1811 DS 1 ; HSTPSC - DESIRED PHYSICAL SECTOR ON TRACK
EFC8 (0001) 1812 DS 1 ; HSTACT - HOST SECTOR ACTIVE FLAG
EFC9 (0001) 1813 DS 1 ; HSTWRT - WRITE TO HOST PENDING FLAG
EFCA (0001) 1814 DS 1 ; UNACNT - UNALLOCATED RECORD COUNT IN BLOCK
EFCB (0001) 1815 DS 1 ; UNADSK - LAST DRIVE NUMBER OF BLOCK
1816 ; CONTAINING UNALLOCATED RECORD
EFC (0002) 1817 DS 2 ; UNATRK - LAST TRACK NUMBER OF BLOCK
1818 ; CONTAINING UNALLOCATED RECORD
EFCE (0001) 1819 DS 1 ; UNASEC - LAST UNALLOCATED SECTOR NUMBER
EFCF (0002) 1820 DS 2 ; SPARES
1821 ;
1822 ; THE FOLLOWING IS THE WRITE HOST CONTROL TABLE.
1823 ;
EFD1 (0001) 1824 WRITCL: DS 1 ; HSTDSK - HOST DRIVE NUMBER
EFD2 (0002) 1825 DS 2 ; HSTTRK - HOST TRACK NUMBER
EFD4 (0001) 1826 DS 1 ; HSTSEC - HOST SECTOR NUMBER
EFD5 (0001) 1827 DS 1 ; HSTPSC - DESIRED PHYSICAL SECTOR ON TRACK
EFD6 (0001) 1828 DS 1 ; HSTACT - HOST SECTOR ACTIVE FLAG
EFD7 (0001) 1829 DS 1 ; HSTWRT - WRITE TO HOST PENDING FLAG
EFD8 (0001) 1830 DS 1 ; UNACNT - UNALLOCATED RECORD COUNT IN BLOCK

```



```

EFD9 (0001) 1831 DS 1 ; UNADSK - LAST DRIVE NUMBER OF BLOCK
1832 ; CONTAINING UNALLOCATED RECORD
EFDA (0002) 1833 DS 2 ; UNATRK - LAST TRACK NUMBER OF BLOCK
1834 ; CONTAINING UNALLOCATED RECORD
EFDC (0001) 1835 DS 1 ; UNASEC - LAST UNALLOCATED SECTOR NUMBER
EFDD (0002) 1836 DS 2 ; SPARES
EFDF (0001) 1837 SEKHST: DS 1 ; ; RELATIVE RECORD NUMBER DESIRED IN
1838 ; MULTI-RECORD SECTOR
EFE0 (0001) 1839 RSFLAG: DS 1 ; ; FLAG INDICATES SECTOR PRE-READ REQUIRED
1840 ; BEFORE OPERATION
EFE1 (0001) 1841 READOP: DS 1 ; ; READ OPERATION FLAG
EFE2 (0001) 1842 WRTYPE: DS 1 ; ; TYPE OF WRITE 0 = NORMAL, 1 = DIRECTORY,
1843 ; 2 = NEW BLOCK
EFE3 (0001) 1844 STEP: DS 1 ; ; LAST STEP COMMAND (OUT/IN)
EFE4 (0001) 1845 ERFLAG: DS 1 ; ; LAST OPERATION ERROR FLAG OFFH = ERROR
1846 ; OCCURRED ON LAST OPERATION E. G. LAST SELECT
1847 ; COMMAND 1 = UNRECOVERABLE ERROR WRITING
1848 ; OR READING
EFE5 (0002) 1849 DMADDR: DS 2 ; ; CURRENT DMA ADDRESS
EFE7 (0400) 1850 READBF: DS 0400H ; ; READ SECTOR HOST BUFFER
F3E7 (0400) 1851 WRITBF: DS 0400H ; ; WRITE SECTOR HOST BUFFER
F7E7 (0080) 1852 DIRBUF: DS 128 ; ; BDOS DIRECTORY SCRATCH PAD (BUFFER)
F867 (002C) 1853 ALV0: DS 44 ; ; DRIVE A BDOS DISK STORAGE ALLOCATION SCRATCH
F893 (0040) 1854 CSV0: DS 64 ; ; DRIVE A BDOS CHANGE DISK CHECK SCRATCH PAD
F8D3 (002C) 1855 ALV1: DS 44 ; ; DRIVE B BDOS DISK STORAGE ALLOCATION SCRATCH
F8FF (0040) 1856 CSV1: DS 64 ; ; DRIVE B BDOS CHANGE DISK CHECK SCRATCH PAD
F93F (002C) 1857 ALV2: DS 44 ; ; DRIVE C BDOS DISK STORAGE ALLOCATION SCRATCH
F96B (0040) 1858 CSV2: DS 64 ; ; DRIVE C BDOS CHANGE DISK CHECK SCRATCH PAD
F9AB (002C) 1859 ALV3: DS 44 ; ; DRIVE D BDOS DISK STORAGE ALLOCATION SCRATCH
F9D7 (0040) 1860 CSV3: DS 64 ; ; DRIVE D BDOS CHANGE DISK CHECK SCRATCH PAD
FA17 (0012) 1861 BFP: DS LSTSE-LSTS2 ; ; BUFFER TO SAVE RAM DATA WHERE PRINTER %
1862 ; SUBROUTINE IS TO BE MOVED %
FA29 (0000) 1863 END

```

```

Errors 0
Range Count 6

```

Symbol	Value	Defn	References
ALLOC	EDC0	1422	1369 1373 1377 1381
ALV0	F867	1853	0218
ALV1	F8D3	1855	0222
ALV2	F93F	1857	0226
ALV3	F9AB	1859	0230
BDOS	D806	0037	0532
BFP	FA17	1861	1678 1694
BNXTCH	E838	0420	0423
BOOT	E833	0418	0169
BS32	ED4F	1351	1346
BSTACK	0100	0110	0434
BUSYM	0001	0090	0737 0752
CBASE	9C00	0028	0036
CBDOS	0005	0167	0531 0533
CBIOS	E600	0038	0040
CBUSY1	EA12	0751	0690 0725 0753 1088
CKBSYL	E9FD	0736	0738
CKHDL	EB25	0921	0678 0713
CKUNA	ED64	1367	1339
CMDP	0064	0052	0680 0715 0744 0923 1083
CONIN	EEA4	1587	1214 1589
CONOUT	EEB0	1598	0173 0421 1601
CONST	EE9C	1578	0171
CPMB	D000	0036	0037 0038 0454 0462 0475 0498 0547
CR	000D	0042	0189 0189 0190
CSV0	F893	1854	0218
CSV1	F8FF	1856	0222
CSV2	F96B	1858	0226
CSV3	F9D7	1860	0230
DABLO2	EF7D	1748	1741 1745
DABLOST	EF7E	1754	1660 1734
DABLOST1	EFA3	1772	1756 1760 1763 1768
DATAP	0067	0051	0681 0716 0844 1084 1086 1087
DBLOACT	EFBE	1800	1747 1757 1770
DCONI	EC86	1212	0172
DDSKTB	E6E1	0254	0218 0222 0226 0230 0590 1405
DELAYI	EB86	1006	1007
DELAYO	EB84	1005	1009
DENSB	0006	0101	0775 0780 0786 0960

WTEM 00FC 0077 0722
 WTSEC 00A8 0068 0710
 X 0060 0046 0047 0048 0049 0050 0051 0052

Opcode Defn References

ADD 0490 0567 0568 0569 0570 0573 0621 0663 0992 1054 1065 1101 1359 1535 1540
 AND 0696 0737 0752 0758 0796 0802 0885 0913 0928 0937 0968 1012 1031 1060 1130 1164 1579 1588 1592
 1599 1619
 1658 1711 1740 1743 1755 1762 1765 1792
 CALL 0421 0441 0442 0486 0507 0515 0526 0535 0577 0584 0598 0604 0670 0677 0678 0705 0712 0713 0774
 0777 0806
 0807 0819 0822 0830 0848 0854 0870 0897 0902 0954 0958 0962 1020 1022 1026 1079 1082 1112 1121
 1134 1144
 1155 1169 1198 1202 1225 1255 1326 1344 1348 1383 1389 1397 1431 1514 1525 1561 1631 1642 1734
 1778
 CP 0446 0493 0509 0562 0579 0593 0803 0836 0871 0877 0886 0903 0955 0996 1047 1191 1257 1283 1287
 1290 1338
 1345 1372 1376 1380 1384 1432 1490 1493 1496 1499 1502 1554 1620 1622 1624 1659 1661 1663 1735
 1744 1759
 1766
 CPIR 1408
 CPL 0944 1758
 DB 0189 0190 0234 0235 0236 0237 0238 0239 0254 0255 0256 0257 0258 0259 0260 0261 0262 0263 0280
 0281 0282
 0283 0284 0285 0286 0287 0288 0289 0290 0291 0358 0360 0367 0370 0378 0380 0388 0391 0400 0402
 0410 0412
 1800
 DEC 0543 0550 0685 0720 1006 1008 1206 1275 1367 1452 1464 1478 1536 1710 1714 1795
 DI 1120 1154
 DJNZ 0423 0506 0857 0943 1023 1033 1133 1167 1279 1456 1472
 DS 1802 1803 1804 1808 1809 1810 1811 1812 1813 1814 1815 1817 1819 1820 1824 1825 1826 1827 1828
 1829 1830
 1831 1833 1835 1836 1837 1839 1841 1842 1844 1845 1849 1850 1851 1852 1853 1854 1855 1856 1857
 1858 1859
 1860 1861
 DW 0218 0222 0226 0230 0357 0359 0361 0366 0369 0371 0377 0379 0381 0387 0390 0392 0399 0401 0403
 0409 0411
 0413
 EI 1122 1156
 END 1863
 ENDF 0029 0032 0035
 EQU 0015 0016 0018 0020 0022 0024 0028 0036 0037 0038 0042 0043 0044 0046 0047 0048 0049 0050 0051
 0052 0054
 0055 0056 0058 0062 0063 0064 0065 0066 0067 0068 0069 0070 0072 0074 0077 0080 0083 0085 0087
 0088 0089
 0090 0092 0093 0095 0096 0097 0098 0099 0101 0103 0106 0108 0110 0112 0114 0116 0117 0119 0120
 0121 0122
 0123 0124 0125 0126 0127 0128 0129 0130 0131 0135 0136 0137 0138 0139 0140 0141 0142 0143 0144
 0145 0146
 0147 0148 0153 0154 0155 0156 0157 0158 0159 0160 0162 0164 0166 0167
 EX 0662 1361 1551 1706 1709
 EXX 1670 1687 1691 1697 1701
 HALT 0555
 IF 0027 0030 0033
 IN 0736 0739 0751 0754 0757 0795 0835 0883 0912 0927 0967 0970 1011 1057 1086 1087 1578 1587 1590
 1598 1705
 1739 1742 1754 1761 1764 1789 1790 1791
 INC 0422 0501 0502 0613 0617 0625 0798 1127 1161 1178 1280 1310 1415 1424 1458 1636 1639 1704
 INIR 0683 1085
 JP 0169 0170 0171 0172 0173 0174 0175 0176 0177 0178 0179 0180 0181 0182 0183 0184 0185 0444 0547
 0551 0554
 0859 0887 0975 1007 1088 1214 1254 1256 1315 1324 1327 1369 1411 1589 1601 1621 1623 1660 1662
 1690 1715
 1717 1779
 JR 0447 0456 0487 0494 0510 0518 0580 0585 0588 0594 0599 0602 0605 0686 0690 0721 0725 0738 0753
 0779 0785
 0800 0804 0809 0817 0820 0827 0837 0839 0851 0868 0873 0878 0879 0898 0901 0904 0906 0941 0966
 0969 0972
 1009 1029 1032 1038 1048 1061 1131 1136 1138 1165 1171 1189 1193 1221 1226 1246 1258 1267 1285
 1288 1291
 1339 1346 1349 1373 1377 1381 1385 1390 1394 1398 1404 1420 1433 1442 1480 1486 1491 1494 1497
 1500 1503
 1508 1534 1537 1546 1625 1628 1633 1643 1664 1667 1676 1713 1736 1741 1745 1756 1760 1763 1768
 1794
 LD 0418 0419 0420 0424 0425 0426 0427 0434 0435 0436 0437 0438 0439 0440 0449 0450 0452 0454 0457
 0458 0460
 0462 0464 0466 0468 0469 0470 0471 0472 0475 0476 0477 0478 0479 0481 0483 0488 0489 0491 0492
 0498 0500

0544 0545	0503 0513 0514 0521 0523 0527 0528 0529 0530 0531 0532 0533 0534 0537 0538 0539 0540 0541 0542
0623 0627	0546 0559 0561 0564 0565 0566 0571 0581 0583 0587 0590 0592 0595 0597 0601 0607 0611 0615 0619
0710 0711	0636 0637 0641 0642 0648 0649 0654 0660 0664 0665 0675 0676 0681 0682 0684 0687 0695 0697 0700
0801 0810	0716 0717 0719 0722 0730 0735 0743 0763 0764 0765 0766 0773 0776 0778 0781 0782 0784 0794 0797
0858 0861	0814 0816 0821 0824 0825 0826 0828 0831 0833 0834 0838 0840 0841 0842 0843 0845 0846 0852 0856
0908 0921	0862 0863 0864 0865 0867 0872 0876 0880 0882 0884 0888 0891 0892 0894 0895 0896 0899 0905 0907
1030 1034	0922 0929 0939 0940 0957 0973 0980 0981 0988 0989 0990 0991 0993 1004 1005 1018 1019 1021 1024
1096 1098	1040 1046 1049 1050 1051 1052 1053 1055 1056 1062 1063 1067 1068 1074 1075 1076 1080 1081 1084
1192 1194	1099 1100 1103 1114 1118 1119 1125 1128 1129 1150 1152 1153 1159 1162 1163 1179 1186 1188 1190
1292 1294	1200 1203 1204 1207 1212 1219 1231 1232 1233 1235 1237 1241 1252 1262 1269 1274 1282 1286 1289
1351 1352	1296 1299 1300 1302 1304 1306 1309 1311 1312 1313 1314 1321 1329 1331 1334 1335 1337 1341 1342
1430 1436	1357 1362 1364 1371 1375 1379 1387 1392 1395 1400 1402 1405 1407 1410 1413 1417 1419 1423 1425
1510 1516	1439 1440 1445 1448 1449 1451 1460 1461 1463 1481 1485 1487 1488 1489 1492 1495 1498 1501 1505
1562 1602	1517 1518 1519 1520 1521 1522 1523 1527 1528 1530 1531 1542 1543 1544 1547 1548 1553 1555 1559
1684 1685	1617 1618 1626 1627 1629 1634 1637 1644 1657 1665 1666 1668 1669 1672 1673 1677 1678 1680 1683
LDIR	1689 1693 1694 1695 1703 1707 1723 1737 1746 1747 1757 1767 1769 1770 1771 1780
NOP	0428 0524 1239 1297 1303 1365 1552 1681 1686 1696
OR	1572
1511 1524	0741 0755 0767 0783 0866 0900 0931 0938 0971 1106 1187 1220 1253 1276 1322 1453 1474 1483 1507
ORG	1533 1545 1557 1581 1674 1699
OTIR	0040
OUT	0718
1640 1645	0680 0715 0742 0744 0756 0787 0815 0829 0844 0890 0893 0914 0923 1003 1083 1213 1603 1635 1638
POP	1708 1738 1786
1157 1205	0482 0504 0516 0519 0548 0608 0629 0746 0748 0750 0850 0855 0926 0994 0995 1028 1104 1105 1123
PUSH	1244 1356 1539
1151 1197	0473 0484 0511 0512 0552 0575 0609 0745 0747 0749 0847 0853 0925 0986 0987 1025 1094 1095 1116
RES	1243 1355 1538
RET	0775 0960 0961
1013 1069	0563 0631 0643 0650 0655 0666 0674 0698 0701 0709 0759 0768 0788 0916 0930 0932 0946 0956 0998
1728 1748	1107 1124 1158 1180 1208 1240 1556 1558 1563 1574 1580 1582 1593 1604 1641 1646 1700 1719 1724
RLCA	1772 1787 1796 1798
RRA	0942
RRCA	0799 1277 1454 1476
SBC	1469
SET	1675
SUB	0699 0780 0786
TITLE	0881
XOR	0001
1797	0480 0495 0536 0635 0793 0889 0915 1177 1201 1308 1333 1418 1422 1429 1482 1526 1712 1716 1793