

APPENDIX E

SAMPLE MP/N II BANKED XIOS

page 0

TITLE 'XIOS200, Copyright 1980, ALTOS COMPUTER SY

ALTOS COMPUTER SYSTEMS
2360 BERING DRIVE
SAN JOSE, CALIFORNIA 95131

Copyright 1980, ALTOS COMPUTER SYSTEMS

This program is a copyright program product of ALTOS COMPUTER SYSTEMS and is distributed to the owners of ALTOS SUN SERIES 8000 computers for use on those systems only. Any other use of this software constitutes a breach of the copyright license to the purchaser. However, permission is granted to use this listing as a sample for the construction of the reader's own XIOS.

VERSION NUMBER: 1.12*
VERSION DATE: June 28, 1980
 Add support for CP/M version 2.0
 Add support for Hard disk drives
 Add support for disk MODE selection
 Provide compatability MODE for 1.4 operatio
 Remove CTC/1791 counter reset
 CORRECT HARD DISK SEEK PROBLEM
 Add code to recover from WD1791 going to sl
 Initialize parallel port for Centronics pri
VERSION DATE: March 17, 1981
 Virtual disk in banks 1,2,3: M DISK
VERSION DATE: April 11, 1981
 Conditional assembly for virtual disks
 Conditional assembly for MP/M 2.0
VERSION DATE: April 14, 1981
 Equates added for LDRBIOS hooks
VERSION DATE: April 16, 1981
 Testing for bank setup added

Mode 0 IBM single density
 1 ALTOS double density Version 2.0
 2 ALTOS double density Version 1.4
 3 ALTOS hard disk Version 2.0 (8 MEG
 4 ALTOS HARD DISK VERSION 2.0 (8 MEG
 5 ALTOS HARD DISK VERSION 2.0 (8 MEG
 6 ALTOS HARD DISK VERSION 2.0 (4 MEG

ASSEMBLER CONTROL STATEMENTS

MACLIB DISKDEF
MACLIB Z80S

```

FFFF = TRUE EQU OFFFHH ;VALUE FOR TRUE
0000 = FALSE EQU NOT TRUE ;VALUE FOR FALSE
0000 = mdisk equ false ;Virtual Disk cond asm bool
FFFF = mpm20 equ true ;MP/M 2.0 cond asm boolean

; -----

1700 = ldrbiosbase equ 1700h ; for M
0037 = density$mask$offset equ 37h ;density mask offset from LDRBI
00BB = misc$params$offset equ 0bbh ;misc. parameters offset from L

```

THE FOLLOWING EQUATES ARE USER MODIFIABLE BASED ON
PARTICULAR USER SYSTEM AND OPTIONS SELECTED.

```

FFFF = DMA EQU TRUE ;DMA HARDWARE SUPPORT ??
FFFF = HARDSK EQU TRUE ;HARD DISK SUPPORT

```

```

; -----

```

THE FOLLOWING CONSTANTS APPLY TO THE DEBLOCKING OF
SECTORS LARGER THAN 128 FOR THE ALTOS DOUBLE DENSIT
AND THE ALTOS HARD DISK.

```

;-----

4000 = BLKSIZ EQU 16384 ;CP/M ALLOCATION SIZE
0400 = HSTSIZ EQU 1024 ;HOST DISK SECTOR SIZE
0010 = HSTSPT EQU 16 ;HOST DISK SECTORS PER TRAC
0008 = HSTBLK EQU HSTSIZ/128 ;CP/M SECTORS PER HOST BUFF
0080 = CPMSPT EQU HSTBLK HSTSPT ;CP/M SECTORS PER TRACK
0007 = SECMSK EQU HSTBLK 1 ;SECTOR MASK
0003 = SECSHF EQU 3 ;LOG2(HHSTBLK)
;-----

```

; THE FOLLOWING EQUATES APPLY TO THE RELOCATABILITY
; OF THE CBIOS AND SHOULD NOT BE USER ALTERED.

```

FFFF RELOC EQU TRUE ;RELOCATABLE VERSION ??

if mdisk
maxdisk equ 13
else
IF HARDSK
000C MAXDSK EQU 12 ;MAXIMUM NUMBER OF LOGICAL
ELSE
MAXDSK EQU 4 ;MAXIMUM NUMBER OF LOGICAL
ENDIF
endif
IF RELOC
0000 ORG 0000H
ELSE
ORG 0C000H
ENDIF
0000 BASE EQU $

0000 = WRALL EQU 0 ; WRITE TO ALLOCATED
0001 = WRDIR EQU 1 ; WRITE TO DIRECTORY

```

0002 =	WRUAL	EQU	2	; WRITE TO UNALLOCATED
0004 =	NMBCNS	EQU	4	; NUMBER OF CONSOLES
0083 =	POLL	EQU	131	; XDOS POLL FUNCTION
0084 =	FLAGWT	EQU	132	; XDOS FLAG WAIT FUNCTION
0085 =	FLAGST	EQU	133	; XDOS FLAG SET FUNCTION
0005 =	HDFLAG	EQU	5	;HARD DISK FLAG FOR WAIT & SET
0006 =	FPYFLAG	EQU	6	;FLOPPY DISK FLAG FOR WAIT & SET
0000 =	PLLPT	EQU	0	; POLL PRINTER
0001 =	PLC00	EQU	PLLPT+1	; POLL CONSOLE OUT #0 (CRT:)
0002 =	PLC01	EQU	PLC00+1	; POLL CONSOLE OUT #1 (CRT:)
0003 =	PLC02	EQU	PLC01+1	; POLL CONSOLE OUT #2 (CRT:)
0004 =	PLC03	EQU	PLC02+1	; POLL CONSOLE OUT #3 (CRT:)
0005 =	PLCIO	EQU	PLC03+1	; POLL CONSOLE IN #0 (CRT:)
0006 =	PLCI1	EQU	PLCIO+1	; POLL CONSOLE IN #1 (CRT:)
0007 =	PLC12	EQU	PLCI1+1	; POLL CONSOLE IN #2 (CRT:)
0008 =	PLC13	EQU	PLC12+1	; POLL CONSOLE IN #3 (CRT:)
0009 =	MEMPORT	EQU	009H	; MEMORY SELECT PORT
0002 =	MEMSK	EQU	002H	; MEMORY SELECT MASK

```
; JUMP VECTORS FOR ENTRIES TO CBIOS ROUTINES
```

```
; EXTERNAL JUMP TABLE (BELOW XIOS BASE)
```

```
;PDISP EQU      $-3
;XDOS EQU       PDISP-3
                if mpm20
0000 C3040B      jmp      commonbase
                else
                jmp      COLDSTART          ;COLD START
                endif

                WBOTE:
0003 C3150B      jmp      WARMSTART          ;WARM START
0006 C3790B      jmp      CONST              ;CONSOLE STATUS
0009 C3840B      jmp      CONIN              ;CONSOLE CHARACTER IN
000C C38F0B      jmp      CONOUT             ;CONSOLE CHARACTER OUT
000F C3A90C      jmp      LIST               ;LIST CHARACTER OUT - TI
                                ;"CLIST" IF SETUP PROI
                                ; PARALLEL PRINTER POR'
0012 C31A0B      jmp      RTNEMPTY           ;PUNCH NOT IMPLEMENTED
0015 C31A0B      jmp      RTNEMPTY           ;READER NOT IMPLEMENTED
0018 C3F902      jmp      HOMEIT             ;MOVE HEAD TO HOME
001B C30302      jmp      SELDSK             ;SELECT DISK
001E C36D02      jmp      SETTRK             ;SET TRACK NUMBER
0021 C37302      jmp      SETSEC             ;SET SECTOR NUMBER
0024 C3550       jmp      SETDMA             ;SET DMA ADDRESS
0027 C38B02      jmp      READ               ;READ DISK
002A C39602      jmp      WRITE              ;WRITE DISK
002D C3BC0C      jmp      POLLPT             ;LIST STATUS
0030 C3D605      jmp      SECTTRAN           ;SECTOR TRANSLATE
```

```
EXTENDED 1/0 SYSTEM JUMP VECTOR
```

```
0033            SELMEMORY          SELECT MEMORY
0036 C3CB0C      jmp      POLLDEVICE         ;POLL DEVICE
0039 C3050D      jmp      STARTCLOCK         ;START CLOCK
003C C30B0D      jmp      STOPCLOCK          ;STOP CLOCK
003F C3100D      jmp      EXITREGION         ;EXIT REGION
0042 C3170D      jmp      MAXCONSOLE         ;MAXIMUM CONSOLE NUMBE

0045 C39D12      jmp      SYSTEMINIT         ;SYSTEM INITIALIZATION
0048 00          NOP                      ;NO JMP HERE
0049 00          NOP                      ;FOR MP/M DELAY
004A 00          NOP
004B C3A102      jmp      SETMOD             ;ROUTINE TO SET DISK MODE
004E C3EE02      jmp      RETMOD             ;ROUTINE TO RETURN CURRENT
                if not mpm20
                COLDSTART:
                WARMSTART:
                        MVI C, 0            ;SEE SYSTEM INIT
                                                ;COLD & WARM START INCLUDE
                                                ;FOR COMPATIBILITY WITH CP
                                                ;SYSTEM RESET, TERMINATE P
                RTNEMPTY:      jmp      XDOS
                                XRA      A            ;NOT USED
                                RET
```

endif

LAST:

005E ORG (((LAST-BASE)+0A2H) AND 0FF00H) +05EH

INTERUPT:

```
005E 470B      DW      FLOPPY$INT      ;FLOPPY DISK INTERR
0060 1COB      DW      NULL$INT
0062 1COB      DW      NULL$INT
0064 1COB      DW      NULL$INT
0066 1A0D      DW      INTLHND      ;CTC INTERRUPT
0068 1COB      DW      NULL$INT
006A 5E0B      DW      HARD$INT      ;HARD DISK INTERRUPT
006C 1COB      DW      NULL$INT
006E 1COB      DW      NULL$INT
         if not mpm20
```

NULL\$INT:

```
EI
RETI
endif
PAGE
```

; WORK AND CONTROL AREAS FOR CBIOS SERVICES

```
0070 FFFFFFFF TRK0: DB      0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FF
007C 0408102010SELO: DB      004H,008H,010H,020H,010H,010H,010H,020H,020
0088 00000000003MODE: DB      000H,000H,000H,000H,003H,004H,005H,003H,004
0094 00000000000TCNT: DB      000H,000H,000H,000H,000H,000H,000H,000H,000
00A0 00000000000PCNT: DB      000H,000H,000H,000H,000H,000H,000H,000H,000
```

```
0,0AC00 DISKNO: DB      000H      ;CURRENT DR
HAD 00 TRAKNO: DB      000H      ;CURRENT TR
00AE 00 HEADNO: DB      000H      ;CURRENT HE
00AF 0000 DMAADR: DW      000H      ;CURRENT DM
00B1 00 SECTNO: DB      000H      ;CURRENT SE
00B2 0000 DPEPTR: DW      000H      ;CURRENT DP
00B4 0000 DBLKAD: DW      000H      ;CURRENT EX
00B6 0000 MPARMS: DW      000H      ;MISC. PARA
00B8 10 HTK1: DB      10H      ;HARD DISK
00B9 20 HTK2: DB      20H      ;HARDDISK
```

PARAMETER FLAGS

```
0100H = DOUBLE HEADED DRIVES
0200H = CENTRONICS PRINTER FOR LIST DEVICE
0400H = FOUR DRIVE SYSTEM [ A B C D ]
```

NOTE:

NO CHANGES ARE TO BE MADE TO THE ABSOLUTE LOCATIONS
ANY FIELDS PRIOR TO THIS POINT. EXTERNAL PROGRAMS A
DEPENDENT UPON THE LOCATION OF THE PRECEEDING DATA.

```

                IF NOT DMA
NMIRTN:  DB      0EDH,0A2H,0EDH,045H ;FAKE INI A
ENDIF

00BA C37D DMAS1:  DB      0C3H,07DH      ;FIRST PART
00BC 0000 DMASA:  DW      000H           ;ADDRESS FO
00BE 0004 DMALEN: DW      1025-1         ;LENGTH FOR
00C0 54CE68CEA5DMAS2H: DB  054H,0CEH,068H,0CEH,0A5H,020H ;HARD DISK

00C6 14288507 DMAS2F: DB  014H,028H,085H,007H ;FLOPPY DIS
00CA 8ACF01CF DMAS3:  DB  08AH,0CFH,001H,0CFH ;LAST PART
00CE 01  DMAS3F:  DB  001H           ;001=READ,
00CF CF87      DB      0CFH,087H      ;SETUP DMA,

; CONTROL BLOCKS FOR DISK DRIVER

00D1 =  DPBASE  EQU      $              ;START OF DISK PARAMETER BL

00D1 B5010000 DPE0:DW      XLTO,0000H    ;TRANSLATE TABLE AND WORK A
00D5 00000000      DW      0000H,0000H    ;SCRATCH AREA
00D9 9D12D40D      DW      DIRBUF,DPBO    ;DIR BUFF, PARM BLOCK
00DD 3E081E08      DW      CSV0,ALV0      ;CHECK VECTOR, ALLOC VECTOR
00E1 B5010000 DPE1:DW      XLTO,0000H    ;TRANSLATE TABLE AND WORK A
00E5 00000000      DW      0000H,0000H    ;SCRATCH AREA
00E9 9D12D40D      DW      DIRBUF,DPBO    ;DIR BUFF, PARM BLOCK
00ED 7E085E08      DW      CSV1,ALV1      ;CHECK VECTOR, ALLOC VECTOR
00F1 B5010000 DPE2:DW      XLTO,0000H    ;TRANSLATE TABLE AND WORK A
00F5 00000000      DW      0000H,0000H    ;SCRATCH AREA
00F9 9D12D40D      DW      DIRBUF,DPBO    ;DIR BUFF, PARM BLOCK
00FD BE089E08      DW      CSV2,ALV2      ;CHECK VECTOR, ALLOC VECTOR

0101 B5010000 DPE3:DW      XLTO,0000H    ;TRANSLATE TABLE AND WORK A
0105 00000000      DW      0000H,0000H    ;SCRATCH AREA
0109 9D12D40D      DW      DIRBUF,DPBO    ;DIR BUFF, PARM BLOCK
010D FE08DE08      DW      CSV3,ALV3      ;CHECK VECTOR, ALLOC VECTOR
                IF      HARDSK

0111 00000000 DPE4:DW      0000H,0000H    ;TRANSLATE TABLE AND WORK A
0115 00000000      DW      0000H,0000H    ;SCRATCH AREA
0119 9D12010E      DW      DIRBUF,DPB3    ;DIR BUFF, PARM BLOCK
011D 5E091E09      DW      CSV4,ALV4      ;CHECK VECTOR, ALLOC VECTOR
0121 00000000 DPE5:DW      0000H,0000H    ;TRANSLATE TABLE AND WORK A
0125 00000000      DW      0000H,0000H    ;SCRATCH AREA
0129 9D12100E      DW      DIRBUF,DPB4    ;DIR BUFF, PARM BLOCK
012D 9E095E09      DW      CSV5,ALV5      ;CHECK VECTOR, ALLOC VECTOR
0131 00000000 DPE6:DW      0000H,0000H    ;TRANSLATE TABLE AND WORK A
0135 00000000      DW      0000H,0000H    ;SCRATCH AREA
0139 9D121F0E      DW      DIRBUF,DPB5    ;DIR BUFF, PARM BLOCK
013D DE099E09      DW      CSV6,ALV6      ;CHECK VECTOR, ALLOC VECTOR
0141 00000000 DPE7:DW      0000H,0000H    ;TRANSLATE TABLE AND WORK A
0145 00000000      DW      0000H,0000H    ;SCRATCH AREA
0149 9D12010E      DW      DIRBUF,DPB3    ;DIR BUFF, PARM BLOCK
014D 1E0ADE09      DW      CSV7,ALV7      ;CHECK VECTOR, ALLOC VECTOR
0151 00000000 DPE8:DW      0000H,0000H    ;TRANSLATE TABLE AND WORK A
0155 00000000      DW      0000H,0000H    ;SCRATCH AREA

```

```

0159 9D12100E    DW    DIRBUF,DPB4    ;DIR BUFF, PARM BLOCK
015D 5E0A1E0A    DW    CSV8,ALV8     ;CHECK VECTOR, ALLOC VECTOR

0161 00000000 DPE9:DW    0000H,0000H    ;TRANSLATE TABLE AND WORK A
0165 00000000    DW    0000H,0000H    ;SCRATCH AREA
0169 9D121F0E    DW    DIRBUF,DPB5    ;DIR BUFF, PARM BLOCK
016D 9E0A5E0A    DW    CSV9,ALV9     ;CHECK VECTOR, ALLOC VECTOR

```

```

0171 00000000 DPEA:DW    0000H,0000H    ;TRANSLATE TABLE AND WORK A
0175 00000000    DW    0000H,0000H    ;SCRATCH AREA
0179 9D122E0E    DW    DIRBUF,DPB6    ;DIR BUFF, PARM BLOCK
017D C20A9E0A    DW    CSVA,ALVA     ;CHECK VECTOR, ALLOC VECTOR
0181 00000000 DPEB:DW    0000H,0000H    ;TRANSLATE TABLE AND WORK A
0185 00000000    DW    0000H,0000H    ;SCRATCH AREA
0189 9D122E0E    DW    DIRBUF,DPB6    ;DIR BUFF, PARM BLOCK
018D E60AC20A    DW    CSVB,ALVB     ;CHECK VECTOR, ALLOC VECTOR

```

```

        ENDIF

```

```

        if mdisk

```

```

;Virtual disk parameter header

```

```

DPEC:    DW    0000H,0000H    ;TRANSLATE TABLE AND WORK A
         DW    0000H,0000H    ;SCRATCH AREA
         DW    DIRBUF,DPB7    ;DIR BUFF, PARM BLOCK
         DW    CSVC,ALVC     ;CHECK VECTOR, ALLOC VECTOR
        endif

```

```

-----

0191 B5010000 MODL0:DW    XLT0,000H    ;MODEL DPE FOR MODE 0
0195 00000000    DW    000H,000H
0199 9D12D40D    DW    DIRBUF,DPB0
019D CF010000 MODL1:DW    XLT1,0000H    ;MODEL DPE FOR MODE 1
01A1 00000000    DW    0000H,0000H
01A5 9D12E30D    DW    DIRBUF,DPB1
01A9 CF010000 MODL2:DW    XLT2,0000H    ;MODEL DPE FOR MODE 2
01AD 00000000    DW    0000H,0000H
01B1 9D12F20D    DW    DIRBUF,DPB2

```

```

; -----

```

```

01B5 01070D1319 XLT0: DB    1, 7, 13, 19, 25, 5, 11, 17, 23, 3, 9, 15, 21
01C2 02080E141A DB    2, 8, 14, 20, 26, 6, 12, 18, 24, 4, 10, 16, 22
        XLT1:
        XLT2:
01CF 0102030405 DB    01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13
01DC 0E0F101112 DB    14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26
01E9 1B1C1D1E1F DB    27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
01F6 28292A2B2C DB    40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52

```

```

; DISK ACCESS ROUTINES

```

```

; -----

```

```

        SELDSK:

```

```

0203 79      MOV      A,C          ;LIMIT SELECT TO REAL OPTIO
0204          FEOC      CPI MAXDSK
          JRNC      SELERR        ;INVALID DRIVE
0206+303A    DB        030H,SELERR-$-1 ;---- FAKE JRNC INSTRUCTION
          Mov      A,E          ;TEST FOR INITIAL SELECT
          ANI      1            ;E = 0 IS FIRST TIME
          PUSH     PSW
0208 1600    MVI      D,0
020A 59      MOV      E,C          ;TRANSLATE TABLE
020B 214602  LXI      H,DTBLT     ;FOR LOGICAL TO PHYSICAL
020E 19      DAD      D
020F 4E      MOV      C,M          ;C = PHYSICAL DRIVE
0210 79      MOV      A,C          ;M translates to the 12 di
          if      mdisk
          CPI      12
          jz      VIRTUAL
          endif

          POP      PSW          ;RESTORE TEST
          JRNZ     SELSDP       ;BYPASS SELECT

      SETDSK:
0211 0600    MVI      B,0
0213 217C00  LXI      H,SELO      ;BASE OF SELECT MASKS
0216 09      DAD      B
0217 7E      MOV      A,M          ;GET SELECT BYTE
0218 A7      ANA      A          ;CHECK FOR VALID DRIVE
          JRZ      SELERR        ;DRIVE NOT CONFIGURED
          ;FAKE JRZ INSTRUCTION
0219+2827    DB        028H,SELERR-$-1
021B 79      MOV      A,C
021C FE04    CPI      4          ;CHECK FOR FLOPPY
          JRC      SELSDP
021E+380F    DB        038H,SELSDP-$-1 ; FAKE JRC INSTRUCTION
0220 7E  CHKHRD: MOV    A,M          ;RESTORE SELECT BYTE
0221 D320    OUT     20H
0223 C5      PUSH     B
0224 0E01    MVI     C,1          ;DELAY FOR 1 MS
0226 CD8207  CALL    DELAY
0229 C1      POP      B
022A DB24    IN       24H          ;CHECKFOR HARD DISK REAr.,Y
022C 17      RAL          ;80H = READY
          JRNC     SELERR
022D+3013    DB        030H,SELERR-$-1 ;----FAKE JRNC INSTRUCTION
      SELSDP:
022F 79      MOV      A,C
          if      mdisk
      VIRTUAL:
          endif

0230 32E60A  STA      NEWDSK      ;SAVE FOR 1/0 LATER
0233 2600    MVI      H,0
0235 69      MOV      L,C          ; COMPUTE DP HEADER ADDRESS
0236 29      DAD      H          ; 2
0237 29      DAD      H          ; 4
0238 29      DAD      H          ; 8
0239 29      DAD      H          ; 16 (DP HEADER SIZE)
023A 11D100  LXI      D,DPBASE    ;START OF DP HEADERS
023D 19      DAD      D          ;POINT TO CORRECT ONE

```



```

023E 22B200      SHLD      DPEPTR      ;SAVE ADDRESS OF CURRENT DP
0241 C9          RET
0242 210000 SELERR:LXI      H,0          ; INDICATE ERROR
0245 C9          RET                  ; AND RETURN

```

A, B, C, D, E, F, G, H, I, J, K, L, M, N, O

```

0246 0001020304D TBLT: DB 0,1,2,3,4,5,6,7,8,9,10,11,12,0,0

```

SETDMA:

```

0255 60          Mov       H,B          ;TO ALLOW SAVING
0256 69          Mov       L,C
0257 22A         FOOSHL      DMAADR
                if        mpm20
025A 23          inx        h            ;test for flush buffers
025B 7D          mov       a,l
025C B4          ora       h
025D C0          rnz              ;HL = FFFFh is flush buffer
025E 21F00A      lxi       h,hstwrt
0261 7E          mov       a,m
0262 3600        mvi       m,0
0264 B7          ora       a
0265 C8          rz
0266 CD6D04      call writehst          ;flush host write if pendin
0269 B7          ora       a
026A C8          rz              ;return if no error
026B E1          POP       h
                endif
026C C9          ret

```

SETTRK:

```

026D 60          Mov       H,B          ;TO ALLOW SAVE
026E 69          MOV       L,C
026F 22E70A      SHLD      NEWTRK      ;SAVE NEXT TRACK NUMBER
0272 C9          RET                  ;RETURN TO CALLER

```

SETSEC:

```

0273 79          MOV       A,C          ;FOR SAVE
0274 32E90A STA  NEWSEC
0277 C9          RET                  ;RETURN TO CALLER

```

SETDEN:

```

0278 117C00      LXI       D,SELO      ;START OF SELECT/DENSITY MA
027B 2AE60A      LHL      NEWDSK      ;NEXT DRIVE ADDRESS
027E 2600        MVI       H,000H     ;ENSURE ZERO FOR SINGLE BYT

0280 19          DAD       D          ;POINT TO CORRECT MASK
0281 79          MOV       A,C          ;ISOLATE DENSITY BIT
0282 E601        ANI       00000001B
0284 4F          MOV       C,A          ;SAVE FOR NOW
0285 7E          MOV       A,M          ;LOAD SELECT DENSITY MASK
0286 ME          ANI       11111110B   ;RESET CURRENT DENSITY SETT
0288 B1          ORA       C          ;SET NEW VALUE
0289 77          MOV       M,A          ;RESTORE MASK IN TABLE
028A C9          RET                  ;RETURN TO CALLER

```

```

                if        mdisk
MREADSECTOR:

```

```

        call    compbank          ;compute bank
        di
        call    chgbank
        lxi     b,128
        lxi     d,localbuf
        lhld    addroff
        ldir
                                ;block move into the dma ar
        mvi     a,02h              ;select bank 0
        out     09h
        ei
        lxi     b,128
        lhld    dmaadr
        xchg
        lxi     h,localbuf
        ldir
        xra     a
        ret

mbankno db      0
addroff dw      0
localbuf ds     128

compbank:
        lda     newtrk
        mov     h,a
        ani     0fh                ;save track rem 16
        mov     l,a
        mov     a,h                ;restore track
        mvi     h,0
        ani     0f0h              ;bank is high order nibble
        rar     !rar ! Rar ! rar
        inr     a
        sta     mbankno           ; which bank we want
        dad     h                  ;trk 0-15
        dad     h                  ; 2
        dad     h                  ; 4
        mov     e,l
        mov     d,h
        dad     d
        dad     d                  ; 24:
        lda     newsec            ; figure offset within the

        mov     e,a
        mvi     d,0
        dad     d                  ; add sector offset within
        dad h ! dad h 1 dad h ! dad h ! dad h ! dad h
        shld    addroff           (track 24 + sector)    1
        ret
    endif

READ:
        if mdisk
        LDA     NEWDSK
        CPI     12                  ;VIRTUAL DISK ?
        iz      MREADSECTOR
        endif

```

```

028B CDEE02      CALL    RETMOD          ;WHAT TYPE OF 1/0 ??
028E FE03        CPI      003H
0290 DAE405      ic       READSOFT       ;FLOPPY DISK DRIVE
0293 C36B03      jmp      READHARD       ;HARD DISK 1/0
              if mdisk
mwritesector:
              call    compbank
              lhld    dmaadr
              lxi     d,localbuf
              lxi     b,128
              ldir
              di
              call    chgbank
              lxi     d,localbuf
              lxi     b,128
              lhld    addroff
              xchg
              ldir
              mvi     a,02h              ; select bank 0
              out     09h
              ei
              xra      a
              ret
chgbank:
              lda     mbankno
              ral
              ral
              ral
              ani     018h
              ori     memsk
              out     009h
              ret
endif
WRITE:
              if      mdisk
              lda     newdsk
              cpi     12
              jz      mwritesector
              endif

0296 CDEE02      CALL    RETMOD          ;WHAT TYPE OF 1/0 ??
0299 FE03        CPI      003H
029B DAF205      ic       WRITESOFT      ;FLOPPY DISK
029E C37E03      jmp      WRITEHARD      ;HARD DISK 1/0

```

; -----

ROUTINES TO SET AND RETURN THE CURRENT DRIVE MODE

```

      SETMOD:
02A1 21E60A      LXI      H,NEWDSK      ;SAVE NEWDSK IN STACK
02A4 7E          MOV      A,M
02A5 F5          PUSH     PSW
02A6 70          MOV      M,B
02A7 C5          PUSH     B
                        MVI      E,0      ;INDICATE INITIAL SELECT
02A8 48          MOV      C,B      ;CALL DISK SELECT
02A9 CD0302      CALL     SELDSK
02AC C1          POP      B
02AD 7C          MOV      A,H      ;CHECK FOR BAD SELECT
02AE B5          ORA      L
                        JRZ      SMERR      ; YES ABORT CHANGING
02AF+2832        DB      028H,SMERR-$-1 ; FAKE JRZ INSTRUCTION
02B1 68          MOV      L,B      ; B AND L DRIVE #
02B2 2600        mvi      H,000H
02B4 78          MOV      A,B      ;CHECK MODE SET VALIDITY
02B5 FE04        CPI      004H      ;ONLY VALID FOR FLOPPY DISK
                        JRNC     SMERR      ;INVALID DRIVE FOR MODE SET
02B7+302A        DB      030H,SMERR-$-1 ; ---- FAKE JRNC INSTRUCTION
02B9 118800      LXI      D,MODE      ;START OF MODE BYTES
02BC 19          DAD      D
02BD 71          MOV      M,C      ;SAVE NEW MODE BYTE
02BE E5          PUSH     H      ;SAVE MODE BYTE ADDRESS
02BF 79          MOV      A,C      ;SETUP FOR DENSITY CHANGE
02C0 B7          ORA      A
02C1 0E00        mvi      C,000H      ;ASSUME SINGLE DENSITY MODE
                        JRZ      SETSEL      ;VERIFY ASSUMPTION
02C3+2802        DB      028H,SETSEL-$-1 ; ---- FAKE JRZ INSTRUCTION
02C5 0E01        mvi      C,001H      ;SET FOR DOUBLE DENSITY MOD
02C7 CD7802 SETSEL: CALL SETDEN      ;SET DENSITY BASED ON LOW B
02CA E1          POP      H      ;RESTORE
02CB 6E          MOV      L,M      ;PICKUP MODE AGAIN
02CC 2600        mvi      H,000H      ;FOR SINGLE BYTE PRECISION
02CE 7D          MOV      A,L      ;SAVE MODE IN ACCUMULATOR F
02CF 29          DAD      H      2
02D0 29          DAD      H      4
02D1 E5          PUSH     H      ;SAVE * 4
02D2 29          DAD      H      ; * 8
02D3 D1          POP      D      ;REGAIN * 4
02D4 19          DAD      D      ; * 12
02D5 119101      LXI      D,MODLO      ;FIRST MODEL DPE
02D8 19          DAD      D      ;POINT TO THIS ONE
02D9 EB          XCHG      ;SETUP TEMPORARILY AS DESTI
02DA 2AB200      LHLD      DPEPTR      ;ADDRESS OF CURRENTLY SELEC
02DD EB          XCHG      ;SETUP TO ALTER
02DE 010C00      LXI      B,12      ;LENGTH FOR MOVE
                        LDIR      ;DO MOVE
02E1+EDB0        DB      0EDH,0BOH      ---- FAKE LDIR INSTRUCTION
02E3 F1 SMERR:   POP      PSW
02E4 E5          PUSH     H
02E5 32E60A      STA      NEWDSK      RESTORE ORIGINAL NEWDSK
02E8 4F          MOV      C,A

```

02E9 CD0302	CALL	SELDISK	
02EC E1	POP	H	
02ED C9	RET		;RETURN TO CALLER
RETMOD:			
02EE 118800	LXI	D,MODE	;START OF MODE BYTES
02F1 2AE60A	LHLD	NEWDSK	;NEXT DRIVE FOR 1/0
02F4 2600	MVI	H,000H	;RESET FOR SINGLE BYTE QUAN
02F6 19	DAD	D	;POINT TO IT
02F7 7E	MOV	A,M	;LOAD IT FOR CALLER
02F8 C9	RET		;RETURN, WITH CURRENT MODE

PAGE

THIS IS THE HOME DEVICE ROUTINE

```

02F9 3AE60A HOMEIT:LDA    NEWDSK          ; CHECK FOR FIRST HOME
02FC FE0C          CPI     12              ; CHECK FOR VIRTUAL DISK
02FE C20803        JNZ     REALDISK
0301 AF            XRA     A              ; VIRTUAL DISK
0302 67            MOV     H,A            ; SET TRACK TO ZERO
0303 6F            MOV     L,A
0304 22E70A        SHLD    NEWTRK
0307 C9            RET
      REALDISK:
0308 FE04          CPI     4              ; CHECKFOR FLOPPY
          JRC     HOME                  ; DO NOT BYPASS FLOPPY HOM
030A+380E          DB      038H,HOME-$-1  ; ---- FAKE JRC INSTRUCTION
030C 4F            MOV     C,A
030D 0600          MVI     B,0            ; POINT TO PRESENT TRACK ST
030F 217000        LXI     H,TRKO
0312 09            DAD     B
0313 7E            MOV     A,M          CHECK IF INITIALIZED
0314 FEFF          CPI     OFFH
0316 3E00          MVI     A,0
0318 C0            RNZ
0319 77            MOV     M,A          ; YES RETURN WITH NO ER

HOME:
031A 3AE60A        LDA     NEWDSK          ;GET VALUE OF DRIVE FOR HOM
031D FE04          CPI     004H          ;IS IT A HARD DISK ??
          JRNC    HOMEHARD              ;YES, PROCESS
031F+3022          DB      030H,HOMEHARD-$-1 ; ----FAKE JRNC INS

      HOMESOFT:
0321 CD5205        CALL    DSKSEL          ;SELECT CORRECT DRIVE (IN A
0324 3AF60A        LDA     ERFLAG
0327 B7            ORA     A              ;CHECK FOR ERRORS DURING SE
          JRNZ    HOME1A                ;EXIT IF ERRORS
0328+2016          DB      020H,HOME1A-$-1 ; ---- FAKE JRNZ INSTRUCTION
032A CDB305        CALL    POINT          ;POINT TO TRACK REGISTER SA
032D 3600          MVI     M,000H        ;RESET TO TRACK ZERO
032F CD1905        CALL    DBL$UPDATE
0332 3E0A          MVI     A,00AH        ;HOME COMMAND
0334 CD6307        CALL    FINTFIX        ;CLEAR ANY PENDING INTERRUPT
          ;AND ISSUE COMMAND
0337 CD3A07 HOME1:CALL    FPYWAIT        ;WAIT UNTIL 1/0 COMPLETE
033A 3AFC0A        LDA     STATUS        ;PICKUP STATUS BYTE
033D E698          ANI     10011000B     ;CHECK STATUS
033F C8            RZ                    ;RETURN WITH GOOD ESULT
0340 3E01 HOME1A: MVI     A,001H        ;SET ERROR ON HOME
0342 C9            RET                    ;AND RETURN

      HOMEHARD:
          IF HARDSK
0343 CD5205        CALL    DSKSEL          ;SELECT CORRECT DRIVE (IN A
0346 CDB305        CALL    POINT          ;POINT TO SAVE AREA
0349 3600          MVI     M,000H        ;SET TO TRACK ZERO
034B EB            XCHG                    ;POINT TO SELECT WORD

```

```
034C 7E          MOV      A,M          ;LOAD SELECT MASK
034D E6F0        ANI      11110000B    ;RESET HEAD MASK
034F 77          MOVM,A    ;SAVE
0350 D320        OUT      020H         ;WRITE HEAD/SELECT MASK
0352 3E20        MVI      A,020H       ;HOME COMMAND
0354 CD2107      CALL     INTFIX        ;CLEAR ANY PENDING INTERRUPT
                                           ;AND ISSUE COMMAND
0357 CD1707 HOME2:CALL    WAIT0         ;WAIT UNTIL 1/0 COMPLETE
035A 0E14        MVI      C,20         ;DELAY FOR 20 MILLISECONDS
035C CD8207      CALL     DELAY
035F AF          XRA      A            ;SET NEW TRACK REGISTER TO
0360 D322        OUT      022H         ;FOR CONTROLLER
                                           ;***DEBUG***
                                           ;***DEBUG***
0362 3AFC0A      LDA      STATUS        ;PICKUP STATUS BYTE
0365 E65D        ANI      01011101B    ;CHECK STATUS
0367 C8          RZ
0368 3E01        MVI      A,001H       ;SET ERROR ON HOME
                                           ;
                                ENDIF
036A C9          RET                  ;AND RETURN
```

PAGE

; THESE ARE THE HARD DISK UNBLOCK/REBLOCK AND READ AN
; ROUTINES CALLED BY THE BDOS SOFTWARE.

READHARD:

```

                IF      HARDSK
036B AF         XRA     A                ;RESET UNALLOCATED COUNT
036C 32F10A     STA     UNACNT
036F 3E01       MVI     A,001H          ;READ THE SELECTED CP/M SEC
0371 32F80A     STA     READOP
0374 32F70A     STA     RSFLAG          ;MUST READ DATA
0377 3E02       MVI     A,WRUAL
0379 32F90A     STA     WRTYPE          ;TREAT AS UNALLOCATED
                JR      RWOPER          ;TO PERFORM THE READ
037C+1864       DB      018H,RWOPER-$-1 ; ---- FAKE JR INSTRUCTION -
                ENDIF

```

WRITEHARD:

```

                IF      HARDSK
037E AF         XRA     A                ;WRITE THE SELECTED CP/M SE
037F 32F80A     STA     READOP          ;NOT A READ OPERATION
0382 79         Mov     A,C            ;WRITE TYPE IS PASSED IN RE
0383 32F90A     STA     WRTYPE
                if      mpm20
0386 E602       ani     WRUAL          ;IS IT WRITE UNALLOCATED ??
                JRZ     CHKUNA          ;CHECK FOR UNALLOCATED
0388+2817       DB      028H,CHKUNA-$-1 ; ---- FAKE JRZ INSTRUCTION
                else
                CPI     WRUAL          ;IS IT WRITE UNALLOCATED ??
                JRNZ     CHKUNA          ;CHECK FOR UNALLOCATED
                endif

```

; WRITE TO UNALLOCATED, SET PARAMETERS

```

038A 3E80       MVI     A,BLKSIZ/128    ;NEXT UNALLOC RECS
038C 32F10A     STA     UNACNT
038F 3AE60A     LDA     NEWDSK          ;DISK FOR 1/0
0392 32F20A     STA     UNADSK          ;UNADSK = NEWDSK
0395 2AE70A     LHLD    NEWTRK
0398 22F30A     SHLD    UNATRK          ;UNATRK = NEWTRK
039B 3AE90A     LDA     NEWSEC
039E 32F50A     STA     UNASEC          ;UNASEC = NEWSEC

```

; CHECK FOR WRITE TO UNALLOCATED SECTOR

CHKUNA:

```

03A1 3AF10A     LDA     UNACNT          ;ANY UNALLOCATED REMAIN ??
03A4 B7 ORA     A
                JRZ     ALLOC          ;SKIP IS NOT
03A5+2833       DB      028H,ALLOC-$-1  ---- FAKE JRZ INSTRUCTION

```

; MORE UNALLOCATED RECORDS REMAIN

```

03A7 3D         DCR     A                ;UNACNT = UNACNT - 1
03A8 32F10A     STA     UNACNT
03AB 3AE60A     LDA     NEWDSK          ;SAME DISK ??
03AE 21F2 0A     LXI     H,UNADSK
03B1 BE         CMP     M                ;NEWDSK = UNADSK ??

```



```

03B2+2026      JRNZ      ALLOC      ;SKIP IF NOT
                DB        020H,ALLOC-$-1  ---- FAKE JRNZ INSTRUCTION

; DISKS ARE THE SAME

03B4 21F30A     LXI       H,UNATRK
03B7 CD6104     CALL      NEWTRKCMP      ;NEWTRK = UNATRK ??
                JRNZ      ALLOC      ;SKIP IF NOT
03BA+201E      DB        020H,ALLOC-$-1  ; ---- FAKE JRNZ INSTRUCTION

; TRACKS ARE THE SAME

03BC 3AE90A     LDA       NEWSEC          ;SAME SECTOR ??
03BF 21F50A     LXI       H,UNASEC
03C2 BE        CMP       M              ;NEWSEC = UNASEC ??
                JRNZ      ALLOC      ;SKIP IF NOT
03C3+2015      DB        020H,ALLOC-$-1  ; ---- FAKE JRNZ INSTRUCTION

; MATCH, MOVE TO NEXT SECTOR FOR FUTURE REFERENCE

03C5 34        INR       M              ;UNASEC = UNASEC + 1
03C6 7E        MOV      A,M            ;END OF TRACK ??
03C7 FE80      CPI       CPMSPT        ;COUNT CP/M SECTORS
                JRC       NOOVF        ;SKIP IF NO OVERFLOW
03C9+3809      DB        038H,NOOVF-$-1  ; ---- FAKE JRC INSTRUCTION

```

OVERFLOW TO NEXT TRACK

```

03CB 3600      MVI       M,000H          ;UNASEC = 0
03CD 2AF30A     LHL      UNATRK
03D0 23        INX       H
03D1 22F30A     SHLD     UNATRK          ;UNATRK = UNATRK + 1

```

MATCH FOUND, MARK AS UNNECESSARY READ

NOOVF:

```

03D4 AF        XRA       A              ;ZERO TO ACCUMULATOR
03D5 32F70A     STA      RSFLAG          ;RSFLAG = 0
                JR        RWOPER        ;TO PERFORM THE WRITE
03D8+1808      DB        018H,RWOPER-$-1 ;---- FAKE JR INSTRUCTI

```

; NOT AN UNALLOCATED RECORD, REQUIRES PRE-READ

ALLOC:

```

03DA AF        XRA       A              ;ZERO TO ACCUMULATOR
03DB 32F10A     STA      UNACNT          ;UNACNT = 0
03DE 3C        INR       A              ;ONE TO ACCUMULATOR
03DF 32F70A     STA      RSFLAG          ;RSFLAG = 1

```

; THE FOLLOWING CODE IS COMMON TO BOTH READ AND W

RWOPER:

```

03E2 AF        XRA       A              ;ZERO TO ACCUMULATOR

```

```

03E3 32F60A      STA      ERFLAG      ;NO ERRORS YET
03E6 3AE90A      LDA      NEWSEC      ;COMPUTE HOST SECTOR
                REPT      SECSHF      ;COMPUTE HOST SECTOR
                ORA      A              ;CARRY = 0
                RAR              ;SHIFT RIGHT
                ENDM
03E9+B7          ORA      A              ;CARRY = 0
03EA+1F          RAR              ;SHIFT RIGHT
03EB+B7          ORA      A              ;CARRY = 0
03EC+1F          RAR              ;SHIFT RIGHT
03ED+B7          ORA      A              ;CARRY = 0
03EE+1F          RAR              ;SHIFT RIGHT
03EF 32EE0A      STA      NEWHST      ;HOST SECTOR TO SEEK

; ACTIVE HOST SECTOR ??

03F2 21EF0A      LXI      H,HSTACT      ;HOST ACTIVE FLAG
03F5 7E          MOV      A,M
03F6 3601        MVI      M,001H        ;ALWAYS BECOMES 1
03F8 B7          ORA      A              ;WAS IT ALREADY ??
                JRZ      FILLHST      ;FILL HOST IF NOT
03F9+2821        DB      028H,FILLHST-$-1 ; ---- FAKE JRZ

; HOST BUFFER ACTIVE, SAME AS SEEK BUFFER

03FB 3AE60A      LDA      NEWDSK
03FE 21EA0A      LXI      H,HSTDSK      ;SAME DISK ??
0401 BE          CMP      M              ;NEWDSK HSTDSK ??
                JRNZ      NOMATCH
0402+2011        DB      020H,NOMATCH-$-1 ; ---- FAKE JRNZ INS

; SAME DISK, SAME TRACK ??

0404 21EB0A      LXI      H,HSTTRK
0407 CD6104      CALL     NEWTRKCMP      ;NEWTRK HSTTRK ??
                JRNZ      NOMATCH
040A+2009        DB      020H,NOMATCH-$-1 ---- FAKE JRNZ INS

; SAME DISK, SAME TRACK, SAME BUFFER ??

040C 3AEE0A      LDA      NEWHST
040F 21ED0A      LXI      H,HSTSEC ;NEWHST = HSTSEC ??
0412 BE          CMP      M
                JRZ      MATCH ;SKIP IF MATCH
0413+2824        DB      028H,MATCH-$-1 ---- FAKE JRZ INSTRUCTION

; PROPER DISK, BUT NOT CORRECT SECTOR

                NOMATCH:
0415 3AF00A      LDA      HSTWRT ;HOST WRITTEN ??
0418 B7 ORA      A
0419 C46D04      CNZ      WRITEHST ;CLEAR HOST BUFFER

; MAY HAVE TO FILL HOST BUFFER

                FILLHST:
041C 3AE60A      LDA      NEWDSK

```

```

041F    32EA0A    STA    HSTDSK
0422    2AE70A    LHLD    NEWTRK
0425    22EB0A    SHLD    HSTTRK
0428    3AEE0A    LDA     NEWHST
042B    32ED0A    STA     HSTSEC
042E    3AF70A    LDA     RSFLAG    ;NEED TO READ ??
0431    B7        ORA     A
0432    C47F04    CNZ     READHST    ;YES, IF 1
0435    AF        XRA     A          ;ZERO TO ACCUMULATOR
0436    32F00A    STA     HSTWRT    ;NO PENDING WRITE

```

MATCH:

```

0439 3AE90A    LDA     NEWSEC    ;MASK BUFFER NUMBER
043C E607      ANI     SECMSK    ;LEAST SIGNIF BITS
043E 6F        mov    L,A        ;READY TO SHIFT

```

```

043F 2600      MVI     H,000H    ;DOUBLE COUNT
                REPT    7
                DAD     H
                ENDM

```

```

0441+29      DAD     H
0442+29      DAD     H
0443+29      DAD     H
0444+29      DAD     H
0445+29      DAD     H
0446+29      DAD     H
0447+29      DAD     H

```

; HL NOW HAS RELATIVE HOST BUFFER ADDRESS

```

0448 119C0E    LXI     D,HSTBUF
044B 19        DAD     D          ;HL = HOST ADDRESS
044C EB        XCHG                ;NOW IN DE
044D 2AAF00    LHLD    DMAADR    ;GET/PUT CP/M DATA
0450 EB        XCHG                ;SET FOR Z80 LDIR INSTR
                LXI     B,128     ;LENGTH OF MOVE
0451 3AF80A    LDA     READOP    ;WHICH WAY ??
0454 B7        ORA     A
0455 C23D0E    JNZ     RWMOVE    ;SKIP IF READ

```

; WRITE OPERATION, MARK AND SWITCH DIRECTION

```

0458 3E01      MVI     A,001H
045A 32F00A    STA     HSTWRT    ;HSTWRT = 1
045D EB        XCHG                ;SWAP DIRECTION
045E C33D0E    jmp     rwmov
                endif

```

PAGE

; UTILITY SUBROUTINE FOR 16 BIT COMPARE

```
                IF HARDSK
NEWTRKCMP:
0461 EB         XCHG                ;HL = UNATRK OR HSTTR
0462 21E70A     LXI      H,NEWTRK
0465 1A         LDAX    D           ;LOW BYTE COMPARE
0466 BE         CMP     M           ;SAME ??
0467 C0         RNZ                    ;RETURN IF NOT
0468 13         INX     D           ;TO CHECK HIGH BYTE
0469 23         INX     H
046A 1A         LDAX    D
046B BE         CMP     M           ;SETS FLAGS
046C C9         RET
```

PAGE

```
;WRITEHST PERFORMS THE PHYSICAL WRITE TO THE HOST DI
;READHST PERFORMS THE PHYSICAL READ FROM THE HOST DI
```

```
;HSTDSK = HOST DISK NUMBER
;HSTTRK = HOST TRACK NUMBER
;HSTSEC = HOST SECTOR NUMBER
;RETURN ERROR FLAG IN ERFLAG
```

WRITEHST:

```
046D 3E05      MVI      A,005H      ;SETUP DMA FOR WRITE
046F 32CE00     STA      DMAS3F
0472 3E02      MVI      A,002H      ;WRITE COMMAND
0474 32FA0A     STA      CMD        ;SAVE FOR LATER
0477 219B0E     LXI      H,HSTBUF-1 ;WRITE MUST WRITE CONTROL B
047A 22BC00     SHLD     DMASA
              JR      HRWO
047D+1810      DB      018H,HRWO-$-1  ---- FAKE JR INSTRUCTION -
```

READHST:

```
047F 3E01      MVI      A,001H      ;SETUP DMA FOR READ
0481 32CE00     STA      DMAS3F
0484 3E04      MVI      A,004H      ;READ COMMAND
0486 32FA0A     STA      CMD        ;SAVE FOR LATER
0489 219C0E     LXI      H,HSTBUF   ;READ ONLY DATA BYTES
048C 22BC00     SHLD     DMASA
```

HRWO:

```
048F 3E05      MVI      A,05        ;FIVE RETRIES
0491 32020B     STA      T$RETRIES  ;SETUP TEMPORARY RETRIES CO
0494 3EFF       MVI      A,0FFH     ;INIT TOGGLE SO THAT NO HOM
0496 32030B     STA      HOME$TOGGLE;ALTERNATE RETRIES WILL BE
                                ;OTHER RETRIES WILL BE DONE
```

HRW1:

```
0499 3AED0A     LDA      HSTSEC     ;HOST SECTOR NUMBER
049C 32B100     STA      SECTNO     ;SAVE SECTOR NUMBER
049F 3AEA0A     LDA      HSTDSK     ;PICKUP DRIVE ID FOR SELECT
04A2 CD5205     CALL     DSKSEL     ;SELECT CORRECT DRIVE FOR I
04A5 CDB305     CALL     POINT      ;POINT TO TRACK REGISTER SA
04A8 EB        XCHG                ;POINT TO SELECT MASK
04A9 3EF0       MVI      A,11110000B;TO REMOVE CURRENT HEAD SEL
04AB A6        ANA      M
04AC 77        mov     M,A
04AD E5        PUSH     H           ;SAVE MASK ADDRESS
04AE CD3205     CALL     SETHED     ;COMPUTE CORRECT HEAD NUMBE
04B1 7D        MOV      A,L        ;TRACK NUMBER AFTER HEAD CA
```

```
04B2 32AD00     STA      TRAKNO
04B5 E1        POP      H           ;RESTORE MASK ADDRESS
04B6 3AAE00     LDA      HEADNO     ;TO OR IN NEW HEAD NUMBER
04B9 B6        ORA      M
04BA 77        MOV      M,A        ;SAVE NEW DRIVE/HEAD SELECT
04BB E67F      ANI      07FH       ; MASK OFF LARGE DRIVE FLAG
04BD D320      OUT      020H       ;WRITE IT TO SELECT NEW HEA
04BF 0E01      MVI      C,1        ;DELAY FOR 1 MILLISECOND
04C1 CD8207     CALL     DELAY
```

HRW2:

```

04C4 CDB305      CALL    POINT          ;IS A SEEK NECESSARY ??
04C7 3AAD00      LDA     TRAKNO         ;CHECK
04CA BE          CMP     M              ;WELL ??
                                JRZ      HRW5         ;NO SEEK NECESSARY
04CB+2814        DB      028H,HRW5-$-1  ---- FAKE JRZ INSTRUCTION

                                HRW3:
04CD D322        OUT     022H           ;WRITE NEW TRACK NUMBER
04CF 46          MOV     B,M            ;SAVE TEMPORARILY
04D0 77          MOV     M,A            ;UPDATE TRACK REGISTER SAVE
04D1 78          MOV     A,B            ;OLD TRACK NUMBER
04D2 D321        OUT     021H           ;TO OLD TRACK REGISTER
04D4 3E10        MVI     A,010H         ;SEEK COMMAND
04D6 CD2107      CALL    INTFIX         ;CLEAR ANY PENDING INTERRUPT
                                ;AND ISSUE COMMAND
04D9 CD1707 HRW4: CALL    WAIT0         ;WAIT FOR 1/0
04DC 0E14        MVI     C,20           ;DELAY AFTER SEEK FOR 20 MI
04DE CD8207      CALL    DELAY

                                HRW5:
04E1 3AB100      LDA     SECTNO         ;SET SECTOR
04E4 D321        OUT     021H

                                HRW6:
04E6 21BA00      LXI     H,DMAS1        ;SETUP DMA FOR HARD DISK I/
04E9 010006      LXI     B,0600H
                                OUTIR
04EC+EDB3        DB      0EDH,0B3H     ---- FAKE OTIR INSTRUCTION
04EE 21C000      LXI     H,DMAS2H
04F1 010006      LXI     B,0600H
                                OUTIR
04F4+EDB3        DB      0EDH,0B3H     ---- FAKE OTIR INSTRUCTION
04F6 21CA00      LXI     H,DMAS3
04F9 010007      LXI     B,0700H
                                OUTIR
04FC+EDB3        DB      0EDH,0B3H     ---- FAKE OTIR INSTRUCTION
04FE 3AFA0A      LDA     CMD            ;PICKUP 1/0 COMMAND
0501 CD2107      CALL    INTFIX         ;CLEAR ANY PENDING INTERRUPT
                                ;AND ISSUE COMMAND
0504 CD1707 RW7: CALL    WAIT0         ;WAIT FOR COMPLETION
0507 3E5D        MVI     A,01011101B   ;SETUP STATUS AND MASK

0509 32FB0A      STA     MASK            ;SAVE FOR STATUS CHECK
050C CDAE06      CALL    CHECK$STAT     ;CHECK STATUS FROM 1/0
050F C8          RZ                     ;OK ??
0510 3A030B      LDA     HOME$TOGGLE
0513 2F          CMA
0514 32030B      STA     HOME$TOGGLE
                                ;CHANGE TOGGLE SO THAT HOME
                                JR      HRW1         ;RETRY 1/0
0517+1880        DB      018H,HRW1-$-1  ---- FAKE JR INSTRUCTION -

                                ENDIF
                                PAGE

```

 ;DOUBLE SIDED TRACK REGISTER UPDATE ROUTINE

DBL\$UPDATE:

0519 3AB600	LDA	MPARMS	;CHECK FOR DOUBLE SIDED DRI
051C E601	ANI	1	; IS FLAG SET
051E C8	RZ		; NO -- SO RETURN
051F 3AAC00	LDA	DISKNO	;CURRENT DISK DRIVE
0522 FE04	CPI	004H	;IS IT A FLOPPY
0524 D0	RNC		;NO, RETURN WITHOUT UPDATE
0525 E602	ANI	00000010B	;IS THIS DRIVE 2 OR 3 ??
0527 7E	MOV	A,M	;WE WERE CALLED WITH (HL) P
	JRZ	DBL\$LOW	;IT MUST BE DRIVE ZERO OR 0
0528+2804	DB	028H, DBL\$LOW-\$-1	;---- FAKE JRZ INST
052A 2B	DCX	H	;BACKUP TO OTHER SIDE POINT
052B 2B	DCX	H	
	JR	DBL\$SAVE	
052C+1802	DB	018H, DBL\$SAVE-\$-1	---- FAKE JR INSTR

DBL\$LOW:

052E 23	INX	H	;BUMP UP TO DRIVE TWO OR TH
052F 23	INX	H	

DBL\$SAVE:

0530 77	MOV	M,A	;UPDATE OTHER SIDE REGISTER
0531 C9	RET		

PAGE

ROUTINE TO COMPUTE HEAD NUMBER FROM TRACK NUMBER
 TRACK NUMBER IS IN HL ON ENTRY

	IF	HARDSK	
0532 2AEBOA	LHLD	HSTTRK	;CP/M TRACK NUMBER (0-800)
0535 E680	ANI	80H	; CHECK FOR LARGE DRIVE
0537 7D	MOV	A, L	;LOW ORDER
	JRZ	SETH14	; SMALL DRIVE
0538+2806	DB	028H, SETH14-\$-1	;---- FAKE JRZ INSTRUCTION
053A E607	ANI	00000111B	;GET TRACK MOD 8 (HEAD NUMB
053C 0E03	MVI	C, 3	;LIMIT LOOP FOR DIVIDE BY E
	JR	SETDVD	
053E+1804	DB	018H, SETDVD-\$-1	---- FAKE JR INSTRUCTION -
0540 E603 SETH14:	ANI	00000111B	;GET TRACK MOD 4 (HEAD NUMB
0542 0E02	MVI	C, 2	;LIMIT LOOP FOR DIVIDE BY F
0544 32AE00 SETDVD:	STA	HEADNO	;SAVE AS HEAD NUMBER
0547 B7 SHD1:	ORA	A	;ENSURE CARRY IS ZERO
0548 7C	MOV	A, H	;FOR SHIFT
0549 1F	RAR		;ONE BIT
054A 67	MOV	H, A	
054B 7D	MOV	A, L	;LOW ORDER
054C 1F	RAR		;CARRY PARTICIPATES FROM HI
054D 6F	MOV	L, A	
054E 0D	DCR	C	;END OF DIVIDE YET ??
	JRNZ	SHD1	;NO, CONTINUE
054F+20F6	DB	020H, SHD1-\$-1	;---- FAKE JRNZ INSTRUCTION
0551 C9	RET		;RETURN TO CALLER, TRACK IN
	ENDIF		
	PAGE		

DISK DRIVE SELECT ROUTINE

ON ENTRY, THE ACCUMULATOR CONTAINS THE DRIVE
 RETURNS CARRY SET FOR HARD DISK SELECTED
 RETURNS CARRY RESET FOR FLOPPY DISK SELECTED

DSKSEL:

```
0552 FE04      CPI      004H      ;IS IT HARD DISK ??
              JRNC     SELHARD    ;YES, GO PROCESS
0554+3045      DB       030H,SELHARD-$-1 ; ---- FAKE JRNC INS
```

SELISOFT:

```
0556 21AC00     LXI      H,DISKNO  ;CURRENT DRIVE NUMBER
0559 BE         CMP      M          ;SAME DRIVE AS LAST TIME ??
              JRZ      SLS3        ;YES, DONT BOTHER WITH UNLO
055A+2819      DB       028H,SLS3-$-1 ; ---- FAKE JRZ INSTRUCTION
055C 77         MOV      M,A        ;UPDATE WITH CURRENT DRIVE
```

WE WILL NOW FORCE THE HEAD TO UNLOAD PRIOR TO THE S
 TO ENSURE THAT WHEN WE RETURN TO THIS DISK WE WILL
 LOAD AND WAIT FOR THE HEAD TO SETTLE.

SLS1:

```
055D DB04      IN       004H      ;ENSURE FLOPPY PORT NOT BUS
055F 1F         RAR
              JRC      SLS1
0560+38FB      DB       038H,SLS1-$-1 ---- FAKE JRC INSTRUCTION
0562 DB05      IN       005H      ;READ THE TRACK REGISTER
0564 D307      OUT      007H      ;ENSURE WE DONT MOVE THE HE
0566 3E12      MVI      A,012H    ;SEEK AND UNLOAD HEAD
0568 CD6307     CALL     FINFIX    ;CLEAR ANY PENDING INTERRUPT
              ; AND ISSUE COMMAND
056B CD3A07 SLS2: CALL     FPYWAIT  ;WAIT HERE FOR INTERRUPT
056E 3AFC0A     LDA      STATUS    ;HOW DID THE 1/0 GO?
0571 E698      ANI      10011000B ; CHECK
              JRNZ     SLSERR      ;EXIT IF ERROR
0573+2020      DB       020H,SLSERR-$-1 ; ---- FAKE JRNZ INSTRUCTION
```

;WE WILL NOW LOAD THE SELECT MASK AND SELECT THE DRIVE
 ;EVEN IF ITS THE SAME DRIVE BECAUSE THE DENSITY MAY
 ;HAVE CHANGED.

SLS3:

```
0575 CDB305     CALL     POINT     ;POINT TO TRACK SAVE AREA
0578 EB         XCHG
              ;POINT TO SELECT MASK
0579 3AAD00     LDA      TRAKNO    ;NEXT TRACK FOR 1/0
057C FE02      CPI      002H      ;IS IT TRACK ZERO OR ONE
057E 3EFF      MVI      A,11111111B ;ASSUME NO
              JRNC     SLS4        ;VERIFY ASSUMPTION
0580+3002      DB       030H,SLS4-$-1 ; ---- FAKE JRNC INSTRUCTION
```

```

0582 3EFE          MVI      A,11111110B      ;FORCE SINGLE DENSITY FOR 0

SLS4:
0584 A6           ANA       M                ;LOAD MASK AND CORRECT IF N
0585 D308          OUT      008H             ;SELECT IT
0587 DB04          IN       004H             ;IS DRIVE READY?
0589 17           RAL
058A+3809          JRC       SLSERR           ;IF NOT BRANCH
058C EB           DB        038H,SLSERR-$-1  ;---- FAKE JRC INSTRUCTION
058D 7E           XCHG
058E D305          MOV      A,M              ;RESTORE TRACK REGISTER ADD
0590 AF           XRA       A                ;PICK UP TRACK NUMBER
0591 32F60A        STA      ERFLAG           ;GIVE IT TO CONTROLLER
0594 C9           RET
0595 AF SLSERR:    XRA       A                ;ENSURE CARRY IS RESET
0596 3C           INR      A                ;ALSO ZERO ERROR INDICATOR
0597 32F60A        STA      ERFLAG
059A C9           RET

```

THIS ROUTINE SETS UP THE HARD DISK BY SELECTING THE
DRIVE AND RELOADING THE HEAD AND TRACK REGISTERS IN
HARD DISK CONTROLLER READY FOR 1/0 LATER.

SELHARD:

```

059B 21AC00        LXI      H,DISKNO        ;CURRENT DRIVE SELECTED
059E BE           CMP      M                ;SAME ??
059F C8           RZ
05A0 77           MOV      M,A              ;YES, NO NEW SELECT NECESSA
05A1             ;UPDATE DISKNO

SLH1:
05A1 CDB305        CALL     POINT            ;TRACK SAVE REGISTER
05A4 EB           XCHG
05A5 7E           MOV      A,M              ;POINT TO SELECT MASK
05A6 D320          OUT      020H             ;LOAD DRIVE/HEAD VALUE
05A8 EB           XCHG
05A9 7E           MOV      A,M              ;WRITE IT TO SELECT PORT
05AA D322          OUT      022H             ;REGAIN ADDRESS OF TRACK RE
05AC 0E14          MVI      C,20             ;LOAD OLD TRACK NUMBER
05AE CD8207        CALL     DELAY            ;WRITE IT TO OLD TRACK REGI
05B1 37           STC
05B2 C9           RET                       ;DELAY FOR 20 MILLISECONDS
05B3             ;SET CARRY TO SHOW HARD DIS
05B4             ;RETURN TO CALLER

```

PAGE

; SUBROUTINE TO POINT TO CURRENT TRACK REGISTER SAVE

POINT:

```

05B3 2AAC00      LHL      DISKNO      ;PICKUP CURRENT DISK
05B6 7D          MOV      A,L
05B7 2600        MVI      H,0          ;RESET HIGH ORDER HALF
05B9 117000      LXI      D,TRK0      ;LOAD TRACK POINTER
05BC 19          DAD      D           ;POINT TO CURRENT TRACK PTR
05BD 54          MOV      D,H         DE = TRACK
05BE 5D          MOV      E,L
05BF 010C 00     LXI      B,12
05C2 09          DAD      B           HL = SELECT
                                IF      HARDSK
05C3 FE04        CPI      4
                                JRC      PNTFN      FLOPPY DISK
05C5+380D        DB      038H,PNTFN-$-1  ---- FAKE JRC INSTRUCTION
05C7 3E10        MVI      A,10H
05C9 A6          ANA      M           CHECK DRIVE SELECT
                                JRZ      PNTH2      MUST BE DRIVE # 2
05CA+2805        DB      028H,PNTH2-$-1  ---- FAKE JRZ INSTRUCTION

05CC 11B800      LXI      D,HTK1      POINT TO DRIVE 1
                                JR      PNTFN
05CF+1803        DB      018H,PNTFN-$-1  ---- FAKE JR INSTRUCTION -
05D1 11B900      PNTH2: LXI      D,HTK2 POINT TO DRIVE 2
                                ENDIF
05D4 EB  PNTFN:  XCHG
05D5 C9          RET      HL = TRACK  DE = SELECT

```

; ROUTINE TO TRANSLATE SECTOR NUMBER

SECTRAN:

```

05D6 EB          XCHG      ;TABLE ADDRESS IS IN DE (NO
05D7 7C          MOV      A,H      ;IS THERE A TABLE ADDRESS ?
05DB B5          ORA      L
                                JRZ      STRN2      ;NO, JUST RETURN ENTERED QU
05D9+2807        DB      028H,STRN2-$-1  ---- FAKE JRZ INSTRUCTION

```

STRN1:

```

05DB 0600        MVI      B,000H      ;ENSURE OK FOR SINGLE BYTE
05DD 09          DAD      B           ;ADD SECTOR NUMBER
05DE 6E          MOV      L,M         ;LOAD TRANSLATED VALUE
05DF 2600        MVI      H,000H
05E1 C9          RET
                                ;NEW VALUE RETURNED IN HL

```

STRN2:

```

05E2 09          DAD      B           ;RETURN SAME VALUE AS ENTER
05E3 C9          RET

```

; ROUTINES TO DO FLOPPY 1/0

READSOFT:

```

05E4 3E9F        Mvi      A,09FH      ;MASK FOR READ STATUS
05E6 32FB0A      STA      MASK
05E9 3E01        MVI      A,001H      ;SETUP DMA FOR READ

```

```

05EB 32CE00      STA      DMAS3F
05EE 3E8C        MVI      A,08CH      ;READ COMMAND
                                JR      SRW1
05F0+180F        DB      018H,SRW1-$-1  ---- FAKE JR INSTRUCTION

        WRITESOFT:
05F2 3EFF        Mvi      A,0FFH      ;MASK FOR WRITE STATUS
05F4 32FB0A      STA      MASK
05F7 CD6B0E      CALL     MVDTB
05FA 3E05        MVI      A,005H      ;SETUP DMA FOR WRITE
05FC 32CE00      STA      DMAS3F
05FF 3EAC        Mvi      A,0ACH      ;WRITE COMMAND

        SRW1:
0601 32FA0A      STA      CMD
0604 21ID13      LXI      H,FPYBUF
0607 22BC00      SHLD     DMASA
060A 3AE60A      LDA      NEWDSK
060D CD5205      CALL     DSKSEL      ;SELECT DRIVE FOR 1/0
0610 3AF60A      LDA      ERFLAG      ;CHECK FOR SELECT ERROR
0613 B7          ORA      A
0614 C0          RNZ          ;RETURN IF ERROR

SRW2:
0615 3E0A        MVI      A,10        ;SET NUMBER OF TRIALS
0617 32020B      STA      T$RETRIES    ;SAVE FOR RETRY ROUTINE
061A AF          XRA      A
061B 32030B      STA      HOME$TOGGLE   ;FORCE HOME PRIOR TO EACH R

        LOAD$HEAD:
061E DB08        IN       008H        ;IS HEAD LOADED ??
0620 E602        ANI      00000010B    ;CHECK IT
                                JRNZ     REMOVE$LD      ;YES, ITS LOADED, DONT RELO
0622+201F        DB      020H,REMOVE$LD-$-1 ;---- FAKE JRNZ INS
0624 DB05        IN       005H        ;DUMMY SEEK TO START HEAD L
0626 D307        OUT      007H        ;KEEP IT SHORT
0628 3E1A        MVI      A,01AH      ;START HEAD LOADING
062A CD6307      CALL     FINTFIX      ;CLEAR ANY PENDING INTERRUPT
                                ;AND ISSUE COMMAND
062D CD3A07 LDH1: CALL     FPYWAIT      ;WAIT FOR 1/0 TO COMPLETE
0630 3AFC0A      LDA      STATUS      ;HOW DID IT GO?
0633 E698        ANI      10011000B    ;CHECK
                                JRNZ     CHECKIT         ;DO NOT GO ON IF ERROR
0635+2044        DB      020H,CHECKIT-$-1 ;---- FAKE JRNZ INS

0637 0E10        MVI      C,16        ;WAIT HERE FOR 16 MS
0639 CD8207      CALL     DELAY        ;CALL WAIT ROUTINE
063C CDB305      CALL     POINT        ;REESTABLISH TRACK REGISTER
063F 36FE        MVI      M,254        ;ENSURE FURTHER SEEK AND DE
                                JR      TRKTST
0641+1807        DB      018H,TRKTST-$-1 ---- FAKE JR INSTRUCTION -

        REMOVE$LD:
0643 21FA0A      LXI      H,CMD        ;POINT TO 1/0 COMMAND
0646 3EFB        MVI      A,11111011B  ;REMOVE HEAD LOAD BIT
0648 A6          ANA      M           ;DO IT
0649 77          MOV      M,A         ;SAVE IT BACK INTO CMD

```

TRKTST:

```

064A CDB305      CALL    POINT          ;RESTORE TRACK REGISTER POI
064D 3AE70A      LDA     NEWTRK         ;GET NEW TRACK NUMBER
0650 32AD00      STA     TRAKNO         ;SAVE IN COMMON PLACE
0653 BE          CMP     M              ;SAME AS LAST TIME ??
          JRZ       FSECSET             ;YES, DONT BOTHER WITH SEEK
0654+281A        DB      028H,FSECSET-$-1 ; ---- FAKE JRZ INST
0656 77          MOV     M,A            ;SAVE IT
0657 D307        OUT     007H           ;ALSO SEND IT TO CONTROLLER
0659 CD1905      CALL    DBL$UPDATE     ;DOUBLE SIDED SUPPORT

```

FLOPPY\$SEEK:

```

065C MA          MVI     A,01AH         ;SEEK COMMAND WITH HEAD LOA
065E CD6307      CALL    FINTRFIX       ;CLEAR ANY PENDING INTERRUPT
          ;AND ISSUE COMMAND
0661 CD3A07 FPS1: CALL    FPYWAIT       ;WAIT FOR 1/0 TO COMPLETE
0664 3AFC0A      LDA     STATUS         ;HOW DID IT GO?
0667 E698        ANI     10011000B      ;CHECK
          JRNZ      CHECKIT             ;DO NOT GO ON IF ERROR
0669+2010        DB      020H,CHECKIT-$-1 ; ---- FAKE JRNZ INS

```

```

066B 0E10        MVI     C,16           ;SET FOR 16 MS DELAY
066D CD8207      CALL    DELAY

```

FSECSET:

```

0670 3AE90A      LDA     NEWSEC         ;SET SECTOR
0673 32BI00      STA     SECTNO        ;SAVE IN COMMONN PLACE
0676 D306        OUT     006H
0678 CD8706      CALL    FLOPPYIO      ;DO 1/0

```

CHECKIT:

```

067B CDAE06      CALL    CHECK$STAT     ;CHECK STATUS OF 1/0
067E 3AF60A      LDA     ERFLAG         ;SETUP TO RETURN TO BDOS
0681 CC7E0E      CZ      MVDFB
0684 C8          RZ                     ;EITHER OK OR PERMANENT ERR
          JR        LOAD$HEAD           ;ERROR, JUST RETRY THIS SAM
0685+1897        DB      0JBH,LOAD$HEAD-$-1 ; ---- FAKE JR INSTR

```

THIS IS THE ROUTINE THAT DOES THE FLOPPY DISK 1/0

```
; -----

FLOPPYIO:
    IF      NOT DMA
    LXI     H,066H          ;MOVE DATA FROM 066H TO SAV
    LXI     D,SAVE1
    LXI     B,004H
    LDIR                    ;MOVE IT
    LXI     H,NMIRTN        ;SET NMI ROUTINE TO NMI ADD
    LXI     D,066H
    LXI     B,004H
    LDIR                    ;MOVE IT
    LDA     CMD              ;IS IT A WRITE ??
    ANI     20H
    iz      FRD              ;NO, LEAVE INI CMD IN LOW M
    LXI     H,067H          ;POINT TO COMMAND AREA
    MVI     M,0A3H          ;MAKE IT AN OTI CMD
    FRD     EQU             $ ;LABEL
    ENDIF

0687 21BA00    IF      DMA
068A 010006    LXI     H,DMA$1      ;INITIALIZE DMA
                LXI     B,0600H
                OUTIR          ;WRITE TO DMA
068D+EDB3     DB      0EDH,0B3H    ---- FAKE OTIR INSTRUCTION
068F 21C600    LXI     H,DMA$2F
0692 010004    LXI     B,0400H
                OUTIR          ;WRITE TO DMA
0695+EDB3     DB      0EDH,0B3H    ---- FAKE OTIR INSTRUCTION
0697 21CA00    LXI     H,DMA$3
069A 010007    LXI     B,0700H
                OUTIR          ;WRITE TO DMA
069D+EDB3     DB      0EDH,0B3H    ---- FAKE OTIR INSTRUCTION
                ENDIF
069F 0E07      MVI     C,007H      ;PORT ADDRESS FOR 1/0
06A1 211D13    LXI     H,FPYBUF    ;DMA ADDRESS
06A4 3AFA0A    LDA     CMD          ;1/0 COMMAND
06A7 CD6307    CALL    FINTEFIX    ;CLEAR ANY PENDING INTERRUPT
                                ;AND ISSUE COMMAND
06AA CD3A07 FWT1: CALL    FWTWAIT    ;WAIT HERE FOR 1/0 TO COMPL
                IF      NOT DMA
                LXI     H,SAVE1      ;SETUP TO REPLACE DATA
                LXI     D,066H      ;COPIED FROM NMI LOCATION
                LXI     B,004H
                LDIR                    ;MOVE IT
                ENDIF

06AD C9        RET                ;RETURN, 1/0 COMPLETED
```

WE WILL NOW CHECK THE STATUS OF THE 1/0 OPERATION
 RETURN WITH CONDITION CODE ZERO = NO RETRY
 RETURN WITH CONDITION CODE NON ZERO = RETRY

CHECK\$STAT:

```

06AE 21F60A      LXI      H,ERFLAG      ;POINT TO ERROR INDICATOR
06B1 3600        Mvi      M,000H        ;ASSUME OK
06B3 21FC0A      LXI      H,STATUS      ;CHECK STATUS
06B6 3AFB0A      LDA      MASK          ;MASK FOR UNWANTED BIT REMO
06B9 A6          ANA      M
06BA 77          MOV      M,A           ;SAVE CLEANED STATUS
06BB C8          RZ                    ;OK, SO RETURN

      CHKS0:
06BC CDEE02      CALL     RETMOD
06BF FE03        CPI      003H          ;HARD DISK ??
06CI 21FC0A      LXI      H,STATUS
06C4 7E          MOV      A,M           ;RELOAD STATUS BYTE
      JRNC       CHKS2                ;YES, CHECK FOR DRIVE READY

06C5+3006        DB       030H,CHKS2-$-1  ---- FAKE JRNC INSTRUCTION

      CHKS1:
06C7 FE80        CPI      080H          ;IS FLOPPY DISK NOT READY ?
      JRZ        BADIO                ;YES, DONT BOTHER WITH RETR
06C9+283D        DB       028H,BADIO-$-1  ;---- FAKE JRZ INSTRUCTION
      JR         CHKS3                ;GO TO BAD MESSAGE ROUTINE
06CB+1819        DB       018H,CHKS3-$-1  ---- FAKE JR INSTRUCTION

      CHKS2:
06CD FE00        CPI      000H          ;IS HARD DISK NOT READY ??
      JRZ        BADIO                ;YES, BYPASS ERROR MESSAGE
06CF+2837        DB       028H,BADIO-$-1  ;---- FAKE JRZ INSTRUCTION
06D1 E640        ANI      01000000B      ;IS IT WRITE FAULT ??
      JRZ        CHKS3                ;NO, CONTINUE ON
06D3+2811        DB       028H,CHKS3-$-1  ;---- FAKE JRZ INSTRUCTION
06D5 CDB305      CALL     POINT          ;POINT TO TRACK REGISTER
06D8 EB          XCHG
06D9 7E          MOV      A,M           ;POINT TO SELECT MASK
06DA F640        ORI      01000000B      ;TURN ON WRITE FAULT CLEAR
06DC D320        OUT      020H
06DE 7E          MOV      A,M           ;RESET CLEAR
06DF D320        OUT      020H
06E1 0E14        MVI      C,20          ;DELAY JUST TO BE SAFE
06E3 CD8207      CALL     DELAY

CHKS3:
06E6 3A030B      LDA      HOME$TOGGLE
06E9 B7          ORA      A             ;IS A HOME NEEDED ON THIS R
      JRNZ       CHKS4
06EA+200B        DB       020H,CHKS4-$-1  ---- FAKE JRNZ INSTRUCTION
06EC 3AF0A      LDA      STATUS          ;SAVE STATUS OVER HOME
06EF F5          PUSH     PSW
06F0 CD1A03      CALL     HOME           ;RESET DEVICE TO HOME
06F3 F1          POP      PSW
06F4 32FC0A      STA      STATUS          ;SAVE FOR ERROR MESSAGE

CHKS4:
06F7 119400      LXI      D,TCNT        ;BUMP TEMP ERROR COUNT
06FA CD0F07      CALL     ADDERRORS
06FD 21020B      LXI      H,T$RETRIES   ;PICKUP RETRY COUNT
0700 35          DCR      M             ;DECREMENT COUNT OF RETRIES
0701 C0          RNZ

```

```

0702 11A000      LXI      D,PCNT      ;BUMP PERMANENT ERROR COUNT
0705 CDOF07      CALL     ADDERRORS

```

BADIO:

```

0708 21F60A      LXI      H,ERFLAG    ;SET PERMANENT ERROR
070B 3601        MVI      M,001H      ;DO IT
070D AF          XRA      A           ;RESET T6*PRECLUDE RETRIES
070E C9          RET                ;RETURN TO CALLER

```

ADDERRORS:

```

070F2AAC00      LHL      DISKNO      ;BUMP COUNT OF DISK ERRORS
0712 2600        MVI      H,000H
0714 19          DAD      D           ;POINT TO ERROR REGISTER
0715 34          INR      M
0716 C9          RET

```

PAGE

; THIS IS HARD DISK WAIT ENTRY

WAIT0:

```

0717 C5  PUSH     B                SAVE RETRY COUNT
0718 0E84        MVI      C,FLAGWT  FUNCTION FLAG WAIT
071A 1E05        MVI      E,HDFLAG  DEVICE IS HARD DISK
071C CD100B      CALL     XDOS
071F C1          POP      B                RESTORE RETRY COUNTER IN

```

READ OR WRITE IS OK, ACCUMULATOR CONTAINS ZERO

```

0720 C9          RET

```

THE FOLLOWING CODE GUARANTEES THAT HARD DISK FLAG I
IT APPEARS THAT WE OCCASIONALLY GET FLAG SET AS A R
OF AN INTERRUPT FROM THE HARD DISK, WHEN WE DO
NOT EXPECT IT.

INTFIX:

```

0721 F5          PUSH     PSW
0722 C5          PUSH     B
0723 D5          PUSH     D
0724 E5          PUSH     H
0725 0E85        MVI      C,FLAGST
0727 1E05        MVI      E,HDFLAG
0729 CD100B      CALL     XDOS      ;EITHER FLAG 5 WILL BE SET
                                      ;IT IS ALREADY SET - IN WHI
                                      ;THIS REQUEST WILL BE IGNOR

072C 0E84        MVI      C,FLAGWT
072E 1E05        MVI      E,HDFLAG
0730 CD100B      CALL     XDOS      ;NOW CLEAR THE FLAG
0733 E1          POP      H
0734 D1          POP      D

0735 C1          POP      B
0736 F1          POP      PSW      ;RESTORE REGISTERS
0737 D323        OUT      023H     ;ISSUE COMMAND TO HARD DISK
0739 C9          RET
PAGE

```


THIS IS FLOPPY DISK WAIT ENTRY

```

        FPYWAIT:
073A C5          PUSH      B                ;SAVE RETRY COUNT
073B E5          PUSH      H
073C 0E84        Mvi       C,FLAGWT        ; FUNCTION IS FLAG WAIT
073E 1E06        Mvi       E,FPYFLAG      ; WAIT FOR FLOPPY
0740 CD100B      CALL      XDOS
0743 F5          PUSH      PSW
0744 3AD00D      LDA       FPYTIME        ;DID WD1791 GO TO SLEEP?
0747 B7          ORA       A
0748+2015        JRNZ      NOPPYRST        ;IF STILL AWAKE, SKIP RESET
                                DB          020H,NOPPYRST-$-1 ;---- FAKE JRNZ INS

074A DB09        IN        009H           ;GET CURRENT BANK NUMBER
074C E618        ANI       00011000B      ;REMOVE OTHER INFO
074E D309        OUT       009H           ;RESET WD1791
0750 0E01        Mvi       C'l           ;DELAY 1 MILLISEC
0752 CD8207      CALL      DELAY
0755 F602        ORI       00000010B      ;END RESET
0757 D309        OUT       009H
0759 3AE60A      LDA       NEWDSK         ;MAKE SURE CURRENT DISK AND
075C 32AC00      STA       DISKNO         THE SAME

        NOPPYRST:
075F Fl         POP       PSW
0760 El         POP       H
0761 Cl         POP       B
0762 C9         RET                     ;RESTORE RETRY COUNT IN <C>

```

THE FOLLOWING CODE GUARANTEES THAT FLOPPY DISK FLAG

```

        FINTFIX:
0763 F5          PUSH      PSW
0764 C5          PUSH      B
0765 D5          PUSH      D
0766 E5          PUSH      H

0767 0E85        MVI       C,FLAGST
0769 1E06        Mvi       E,FPYFLAG
076B CD100B      CALL      XDOS
076E 0E84        Mvi       C,FLAGWT
0770 1E06        MVI       E,FPYFLAG
0772 CD100B      CALL      XDOS
0775 210301      LXI       H,00103H      ;SET TIME OUT INDICATOR
0778 22D00D      SHLD      FPYTIME      TIME TO BE BETWEEN 2
077B El         POP       H
077C Dl         POP       D
077D Cl         POP       B
077E Fl         POP       PSW
077F D304        OUT       004H          ;ISSUE COMMAND TO FLOPPY
0781 C9         RET

        if not mpm20
        FPYTIME: DW          0
        FPYTCNT: DW          0
        endif

```

THIS IS THE DELAY ROUTINE. IT WILL LOOP HERE FOR
NUMBER OF MILLISECONDS SPECIFIED IN REGISTER C.

```

-----
                DELAY:
0782 0664 DEL1:  Mvi B,100          ;FORCE DELAY FOR 1 MILL7
0784 00  DEL2:   NOP              ;INSTRUCTIONS TO FILL It
0785 29          DAD             H
0786 29          DAD             H
0787 05          DCR             B          ;AT ONE MILLISECOND YET
0788 C28407      JNZ             DEL2      ;NO, KEEP ON LOOPING
078B 0D          DCR             C          ;END OF REQUESTED INTERI
078C C28207      JNZ             DEL1      ;NO, KEEP ON
078F C9          RET              ;RETURN TO CALLER

```

```

; NOTE: THE INITIALIZATION CODE WILL BE
; OVERWRITTEN BY DIRBUF & FPYBUF

```

```

                if          not mpm20
DIRBUF EQU $
                endif

```

DISK CONFIGURATION TABLE

```

-----
                IF          HARDSK
                PIN C
0790 0000000000 DSCNO: DB 00H,00H,00H,00H,00H,00H,00H,00H
0798 1000000000 DB 10H,00H,00H,00H,00H,00H,10H,00H
07A0 9090900000 DB 90H,90H,90H,00H,00H,00H,00H,00H
07A8 0000000000 DB 00H,00H,00H,00H,00H,00H,00H,00H
07B0 1000002000 DB 10H,00H,00H,20H,00H,00H,10H,20H
07B8 0000000000 DB 00H,00H,00H,00H,00H,00H,00H,00H
07C0 9090902000 DB 90H,90H,90H,20H,00H,00H,00H,20H
07C8 909090A0A0 DB 90H,90H,90H,0A0H,0A0H,0A0H,0H,0H
                ENDIF

```

SET UP DISK CONFIGURATON

```

; [ THIS CODE EXECUTED ONLY ONCE

```

```

-----
07D0 217E00 SDCONF: LXI      H,SELO+2          ;POINT TO DRIVE C:
07D3 3AB600      LDA      MPARMS
07D6 E605        ANI      05H          ; TEST FOR FOUR FLOPPIES
07D8 C3DE07      jmp      SDDBL      ; YES SKIP THE ZAP
07DB 77          MOV      M,A
07DC 23          INX      H          ZAP C: AND D:
07DD 77          MOV      M,A
                SDDBL:
07DE 118000      LXI      D,SELO+4          ;POINT TO DRIVE E:

```

ADDRESS	HEX	ASSEMBLY	OPERATION	REMARKS
07E1	DB25	IF	HARDSK	
07E3	E607	IN	025H	;READ CONFIGURATION PORT
07E5	17	ANI	07H	;STRIP OFF HIGH PART
07E6	17	RAL		
07E7	17	RAL		
07E8	0600	MVI	B,0	
07EA	4F	MOV	C,A	;POINT TO CONFIGURATION TAB
07EB	219007	LXI	H,DSCNO	
07EE	09	DAD	B	INDEX TO RIGHT ENTRY
07EF	0608	MVI	B,8	
07F1	7E SDL1:	MOV	A,M	CHANGE ALL SELECT MASKS
07F2	12	STAX	D	
07F3	13 SDOK:	INX	D	NEXT
07F4	23	INX	H	DRIVE
07F5+10FA		DJNZ	SDL1	
		DB	010H,SDL1-\$-1	---- FAKE DJNZ INSTRUCTION
		ENDIF		
		IF	NOT HARDSK	
		XCHG		
		MVI	B,8	
		XRA	A	
	SDL2:	MOV	M,A	ZAP ALL HARD DRIVES
		INX	H	
		DJNZ	SDL2	
		ENDIF		
07F7	C9	RET		
07F8 =	INITEND	EQU	\$	
07F8 E5	XETMOD:	PUSH	H	;SAVE MODE BYTE ADDRESS
07F9 79		MOV	A,C	;SETUP FOR DENSITY CHANGE
07FA B7		ORA	A	
07FB 0E00		MVI	C,000H	;ASSUME SINGLE DENSITY MODE
		JRZ	XETSEL	;VERIFY ASSUMPTION
07FD+2802		DB	028H,XETSEL-\$-1	---- FAKE JRZ INSTRUCTION
07FF 0E01		MVI	C,001H	;SET FOR DOUBLE DENSITY MOD
0801 CD7802	XETSEL: CALL	SETDEN		;SET DENSITY BASED ON LOW B
0804 E1		POP	H	;RESTORE
0805 6E		MOV	L,M	;PICKUP MODE AGAIN
0806 2600		MVI	H,000H	;FOR SINGLE BYTE PRECISION
0808 7D		MOV	A,L	;SAVE MODE IN ACCUMULATOR F
0809 29		DAD	H	2
080A 29		DAD	H	4
080B E5		PUSH	H	;SAVE * 4
080C 29		DAD	H	; * 8
080D D1		POP	D	;REGAIN * 4
080E 19		DAD	D	; * 12
080F 119101		LXI	D,MODLO	;FIRST MODEL DPE
0812 19		DAD	D	;POINT TO THIS ONE
0813 EB		XCHG		;SETUP TEMPORARILY AS DESTI
0814 2AB200		LHLD	DPEPTR	;ADDRESS OF CURRENTLY SELEC
0817 EB		XCHG		;SETUP TO ALTER
0818 010C00		LXI	B,12	;LENGTH FOR MOVE LDIR DO MOVE
081B+EDB0		DB	0EDH,0B0H	---- FAKE LDIR INSTRUCTION
081D C9		RET		;RETURN TO CALLER

THE FOLLOWING AREA CONTAINS THE DISK/WORK SAVE AREA
USED BY THE CBIOS IN THE NORMAL COURSE OF ACTIVITY.

```

                if      mpm20
;tempbuf      equ      (dirbuf-base)+128
                else
TEMPBUF EQU      (DIRBUF-BASE)+256
                ORG TEMPBUF+((INITEND-BASE)/TEMPBUF)*((INITEND-BASE
                endif

081E    BEGDAT    EQU      $          ;START OF BDOS AREA
;DIRBUF:    DS      128          ;OVERLAYS SYSTEMINIT CODE
081E    ALV0:    DS      32
083E    CSV0:    DS      32
085E    ALV1:    DS      32
087E    CSV1:    DS      32
089E    ALV2:    DS      32
08BE    CSV2:    DS      32
08DE    ALV3:    DS      32
08FE    CSV3:    DS      32
                IF      HARDSK
091E    ALV4:    DS      64
095E    CSV4:    DS      0
095E    ALV5:    DS      64
099E    CSV5:    DS      0
099E    ALV6:    DS      64
09DE    CSV6:    DS      0
09DE    ALV7:    DS      64
0A1E    CSV7:    DS      0
0A1E    ALV8:    DS      64
0A5E    CSV8:    DS      0
0A5E    ALV9:    DS      64
0A9E    CSV9:    DS      0
0A9E    ALVA:    DS      36
0AC2    CSVA:    DS      0
0AC2    ALVB:    DS      36
0AE6    CSVB:    DS      0
                endif
                if      mdisk
ALVC:    DS      32          ;VIRTUAL DISK
CSVC:    DS      0
                endif

                if      not mpm20
                if      hardsk
                DS      1          ;MUST PRECEDE HSTBU
HSTBUF:  DS      1024          ;HOST BUFFER AREA
                DS      1          ;MUST FOLLOW HSTBUF
                ENDIF
FPYBUF    EQU      DIRBUF+128      FLOPPY 1/0 BUFFER
                endif

0AE6    NEWDSK:  DS      1          ;SEEK DISK NUMBER
0AE7    NEWTRK:  DS      2          ;SEEK TRACK NUMBER
0AE9    NEWSEC:  DS      1          ;SEEK SECTOR NUMBER

```

0AEA	HSTDSK:	DS	1	;HOST DISK NUMBER
0AEB	HSTTRK:	DS	2	;HOST TRACK NUMBER
0AED	HSTSEC~.	DS	1	;HOST SECTOR NUMBER
0AEE	NEWHST:	DS	1	;SEEK SHR SECSHF
0AEF	HSTACT:	DS	1	;HOST ACTIVE FLAG
0AF0	HSTWRT:	DS	1	;HOST WRITTEN FLAG
0AF1	UNACNT:	DS	1	;UNALLOCATED RECORD
0AF2	UNADSK:	DS	1	;LAST UNALLOCATED D
0AF3	UNATRK:	DS	2	;LAST UNALLOCATED T
0AF5	UNASEC:	DS	1	;LAST UNALLOCATED S
0AF6	ERFLAG:	DS	1	;ERROR REPORTING
0AF7	RSFLAG:	DS	1	;READ SECTOR FLAG
0AF8	READOP:	DS	1	;1 IF READ OPERATIO
0AF9	WRTYPE:	DS	1	;WRITE OPERATION TY
0AFA 00	CMD:	DB	0	;COMMANDS FOR NEXT
0AFB 00	MASK:	DB	0	;STATUS MASKS BUF
0AFC 00	STATUS:	DB	0	;STATUS SAVE LOCATI
0AFD 00000000	SAVE1:	DB	000H,000H,000H,000H	;SAVE AREA FOR NMI
0B01 00	P\$RETRIES:	DB	000H	;COUNTER FOR PERMAN
0B02 00	T\$RETRIES:	DB	000H	;COUNTER FOR TEMPOR
0B03 00	HOME\$TOGGLE:	DB	000H	;INDICATOR TO TELL IF HOME SHOULD

page

```

                if          mpm20

; M P / M      2 0 C 0 M M 0 N   B A S E

        commonbase:
0B04 C3150B          jmp          coldstart
0B07 C30000          swtuser: jmp          $-$
0B0A C30000          swtsys:  jmp          $-$
0B0D C30000          pdisp:   jmp          $-$
0B10 C30000          xdos:    jmp          $-$
0B13 0000            sysdat:  dw          $-$

        COLDSTART:
        WARMSTART:
0B15 0E00            mvi        C'o          SEE SYSTEM INIT
                                           COLD & WARM START INCLUDE
                                           FOR COMPATIBILITY WITH CP
0B17 C3100B          jmp          XDOS        SYSTEM RESET, TERMINATE P

        rtnempty:
0B1A AF              xra          a
0B1B C9              ret

        NULL$INT:
0B1C FB              EI
                        RETI
0B1D+ED4D            DB          0EDH,04DH      ---- FAKE RETI INSTRUCTION
                        endif

CENTRONICS PRINTER ROUTINE (WITH SEPARATE BUSY TEST

0B1F 3E01CNSTAT:    MVI          A,001H          ;TO SET STROBE HIGH
0B21 D310            OUT          010H
0B23 DB10            IN           010H          ;READ PRINTER STATUS
0B25 E620            ANI          020H          ;REMOVE ALL BUT BUSY BIT
0B27 3EFF            MVI          A,0FFH          ;ASSUME NOT BUSY
0B29 C8              RZ              ;CHECK ASSUMPTION
0B2A AF              XRA          A              ;SET TO SHOW STILL BUSY
0B2B C9              RET

        CLIST:
0B2C CD1F0B          CALL         CNSTAT          ;IS PRINTER READY NOW?
0B2F B7              ORA          A
                        JRNZ        CLIST1          ;IF READY, SKIP POLL
0B30+2009            DB          020H,CLIST1-$-1  ---- FAKE JRNZ INSTRUCTION
0B32 C5              PUSH         B
0B33 0E83            MVI          C,POLL          POLL DEVICE
0B35 1E00            Mvi          E,PLLPT        PRINTER
0B37 CD100B          CALL         XDOS          ;WAIT FOR PRINTER TO FREE U
0B3A C1              POP          B

        CLIST1:
0B3B 79              MOV          A,C              ;CHARACTER TO PRINT
0B3C D311            OUT          011H          ;WRITE IT TO DATA PORT
0B3E 3E00            MVI          A,000H          ;TO FORCE STROBE LOW
0B40 D310            OUT          010H
0B42 3E01            MVI          A,001H          ;TO FORCE STROBE HIGH
0B44 D310            OUT          010H
0B46 C9              RET

```

PAGE

DISK INTERRUPT ROUTINE

```

-----

FLOPPY$INT:
0B47 22C80D      SHLD      SVDHL
0B4A 21500B      LXI       H,FDINTH
0B4D C37F0D      jmp      INTINIT
      FDINTH:
0B50 DB04        IN        004H          ;GET STATUS
0B52 32FC0A      STA      STATUS        ;SAVE FOR 1/0 ROUTINE
0B55 3E00        MVI      A,0          ;STOP TIMING OF RESPON~
0B57 32D10D      STA      FPYTIME+1
0B5A 1E06        MVI      E,FPYFLAG    ;SHOW 1/0 COMPLETED
      JR        HDSTFLG
0B5C+1813        DB 018H,HDSTFLG-$-1  ---- FAKE JR I]
      HARD$INT:
0B5E 22C80D      SHLD      SVDHL
0B61 21670B      LXI       H,HDINTH
0B64 C37F0D      jmp      INTINIT
      HDINTH:
0B67 DB24        IN        024H          ;GET STATUS
0B69 32FC0A      STA      STATUS        ;SAVE FOR CHECK LATER
0B6C AF          XRA      A
0B6D D323        OUT      023H          ;RESET INTERRUPT BY REL(
0B6F 1E05        MVI      E,HDFLAG    ;SHOW 1/0 COMPLETED
      HDSTFLG:
0B71 0E85        MVI      C,FLAGST
0B73 CD100B      CALL     XDOS
0B76 C3670D      jmp      INTDONE

```

PAGE

CONSOLE DISPLAY ROUTINES

CONST:			CONSOLE STATUS	
0B79	CD9A0B	CALL	PTBLJMP	COMPUTE AND JUMP TO HNDLR
0B7C	AD0B	DW	PT0ST	CONSOLE #0 STATUS ROUTINE
0B7E	EC0B	DW	PT1ST	CONSOLE #1 STATUS ROUTINE
0B80	2B0C	DW	PT2ST	CONSOLE #2 STATUS ROUTINE
0B82	6A0C	DW	PT3ST	CONSOLE #3 STATUS ROUTINE
CONIN:			CONSOLE INPUT	
0B84	CD9A0B	CALL	PTBLJMP	COMPUTE AND JUMP TO HNDLR
0B87	B80B	DW	PT0IN	CONSOLE #0 INPUT
0B89	F70B	DW	PT1IN	CONSOLE #1 INPUT
0B8B	360C	DW	PT2IN	CONSOLE #2 INPUT
0B8D	750C	DW	PT3IN	CONSOLE #3 INPUT
CONOUT:			CONSOLE OUTPUT	
0B8F	CD9A0B	CALL	PTBLJMP	; COMPUTE AND JUMP TO HNDLR
0B92	CA0B	DW	PT0OUT	; CONSOLE #0 OUTPUT
0B94	090C	DW	PT1OUT ;	CONSOLE #1 OUTPUT
0B96	480C	DW	PT2OUT	CONSOLE #2 OUTPUT
0B98	870C	DW	PT3OUT	CONSOLE #3 OUTPUT
PTBLJMP:			COMPUTE AND JUMP TO HANDLER	
			D = CONSOLE #	
			DO NOT DESTROY <D>	
0B9A	7A	Mov	A, D	
0B9B	FE04	CPI	NMBCNS	
		JRC	TBLJMP	
0B9D+3803		DB	038H, TBLJMP-\$-1	;----- FAKE JRC INSTRUCTION
0B9FF1		POP	PSW	THROW AWAY TABLE ADDRESS
0BA0AF		XRA	A	
0BA1C9		RET		
TBLJMP:			COMPUTE AND JUMP TO HANDLER	
			A = TABLE INDEX	
0BA287		ADD	A	DOUBLE TABLE INDEX FOR ADR OFFST
0BA3E1		POP	H	RETURN ADR POINTS TO JUMP TBL
0BA45F		Mov	E, A	
0BA51600		MVI	D, 0	
0BA7 19		DAD	D	ADD TABLE INDEX * 2 TO TBL BASE
0BA8 5E		Mov	E, M	GET HANDLER ADDRESS
0BA9 23		INX	H	
0BAA 56		MOV	D, M	
0BAB EB		XCHG		
0BAC E9		PCHL		JUMP TO COMPUTED CNS HANDLER

SERIAL PORT ADDRESS EQUATES

001C =	DATA0	EQU	01CH	;CONSOLE #0 DATA
001D =	STS0	EQU	DATA0+1	;CONSOLE #0 STATUS
002C =	DATA1	EQU	02CH	;CONSOLE #1 DATA
002D =	STS1	EQU	DATA1+1	;CONSOLE #1 STATUS
002E =	DATA2	EQU	02EH	;CONSOLE #2 DATA
002F =	STS2	EQU	DATA2+1	;CONSOLE #2 STATUS
002A =	DATA3	EQU	02AH	;CONSOLE #3 DATA
002B =	STS3	EQU	DATA3+1	;CONSOLE #3 STATUS
001E =	LPTPRT0	EQU	01EH	;PRINTER #0 DATA
001F =	LPTSTS0	EQU	LPTPRT0+1	;PRINTER #0 STATUS
0028 =	LPTPRT1	EQU	028H	;PRINTER #1 DATA
0029 =	LPTSTS1	EQU	LPTPRT1+1	;PRINTER #1 STATUS

PAGE

POLL CONSOLE # 0 INPUT

	POLCIO:			TEST CONSOLE STATUS
	PTOST:			RETURN OFFH IF READY
OBAD AF	XRA	A		000H IF NOT
OBAE D31D	OUT	STSO		
OBBO DB1D	IN	STSO		
OB B2 E601	ANI	1		RX CHAR ?
OB B4 C8	RZ			NO
OB B5 3EFF	MVI	A, OFFH		YES- SET FLAG
OB B7 C9	RET			

CONSOLE # 0 INPUT

- - - - -

	PTOIN:			RETURN CHAR IN REG A
OB B8 CDAD0B	CALL	POLCIO		;IS IT READY NOW?
OB B9 B7	ORA	A		
	JRNZ	PTOIN1		;IF READY, SKIP POLL
OB B C +2007	DB	020H, PTOIN1-\$-1		---- FAKE JRNZ INSTRUCTION
OB B E 0E83	MVI	C, POLL		
OB C 0 1E05	MVI	E, PLCIO		POLL CONSOLE #0 INPUT
OB C 2 CD100B	CALL	XDOS		
OB C 5 Db1C PTOIN1:	IN	DATA0		READ CHARACTER
OB C 7 E67F	ANI	7FH		STRIP PARITY
OB C 9 C9	RET			

CONSOLE # 0 OUTPUT

	PTOOUT:	REG C = CHAR TO OUTPUT		---
OB C A CDDD0B	CALL	POLC00		;IS IT READY NOW?
OB C D B7	ORA	A		
	JRNZ	PTOOUT1		;IF READY, SKIP POLL
OB C E +2009	DB	020H, PTOOUT1-\$-1		---- FAKE JRNZ INS
OB D 0 C5	PUSH	B		
OB D I 0E83	MVI	C, POLL		
OB D 3 1E01	MVI	E, PLC00		
OB D 5 CD100B	CALL	XDOS		POLL CONSOLE #0 OUTPUT
OB D 8 C1	POP	B		
	PTOOUT1:			
OB D 9 79	Mov	A, C		
OB D A D31C	OUT	DATA0		TRANSMIT CHARACTER
OB D C C9	RET			

POLL CONSOLE # 0 OUTPUT

```

POLC00:
OBDD 3E10      MVI      A,10H      RETURN OFFH IF READY
OBDF D31D      OUT      STS0      000H IF NOT
OBE1 DB1D      IN       STS0      RESET INT BIT
OBE3 E60C      ANI      0CH       READ STATUS
OBE5 FE0C      CPI      0CH       MASK FOR DTR AND TXE
OBE7 3E00      MVI      A,0        MUST HAVE BOTH
OBE9 C0        RNZ
OBEA 3D        DCR      A          RETURN NOT READY
OBEB C9        RET              ;CHANGE "A" TO OFFH
                                RETURN READY
```

PAGE

POLL CONSOLE # 1 INPUT

```

                POLCIL:
                PT1ST:
OBEC AF          XRA      A          TEST CONSOLE STATUS
OBED D32D         OUT     STS1       RETURN OFFH IF READY
OBEF DB2D         IN      STS1       000H IF NOT
OBF1 E601         ANI     1          RX CHAR ?
OBF3 C8           RZ              NO
OBF4 3EFF         MVI     A,OFFH     YES      - SET FLAG
OBF6 C9           RET

```

CONSOLE # 1 INPUT

```

                PT1IN:
OBF7 CDEC0B       CALL    POLCIL     RETURN CHAR IN REG A
OBFA B7           ORA     A          ;READY NOW?
                JRNZ     PT1IN1      ;IF READY, SKIP POLL
OBF8+2007         DB      020H,PT1IN1-$-1  ---- FAKE JRNZ INSTRUCTION
OBF9 0E83         MVI     C,POLL
OBF8 1E06         MVI     E,PLCIL
OC01 CD100B       CALL    XDOS
OC04 DB2C PT1IN1: IN     DATA1      POLL CONSOLE #1 INPUT
OC06 E67F         ANI     7FH        READ CHARACTER
OC08 C9           RET              STRIP PARITY

```

CONSOLE # 1 OUTPUT

```

                PT1OUT:
OC09 CD1C0C       CALL    POLC01     REG C = CHAR TO OUTPUT
OC0C B7           ORA     A          ;ARE WE READY NOW?
                JRNZ     PT1OUT1      ;IF READY, SKIP POLL
OC0D+2009         DB      020H,PT1OUT1-$-1  ---- FAKE JRNZ INS
OC0F C5 PUSH      B
OC10 0E83         MVI     C,POLL
OC12 1E02         MVI     E,PLC01
OC14 CD100B       CALL    XDOS        POLL CONSOLE #1 OUTPUT
OC17 C1           POP     B
PT1OUT1:
OC18 79           MOV     A,C
OC19 D32C         OUT     DATA1      TRANSMIT CHARACTER
OC1B C9 RET

```

POLL CONSOLE # 1 OUTPUT

POLC01:				
0C1C	3E10	MVI	A,10H	RETURN OFFH IF READY
0C1E	D32D	OUT	STS1	000H IF NOT
0C20	DB2D	IN	STS1	RESET INT BIT
0C22	E60C	ANI	0CH	READ STATUS
0C24	FE0C	CPI	0CH	MASK FOR DTR AND TXE
0C26	3E00	MVI	A,0	MUST HAVE BOTH
0C28	C0	RNZ		RETURN NOT READY
0C29	3D	DCR	A	;CHANGE "A" TO OFFH
0C2A	C9	RET		RETURN READY

PAGE

POLL CONSOLE # 2 INPUT

; -----

		POLCI2:			
		PT2ST:			TEST CONSOLE STATUS
0C2B	AF	XRA	A		RETURN OFFH IF READY
0C2C	D32F	OUT	STS2		000H IF NOT
0C2E	DB2F	IN	STS2		
0C30	E601	ANI	1		RX CHAR ?
0C32	C8	RZ			NO
0C33	3EFF	MVI	A, OFFH		YES -SET FLAG
0C35	C9	RET			

; CONSOLE # 2 INPUT

		PT21N:			RETURN CHAR IN REG A
0C36	CD2B0C	CALL	POLCI2		;READY NOW?
0C39	B7	ORA	A		
		JRNZ	PT21N1		;IF READY, SKIP POLL
0C3A+2007		DB	020H, PT21N1-\$-1		---- FAKE JRNZ INSTRUCTION
0C3C	0E83	MVI	C, POLL		
0C3E	1E07	MVI	E, PLC12		POLL CONSOLE #2 INPUT
0C40	CD100B	CALL	XDOS		
0C43	DB2E PT21N1:	IN	DATA2		READ CHARACTER
0C45	E67F	ANI	7FH		STRIP PARITY
0C47	C9	RET			

CONSOLE # 2 OUTPUT

		PT20UT:			REG C = CHAR TO OUTPUT
0C48	CD5B0C	CALL	POLC02		;READY NOW?
0C4B	B7	ORA	A		
		JRNZ	PT20UT1		;IF READY, SKIP POLL
0C4C+2009		DB	020H, PT20UTI-\$-1		---- FAKE JRNZ INS
0C4E	C5	PUSH	B		
0C4F	0E83	MVI	C, POLL		
0C51	1E03	MVI	E, PLC02		
0C53	CD100B	CALL	XDOS		POLL CONSOLE #2 OUTPUT
0C56	C1	POP	B		
		PT20UTI:			
0C57	79	Mov	A, C		
0C58	D32E	OUT	DATA2		TRANSMIT CHARACTER
0C5A	C9	RET			

POLL CONSOLE # 2 OUTPUT

- - - - -

```

POLC02:
0C5B 3E10      MVI      A,10H      ; RETURN OFFH IF READY
0C5D D32F      OUT      STS2      000H IF NOT
0C5F DB2F      IN       STS2      RESET INT BIT
0C61 E60C      ANI      0CH       READ STATUS
0C63 FEW       CPI      0CH       MASK FOR DTR AND TXE
0C65 3E00      MVI      A,0       MUST HAVE BOTH
0C67 C0        RNZ                RETURN NOT READY
0C68 3D        DCR      A         ;CHANGE "A" TO OFFH
0C69 C9        RET                RETURN READY
```

PAGE

```
; POLL CONSOLE # 3 INPUT
```

11

POLCI3:

PT3ST: ; TEST CONSOLE STATUS

0C6A AF	XRA	A	RETURN OFFH IF READY
0C6B D32B	OUT	STS3	000H IF NOT
0C6D DB2B	IN	STS3	
0C6F E601	ANI	1	RX CHAR ?
0C71 C8	RZ		NO
0C72 3EFF	MVI	A, OFFH	YES - SET FLAG
0C74 C9	RET		

CONSOLE # 3 INPUT

PT31N:

RETURN CHAR IN REG A

0C75	CD6A0C	CALL	POLCI3	;READY NOW?
0C78	B7	ORA	A	
		JRNZ	PT31N1	;IF READY, SKIP POLL
0C79-~-2007		DB	020H,PT3IN1-\$-1	---- FAKE JRNZ INSTRUCTION
0C7B	0E83	MVI	C,POLL	
0C7D	1E08	MVI	E,PLC13	POLL CONSOLE #3 INPUT
0C7F	CD100B	CALL	XD0S	
0C82	DB2A PT31N1:	IN	DATA3	READ CHARACTER
0C84	E67F	ANI	7FH	STRIP PARITY
0C86	C9	RET		

CONSOLE # 3 OUTPUT

PT30UT:

```
; REG C = CHAR TO OUTPUT
```

0C87	CD9A0C	CALL	POLC03	;READY NOW?
0C8A	B7	ORA	A	
		JRNZ	PT30UTI	;IF READY, SKIP POLL
0C8B+2009		DB	020H,PT30UT1-\$-1	---- FAKE JRNZ INS
0C8D	C5	PUSH	B	
0C8E	0E83	MVI	C,POLL	
0C90	1E04	MVI	E,PLC03	
0C92	CD100B	CALL	XD0S	POLL CONSOLE #3 OUTPUT
0C95	C1	POP	B	

PT30UT1:

```

0C96 79          MOV     A,C
0C97 D32A        OUT     DATA3      TRANSMIT CHARACTER
0C99 C9         RET

```

POLL CONSOLE # 3 OUTPUT

```
      POLC03:
0C9A 3E10      MVI      A,10H      RETURN OFFH IF READY
0C9C D32B      OUT      STS3      000H IF NOT
0C9E DB2B      IN       STS3      RESET INT BIT
0CA0 E60C      ANI      0CH      READ STATUS
0CA2 FE0C      CPI      0CH      MASK FOR DTR AND TXE
0CA4 3E00      MVI      A,0       MUST HAVE BOTH
0CA6 C0        RNZ
0CA7 3D        DCR      A         RETURN NOT READY
0CA8 C9        RET              ;CHANGE "A" TO OFFH
                                RETURN READY
```

PAGE

LINE PRINTER #0 DRIVER

LIST:		;LIST OUTPUT #0	
OCA9 CDBC0C	CALL	POLLPT	
OCAC B7	ORA	A	;IS PRINTER READY NOW?
	JRNZ	LIST1	;IF READY, SKIP POLL
OCAD+2009	DB	020H,LIST1-\$-1	; ---- FAKE JRNZ INSTRUCTION
OCAF C5	PUSH	B	
OCB0 0E83	Mvi	C,POLL	; POLL PRINTER STATUS
OCB2 1E00	MVI	E,PLLPT	
OCB4 CD100B	CALL	XDOS	
OCB7 C1	POP	B	

LIST1:			
OCB8 79	MOV	A,C	CHARACTER TO PRINT
OCB9 D31E	OUT	LPTPRTO	
OCBB C9	RET		

POLL PRINTER OUTPUT

POLLPT:		RETURN OFFH IF READY	
OCBC 3E10	MVI	A,10H	000H IF NOT
OCBE D31F	OUT	LPTSTS0	RESET INT BIT
OCC0 DB1F	IN	LPTSTS0	READ STATUS
OCC2 E60C	ANI	0CH	MASK FOR DTR AND TXE
OCC4 FE0C	CPI	0CH	MUST HAVE BOTH
OCC6 3E00	MVI	A,0	
OCC8 C0	RNZ		RETURN NOT READY
OCC9 3D	DCR	A	;CHANGE "A" TO OFFH
OCCA C9	RET		RETURN READY

PAGE

```
; MP/M 1.0          EXTENDED 1/0 SYSTEM
```

```
POLLDEVICE:
```

```
    REG C = DEVICE # TO BE POLLED
    RETURN OFFH IF READY,
    000H IF NOT
```

```
OCCB 79          MOV     A,C
OCCC FE09        CPI     NMBDEV
                  JRC     DEVOK
OCCE+3802        DB      038H,DEVOK-$-1    ---- FAKE JRC INSTRUCTI
OCD0 3E09        MVI     A,NMBDEV;          IF DEV # >= NMBDEV,
                                          SET TO NMBDEV
```

```
    DEVOK:
```

```
OCD2 CDA20B      CALL    TBLJMP            JUMP TO DEV POLL CODE
```

```
DEVTBL:
```

```
OCD5 BC0C        DW      POLLPT           POLL PRINTER OUTPUT - THIS WIL
                                          SPECIFIED PARALLEL PORT FOR P
OCD7 DD0B        DW      POLC00           POLL CONSOLE #0 OUTPUT
OCD9 1C0C        DW      POLC01           POLL CONSOLE #1 OUTPUT
OCDB 5B0C        DW      POLC02           POLL CONSOLE #2 OUTPUT
OCDD 9A0C        DW      POLC03           POLL CONSOLE #3 OUTPUT
OCDF ADOB        DW      POLCIO          POLL CONSOLE #0 INPUT
OCE1 ECOB        DW      POLCI1          POLL CONSOLE #1 INPUT
OCE3 2B0C        DW      POLCI2          POLL CONSOLE #2 INPUT
OCE5 6A0C        DW      POLCI3          POLL CONSOLE #3 INPUT
0009            NMBDEV EQU      ($-DEVTBL)/2
OCE7 1A0B        DW      RTNEMPTY         ; BAD DEVICE HANDLER
```

```
PAGE
```

```

;SELECT / PROTECT MEMORY
;
; REG BC = ADDR OF MEM. DESCRIPTOR
; BC ->  BASE   1 BYTE,
;        SIZE   1 BYTE,
;        ATTRIB 1 BYTE,
;        BANK   1 BYTE.
;
; BIOS TABLE MODIFIED

      SELMEMORY:
0CE9 FE20      CPI      20H
0CEB CAEBOC    JZ       $      ; tomj MAKES NO SENSE!
0CEE 210300    LXI      H,3    ; POINT TO BANK
0CF1 09        DAD      B
0CF2 7E        MOV      A,M    ; GET IT
0CF3 32030D    STA      BANKNO ; SAVE BANK NUMBER
0CF6 17        RAL
0CF7 17        RAL
0CF8 17        RAL
0CF9 E618      ANI      018H   ; MASK FOR PIO
0CFB F602      ORI      MEMSK
0CFD 32040D    STA      CURMEM ; STORE CURRENT BANK MASK
0D00 D309      OUT      009H   ; SET PIO
0D02 C9        RET

0D03 00 BANKNO: DB      0      ; LAST SELECTED MEMORY BANK NUMBER
0D04 00 CURMEM: DB      0      ; LAST SELECTED MEMORY BANK MASK

; START CLOCK
      STARTCLOCK:
; WILL CAUSE FLAG #1 TO BE SET
; AT EACH SYSTEM TIME UNIT TICK
0D05 3EFF      MVI      A,OFFH
0D07 32CE0D    STA      TICKN
0DOA C9        RET

; STOP CLOCK
      STOPCLOCK:
; WILL STOP FLAG #1 SETTING AT
; SYSTEM TIME UNIT TICK
0DOB AF        XRA      A
0DOC 32CE0D    STA      TICKN
0DOF C9        RET

; EXIT REGION
      EXITREGION:
; EI IF NOT PREEMPTED
0D10 3ACF0D    LDA      PREEMP
0D13 B7        ORA      A
0D14 C0        RNZ
0D15 FB        EI
0D16 C9        RET

```

; MAXIMUM CONSOLE NUMBER

MAXCONSOLE:

```
OD17 3E04      MVI      A,NMBCNS
OD19 C9        RET
```

; MP/M 1.0 INTERRUPT HANDLERS

```
008E          DSPTCH   EQU      142
```

INT1HND:

; INTERRUPT 1 HANDLER ENTRY POINT

T20MS:

```
OD1A 22C80D    SHLD     SVDHL
OD1D 21220D    LXI      H,TIMERINT
                JR       INTINIT
OD20+185D      DB 018H,INTINIT-$-1
```

----- FAKE JR INSTR

TIMERINT:

```
OD22 3ACE0D    LDA      TICKN
OD25 B7        ORA      A
                ; TEST TICKN, INDICATES
                ; DELAYED PROCESS (ES)
```

```
                JRZ      NOTICKN
OD26+2807      DB 028H,NOTICKN-$-1
OD28 0E85      MVI      C,FLAGST
```

----- FAKE JRZ INST

```
OD2A 1E01      MVI      E,1
OD2C CD100B    CALL     XDOS
                ; SET FLAG #1 EACH TICK
```

NOTICKN:

```
OD2F 219D0D    LXI      H,CNTX
OD32 35        DCR      M
                ; DEC TICK CNTR
```

```
                JRNZ     NOT1SEC
OD33+2032      DB 020H,NOT1SEC-$-1
OD35 3E7D      MVI      A,125
```

----- FAKE JRNZ INS

```
OD37 2B        DCX      H
OD38 96        SUB      M
OD39 77        MOV      M,A
                ; TOGGLE COUNT 62 <-> 6
```

```
OD3A 23        INX      H
OD3B 77        MOV      M,A
                ; ACTUAL #ISEC = 62.5
```

```
OD3C 0E85      MVI      C,FLAGST
OD3E 1E02      MVI      E,2
```

```
OD40 CD100B    CALL     XDOS
OD43 2AD00D    LHLD     FPYTIME
                ; SET FLAG #2 @ 1 SEC
                ; IS FLOPPY TIME CHECK IN EF
```

```
OD46 7C        MOV      A,H
OD47 B7        ORA      A
```

```
                JRZ      NOT1SEC
OD48+281D      DB 028H,NOT1SEC-$-1
                ; IF NOT IN EFFECT, FINISH
                ; ----- FAKE JRZ INST
```

```
OD4A 2D        DCR      L
OD4B 22D00D    SHLD     FPYTIME
                ; SUBTRACT A SECOND
                ; SAVE FOR NEXT TIME
```

```
                JRNZ     NOT1SEC
                ; IF NOT TOO LONG, FINISH
```

```
OD4E+2017      DB 020H,NOT1SEC-$-1
OD50 65        MOV      H,L
                ; ----- FAKE JRNZ INS
                ; ZERO OUT INDICATOR
```

```
OD51 22D00D    SHLD     FPYTIME
OD54 0E85      MVI      C,FLAGST
OD56 1E06      MVI      E,FPYFLAG
```

```
OD58 CD100B    CALL     XDOS
OD5B 3E90      MVI      A,10010000B
                ; CAUSE 1/0 FOR FLOPPY TO CO
```

```
OD5D 32FC0A    STA      STATUS
                ; SHOW ERROR IN FLOPPY 1/0
```

```

0D60 2AD20D      LHLD    FPYTCNT
0D63 23          INX     H           ;COUNT TIMES WD1791 GOES TO
0D64 22D20D      SHLD    FPYTCNT

```

NOT1SEC:

INTDONE:

```

0D67 AF          XRA     A
0D68 32CF0D      STA     PREEMP      CLEAR PREEMPTED FLAG
0D6B C1          POP     B
0D6C D1          POP     D
0D6D 2ACA0D      LHLD    SVDSP
0D70 F9          SPHL    RESTORE STK PTR
0D71 F1          POP     PSW
0D72 2ACC0D      LHLD    SVDRET
0D75 E5          PUSH    H
0D76 210D0B      LXI     H,PDISP      MP/M DISPATCH
0D79 E5          PUSH    H           PUT ON STACK FOR RETURN
0D7A 2AC80D      LHLD    SVDHL

```

THE FOLLOWING DISPATCH CALL WILL FORCE ROUND ROBIN
SCHEDULING OF PROCESSES EXECUTING AT THE SAME PRIORITY
EACH 1/32ND OF A SECOND.

NOTE: INTERRUPTS ARE NOT ENABLED UNTIL THE DISPATCHER
RESUMES THE NEXT PROCESS. THIS PREVENTS INTERRUPT
OVER-RUN OF THE STACKS WHEN STUCK OR HIGH FREQUENCY
INTERRUPTS ARE ENCOUNTERED.

```

                                RETI           ; DISPATCH
0D7D+ED4D          DB       0EDH,04DH        ;---- FAKE RETI INSTRUCTION
                                INTINIT: ;SAVE MACHINE STATE FOR INTRPT HNDL
0D7F 22C60D          SHLD    ADRINTHD
0D82 E1            POP     141
0D83 22CC0D          SHLD    SVDRET
0D86 F5            PUSH    PSW
0D87 210000          LXI     H,0
0D8A 39            DAD     SP
0D8B 22CA0D          SHLD    SVDSP          SAVE USERS STK PTR
0D8E 31C60D          LXI     SP,LSTINTSTK    LCL STK FOR INTR HNDL
0D91 D5            PUSH    D
0D92 C5            PUSH    B
0D93 3EFF          MVI     A,OFFH
0D95 32CF0D          STA     PREEMP          SET PREEMPTED FLAG
0D98 2AC60D          LHLD    ADRINTHD
0D9B E9            PCHL    ;JUMP TO INTERRUPT HANDLER

```

; BIOS DATA SEGMENT

```

0D9C 3E    TOGCNT:  DB       62          TOGGLE COUNTER 62 <-> 63
0D9D 3E    CNTX:    DB       62          TICK CNTR TO 1 SEC
                                INTSTK:   LOCAL INTRPT STK
0D9E C7C7C7C7C7    DW       0C7C7H,0C7C7H,0C7C7H,0C7C7H,0C7C7H
0DA8 C7C7C7C7C7    DW       0C7C7H,0C7C7H,0C7C7H,0C7C7H,0C7C7H
0DB2 C7C7C7C7C7    DW       0C7C7H,0C7C7H,0C7C7H,0C7C7H,0C7C7H
0DBC C7C7C7C7C7    DW       0C7C7H,0C7C7H,0C7C7H,0C7C7H,0C7C7H
                                LSTINTSTK:
0DC6 0000 ADRINTHD: DW       0           INTERRUPT HANDLER ADDRESS

```

MP/M II System Guide

Appendix E MP/M Banked XIOS

ODC8 0000	SVDHL:	DW	0	SAVED REGS HL DURING INT HNDL
ODCA 0000	SVDSP:	DW	0	SAVED SP DURING INT HNDL
ODCC 0000	SVDRET:	DW	0	SAVED RETURN DURING INT HNDL
ODCE 00	TICKN:	DB	0	TICKING BOOLEAN, TRUE = DELAYEI
ODCF 00	PREEMP:	DB	0	PREEMPTED BOOLEAN
		if	mpm20	
	FPYTIME:			
ODD0 0000		DW	0	
	FPYTCNT:			
ODD2 0000		DW	0	
		endif		
		PAGE		

THESE ARE THE DISK TYPE DEFINITION BLOCKS
EACH OF WHICH CORRESPONDS TO A PARTICULAR MODE.

```

ODD4      DPB0:    EQU      $          ;VERSION 2.0, IBM SINGLE
ODD4 1A00      DW      26          ;SECTORS PER TRACK
ODD6 03        DB      3           ;BLOCK SHIFT
ODD7 07        DB      7           ;BLOCK SHIFT MASK
ODD8 00        DB      0           ;EXTENT MASK
ODD9 F200      DW      242         ;DISK SIZE MINUS 1
ODDB 3F00      DW      63          ;DIRECTORY MAX
ODDD C0        DB      192         ;ALLOCO
ODDE 00        DB      0           ;ALLOC1
ODDF 1000      DW      16          ;CHECK AREA SIZE
ODE1 0200      DW      2           ;OFFSET TO START TRACK

ODE3 = DPB1:    EQU      $          ;VERSION 2.0, IBM DOUBLE
ODE3 3400      DW      52          ;SECTORS PER TRACK
ODE5 04DB      DB      4           ;BLOCK SHIFT

ODE6 0F        DB      15          ;BLOCK SHIFT MASK
ODE7 01        DB      1           ;EXTENT MASK
ODE8 F200      DW      242         ;DISK SIZE MINUS 1
ODEA 7F00      DW      127         ;DIRECTORY MAX
ODEC C0        DB      192         ;ALLOCO
ODED 00        DB      0           ;ALLOC1
ODEE 2000      DW      32          ;CHECK AREA SIZE
ODF0 0200      DW      2           ;OFFSET TO START TRACK

ODF2 = DPB2:    EQU      $          ;VERSION 1.4 ALTOS DOUBLE D
ODF2 3000      DW      48          ;SECTORS PER TRACK
ODF4 04        DB      4           ;BLOCK SHIFT
ODF5 0F        DB      15          ;BLOCK SHIFT MASK
ODF6 00        DB      0           ;EXTENT MASK (1.4 COMPATABI
ODF7 E000      DW      224         ;DISK SIZE MINUS 1
ODF9 5F00      DW      95          ;DIRECTORY MAX
ODFB C0        DB      192         ;ALLOCO
ODFC 00        DB      0           ;ALLOC1
ODFD 1800      DW      24          ;CHECK AREA SIZE
ODFF 0200      DW      2           ;OFFSET TO START TRACK

                                IF      HARDSK
                                if      mpm20
                                DPB3:  DISKDEF 3,0,127,,16384,512,512,0,1,,0
0E01+=      DPB3    EQU      $ ;DISK PARM BLOCK
0E01+8000    DW      128          ;SEC PER TRACK
0E03+07      DB      7           ;BLOCK SHIFT
0E04+7F      DB      127         ;BLOCK MASK
0E05+07      DB      7           ;EXTNT MASK
0E06+FF01    DW      511          ;DISK SIZE-1
0E08+FF01    DW      511          ;DIRECTORY MAX
0E0A+80      DB      128          ;ALLOCO
0E0B+00      DB      0           ;ALLOC1
0E0C+0080    DW      8000H+CKSZ ;PERMANENT DISK WIT
0E0E+0100    DW      1           ;OFFSET
0000+=      XLT3    EQU      0           ;NO XLATE TABLE

```



```

        DPB4:    DISKDEF 4,0,127,,16384,512,512,0,513,,0
0E10+=    DPB4    EQU      $          ;DISK PARM BLOCK
0E10+8000    DW      128          ;SEC PER TRACK
0E12+07      DB      7           ;BLOCK SHIFT
0E13+7F      DB      127         ;BLOCK MASK
0E14+07      DB      7           ;EXTNT MASK
0E15+FF01    DW      511         ;DISK SIZE-1
0E17+FF01    DW      511         ;DIRECTORY MAX
0E19+80      DB      128         ;ALLOCO
0E1A+00      DB      0           ;ALLOC1
0E1B+0080    DW      8000H+CKSZ ;PERMANENT DISK WIT
0E1D+0102    DW      513         ;OFFSET
0000+=    XLT4    EQU      0           ;NO XLATE TABLE

        DPB5:    DISKDEF          5,0,127,,16384,512,512,0,1025,,0
0E1F+=    DPB5    EQU      $          ;DISK PARM BLOCK
0E1F+8000    DW      128         ;SEC PER TRACK
0E21+07      DB      7           ;BLOCK SHIFT
0E22+7F      DB      127         ;BLOCK MASK
0E23+07      DB      7           ;EXTNT MASK
0E24+FF01    DW      511         ;DISK SIZE-1
0E26+FF01    DW      511         ;DIRECTORY MAX
0E28+80      DB      128         ;ALLOCO
0E29+00      DB      0           ;ALLOC1
0E2A+0080    DW      8000H+CKSZ ;PERMANENT DISK
0E2C+0104    DW      1025        ;OFFSET
0000+=    XLT5    EQU      0           ;NO XLATE TABLE

        DPB6:    DISKDEF 6,0,127,,16384,288,512,0,513,,0
0E2E+=    DPB6    EQU      $          ;DISK PARM BLOCK
0E2E+8000    DW      128         ;SEC PER TRACK
0E30+07      DB      7           ;BLOCK SHIFT
0E31+7F      DB      127         ;BLOCK MASK
0E32+07      DB      7           ;EXTNT MASK
0E33+1F01    DW      287         ;DISK SIZE-1
0E35+FF01    DW      511         ;DIRECTORY MAX
0E37+80      DB      128         ;ALLOCO
0E38+00      DB      0           ;ALLOC1
0E39+0080    DW      8000H+CKSZ ;PERMANENT DISK
0E3B+0102    DW      513         ;OFFSET
0000+=    XLT6    EQU      0           ;NO XLATE TABLE
        else
DPB3:      DISKDEF 3,0,127,,16384,512,512,0,1
DPB4:      DISKDEF 4,0,127,,16384,512,512,0,513
DPB5:      DISKDEF 5,0,127,,16384,512,512,0,1025
DPB6:      DISKDEF 6,0,127,,16384,288,512,0,513
        endif
ENDIF

        if      mdisk
DPB7:      EQU      $          ;VIRTUAL DISK
            DW      24          ;SECTORS PER TRACK
            DB      3           ;BLOCK SHIFT
            DB      7           ;BLOCK SHIFT MASK
            DB      0           ;EXTENT MASK
            DW      142         ;DISK SIZE MINUS 1

```

```
DW      63      ;DIRECTORY MAX
DB      0COH    ;ALLOCO
DB      0        ;ALLOC1
DW      0        ;CHECK AREA SIZE
DW      0        ;OFFSET TO START TRACK
endif
```

;MOVE SUBROUTINE

```

                if      hardsk
RWMOVE:
0E3D D5          push   d
ME E5           push   h
0E3F CD070B      call   swtuser      ;switch in user bank
0E42 E1          POP    h
0E43 DI          POP    d
0E44 018000      lxi     b,128
                LDIR                    ;MOVE DATA TO/FROM BUFFER
0E47+EDB0        DB      0EDH,0BOH   ; ---- FAKE LDIR INSTRUCTION
0E49 CD0A0B      call   swtsys      ;switch system back in

```

;DATA HAS BEEN MOVED TO/FROM HOST BUFFER

```

0E4C 3AF90A      LDA     WRTYPE      ;WRITE TYPE ??
                if      mpm20
0E4F E601        ani     WRDIR      ;TO DIRECTORY ??
                JRZ      RWEND      ;NO, JUST END UP HERE
0E51+280D        DB      028H,RWEND-$-1  ---- FAKE JRZ INSTRUCTION
                else
                CPI     WRDIR      ;TO DIRECTORY ??
                JRNZ     RWEND      ;NO, JUST END UP HERE
                endif

```

CLEAR HOST BUFFER FOR DIRECTORY WRITE

```

0E53 3AF60A      LDA     ERFLAG      ;CHECK PRIOR TO DIR ACTIVIT
0E56 B7          ORA     A           ;ERRORS ??
                JRNZ     RWEND      ;SKIP IF SO
0E57+2007        DB      020H,RWEND-$-1  ; ---- FAKE JRNZ INSTRUCTION
0E59 AF          XRA     A           ;ZERO TO ACCUMULATOR
0E5A 32F00A      STA     HSTWRT      ;BUFFER WRITTEN
0E5D CD6D04      CALL    WRITEHST

```

RWEND:

```

0E60 3AF60A      LDA     ERFLAG
0E63 B7          ORA     A           ;IF ERRORS, RESET SO NO MAT
0E64 C8          RZ                ;NONE, JUST RETURN
0E65 21EA0A      LXI     H,HSTDSK
0E68 36FF        MVI     M,OFFH     ;CANT POSSIBLY MATCH, MUST
                ENDIF
0E6A C9          RET

```

MVDTB:

```

0E6B 2AAF00      LHLD    DMAADR      MOVE DATA TO FLOPPY BUF
0E6E E5          push   h
0E6F CD070B      call   swtuser      ;switch in user bank,
0E72 E1          POP    h           cannot access non-common BNKXIO
0E73 111D13      LXI     D,FPYBUF
0E76 018000      LXI     B,128      128 BYTES
                LDIR
0E79+EDB0        DB      0EDH,0BOH   ----FAKE LDIR INSTRUCTION
0E7B C30A0B      jmp     swtsys      ;switch system back in

```

		RET		
0E7E F5	MVDFB:	PUSH	PSW	MOVE DATA FROM FLOPPY BUF
0E7F 3AFA0A		LDA	CMD	
0E82 E620		ANI	20H	CHECK FOR READ
		JRNZ	MVDFX	NO - BYPAS MOVE
0E84+2013		DB	020H, MVDFX-\$-1	---- FAKE JRNZ INSTRUCTION
0E86 2AAF00		LHLD	DMAADR	
0E89 E5		push	h	
0E8A CD070B		call	swtuser	;switch in user bank,
0E8D D1		POP	d	cannot access non-common BNKXIO
0E8E 211D13		LXI	H,FPYBUF	
0E91 018000		LXI	B,128	128 BYTES
		LDIR		
0E94+EDB0		DB	0EDH,0BOH	FAKE LDIR INSTRUCTION
0E96 CDOA0B		call	swtsys	;switch system back in
0E99 F1	MVDFX:	POP	PSW	
0E9A C9		RET		
		IF	HARDSK	
0E9B		DS	1	;MUST PRECEDE HSTBU
0E9C	HSTBUF:	DS	1024	;HOST BUFFER AREA
129C		DS	1	;MUST FOLLOW HSTBUF
		ENDIF		
		PAGE		

INITIALIZE MP/M: REAL TIME CLOCK & DISKS

```

-----

129D =      dirbuf      if      mpm20
131D =      fpybuf      equ      $
                        equ      dirbuf+128
                        endif

      SYSTEMINIT:
;C      = BREAKPOINT RESTART NUMBER
;DE      = BREAKPOINT RESTART HANDLER ADDRESS
;HL      = DIRECT XIOS INTERCEPT JUMP TABLE ADDRESS

129D 225E13      SHLD      SVDJT
12A0 69          MOV      L,C
12A1 2600        MVI      H,0
12A3 29          DAD      H
12A4 29          DAD      H
12A5 29          DAD      H          ;HL = RESTART JUMP ADDRESS
12A6 226013      SHLD      SVDBPA
                        if      not mdisk
12A9 2A130B      lhld      sysdat
12AC 2E0F        mvi      1,15      ;hl nmbmemsegs
12AE 46          mov      b,m      ;b nmbmemsegs
test$bank$setup$loop:
12AF 23          inx      h
12B0 23          inx      h
12B1 23          inx      h
12B2 23          inx      h          ;hl memseg(i).bank
12B3 7E          mov      a,m
12B4 B7          ora      a
12B5 C2BF12      jnz      bank$setup
12B8 05          dcr      b
12B9 C2AF12      jnz      test$bank$setup$loop
12BC C3CE12      jmp      after$bank$setup
      bank$setup:
12BF 3EIA        Mvi      A,01AH      SELECT      BANK 3
12C1 CD4813      CALL     STMVTR      SET UP      VECTORS
12C4 3E12        MVI      A,012H      SELECT      BANK 2
12C6 CD4813      CALL     STMVTR      SET UP      VECTORS
12C9 3E0A        MVI      A,00AH      SELECT      BANK 1
12CB CD4813      CALL     STMVTR      SET UP      VECTORS
after$bank$setup:
else
                        mvi      a,lah      bank 3 select for directo
                        out      09h
                        lxi      h,0bffeh
                        mvi      a,0e5h
                        cmp      m
                        inx      h
                        jrnz      fill
                        cmp      m
                        jrz      dontfill
fill:
                        mov      m,a      ;set directory initialized
                        dcx      h

```

```

        mov     m,a
        lxi     b,07ffh           ;first 2 k of bank one gets
        lxi     h,0
        lxi     d,l
        mvi     a,0ah             select bank 1
        out     09h
        mvi     m,0e5h
        ldir
dontfill:
        endif
12CE 3E02      MVI     A,002H      SELECTBANK 0
12D0 CD4813    CALL    STMVTR      SET UP VECTORS

12D3 213717    lxi     h,ldrbiosbase+density$mask$offset
        LXI     H,1737H           MOVE PARAMETERS CHANGED B
12D6 117C00    LXI     D,SELO      THE SETUP PROGRAM
12D9 010400    LXI     B,4         4 SELECT MASKS
        LDIR
12DC+EDB0      DB         0EDH,0BOH    ---- FAKE LDIR INSTRUCTION
12DE 118800    LXI     D,MODE
12E1 010400    LXI     B,4         4 MODE BYTES
        LDIR
12E4+EDB0      DB         0EDH,0BOH    ---- FAKE LDIR INSTRUCTION
12E6 2ABB17    1hld     ldbiosbase+misc$params$offset
        LHLD    17BBH           GET MISC. PARAMETERS
12E9 22B600    SHLD    MPARMS
12EC 3AB600    LDA      MPARMS      NOW TEST FOR CENTRONICS P
12EF E602      ANI      2
        JRZ     PRTOK           NO- LEAVE SERIAL
12FJ+2814      DB         028H,PRTOK-$-1    ---- FAKE JRZ INSTRUCTION
12F3 212C0B    LXI     H,CLIST
12F6 221000    SHLD    WBOTE+13     CHANGE PRINTER ROUTINE
12F9 211F0B    LXI     H,CNSTAT     AND STATUS CHECK
12FC 22D50C    SHLD    DEVTBL
12FF 3E03      MVI     A,003H      ;INITIALIZE PARALLEL PORT
1301 D313      OUT     013H
1303 3E0F      MVI     A,00FH
1305 D313      OUT     013H

        PRTOK:
1307 010300    LXI     B,003H      ;SET THE MODE FOR DRIVES IN
        MODESET:
130A CD2F02    CALL    SELSDP      ;SELECT DRIVE FOR MODESET
130D 218800    LXI     H,MODE
1310 09        DAD      B          ;POINT TO CORRECT MODE BY
1311 C5        PUSH    B          ;SAVE COUNT OF DRIVES
1312 41        Mov     B,C        B = DRIVE #
1313 4E        Mov     C,M
1314 C'DF807    CALL    XETMOD      ;SET MODE
1317 C1        POP     B
1318 0D        DCR     C          ;END OF LIST YET ??
1319 F20A13    jP      MODESET     ;SET MODE FOR ALL DRIVES
131C CDD007    CALL    SDCONF      ;SET DISK CONFIGURATION
131F 018000    LXI     B,80H
1322 CD5502    CALL    SETDMA      ;SET DMA ADDRESS
1325 E5        push    h
        if     mpm20

```

```

1326 2A130B      lhld    sysdat
1329 2E07        mvi     1,7
132B 7E          mov     a,m
                else
                lxi     h,INTERUPT
                mov     a,h
                endif

132C E1          POP     h
132D ED47        DB      0EDH,047H      ---- FAKE STAI INSTRUCTION
132F 3E60        MVI     A,60H          SET VECTOR FOR CTC
1331 D330        OUT     30H            CTC CHANNEL 0
1333 3EA7        MVI     A,0A7H        RESET / LOAD TIME CONST
1335 D333        OUT     33H            CHANNEL 3
1337 3EFA        MVI     A,250         TIME CONSTANT
1339 D333        OUT     033H
                IF      HARDSK
133B AF          XRA      A              ;ZERO ACCUMULATOR
133C 32EFOA      STA     HSTACT         ;SET HOST BUFFER INACTIVE
133F 32F10A      STA     UNACNT        ;SET UNALLOCATED COUNT TO Z

1342 219B0E      LXI     H,HSTBUF-1    ;SETUP WRITE CONTROL BYTE F
1345 360D        MVI     M,00DH
                ENDIF
1347 C9          RET

                STMVTR:
1348 D309        OUT     MEMPORT
134A 3EC3        MVI     A,0C3H          SET VECTORS FOR BDOS
134C 320000      STA     0              JMP INSTRUCTION
134F 2A5E13      LHLD    SVDJT
1352 220100      SHLD    1
1355 2A6013      LHLD    SVDBPA
1358 77          MOV     M,A
1359 23          INX     H
135A 73          MOV     M,E
135B 23          INX     H
135C 72          MOV     M,D
135D C9          RET
135E            SVDJT:  DS      2          SAVED DIRECT JUMP TABLE ADDRESS
1360            SVDBPA: DS      2          SAVED BREAK POINT ADDRESS
                if      mpm20
1362 = xiosend    equ     $
139D = fdbuf      equ     (dirbuf-base)+256
139D      org fdbuf+((xiosend-base)/fdbuf)*((xiosend-base)-fd
139D 00          db      0
                endif

139E            END

```

070F	ADDERRORS	0DC6	ADRINTHD	12CE	AFTERBANKS	03DA	ALLOC		
081E	ALV0	085E	ALV1		089E	ALV2	08DE	ALV3	
091E	ALV4	095E	ALV5		099E	ALV6	09DE	ALV7	
0A1E	ALV8	0A5E	ALV9	0A9E	ALVA	0AC2	ALVB,		
0708	BADIO	0D03	BANKNO	12BF	BANKSETUP		0000	BASE	
081E	BEGDAT	4000	BLKSIZ	067B	CHECKIT	06AE	CHECKSTAT		
0220	CHKHRD	06BC	CHKS0	06C7	CHKS1	06CD	CHKS2		
06E6	CHKS3	06F7	CHKS4	03A1	CHKUNA		0B2C	CLIST	
0B3B	CLIST1		0AFA	CMD	0B1F	CNSTAT	0D9D	CNTX	
0B15		COLDSTART		0B04	COMMONBASE		0B84	CONIN	0B8F
	CONOUT								
0B79		CONST	0080	CPMSPT	083E	CSV0	087E	CSV1	
08BE		CSV2	08FE	CSV3	095E	CSV4	099E	CSV5	
09DE		CSV6	0A1E	CSV7	0A5E	CSV8	0A9E	CSV9	
0AC2		CSVA	0AE6	CSVB	0D04	CURMEM	001C	DATA0	
002C		DATA1	002E	DATA2	002A	DATA3	00B4	DBLKAD	
052E		DBLLOW	0530	DBLSAVE	0519	DBLUPDATE	0782	DEL1	
0784		DEL2	0782	DELAY	0037	DENSITYMAS	0CD2	DEVOK	
0CD5	DEVTBL		129D	DIRBUF	00AC	DISKNO	FFFF	DMA	
00AF	DMAADR		00BE	DMALEN	00BA	DMAS1	00C6	DMAS2F	
00C0	DMAS2H		00CA	DMAS3	00CE	DMAS3F	00BC	DMASA	
0DD4		DPB0	0DE3	DPB1	0DF2	DPB2	0E01	DPB3	
0E10		DPB4	0E1F	DPB5	0E2E	DPB6	00D1	DPBASE	
00D1		DPE0	00E1	DPE1	00F1	DPE2	0101	DPE3	
0111		DPE4	0121	DPE5	0131	DPE6	0141	DPE7	
0151		DPE8	0161	DPE9	0171	DPEA	0181	DPEB	
00B2		DPEPTR	0790	DSCNO	0552	DSKSEL	008E	DSPTCH	
0246		DTBLT	0AF6	ERFLAG	0D10	EXITREGION	0000	FALSE	
139D		FDBUF	0B50	FDINTH	041C	FILLHST	0763	FINTFIX	
0085		FLAGST	0084	FLAGWT	0B47	FLOPPYINT	0687	FLOPPYIO	
065C		FLOPPYSEEK		0661	FPS1	131D	FPYBUF	0006	
	FPYFLAG								
0DD2		FPYTCNT	0DD0	FPYTIME	073A	FPYWAIT	0670	FSECSET	
06AA		FWT1	0B5E	HARDINT	FFFF	HARDSK	0005	HDFLAG	
0B67		HDINTH	0B71	HDSTFLG	00AE	HEADNO	031A	HOME	
0337		HOME1	0340	HOME1A	0357	HOME2	0343	HOMEHARD	
02F9		HOMEIT	0321	HOMESOFT	0B03	HOMETOGGLE	048F	HRW0	
0499		HRW1	04C4	HRW2	04CD	HRW3	04D9	HRW4	
04E1		HRW5	04E6	HRW6	0504	HRW7	0AEF	HSTACT	
0008	HSTBLK		0E9C	HSTBUF	0AEA	HSTDSK	0AED	HSTSEC	
0400		HSTSIZ	0010	HSTSPT	0AEB	HSTTRK	0AF0	HSTWRT	
00B8		HTK1	00B9	HTK2	07F8	INITEND	0D1A	INTJHND	
0D67		INTDONE	005E	INTERUPT	0721	INTFIX	0D7F	INTINIT	
0D9E		INTSTK	0051	LAST	062D	LDH1	1700	LDRBIOSBAS	
0CA9		LIST	0CB8	LISTI	061E	LOADHEAD	001E	LPTPRTO	
0028		LPTPRT1	001F	LPTSTSO	0029	LPTSTS1	0DC6	LSTINTSTK	
0AFB	MASK	0439	MATCH		0D17	MAXCONSOLE	000C	MAXDSK	
0000		MDISK	0009	MEMPORT	0002	MEMSK	00BB	MISCPARAMS	
0088		MODE	130A	MODESET	0191	MODLO	019D	MODL1	
01A9		MODL2	00B6	MPARMS	FFFF	MPM20	0E7E	MVDFB	
0E99		MVDFX	0E6B	MVDTB	0AE6	NEWDSK	0AEE	NEWHST	
0AE9		NEWSEC	0AE7	NEWTRK	0461	NEWTRKCMP	0004	NMBCNS	
0009		NMBDEV	075F	NOFPYRST	0415	NOMATCH	03D4	NOOVF	
0D67		NOTLSEC	0D2F	NOTICKN	0B1C	NULLINT	00A0	PCNT	
0B0D		PDISP	0005	PLCIO	0006	PLCI1	0007	PLC12	
0008		PLC13	0001	PLC00	0002	PLC01	0003	PLC02	
0004		PLC03	0000	PLLPT	05D4	PNTFN	05D1	PNTH2	

05B3		POINT	OBAD	POLCIO	OBEC	POLCI1	OC2B	POLCI2
OC6A		POLCI3	OBDD	POLC00	OC1C	POLC01	OC5B	POLC02
OC9A		POLC03	0083	POLL	OCCB	POLLDEVICE	OCBC	POLLPT
ODCF		PREEMP	OB01	PRETRIES	1307	PRTOK	0BB8	PT0IN
OBC5		PT0IN1	OB0A	PT0OUT	0BD9	PT0OUT1	OBAD	PT0ST
0BF7		PTLIN	OC04	PTLIN1	OC09	PT10UT	OC18	PT10UT1
OBEC	PT1ST	OC36 MIN	OC43	PT21N1		OC48PT20UT		
OC57	PT20UT1	OC2B	PT2ST	OC75	PT31N	OC82	PT31N1	
OC87	PT30UT	OC96	PT30UT1	OC6A	PT3ST	OB9A	PTBLJMP	
028B	READ	036B	READHARD	047F	READHST	0AF8	READOP	
05E4	READSOFT	0308	REALDISK	FFFF	RELOC	0643	REMOVED	
02EE	RETMOD	0AF7	RSFLAG	0B1A	RTNEMPTY	0E60	RWEND	
0E3D	RWMOVE	03E2	RWOPER	0AFD	SAVE1	07D0	SDCONF	
07DE	SDDBL	07F1	SDL1	07F3	SDOK	0007	SECMSK	
0003	SECSHF	00B1	SECTNO	05D6	SECTRAN	007C	SELO	
0203	SELDSK	0242	SELERR	059B	SELHARD	0CE9	SELMEMORY	
022F	SELSDP	0556	SELDOFT	0278	SETDEN	0255	SETDMA	
0211	SETDSK	0544	SETDVD	0540	SETH14	0532	SETHED	
02A1	SETMOD	0273	SETSEC	02C7	SETSEL	026D	SETTRK	
0547	SHD1	05A1	SLH1	055D	SLS1	056B	SLS2	
0575	SLS3	0584	SLS4	0595	SLSERR	02E3	SMERR	
0601	SRW1	0615	SRW2	0D05	STARTCLOCK		0AFC	STATUS
1348	STMVTR	0D0B	STOPCLOCK		05DB	STRN1	05E2	STRN2
001D	STSO	002D	STS1	002F	STS2	002B	STS3	
1360	SVDBPA	0DC8	SVDHL	135E	SVDJT	0DCC	SVDRET	
0DCA	SVDSP	0B0A	SWTSYS	0B07	SWTUSER	0B13	SYSDAT	
129D	SYSTEMINIT	0D1A	T20MS	0BA2	TBLJMP	0094	TCNT	
12AF	TESTBANKSE	0DCE	TICKN	0D22	TIMERINT	0D9C	TOGCNT	
HAD	TRAKNO	0B02	TRETRIES	0070	TRK0064A	TRKTST		
FFFF	TRUE	0AF1	UNACNT	0AF2	UNADSK	0AF5	UNASEC	
0AF3	UNATRK	0717	WAIT0	0B15	WARMSTART		0003	WBOTE
0000	WRALL	0001	WRDIR	0296	WRITE	037E	WRITEHARD	
046D	WRITEHST	05F2	WRITESOFT		0AF9	WRTYPE	0002	WRUAL
0B10	XDOS	07F8	XETMOD	0801	XETSEL	1362	XIOSEND	
0IB5	XLT0	01CF	XLT1	01CF	XLT2	0000	XLT3	
1.1000	XLT4	0000	XLT5	0000	XLT6			