A P P E N D I X A

Keyboard Matrix

Keyboard Matrix

	X0	X1	X2	X3	X4	X5	X6	X7
Y9	CAP LOCK							
Y8	SHIFT		SPACE BAR	Z	A	Q	2	F1
Y7	CTRL	= –	¦ \	X	S	W	3	F2
Y6	ⓐ	P	CURSOR UP	C	D	E	4	F3
Y5	1			V	F	R	5	F4
Y4	ESC			B	G	T	6	F5
Y3	TAB	* :	↵	N	H	Y	7	F6
Y2	CURSOR DOWN	· ~ ∧	{ [	M	J	U	8	F7
Y1	CURSOR RIGHT	+ ;	}]	< ,	K	I	9	F8
Y0	CURSOR LEFT	⇦	? /	> .	L	O	0	DEL

A P P E N D I X B

Bondwell 2 BIOS Listing

```
;*****************************************************
;CP/M version 2.2 GRAPHIC BIOS RELEASE 1.0 for
;project BH-026 Handbook
;edited by: LAM CHI KWAN          MARCH, 1985
;*****************************************************
;
CCP     EQU     BIOS-1600H      ;base of ccp
BDOS    EQU     CCP+806H        ;base of bdos
CDISK   EQU     0004H           ;current disk no. 0=a,.
IOBYTE  EQU     0003H           ;intel i/o byte
RESET   EQU     0000H           ;system reset address
;-----------------------------------------------------
;
PPICREG EQU     03H             ;82c55 control reg.
PPIA    EQU     00H             ;82c55 port a
PPIB    EQU     01H             ;82c55 port b
PPIC    EQU     02H             ;82c55 port c
;
CNTCREG EQU     13H             ;82c53 control reg.
CNT0    EQU     10H             ;82c53 counter reg. 0
CNT1    EQU     11H             ;82c53 counter reg. 1
CNT2    EQU     12H             ;82c53 counter reg. 2
;
UTDATA  EQU     40H             ;82c51 data reg.
UTCREG  EQU     41H             ;82c51 control reg.
UTSREG  EQU     41H             ;82c51 status reg.
;
PRINTER EQU     50H             ;printer output port
;
FDCCREG EQU     60H             ;FDC command reg.
FDCSREG EQU     60H             ;FDC status reg.
TR      EQU     61H             ;FDC track reg.
SR      EQU     62H             ;FDC sector reg.
DR      EQU     63H             ;FDC data reg.
;
;-------memory bank---------------------------
;
RAM1    EQU     0       ;ram bank 1
VRAM    EQU     1       ;vram bank
RAM2    EQU     2       ;ram bank 2
RAM3    EQU     3       ;ram bank 3
RAM4    EQU     4       ;ram bank 4
RAM5    EQU     5       ;ram bank 5
RAM6    EQU     6       ;ram bank 6
ROM     EQU     7       ;rom bank
BKMASK  EQU     0F8H    ;bank mask
BKIVMS  EQU     07H     ;bank inverse mask
;
;-------80 coloum x 25 line lcd addr.---------
;
CROM    EQU     800H    ;character rom
LCDIST  EQU     21H     ;lcd instruction reg.
LCDDAT  EQU     20H     ;lcd data reg.
CSRADL  EQU     07H     ;cursor low addr.
CSRADH  EQU     08H     ;cursor high addr.
```

```
VSTADL  EQU     05H     ;vram start addr. low
VSTADH  EQU     06H     ;vram start addr.high
CSRON   EQU     7DH     ;LCD MODE WORD CSR. ON
CSROFF  EQU     4DH     ;LCD MODE WORD CSR. OFF
;
;-------control characters--------------------
;
CR      EQU     0DH     ;carriage return
LF      EQU     0AH     ;line feed
BS      EQU     08H     ;back space
CURRHT  EQU     0CH     ;cursor right
CURUP   EQU     0BH     ;cursor up
CURHOM  EQU     1EH     ;home cursor
BKSPAC  EQU     08H     ;back space
ENDGRF  EQU.    00H     ;end graphic mode
ESC     EQU     1BH     ;escape
SPACE   EQU     20H     ;space
;
;-------esc characters-----------------------
CLEARS  EQU     '*'     ;clear screen & home
CLINE   EQU     'T'     ;clear to end of line
CLEND   EQU     'Y'     ;clear to end of screen
DLINE   EQU     'R'     ;delete line
ILINE   EQU     'E'     ;insert line
STRGRF  EQU     'G'     ;start graphic
INV     EQU     'I'     ;start inverse char.
ENINV   EQU     'N'     ;end inverse char.
;
;-----------------------------------------------
;cp/m to host disk constants
;
BLKSIZ  EQU     2048            ;cp/m block size
HSTSIZ  EQU     256             ;hst disk sec size
HSTSPT  EQU     18              ;phy. secs/trk
HSTBLK  EQU     HSTSIZ/128      ;hst buff/cp/m sec
CPMSPT  EQU     HSTBLK*HSTSPT   ;cp/m secs/trk
SECMSK  EQU     HSTBLK-1        ;sec mask
SECSHF  EQU     1       ;sec shift log2(hstblk)
;-----------------------------------------------
MTRON   EQU     90H     ;set drive motor on
MTROFF  EQU     19      ;oneshot 5s then motor off
MFDBK   EQU     5       ;motor feedback pc5
DRIVE0  EQU     10H     ;drive select 0 pa4
DRIVE1  EQU     20H     ;drive select 1 pa5
DISDRV  EQU     0CFH    ;disable both drive
SEEK    EQU     1CH     ;seek command with verify
RDSEC   EQU     88H     ;rd 1 sector command ,sso=0
WRSEC   EQU     0A8H    ;wr 1 sector command ,sso=0
RECAL   EQU     0CH     ;move rd/wr head to trk 0
RETRY   EQU     10      ;10 retry
NMI     EQU     0066H   ;nonmaskable addr.
ENBNMI  EQU     3       ;enable nmi, pc3
;
;-----------------------------------------------
;bdos constants on entry to write
```

```
;
WRALL   EQU     0       ;write to allocate
WRDIR   EQU     1       ;write to directory
WRUAL   EQU     2       ;write to unallocated
;
;---------------------------------------------
;
PBUSY   EQU     4       ;printer busy flag pc4
PSTB    EQU     7       ;strobe printer data out
;
;---------------------------------------------
;
RPKEY1  EQU     80H     ;repeat key time 1
RPKEY2  EQU     20H     ;repeat key time 2
MASKPA  EQU     11110000B   ;mask ppia
;---------------------------------------------
NSECTS  EQU     1600H/256       ;wboot sec count
;---------------------------------------------
;jump vector for individual subroutines
;
BIOS:
        JP      CBOOT           ;cold start
WBOOTE: JP      WBOOT           ;warm start
        JP      CONST           ;console status
        JP      CONIN           ;console char. in
        JP      CONOUT          ;console char. out
        JP      LIST            ;list char. out
        JP      PUNCH           ;punch char. out
        JP      READER          ;reader char. out
        JP      HOME            ;move head to home
        JP      SELDSK          ;select disk
        JP      SETTRK          ;set track number
        JP      SETSEC          ;set sector number
        JP      SETDMA          ;set dma address
        JP      READ            ;read disk
        JP      WRITE           ;write disk
        JP      LISTST          ;return list status
        JP      SECTRAN         ;sector translate
SWBANK: JP      SWBANK          ;switch bank ----DELETED---
;---------------------------------------------
FUNTBL:
F1:     DB      'DIR',20H,0,0,0,0,0,0,0,0,0,0,0,0
F2:     DB      'ERA',20H,0,0,0,0,0,0,0,0,0,0,0,0
F3:     DB      'TYPE',20H,0,0,0,0,0,0,0,0,0,0,0
F4:     DB      'REN',20H,0,0,0,0,0,0,0,0,0,0,0,0
F5:     DB      'SAVE',20H,0,0,0,0,0,0,0,0,0,0,0
F6:     DB      'STAT',20H,0,0,0,0,0,0,0,0,0,0,0
F7:     DB      'A:',0,0,0,0,0,0,0,0,0,0,0,0,0,0
F8:     DB      'B:',0,0,0,0,0,0,0,0,0,0,0,0,0,0
F9:     DB      'PIP',20H,0,0,0,0,0,0,0,0,0,0,0,0
F10:    DB      'ED',20H,0,0,0,0,0,0,0,0,0,0,0,0,0
F11:    DB      'ASM',20H,0,0,0,0,0,0,0,0,0,0,0,0
F12:    DB      'LOAD',20H,0,0,0,0,0,0,0,0,0,0,0
F13:    DB      'DDT',20H,0,0,0,0,0,0,0,0,0,0,0,0
F14:    DB      'DUMP',20H,0,0,0,0,0,0,0,0,0,0,0
```

```
F15:    DB      'SUBMIT',20H,0,0,0,0,0,0,0,0,0
F16:    DB      'SETUP',0,0,0,0,0,0,0,0,0,0,0
;
BAUDOD: DB      00000101B       ;default 300 baud
UARTOD: DB      01111010B       ;usart 1 stopbit
                ;DATA MASK      ;even parity,7bit/char.
                                ;16x
BITMSK: DB      01111111B       ;7 bit /char.
;-----------------------------------------------------
;fixed data tables (IBM compatible 3.5" disk)
;disk parameter header
DPBASE:
DPB0:   DW      0000H   ;no sec. translate
        DW      0000H   ;bdos use location
        DW      0000H   ;bdos use location
        DW      0000H   ;bdos use location
        DW      DIRBF   ;addr. of 128 byte dir buf
        DW      DPBLK   ;addr. of disk para block
        DW      CHK00   ;change disk chack addr.
        DW      ALL00   ;allocation vector
;
DPB1:   DW      0000H   ;no sec. translate
        DW      0000H   ;bdos use location
        DW      0000H   ;bdos use location
        DW      0000H   ;bdos use location
        DW      DIRBF   ;addr. of 128 byte dir buf
        DW      DPBLK   ;addr. of disk para block
        DW      CHK01   ;change disk chack addr.
        DW      ALL01   ;allocation vector
;
;disk parameter block, common to all disks
;
DPBLK:  DW      36              ;log. sec. per track
        DB      4               ;block shift (2k block)
        DB      15              ;block mask
        DB      1               ;extent mask
        DW      174             ;total block-1(less sys trk)
        DW      127             ;directory max
        DB      0C0H            ;alloc 0
        DB      0               ;alloc 1
        DW      32              ;check size
        DW      2               ;track offset
;
;-----------------------------------------------------------------
;
CBOOT:  LD      A,0FH   ;enable rom
        OUT     (PPIC),A
        JP      RESET           ;jump to reset address
CBOOTE: JP      BOOT
;
;
;-------------------------------------------------
;
WBOOT:  DI
        LD      SP,80H
```

```
        LD      A,(HSTWRT)
        OR      A
        CALL    NZ,WRHST
        XOR     A                       ;load ccp, bdos
        LD      (HSTDSK),A
        LD      (HSTTRK),A
        LD      (HSTSEC),A
        LD      (DR0TRK),A              ;home drv a
        LD      A,MTRON
        OUT     (CNTCREG),A
        IN      A,(PPIA)
        OR      10H
        OUT     (PPIA),A
        CALL    HOLD
        LD      A,0CH                   ;recal command
        OUT     (FDCCREG),A
        CALL    HOLD
;$$$$$$$$$$$$$$$$$$$$$$$
        JP      GOCPM                   ;FOR DEBUG ONLY
;$$$$$$$$$$$$$$$$$$$$$$$
;
NEXT:   LD      HL,CCP-128
        LD      (DMAADR),HL
        LD      B,NSECTS
LOAD:   PUSH    BC
        CALL    RDHST                   ;rd to hstbuf
        LD      A,(ERFLAG)
        OR      A
        JR      Z,NOLOER
        JR      WBOOT                   ;disk load err
NOLOER: LD      DE,(DMAADR)
        LD      HL,HSTBUF
        LD      BC,256
        LDIR
        LD      (DMAADR),DE
        LD      A,(HSTSEC)
        INC     A
        LD      (HSTSEC),A
        CP      18
        JR      NZ,NEXTS
        XOR     A
        LD      (HSTSEC),A
        INC     A
        LD      (HSTTRK),A
NEXTS:  POP     BC
        DJNZ    LOAD

        CALL    RDHST                   ;READ THE LAST RECORD
        LD      HL,HSTBUF
        LD      DE,(DMAADR)
        LD      BC,128
        LDIR
;
GOCPM:  LD      A,0C3H                  ;jmp instruction
        LD      (0000H),A               ;for jmp to wboot
```

```
        LD      HL,WBOOTE       ;wboot entry point
        LD      (0001H),HL
        LD      (0005H),A       ;for jmp to bdos
        LD      HL,BDOS         ;bdos entry point
        LD      (0006H),HL
        LD      BC,0080H        ;default dma address is 80h
        CALL    SETDMA
        LD      SP,0080H        ;boot rom cannot do this
        XOR     A
        LD      (HSTACT),A      ;host buffer inactive
        LD      (UNACNT),A      ;clear unalloc count
;----------DELETED IN GRAPHIC BIOS ----------
;       LD      HL,AUTO         ;set autorun
;       LD      DE,CCP+7
;       LD      BC,9
;       LDIR
;----------DELETED IN GRAPHIC BIOS ----------
        LD      A,(DRV1FL)
        OR      A
        JR      Z,AACTIV
        LD      A,(CDISK)       ;get current disk number
        CP      02
        JR      C,VDRIVE
AACTIV: XOR     A
VDRIVE: LD      C,A             ;send to the ccp
        XOR     A
        LD      (DRV1FL),A      ;recal drv b
;---------- CHANGED IN GRAPHIC BIOS ----------
        LD      HL,(CCPETY)
        LD      DE,CCP+3
        LD      (CCPETY),DE
        JP      (HL)
;-----------------------------------------------
;console status, return 0ffh if
;character ready, 00h if not
;
CONST:  LD      HL,KEYBUF       ;save old buffs.
        LD      DE,KEYBUF2
        LD      BC,4
        LDIR
;
        CALL    SCAN
        LD      A,(KEYBUF)
        CP      -1
        JR      NZ,KEYAVL
        LD      A,(KEYCNT)
        OR      A
        JR      NZ,KEYAVL
                                ;switch back
;       LD      HL,KEYBUF2      ;ctrs.
        LD      DE,KEYBUF
        LD      BC,4
        LDIR
;
        LD      A,(KEYBUF)
```

```
        CP      -1
        JR      NZ,KEYAVL
        LD      A,(KEYCNT)
        OR      A
        RET     Z
KEYAVL: LD      A,-1
        RET
;
;-----------------------------------------------
;console input. return keycode in a
;
CONIN:  LD      A,(KEYBUF)
        CP      -1
        JR      Z,NXTCHK
        LD      HL,KEYBUF
        LD      (HL),-1
        RET
NXTCHK: LD      A,(KEYCNT)
        OR      A
        JR      Z,RDKBD
        DEC     A
        LD      (KEYCNT),A
        LD      HL,(KEYPTR)
        LD      B,(HL)
        INC     HL
        LD      (KEYPTR),HL
        LD      A,(HL)
        OR      A
        JR      NZ,RETKEY
        LD      (KEYCNT),A
RETKEY: LD      A,B
        RET
RDKBD:  LD      HL,00E0H        ;delay 3msec
        CALL    HOLDE
        CALL    SCAN
        JR      CONIN
;
;-----------------------------------------------
;scan keyboard and return keycode in acc,
;keybuf, keyptr, keycnt
;
SCAN:   LD      HL,NEWTBL+1     ;read 9 rows
        LD      C,8
        IN      A,(PPIA)
        AND     MASKPA
        LD      B,A
        LD      E,-1
SCANKB: LD      A,B
        OR      C
        OUT     (PPIA),A
        IN      A,(PPIB)
        LD      (HL),A
        AND     E
        LD      E,A
        INC     HL
```

```
        DEC     C
        JP      P,SCANKB
        INC     E
        JP      Z,VALID
;
        XOR     A
        LD      (CNTRL),A
        LD      (SHIFT),A
        LD      (CAPS),A
;
        LD      A,B             ;caps?
        OR      9
        OUT     (PPIA),A
        IN      A,(PPIB)
        RRA
        JR      C,CKCNTL
        LD      (CAPS),A
CKCNTL: LD      A,B
        OR      7               ;check cntrl
        OUT     (PPIA),A
        IN      A,(PPIB)
        RRA
        JR      C,CKSHFT
        LD      (CNTRL),A
CKSHFT: LD      A,B
        OR      8               ;check shift
        OUT     (PPIA),A
        IN      A,(PPIB)
        RRA
        JR      C,CHKEND
        LD      (SHIFT),A
        LD      A,(CNTRL)
        OR      A
        JP      NZ,INVALI
;
CHKEND: LD      HL,NEWTBL
        LD      (HL),-1
        INC     HL
        SET     0,(HL)
        INC     HL
        SET     0,(HL)
;
        LD      B,10            ;compare 10 rows
        LD      DE,NEWTBL
        LD      HL,OLDTBL
CMPRE:  LD      A,(DE)
        LD      C,(HL)
        XOR     C
        AND     C
        JR      NZ,DEBOUN
        INC     DE
        INC     HL
        DJNZ    CMPRE
;
        LD      (FLAG1),A       ;process old key
```

```
        LD      (FLAG2),A
        LD      B,10            ;compare 10 rows
        LD      DE,NEWTBL
        LD      HL,OLDTBL
COMPRE: LD      A,(DE)
        LD      C,(HL)
        OR      C
        CPL
        OR      A
        JR      NZ,MEMKEY
CONTIN: INC     DE
        INC     HL
        DJNZ    COMPRE          ;process old key
        LD      A,(FLAG2)
        OR      A
        JR      Z,VALID
        LD      A,(TEMP)
        LD      C,A
        LD      A,(NPRESS)
        CP      C
        JR      NZ,VALID
        LD      HL,TIMER
        INC     (HL)
        LD      A,(FLAG3)
        OR      A
        JR      NZ,TIME2
        LD      A,(TIMER)
        CP      RPKEY1
        JR      C,VALID
        LD      (FLAG3),A
        JR      RESTMR
TIME2:  LD      A,(TIMER)
        CP      RPKEY2
        JR      C,VALID
        JR      RESTMR
;
MEMKEY: LD      C,A
        LD      A,(FLAG2)
        OR      A
        JR      NZ,VALID
        INC     A
        LD      (FLAG2),A
        PUSH    BC
        PUSH    DE
        PUSH    HL
        LD      A,C
        CALL    FNDKEY
        LD      (TEMP),A
        POP     HL
        POP     DE
        POP     BC
;&&&&&& release 1.1 &&&&&&&&
        LD      A,(FLAG4)       ;see if bad key
        OR      A
        JR      NZ,INVALI
```

```
;&&&&&& release 1.1 &&&&&&&&
        JR      CONTIN
VALID:  CALL    UPDATE
INVALI: XOR     A
        LD      (KEYCNT),A
        DEC     A
        LD      (KEYBUF),A
        RET
;
DEBOUN: CALL    FNDKEY          ;find key-code
        LD      C,A
;&&&&&&& release 1.1 &&&&&&&
        LD      A,(FLAG4)       ;see if bad key
        OR      A
        JR      NZ,INVALI
;&&&&&& release 1.1 &&&&&&&&
        LD      A,(FLAG1)
        OR      A
        JR      NZ,SCNDRD
NSCN:   LD      A,-1
        LD      (FLAG1),A
        LD      A,C
        LD      (NPRESS),A
        LD      HL,180H         ;delay 5msec
        CALL    HOLDE
        JP      SCAN
;
SCNDRD: LD      A,(NPRESS)      ;npress=newkey?
        CP      C
        JR      NZ,NSCN
        XOR     A
        LD      (FLAG1),A
        LD      (FLAG3),A
RESTMR: XOR     A
        LD      (TIMER),A
;
        LD      A,(NPRESS)
        CP      128
        JR      NC,FUNKEY
        XOR     A
        LD      (KEYCNT),A
        LD      A,(NPRESS)
        LD      (KEYBUF),A
        LD      HL,-1
        LD      (KEYPTR),HL
EXCHG:  CALL    UPDATE
FINISH: LD      A,(KEYBUF)
        RET
FUNKEY: SUB     128
        ADD     A,A
        ADD     A,A
        ADD     A,A
        ADD     A,A
        LD      E,A
        LD      D,0
```

```
        LD      HL,FUNTBL
        ADD     HL,DE
        LD      (KEYPTR),HL
        LD      A,16
        LD      (KEYCNT),A
        LD      A,-1
        LD      (KEYBUF),A
        JR      EXCHG
;
;-----------------------------------------------
;find keycode subroutine. pass a, b
;output--keycode in a
;
FNDKEY: LD      C,A
;
;&&&&&& release 1.1 &&&&&&&&&
        XOR     A
        LD      (FLAG4),A       ;no bad key yet
;&&&&&& release 1.1 &&&&&&&&&
;
        DEC     B
        LD      A,B
        ADD     A,A
        ADD     A,A
        ADD     A,A
        LD      E,A
        LD      D,0
        LD      A,(SHIFT)       ;true shift?
        OR      A
        JR      NZ,ASCII2
        LD      HL,ASCII-1
        ADD     HL,DE
FNDPOS: RLC     C
        INC     HL
        JR      NC,FNDPOS
        LD      A,(CNTRL)
        OR      A
        JR      Z,NOCNTL
        LD      A,(HL)
        CP      60H
        JR      C,BADKEY
        CP      128
        JR      NC,BADKEY
        AND     1FH
        RET
NOCNTL: LD      A,(CAPS)
        OR      A
        JR      Z,GOOUT
        LD      A,(HL)
        CP      61H
        RET     C
        CP      7BH
        RET     NC
        AND     5FH
        RET
```

```
;
ASCII2: LD      HL,SHFTBL-1
        ADD     HL,DE
RDPOS:  RLC     C
        INC     HL
        JR      NC,RDPOS
GOOUT:  LD      A,(HL)
        RET
;
;&&&&&&& release 1.1 &&&&&&&&
BADKEY: LD      A,-1
        LD      (FLAG4),A
        RET
;&&&&&&& release 1.1 &&&&&&&&
;
;-----------------------------------------------
;
ASCII:  DB      7FH,30H,6FH,6CH,2EH,2FH,08H,13H
        DB      135,39H,69H,6BH,2CH,5DH,3BH,04H
        DB      134,38H,75H,6AH,6DH,5BH,5EH,18H
        DB      133,37H,79H,68H,6EH,0DH,3AH,09H
        DB      132,36H,74H,67H,62H,-1,-1,1BH
        DB      131,35H,72H,66H,76H,-1,-1,31H
        DB      130,34H,65H,64H,63H,05H,70H,40H
        DB      129,33H,77H,73H,78H,5CH,2DH,-1
        DB      128,32H,71H,61H,7AH,20H,-1,-1
        DB      -1,-1,-1,-1,-1,-1,-1,-1
;
SHFTBL: DB      7FH,5FH,4FH,4CH,3EH,3FH,08H,13H
        DB      143,29H,49H,4BH,3CH,7DH,2BH,04H
        DB      142,28H,55H,4AH,4DH,7BH,7EH,18H
        DB      141,27H,59H,48H,4EH,0DH,2AH,09H
        DB      140,26H,54H,47H,42H,-1,-1,1BH
        DB      139,25H,52H,46H,56H,-1,-1,21H
        DB      138,24H,45H,44H,43H,05H,50H,60H
        DB      137,23H,57H,53H,58H,7CH,3DH,-1
        DB      136,22H,51H,41H,5AH,20H,-1,-1
        DB      -1,-1,-1,-1,-1,-1,-1,-1
;
;-----------------------------------------------
;subroutine to copy newtbl to oldtbl
;
UPDATE: LD      HL,NEWTBL
        LD      DE,OLDTBL
        LD      BC,10
        LDIR
        RET
;-----------------------------------------------
;GRAPHIC CONSOLE OUT FOR BH-026 HANDBOOK
;CREATED BY C.K.LAM        DATE: FEB 28,1985
;-----------------------------------------------
;FEATURES ADDED:---
;
;ESC 0  GRAPHIC MODE, CLEAR SCREEN (16K VRAM), VSTART=0, HOME TEXT CU
```

```
;       HOME GRAPHIC CURSOR.  (BOTH CURSOR DEFAULT ON)
;ESC 1  ON TEXT CURSOR
;ESC 2  OFF TEXT CURSOR
;ESC 3  ON GRAPHIC CURSOR
;ESC 4  OFF GRAPHIC CURSOR
;ESC 5 Y X  GRAPHIC CURSOR POSITION
;ESC 6  DOT ON, UPDATE GRAPHIC CURSOR
;ESC 7  DOT OFF, UPDATE GRAPHIC CURSOR
;
;-----------------------------------------------
;CONSOLE OUT, PASS (C)
;
CONOUT: LD      IY,0
        ADD     IY,SP
        LD      SP,STACK2
        IN      A,(PPIC)
        LD      (BANKBF),A
        LD      A,(ESC5)
        OR      A
        JP      NZ,G05A
        LD      A,(ESCFLG)      ;esc active?
        OR      A
        JR      NZ,ESCACT
        LD      A,(GRFFLG)      ;graphic mode?
        OR      A
        JR      NZ,GRAPH
        LD      A,C
        LD      HL,CTLTBL
COMPAR: LD      B,(HL)
CNEXT:  INC     HL
        CP      (HL)
        INC     HL
        LD      E,(HL)
        INC     HL
        LD      D,(HL)
        JR      Z,LDJMP
        DJNZ    CNEXT
LDJMP:  EX      DE,HL
        JP      (HL)            ;goto diff. routine
;
ESCACT: LD      A,(ESCFLG)
        DEC     A
        LD      (ESCFLG),A
        CP      2
        JR      NZ,NOT2
        LD      A,C
        CP      '='
        JP      Z,RTPATH
        XOR     A
        LD      (ESCFLG),A
        LD      A,C
        LD      HL,ESCTBL
        JR      COMPAR
;
NOT2:   OR      A
```

```
        JR      Z,CHXY
        LD      A,C
        LD      (CSNY),A
        JP      GOPATH
CHXY:   LD      A,C
        SUB     32
        CP      80
        JR      NC,CRYRTY
        LD      (CUSRX),A
        LD      A,(CSNY)
        SUB     32
        CP      25
CRYRTY: JP      NC,RTPATH
        LD      (CUSRY),A
        CALL    INCSR
        JP      UDCUSR
;
GRAPH:  LD      A,C
        CP      ENDGRF
        JP      NZ,K08          ;display char.
        XOR     A
        LD      (GRFFLG),A
GOPATH: JP      RTPATH
;
CTLTBL: DB      12              ;count
        DB      CR              ;carriage ret.
        DW      K01
        DB      LF              ;line feed
        DW      K02
        DB      CURRHT          ;cursor right
        DW      K03
        DB      CURUP           ;cursor up
        DW      K04
        DB      CURHOM          ;home cursor
        DW      K05
        DB      ESC             ;escape
        DW      K06
        DB      BKSPAC          ;back space
        DW      K07
        DB      1AH             ;form feed for kaypro
        DW      E01
        DB      18H             ;clear to eol for kayp.
        DW      E02
        DB      17H             ;clear to eos for kayp.
        DW      E03
        DB      07H             ;bell
        DW      RTPATH
        DB      SPACE           ;space or char.
        DW      K08
;
ESCTBL: DB      17              ;count
        DB      CLEARS          ;clear screen & home
        DW      E01
        DB      CLINE           ;clear to end of line
        DW      E02
```

```
        DB      CLEND           ;clear to end of screen
        DW      E03
        DB      DLINE           ;delete line
        DW      E04
        DB      ILINE           ;insert line
        DW      E05
        DB      STRGRF          ;start graphic
        DW      E06
        DB      INV             ;start inverse char.
        DW      E07
        DB      ENINV           ;end inverse char.
        DW      E08
        DB      0CH             ;formfeed
        DW      E01
        DB      '1'             ;TEXT CURSOR ON
        DW      G01
        DB      '2'             ;OFF TEXT CURSOR
        DW      G02
        DB      '3'             ;GRAPHIC CURSOR ON
        DW      G03
        DB      '4'             ;OFF GRAPHIC CURSOR
        DW      G04
        DB      '5'             ;GRAPHIC CURSOR POSITION
        DW      G05
        DB      '6'             ;DOT ON
        DW      G06
        DB      '7'             ;DOT OFF
        DW      G07
        DB      SPACE           ;no match
        DW      E09             ;then return
;
;-----------------------------------------------
;
K01:    CALL    INCSR
        XOR     A               ;carriage ret
        LD      (CUSRX),A
        JP      UDCUSR
;
K02:    CALL    INCSR
K02A:   LD      A,(CUSRY)       ;line feed
        CP      24
        JR      Z,BOTTOM
        INC     A
        LD      (CUSRY),A
        JP      UDCUSR
BOTTOM: LD      DE,(VSTART)
        LD      HL,640
        ADD     HL,DE
        EX      DE,HL
        LD      BC,25*640
        ADD     HL,BC
        LD      BC,640
        CALL    CLLINE
        LD      A,D
        AND     3FH
```

```
        LD      D,A
        LD      C,LCDDAT
        LD      A,VSTADL        ;start addr.
        OUT     (LCDIST),A
        OUT     (C),E
        LD      A,VSTADH
        OUT     (LCDIST),A
        OUT     (C),D
        LD      (VSTART),DE
        JP      UDCUSR
;
K03:    CALL    INCSR
K03A:   LD      A,(CUSRX)       ;cursor right
        CP      79
        JR      NZ,MVRHT
        XOR     A
        LD      (CUSRX),A
        JR      K02A
MVRHT:  INC     A
        LD      (CUSRX),A
        JP      UDCUSR
;
K04:    LD      A,(CUSRY)       ;cusor up
        OR      A
        JP      Z,RTPATH
        DEC     A
        LD      (CUSRY),A
        CALL    INCSR
        JP      UDCUSR
;
K05:    CALL    INCSR
K05A:   XOR     A               ;home cursor
        LD      (CUSRY),A
        LD      (CUSRX),A
        JP      UDCUSR
;
K06:    LD      A,3             ;escape
        LD      (ESCFLG),A
        JP      RTPATH
;
K07:    LD      A,(CUSRX)       ;cusor back
        DEC     A
        JP      P,BSNEXT
        LD      A,(CUSRY)
        OR      A
        JP      Z,RTPATH
        DEC     A
        LD      (CUSRY),A
        LD      A,79
BSNEXT: LD      (CUSRX),A
        CALL    INCSR
        JP      UDCUSR
;
K08:    LD      A,(GRFFLG)      ;display char.
        OR      A
```

```
        LD      A,C
        LD      (WORD),A
        JR      NZ,GRF
        RES     7,A             ;mask off bit 7
GRF:    LD      L,A
        LD      H,0
        ADD     HL,HL
        ADD     HL,HL
        ADD     HL,HL
        LD      DE,CROM
        ADD     HL,DE
        LD      DE,WORDBF
        LD      BC,8
        LD      A,(BANKBF)      ;rom bank
        AND     0F8H
        OR      ROM
        OUT     (PPIC),A
        LDIR
;
        LD      A,(GRFFLG)      ;inverse video?
        OR      A
        JR      NZ,CKIFLG
        LD      A,(WORD)
        OR      A
        JP      M,INVMOD
;
CKIFLG: LD      A,(INVFLG)
        OR      A
        JR      Z,WRVRAM
INVMOD: CALL    SETINV
;
WRVRAM: CALL    UPSCRN
        JP      K03A
;
E01A:   CALL    CHVRAM
        LD      HL,3FFFH        ;clear screen
CL:     XOR     A               ;home cusor
        LD      (HL),A
        DEC     HL
        LD      A,H
        OR      L
        JR      NZ,CL
        LD      (VSTART),HL
        XOR     A
        LD      C,LCDIST
        LD      B,VSTADL
        OUT     (C),B
        OUT     (LCDDAT),A
        INC     B
        OUT     (C),B
        OUT     (LCDDAT),A
        RET
;
E02:    CALL    CTEOL           ;clear to end
        JR      RRT             ;of line
```

```
;
CTEOL:  LD      A,(CUSRX)       ;clear to end
        NEG                     ;of line
        ADD     A,80
        JP      Z,RTPATH
        LD      C,A
        LD      B,0
        LD      HL,(CUSRPO)
        LD      A,8
CLMORE: PUSH    AF
        PUSH    BC
        PUSH    HL
        CALL    CLLINE
        POP     HL
        POP     BC
        POP     AF
        LD      DE,80
        ADD     HL,DE
        DEC     A
        JR      NZ,CLMORE
        RET
;
E03:    CALL    CTEOL           ;clear to end
        LD      A,(CUSRY)       ;of screen
        ADD     A,0E8H          ;24-cusry
        NEG
        JP      Z,RTPATH
        CALL    COMPUT
        LD      B,H
        LD      C,L
        LD      DE,(VERTIC)
        LD      HL,640
        ADD     HL,DE
        CALL    CLLINE
RRT:    JP      UDCUSR          ;
;
E04:    XOR     A               ;delete line
        LD      (CUSRX),A
TRANSF: LD      A,(CUSRY)
        ADD     A,0E8H
        NEG
        JR      Z,DELAST
        CALL    COMPUT
        LD      B,H
        LD      C,L
        LD      DE,(VERTIC)
        LD      HL,640
        ADD     HL,DE
        CALL    CHVRAM
LOA:    PUSH    BC
        LD      A,(HL)          ;rd vram
        NOP                     ;screen noise
        NOP
        LD      (DE),A
        INC     HL
```

```
        INC     DE
        POP     BC
        DEC     BC
        LD      A,B
        OR      C
        JR      NZ,LOA
        EX      DE,HL
        JR      LASTLN
DELAST: LD      HL,(VERTIC)
        JR      LASTLN
;
E05:    XOR     A               ;insert line
        LD      (CUSRX),A
        LD      A,(CUSRY)
        ADD     A,0E8H
        NEG
        JR      Z,DELAST
        CALL    COMPUT
        LD      B,H
        LD      C,L
        LD      HL,25*640-1
        LD      DE,(VSTART)
        ADD     HL,DE
        PUSH    HL
        LD      HL,24*640-1
        ADD     HL,DE
        POP     DE
        CALL    CHVRAM
MVDATA: PUSH    BC
        LD      A,(HL)
        NOP                     ;screen noise
        NOP
        LD      (DE),A
        DEC     HL
        DEC     DE
        POP     BC
        DEC     BC
        LD      A,B
        OR      C
        JR      NZ,MVDATA
LASTLN: LD      BC,640
        CALL    CLLINE
        JR      UDCUSR
;
E06:    LD      A,01            ;start graphic
        LD      (GRFFLG),A
        JR      RTPATH
;
E07:    LD      A,01            ;set inverse
        LD      (INVFLG),A
        JR      RTPATH
;
E08:    XOR     A               ;end inverse
        LD      (INVFLG),A
E09:    JR      RTPATH
```

```
;
;
;-----------------------------------------------
;clear vram, pass hl, bc
;
CLLINE: CALL    CHVRAM  ;change to vram
CLEAR:  LD      A,H
        AND     00111111B
        LD      H,A
        XOR     A
        LD      (HL),A
        INC     HL
        DEC     BC
        LD      A,B
        OR      C
        JR      NZ,CLEAR
        RET
;-----------------------------------------------
;calculate cusor addr. and o/p to lcd
;pass cusry, cusrx, vstart
;
UDCUSR: LD      A,(CUSRY)       ;(640xcusry)+560
        CALL    COMPUT          ;+cusrx
        LD      DE,(VSTART)
        ADD     HL,DE
        LD      (VERTIC),HL
        LD      DE,(CUSRX)
        ADD     HL,DE
        LD      (CUSRPO),HL
        LD      DE,560
        ADD     HL,DE
        LD      C,LCDDAT        ;o/p csr addr.
        LD      A,CSRADL
        OUT     (LCDIST),A
        OUT     (C),L
        LD      A,CSRADH
        OUT     (LCDIST),A
        OUT     (C),H
        CALL    INCSR
RTPATH: LD      A,(BANKBF)
        OUT     (PPIC),A
        LD      SP,IY
        RET
;-----------------------------------------------
;inverse char. pat. in cusrpo
;
INCSR:  NOP
        CALL    CHVRAM
        LD      HL,(CUSRPO)
        LD      IX,WORDBF
        LD      DE,80
        LD      B,8
RDPAT:  LD      A,(HL)
        LD      (IX),A
        ADD     HL,DE
```

```
        INC     IX
        DJNZ    RDPAT
        CALL    SETINV
        CALL    UPSCRN
        RET
;-----------------------------------------------
;subroutine to set char. pattern in wordbf to inverse mode
;
SETINV: LD      B,8             ;set inverse
        LD      HL,WORDBF
SETI:   LD      A,(HL)
        CPL
        LD      (HL),A
        INC     HL
        DJNZ    SETI
        RET
;-----------------------------------------------
;subroutine to update pattern on screen from wordbuf
;
UPSCRN: LD      HL,(CUSRPO)
        LD      IX,WORDBF
        LD      B,8
        LD      DE,80
        CALL    CHVRAM
WRPATN: LD      A,H
        AND     00111111B
        LD      H,A
        LD      A,(IX)
        LD      (HL),A
        INC     IX
        ADD     HL,DE
        DJNZ    WRPATN
        RET
;-----------------------------------------------
;subroutine to change to vram bank
;
CHVRAM: LD      A,(BANKBF)      ;change to vram
        AND     0F8H            ;bank
        OR      VRAM
        OUT     (PPIC),A
        RET
;-----------------------------------------------
;subroutine to multiply acc. by 640
;
COMPUT: LD      D,A
        LD      E,0
        LD      H,D
        LD      L,E
        SLA     D
        SRL     H
        RR      L
        ADD     HL,DE
        RET
;-----------------------------------------------
E01:    CALL    E01A    ;ACTIVATE GRAPHIC MODE
```

```
        XOR     A
        LD      (DOTPOSY),A
        LD      (DOTPOSX),A
        LD      (DOTPOSX+1),A
        LD      (DOTBYTE),A
        LD      (DOTBYTE+1),A
        LD      A,0C9H          ;OFF GRF CUSR
        LD      (GRFCSR),A
        LD      A,80H
        LD      (DOTBIT),A
        JP      K05A
;
G01:    LD      A,(INCSR)       ;ON TEXT CURSOR
        OR      A
        JR      Z,ENG01
        XOR     A
        LD      (INCSR),A
        CALL    INCSR
        LD      C,21H
        XOR     A
        OUT     (C),A
        DEC     C
        LD      A,CSRON
        OUT     (C),A
ENG01:  JP      RTPATH
;
G02:    LD      A,(INCSR)       ;OFF TEXT CURSOR
        OR      A
        JR      NZ,ENG01
        CALL    INCSR
        LD      A,0C9H
        LD      (INCSR),A       ;"RET" INST.
        LD      C,21H
        XOR     A
        OUT     (C),A
        DEC     C
        LD      A,CSROFF
        OUT     (C),A
        JR      ENG01
;
G03:    LD      A,(GRFCSR)      ;ON GRF CUSR
        OR      A
        JR      Z,ENG01
        XOR     A
        LD      (GRFCSR),A
        CALL    GRFCSR
        JR      ENG01
;
G04:    LD      A,(GRFCSR)      ;OFF GRF CUSR
        OR      A
        JR      NZ,ENG01
        CALL    GRFCSR
        LD      A,0C9H
        LD      (GRFCSR),A
        JR      ENG01
```

```
;
G05:    LD      A,3             ;MOVE GRF CUSR
        LD      (ESC5),A
        CALL    GRFCSR
        JR      ENG01
G05A:   DEC     A
        LD      (ESC5),A
        CP      2
        JR      NZ,XVALH
        LD      A,C
        CP      200             ;Y RANGE
        JR      C,YRANGE
        LD      A,200
YRANGE: LD      (DOTPOSY),A
        JR      ENG01
XVALH:  OR      A
        JR      Z,XVALL
        LD      A,C
        LD      (DOTPOSX+1),A
        JR      ENG01
XVALL:  LD      A,C
        LD      (DOTPOSX),A
        LD      A,(DOTPOSX+1)
        CP      2H
        JR      C,XRANGE
        JR      NZ,OUTRAN
        LD      A,C
        CP      80H             ;X RANGE
        JR      C,XRANGE
OUTRAN: LD      BC,27FH
        LD      (DOTPOSX),BC
XRANGE: LD      A,(DOTPOSX)
        AND     07H
        LD      B,A
        INC     B
        XOR     A
        SCF
SETDOT: RR      A
        DJNZ    SETDOT          ;DOTBIT IN ACC.
        LD      (DOTBIT),A
        LD      DE,(DOTPOSX)
        LD      B,3
DIVID8: SRL     D
        RR      E
        DJNZ    DIVID8
        PUSH    DE
        LD      A,(DOTPOSY)
        LD      L,A
        LD      H,0
        ADD     HL,HL
        ADD     HL,HL
        ADD     HL,HL
        ADD     HL,HL   ;*16
        PUSH    HL
        ADD     HL,HL
```

```
        ADD     HL,HL   ;*64
        POP     DE
        ADD     HL,DE   ;*80
        POP     DE
        ADD     HL,DE   ;DOTBYTE
        LD      (DOTBYTE),HL
        CALL    GRFCSR
ENG05:  JP      RTPATH
;
G06:    CALL    CHVRAM
        LD      HL,(DOTBYTE)    ;DOT ON
        LD      A,(DOTBIT)
        OR      (HL)
        LD      (HL),A
        JR      ENG05
;
G07:    CALL    CHVRAM          ;DOT OFF
        LD      A,(DOTBIT)
        CPL
        LD      HL,(DOTBYTE)
        AND     (HL)
        LD      (HL),A
        JR      ENG05
;
;-----------------------------------------------
;TOGGLE COLOR OF GRF. CUSOR AT DOT POS Y,X
;
GRFCSR: RET
        CALL    CHVRAM
        LD      HL,(DOTBYTE)
        LD      DE,80
        XOR     A
        SBC     HL,DE
        LD      B,6
        LD      A,(DOTPOSY)     ;TOGGLE COLOR OF
        DEC     A               ;GRF. CUSOR AT
        CP      -1              ;DOT POSY,X
        JR      Z,LOHALF
UPBIT:  DEC     A
        CP      -1
        JR      Z,LOHALF
        PUSH    AF
        LD      DE,80
        XOR     A
        SBC     HL,DE
        LD      C,(HL)
        LD      A,(DOTBIT)
        XOR     C
        LD      C,A
        LD      (HL),C
        POP     AF
        DJNZ    UPBIT
;
LOHALF: LD      HL,(DOTBYTE)
        LD      DE,80
```

```
        ADD     HL,DE
        LD      B,6
        LD      A,(DOTPOSY)
        INC     A
        CP      200
        JR      Z,LFHALF
LOBIT:  INC     A
        CP      200
        JR      Z,LFHALF
        PUSH    AF
        LD      DE,80
        ADD     HL,DE
        LD      C,(HL)
        LD      A,(DOTBIT)
        XOR     C
        LD      C,A
        LD      (HL),C
        POP     AF
        DJNZ    LOBIT
;
LFHALF: LD      HL,(DOTBYTE)    ;LEFT HALF
        DEC     HL
        LD      C,(HL)
        INC     HL
        LD      D,(HL)
        LD      A,(DOTBIT)
        LD      B,0
SHTRT:  SRL     C
        RR      D
        RR      E
        INC     B
        SRL     A
        JR      NC,SHTRT
        LD      A,0FEH
        XOR     D
        LD      D,A
SHTLF:  SLA     E
        RL      D
        RL      C
        DJNZ    SHTLF
        LD      (HL),D
        PUSH    HL
        LD      HL,(DOTPOSX)
        LD      A,H
        OR      L
        POP     HL
        JR      Z,RTHALF
        DEC     HL
        LD      (HL),C
;
RTHALF: LD      HL,(DOTBYTE)    ;RIGHT HALF
        INC     HL
        LD      E,(HL)
        DEC     HL
        LD      D,(HL)
```

```
        LD      A,(DOTBIT)
        LD      B,0
SHTLFT: SLA     E
        RL      D
        RL      C
        INC     B
        SLA     A
        JR      NC,SHTLFT
        LD      A,7FH
        XOR     D
        LD      D,A
SHTRHT: SRL     C
        RR      D
        RR      E
        DJNZ    SHTRHT
        LD      (HL),D
        PUSH    HL
        PUSH    DE
        LD      HL,(DOTPOSX)    ;DOTPOSX=639?
        LD      DE,0FD81H
        ADD     HL,DE
        POP     DE
        POP     HL
        RET     C
        INC     HL
        LD      (HL),E
        RET
;-----------------------------------------------
;list character from register c
;
LIST:   LD      A,(IOBYTE)      ;lst:=?
        BIT     7,A             ;=tty,crt?
        JP      Z,CONOUT
        BIT     6,A             ;=ull
        JR      NZ,PTPOUT
        LD      A,C
        OUT     (PRINTER),A
;
BUSY:   NOP                     ;screen noise
        IN      A,(PPIC)
        BIT     4,A
        JR      Z,BUSY
        CALL    LDELAY
        IN      A,(PPIA)        ;set stb low
        AND     7FH
        OUT     (PPIA),A
        CALL    LDELAY
        OR      80H             ;set stb high
        OUT     (PPIA),A
        RET
;-----------------------------------------------
;return list status (0 if not ready, 0FF if ready)
;
LISTST: LD      A,(IOBYTE)      ;chk lpt?
        AND     11000000B
```

```
        CP      10000000B
        JR      Z,LPTST
LSTRDY: LD      A,0FFH          ;all others ready
        RET
LPTST:  IN      A,(PPIC)
        BIT     4,A
        JR      NZ,LSTRDY
        XOR     A
        RET
;-----------------------------------------------
;punch character from register c
;
PUNCH:  LD      A,(IOBYTE)      ;tty?
        AND     11000000B
        JP      Z,CONOUT
;
PTPOUT: IN      A,(UTSREG)      ;tx rdy?
        RRA
        JP      NC,PTPOUT
        LD      A,C
        OUT     (UTDATA),A
        RET
;-----------------------------------------------
;read character into register a from reader device
;
READER: LD      A,(IOBYTE)      ;tty?
        AND     11000000B
        JP      Z,CONIN
;
TSTRX:  IN      A,(UTSREG)      ;rx rdy?
        BIT     1,A
        JP      Z,TSTRX
        IN      A,(UTDATA)
        LD      HL,BITMSK
        AND     (HL)
        RET
;-----------------------------------------------
;move to the track 00 position of current drive
;
HOME:   LD      C,00H           ;select track 0
        CALL    SETTRK
        LD      A,(HSTWRT)      ;pending write?
        OR      A
        JP      NZ,HOMED
        LD      (HSTACT),A
HOMED:  RET
;-----------------------------------------------
;select disk given by register C
;
SELDSK: LD      HL,0000H        ;error return code
        LD      A,C
        CP      02H             ;must be 0
        RET     NC              ;no carry if 3,4,5,..
        OR      A
        JR      Z,SELDR0
```

```
        LD      A,(DRV1FL)      ;1st time sel drv 1?
        OR      A
        JR      NZ,SELDR1
        IN      A,(PPIA)        ;see if drv1 ready
        LD      E,A
        OR      20H
        DI
        OUT     (PPIA),A
        LD      A,MTRON
        OUT     (CNTCREG),A
        CALL    HOLD
        IN      A,(FDCCREG)
        RLA
        JR      C,NDRV
        LD      A,0D8H          ;FORCE INTERUPT
        OUT     (FDCCREG),A
        CALL    LDELAY
        LD      A,0D0H
        OUT     (FDCCREG),A
        CALL    LDELAY
        LD      A,0CH           ;recal
        OUT     (FDCCREG),A
        CALL    HOLD
        CALL    HOLD
        LD      A,01
        LD      (DRV1FL),A
        DEC     A
        LD      (DR1TRK),A
        LD      A,E
        OUT     (PPIA),A
        LD      A,MTROFF
        OUT     (CNT2),A
SELDR1: LD      A,C
        LD      (SEKDSK),A
        LD      HL,DPB1
        RET
NDRV:   CALL    DISSEL
        LD      HL,0
        RET
SELDR0:
;disk number is in the proper range
;compute proper disk parameter header address
        LD      A,C             ;C=disk number 0,1
        LD      (SEKDSK),A
        LD      HL,DPB0
        RET
;---------------------------------------------------
;set track given by register c
;
SETTRK: LD      A,C
        LD      (SEKTRK),A
        RET
;---------------------------------------------------
;set sector given by register c
;
```

```
SETSEC: LD      A,C
        LD      (SEKSEC),A
        RET
;---------------------------------------------------
SECTRAN: LD     H,B     ;translate sector number bc
        LD      L,C
        RET
;---------------------------------------------------
;set dma address given by registers b and c
;
SETDMA: LD      L,C             ;low order address
        LD      H,B             ;high order address
        LD      (DMAADR),HL     ;save the address
        RET
;---------------------------------------------------
;the read entry point
;
READ:   XOR     A
        LD      (UNACNT),A
        INC     A
        LD      (READOP),A      ;rd operation
        LD      (RSFLAG),A      ;rd 1 hstsec
        INC     A               ;into hstbuf
        LD      (WRTYPE),A      ;treat as unalloc
        JP      RWOPER
;
;---------------------------------------------------
;the write entry point
;
WRITE:  XOR     A
        LD      (READOP),A      ;wr operation
        LD      A,C             ;wrtype in c
        LD      (WRTYPE),A
        CP      WRUAL           ;wr unalloc?
        JR      NZ,CHKUNA
;wr to unalloc, set params
;
        LD      A,BLKSIZ/128    ;next unalloc rec
        LD      (UNACNT),A
        LD      A,(SEKDSK)
        LD      (UNADSK),A      ;unadsk=sekdsk
        LD      A,(SEKTRK)
        LD      (UNATRK),A      ;unatrk=sectrk
        LD      A,(SEKSEC)
        LD      (UNASEC),A      ;unasec=seksec
;
;check for wr to unalloc sec
;
CHKUNA: LD      A,(UNACNT)      ;unalloc remain?
        OR      A
        JR      Z,ALLOC         ;skip if not
;
;more unalloc rec remain
;
        DEC     A               ;unacnt-1
```

```
        LD      (UNACNT),A
        LD      A,(SEKDSK)      ;same disk?
        LD      HL,UNADSK
        CP      (HL)            ;sekdsk=unadsk?
        JR      NZ,ALLOC        ;skip if not
;
        LD      A,(SEKTRK)      ;sektrk=unatrk?
        LD      HL,UNATRK
        CP      (HL)
        JR      NZ,ALLOC        ;skip if not
;
        LD      A,(SEKSEC)      ;same sec?
        LD      HL,UNASEC
        CP      (HL)            ;seksec=unasec?
        JR      NZ,ALLOC        ;skip if not
;
;match, move to next sec for future ref.
;
        INC     (HL)            ;unasec+1
        LD      A,(HL)          ;end of trk?
        CP      CPMSPT
        JR      C,NOOVF         ;skip if no overflow
;
;overflow to next trk
;
        LD      (HL),0          ;unasec=0
        LD      HL,UNATRK       ;unatrk+1
        INC     (HL)
;
;match found, mark as unnecessary pre-rd
;
NOOVF:  XOR     A
        LD      (RSFLAG),A      ;no pre-read
        JR      RWOPER
;
;not an unalloc rec, may need pre-read
;
ALLOC:  XOR     A
        LD      (UNACNT),A      ;unacnt=0
        INC     A               ;may need pre-rd
        LD      (RSFLAG),A      ;so, rsflag=1
;
;-----------------------------------------------
;common code for read and write follows
;enter here to perform the rd/wr
;
RWOPER: XOR     A
        LD      (ERFLAG),A      ;no error yet
        LD      A,(SEKSEC)      ;compute hstsec
        SRL     A               ;for hstsec=256
        LD      (SEKHST),A      ;store hstsec
;
;active hstsec?
;
        LD      HL,HSTACT       ;hst buf active?
```

```
        LD      A,(HL)
        LD      (HL),1          ;always become 1
        OR      A               ;was it already?
        JR      Z,FILHST        ;fill hst if not
;
        LD      A,(SEKDSK)
        LD      HL,HSTDSK
        CP      (HL)            ;sekdsk=hstdsk?
        JR      NZ,NMATCH
;
        LD      A,(SEKTRK)
        LD      HL,HSTTRK
        CP      (HL)            ;sektrk=hsttrk?
        JR      NZ,NMATCH
;
        LD      A,(SEKHST)
        LD      HL,HSTSEC
        CP      (HL)            ;sekhst=hstsec?
        JR      Z,MATCH         ;skip if match
;
NMATCH: LD      A,(HSTWRT)      ;is wr pending
        OR      A               ;flag set?
        CALL    NZ,WRHST        ;if yes wr to disk
;
;may have to fill the hst buf
;
FILHST: LD      A,(SEKDSK)
        LD      (HSTDSK),A
        LD      A,(SEKTRK)
        LD      (HSTTRK),A
        LD      A,(SEKHST)
        LD      (HSTSEC),A
        LD      A,(RSFLAG)      ;need to read?
        OR      A
        CALL    NZ,RDHST        ;yes, if 1
        XOR     A
        LD      (HSTWRT),A      ;no pending write
;
;copy data to or from buf
;
MATCH:  LD      A,(SEKSEC)      ;mask least sig.
        AND     SECMSK          ;bit of seksec
        LD      L,A             ;HL=A
        LD      H,0
        LD      B,7
MUL128: ADD     HL,HL
        DJNZ    MUL128
;
;********* AMENDED IN GRAPHIC BIOS ***********
;       ADD     HL,HL           ;HL*128
;       ADD     HL,HL
;       ADD     HL,HL
;       ADD     HL,HL
;       ADD     HL,HL
;       ADD     HL,HL
```

```
;       ADD     HL,HL
;********* AMENDED IN GRAPHIC BIOS ***********
        LD      DE,HSTBUF
        ADD     HL,DE
;
;hl has hst buf addr. plus offset
;
        LD      DE,(DMAADR)     ;get/put cp/m data
        LD      BC,128          ;length of move
        LD      A,(READOP)      ;which way?
        OR      A
        JR      NZ,RWMOVE       ;skip if read
;
;wr operation, mark and switch direction
;
        LD      A,1             ;set pending write
        LD      (HSTWRT),A
        EX      DE,HL           ;source/dest swap
RWMOVE: LDIR                    ;(hstbuf)<>(dmaadr)
;
;data has moved to/from hstbuf
;
        LD      A,(WRTYPE)      ;wr type
        CP      WRDIR           ;to directory?
        LD      A,(ERFLAG)      ;in case of error
        RET     NZ              ;no more process
;
;clear hstbuf for dir wr
;
        OR      A               ;error?
        RET     NZ              ;skip if so
        XOR     A
        LD      (HSTWRT),A      ;buf written
        CALL    WRHST
        LD      A,(ERFLAG)
        RET
;
;---------------------------------------------
;
;wrhst performs the phy. write to hst disk
;rdhst performs the phy. read from hst disk
;
;hstdsk=host disk#, hsttrk=host trk#,
;hstsec=host sec#. write "hstziz" bytes
;from hstbuf and return err in erflag
;return erflag non-zero if error
;
WRHST:  XOR     A               ;init rd indica
        LD      (RDIND),A
        JR      RDWR            ;phy wr to disk
;
;hstdsk=host disk#, hsttrk=host track#,
;hstsec=host sec#. read "hstziz" bytes
;into hstbuf and return err in erflag
;return erflag non-zero if error
```

```
;
RDHST:  LD      A,0FFH          ;init rd indica
        LD      (RDIND),A
;
RDWR:   DI
        LD      A,RETRY         ;retry count
        LD      (RTYCNT),A
        XOR     A
        LD      (ERFLAG),A
REREAD: IN      A,(PPIC)        ;m/fdbk=1?
        BIT     MFDBK,A
        JR      NZ,MTON         ;if yes goto mton
ONMTR:  LD      A,MTRON         ;set motor on
        OUT     (CNTCREG),A
        CALL    HOLD            ;delay 0.5 sec
;
MTON:   LD      A,MTRON         ;motor on again
        OUT     (CNTCREG),A
;
PWRUP:  LD      A,0D8H          ;force interupt
        OUT     (FDCCREG),A
        CALL    LDELAY          ;delay 16 usec
        LD      A,0D0H          ;force interupt
        OUT     (FDCCREG),A
        CALL    LDELAY
        CALL    WAIT
        LD      A,(HSTDSK)      ;select drive 1?
        CP      01H
        JR      NZ,DSK0         ;if no go to dsk0
;
        IN      A,(PPIA)
        OR      DRIVE1          ;select drive 1
        OUT     (PPIA),A
        LD      A,(DR1TRK)      ;hsttrk=dr1trk?
        OUT     (TR),A
        LD      B,A
        LD      A,(HSTTRK)
        CP      80
        JP      NC,RDERR
RT1:    LD      (DR1TRK),A
        CP      B
        JR      Z,UDSR
        OUT     (DR),A
        JR      FINDTK
;
DSK0:   IN      A,(PPIA)
        OR      DRIVE0
        OUT     (PPIA),A
        LD      A,(DR0TRK)      ;hsttrk=dr0trk?
        OUT     (TR),A
        LD      B,A
        LD      A,(HSTTRK)
        CP      80
        JP      NC,RDERR
RT2:    LD      (DR0TRK),A
```

```
        CP      B
        JR      Z,UDSR
        OUT     (DR),A
;
FINDTK: IN      A,(FDCSREG)     ;FDC busy?
        RRA
        JP      C,RDERR         ;busy go err
;
        LD      A,SEEK          ;seek track
        OUT     (FDCCREG),A
        LD      HL,1250         ;delay 15 msec
        CALL    HOLDE
        CALL    WAIT            ;seek over?
        AND     10011001B       ;error occour?
        JP      NZ,RDERR        ;if yes, go and exit
UDSR:   LD      A,(HSTSEC)      ;update SR in FDC
        CP      18
        JP      NC,NORTY
        OUT     (SR),A
NOTRDY: CALL    WAIT
        BIT     7,A
        JR      NZ,NOTRDY
;-----------------------
        LD      A,(RDIND)       ;see if rd or wr
        OR      A
        JR      Z,WROP          ;no, then it's wr
;-----------------------
RDOP:   LD      HL,NMI          ;save nmi content
        LD      DE,INSTBF
        LD      BC,6
        LDIR
        LD      HL,RSERVE
        LD      DE,NMI
        LD      BC,6
        LDIR
;
        LD      A,RDSEC         ;load rd sec command
SINGL:  LD      HL,HSTBUF       ;set destin. addr.
        LD      B,0             ;transfer 256 bytes
        LD      C,DR            ;FDC data register
        LD      IX,RDWAIT
;
        OUT     (FDCCREG),A     ;rd sec command
RDWAIT: HALT
        JP      (IX)
ALLRD:  POP     DE
;
        LD      HL,INSTBF
        LD      DE,NMI
        LD      BC,6
        LDIR
;
        CALL    WAIT            ;check error
        AND     10011001B
        JP      Z,DISSEL
```

```
GOERR:  JR      RDERR
;-----------------------
WROP:   IN      A,(PPIC)        ;wprot?
        RLA
        JR      NC,NORTY
        LD      HL,NMI          ;save nmi content
        LD      DE,INSTBF
        LD      BC,6
        LDIR
        LD      HL,WSERVE
        LD      DE,NMI
        LD      BC,6
        LDIR
;
        LD      A,WRSEC         ;load wr sec command
ONESID: LD      HL,HSTBUF       ;set source addr.
        LD      B,00            ;transfer 256 bytes
        LD      C,DR            ;FDC data register
        LD      IX,WRWAIT       ;mask status reg.
;
        OUT     (FDCCREG),A     ;wr sec command
WRWAIT: HALT
        JP      (IX)
ALLWR:  POP     DE
;
PRTCT:  LD      HL,INSTBF       ;restore content
        LD      DE,NMI
        LD      BC,6
        LDIR
;
        CALL    WAIT            ;check error
        AND     11011001B
        JP      Z,DISSEL
;-----------------------
RDERR:  LD      A,RECAL         ;restore drive
        OUT     (FDCCREG),A
        CALL    LDELAY
        CALL    WAIT
        LD      A,(HSTDSK)
        RRA
        JR      C,RESDV1
        XOR     A
        LD      (DR0TRK),A
        JR      TSTRTY
RESDV1: XOR     A
        LD      (DR1TRK),A
TSTRTY: LD      HL,RTYCNT       ;get retry count
        DEC     (HL)
        JP      NZ,REREAD
NORTY:  LD      A,01H           ;return error
        LD      (ERFLAG),A
DISSEL: IN      A,(PPIA)        ;dis-select drive 0,1
        AND     DISDRV
        OUT     (PPIA),A
        LD      A,MTROFF        ;oneshot 5sec
```

```
        OUT     (CNT2),A
        RET
;-----------------------------------------
;subroutine to rd FDC status reg. and wait
;until busy flag (bit 0) is reset
;
WAIT:   IN      A,(FDCSREG)     ;rd stat. reg.
        BIT     0,A
        JR      NZ,WAIT
        RET
;-----------------------------------------
;instructions to be copied to nmi
;
RSERVE: INI                     ;6 bytes
        RET     NZ
        JP      ALLRD
;
WSERVE: OUTI                    ;0066H-0067H
        RET     NZ              ;0068H
        JP      ALLWR
;
;-----------------------------------------
HOLD:   LD      HL,0000H        ;delay 0.7 sec
HOLDE:  DEC     HL              ;12 usec/loop
        LD      A,H
        OR      L
        JR      NZ,HOLDE
        RET
;-----------------------------------------
;subroutine to delay a few micro
LDELAY: PUSH    IX              ;delay
        POP     IX
        PUSH    IX
        POP     IX
        RET
;----- DELETED IN GRAPHIC BIOS ------------------
;subroutine to change bank pass c, acc
;if c=0ff ret current bank in acc
;SWCHBK:        LD      A,C
;       INC     A
;       IN      A,(PPIC)
;       JR      NZ,CHBANK
;       AND     BKIVMS
;       RET
;CHBANK: AND    BKMASK
;       OR      C
;       OUT     (PPIC),A
;       RET
;------- CHANGED IN GRAPHIC BIOS ------------------
;NEWTBL: DB     -1,-1,-1,-1,-1
;        DB     -1,-1,-1,-1,-1
;OLDTBL: DB     -1,-1,-1,-1,-1
;        DB     -1,-1,-1,-1,-1
;----this six ptr/ctr must be contiguous----------
KEYBUF: DB      -1      ;keycode                    ö
```

```
KEYCNT: DB      0       ;no. of keycode                 ö
KEYPTR: DW      0       ;point addr.of key-code         ö
KEYBUF2: DB     -1      ;alternate key buf              ö
KEYCNT2: DB     0       ; * no. of keycode              ö
KEYPTR2: DW     0       ; * point addr. of key-codeö
DOTBIT:  DB     10000000B ;POS. OF DOT IN DOTBYTE
;------ DELETED IN GBIOS ----------------------
;
;AUTO:  DB      7,'AUTORUN',0
;-----------------------------------------------
CCPETY: DW      CCP
HSTBUF  EQU     $                       ;HOST BUFFER
;
BOOT:   DI
        LD      A,0C0H                  ;init ppi
        OUT     (PPIA),A
        LD      A,41H                   ;init modem
        OUT     (70H),A
;
;********** DELETED IN RELEASE 1.1 ************
;       LD      A,0D0H                  ;recal drive 0
;       OUT     (PPIA),A
;       LD      A,MTRON
;       OUT     (CNTCREG),A
;       LD      A,01H
;       LD      (RTYCNT),A              ;retry cnt
;       CALL    HOLD
;       CALL    RDERR
;********** DELETED IN RELEASE 1.1 ************
        XOR     A
        LD      HL,FBEGIN
        LD      B,FLENGTH
FDATA:  LD      (HL),A
        INC     HL
        DJNZ    FDATA
        DEC     A
        LD      B,20
FDATA2: LD      (HL),A
        INC     HL
        DJNZ    FDATA2
;********** DELETED IN GBIOS 1.0 **************
;       LD      E,05H                   ;init lcd
;       LD      C,LCDIST
;       OUT     (C),E
;       XOR     A
;       OUT     (LCDDAT),A
;       INC     E
;       OUT     (C),E
;       OUT     (LCDDAT),A
;       INC     E
;       OUT     (C),E
;       OUT     (LCDDAT),A
;       INC     E
;       OUT     (C),E
;       OUT     (LCDDAT),A
```

```
;********** DELETED IN GBIOS 1.0 **************
;       LD      HL,HEADING      ;print heading
;PRINT: LD      A,(HL)
;       OR      A
;       JR      Z,ENPRIN
;       LD      C,A
;       PUSH    HL
;       CALL    CONOUT
;       POP     HL
;       INC     HL
;       JR      PRINT
        LD      C,1AH           ;CLEAR SCREEN
        CALL    CONOUT
ENPRIN: LD      A,94H           ;init iobyte
        LD      (IOBYTE),A
        XOR     A               ;select drive zero
        LD      (CDISK),A
        JP      WBOOT
;-----------------------------------------
;message to be print out in cboot
;
;HEADING: DB    1BH,'*',0AH,0AH,' CP/M VERSION 2.2  Graphic BIOS 1.0',
;       DB      ' Copyright 1985 by Digital Research',0DH,0AH,0
;-----------------------------------------
;
;DIRBF  128 scratch directory area
;
        ORG     HSTBUF+256      ;beginning of data area
DIRBF:  DS      128
ALL00:  DS      24              ;allocation vector 0
ALL01:  DS      24              ;allocation vector 1
CHK00:  DS      32              ;check vector 0
CHK01:  DS      32              ;check vector 1
;
;-------UNINITIALIZE DATA--------------
;
SEKTRK: DS      1       ;seek track no.
SEKSEC: DS      1       ;seek sector no.
;
HSTDSK: DS      1       ;host disk no.
HSTTRK: DS      1       ;host track no.
HSTSEC: DS      1       ;host sector no.
;
SEKHST: DS      1       ;hold the phy. sec no.
HSTACT: DS      1       ;host active flag
;
UNACNT: DS      1       ;no. of unalloc rec left
UNADSK: DS      1       ;last unalloc disk
UNATRK: DS      1       ;last unalloc track
UNASEC: DS      1       ;last unalloc sec
;
RSFLAG: DS      1       ;read sec flag
READOP: DS      1       ;1 if read operation
WRTYPE: DS      1       ;write operation type
DMAADR: DS      2       ;last dma address
```

```
;
CNTRL:   DS      1          ;nonzero if ctrl pressed
SHIFT:   DS      1          ;nonzero if shift pressed
NPRESS:  DS      1          ;new pressed key-code
TEMP:    DS      1          ;put temp. old key
;
CSNY:    DS      1          ;temp. buffer
WORD:    DS      1          ;store ascii from bdos
WORDBF:  DS      8          ;store char. pattern
BANKBF:  DS      1          ;store prev. bank
;
RDIND:   DS      1          ;phy. rd or wr indicator
RTYCNT:  DS      1          ;hold current try no.
INSTBF:  DS      9          ;hold instruction in nmi
INTBF:   DS      4          ;hold instruction in int
ERFLAG:  DS      1
FLAG4:   DS      1          ;indicate no bad key
;
FBEGIN   EQU     $
FLAG1:   DS      1
FLAG2:   DS      1          ;zero if only 1 oldkey
FLAG3:   DS      1          ;timing indicator
TIMER:   DS      1          ;timer
CAPS:    DS      1          ;nonzero if caplock
CUSRX:   DS      2          ;cusor x-position
CUSRY:   DS      2          ;cusor y-position
CUSRPO:  DS      2          ;cusor top coordinate
VERTIC:  DS      2          ;curnt line top left
VSTART:  DS      2          ;vram starting addr.
ESCFLG:  DS      1          ;indica esc typed
GRFFLG:  DS      1          ;indica graphic mode
INVFLG:  DS      1          ;inverse flag
SEKDSK:  DS      1          ;seek disk no.
HSTWRT:  DS      1          ;write pending flag
DR0TRK:  DS      1          ;current pos. of head
DR1TRK:  DS      1          ;current pos. of head
DRV1FL:  DS      1          ;1st time access drv 1
;
;THE FOLLOWING ARE ADDED IN FOR GRAPHIC BIOS
;
DOTPOSY: DS      1          ;DOT POS. Y
DOTPOSX: DS      2          ;DOT POS. X
DOTBYTE: DS      2          ;LOCATION OF DOT IN VRAM
ESC5:    DS      1          ;ESC5=YX FUNCTION
FLENGTH  EQU     $-FBEGIN
;------- THE FOLLOWING 20 BYTES MUST FOLLOW ABOVE ---
NEWTBL:  DS      10
OLDTBL:  DS      10
FLENG2   EQU     $-FLENGTH
;
         DS      20                 ;temp stack for
STACK2   EQU     $                  ;bank change
         END
;-----------------------------------------------
```