This document details the files changes necessary (by module) to enable continuous LBA addressing (No Holes). Please note that at this time only 64 sectors/track > LBA are supported with a maximum drive size of 8MB, this is ideally suited to our S-100 Z80 systems using CF flash drives.

No warranty is given or implied and you use this software at your own risk (Hobbyist warranty ☺)

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
==
PLEASE NOTE - the file HSYSGEN.ASM has not had it's wrlba: section updated as it was not my intention to write to the boot sector in this way.
==
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

==

CPMLDR

==

Changes to **HLDRBIOS.ASM** in order as listed

Insert replacement wrlba: routine

```
;  Convert CP/M Track and Sector requests to LBA and write to drive registers
wrlba:
            LHLD    @TRK        ;Get CPM requested Track Hi&Lo
            MVI     H,00H       ;zero high track byte
            MOV     A,L         ;load low track byte to accumulator
            CPI     00H         ;check for 0 track and skip track loop
            JZ      lbasec
            MVI     B,06H       ;load counter to shift low track value 6 places to left i.e X 64
lbatrk:
            DAD     H           ;Add HL to itself 6 times to multiply by 64
            DJNZ    lbatrk      ;loop around 6 times i.e x 64

lbasec:
            LDA     @SECT       ;Get CPM requested sector
            ADD     L           ;Add value in L to sector info in A
            JNC     lbatot      ;If no carry jump to lba 'total' calculation
            INR     H           ;carry one over to H
lbatot:
            MOV     L,A         ;copy accumulator to L
                                ;HL should now contain correct LBA value


            MVI     D,0         ;Send 0 for upper cyl value
            MVI     E,REGcylinderMSB
            CALL    IDEwr8D     ;Send info to drive
```

```
                MOV     D,H             ;load lba high byte to D from H
                MVI     E,REGcylinderLSB
                CALL    IDEwr8D         ;Send info to drive

                MOV     D,L             ;load lba low byte to D from L
                MVI     E,REGsector
                CALL    IDEwr8D         ;Send info to drive

                MVI     D,1             ;For now, one sector at a time
                MVI     E,REGseccnt
                CALL    IDEwr8D

                RET
```

Updated disk parameter block to specify 64 sectors

```
        ; IDE HARD DISK PARAMETER BLOCK:
HD$DPB:         DPB     512,64,256,2048,1024,1,8000H
```

==========

CPM3.SYS

==========

Changes to **HIDE3.ASM** in order as listed

Updated number of sectors to 40H (64)

```
MAXSEC                  EQU     40H     ;Sectors per track for CF my Memory drive, Kingston CF 8G.
                                        ; (CPM format, 0-3D)
                                        ; Value changed to 40H for 'No Holes' LBA
```

Updated disk parameter block to specify 64 sectors

```
IDEHD$DPB:
;       DPB     512,32,512,2048,1024,2,8000H
;       DPB     256,32,1001,2048,1024,2,8000H
;       DPB     512,60,256,2048,1024,1,8000H
;       DPB     512,62,252,2048,1024,1,8000H
        DPB     512,64,256,2048,1024,1,8000H
```

Insert replacement wrlba: routine

```
; Convert Track & Sector disk requests to LBA addressing & write to drive

wrlba:
                LHLD    @TRK            ;Get CPM requested Track Hi&Lo
                MVI     H,00H           ;zero high track byte
                MOV     A,L             ;load low track byte to accumulator
                CPI     00H             ;check for 0 track and skip track loop
                JZ      lbasec
                MVI     B,06H           ;load counter to shift low track value 6 places to left i.e X 64
lbatrk:
                DAD     H               ;Add HL to itself 6 times to multiply by 64
                DJNZ    lbatrk          ;loop around 6 times i.e x 64

lbasec:
                LDA     @SECT           ;Get CPM requested sector
                ADD     L               ;Add value in L to sector info in A
                JNC     lbatot          ;If no carry jump to lba 'total' calculation
                INR     H               ;carry one over to H
lbatot:
                MOV     L,A             ;copy accumulator to L
                                        ;HL should now contain correct LBA value


                MVI     D,0             ;Send 0 for upper cyl value
                MVI     E,REGcylinderMSB
                CALL    IDEwr8D         ;Send info to drive

                MOV     D,H             ;load lba high byte to D from H
                MVI     E,REGcylinderLSB
                CALL    IDEwr8D         ;Send info to drive

                MOV     D,L             ;load lba low byte to D from L
                MVI     E,REGsector
                CALL    IDEwr8D         ;Send info to drive

                MVI     D,1             ;For now, one sector at a time
                MVI     E,REGseccnt
                CALL    IDEwr8D

                RET
```

==========