Altair CP/M 2.2AT Ver 1.1

CP/M 2.2AT supports an Altair 8800 computer with both an Altair and Tarbell floppy controller installed. Drives A & B are Altair drives 0 & 1. Drives C & D are Tarbell drives 0 & 1. The boot drive is always Altair drive 0. The Altair controller can be either the 8" or the 5.25" (mini-disk) controller as selected by conditional assembly.

The BIOS uses track buffering to improve disk performance. Compared to a non-buffered BIOS, performance of program with substantial disk I/O improves by 25% to 75%. For the Altair drives, the same disk layout used by the original versions of Altair CP/M 1.4 and 2.2 is used. For the Tarbell controller, the standard IBM SSSD soft-sectored format is used.

This BIOS provides full IOBYTE support to allow use of a variety of standard Altair I/O boards. See IOBYTE below.

During cold boot, the BIOS looks for a Teletype on the console port and if detected (based on baud rate), subsequently follows any CR to the console device with a NULL to give the Teletype carriage time to reach the left margin. The send null flag (SNDNULL) is located immediately following the MODE and IOBYTE in memory so it can be over-ridden if needed.

This BIOS adds a disk select timeout for the Altair 8" floppy so if a non-present or empty drive is selected, it will eventually timeout. To still allow the operator time to insert a disk, close the door, and wait for the drive enable one-shot to expire, the default timeout is set to seven seconds. This value can be patched in the SELTIME (select timeout) location. The Altair mini-disk does not support this capability. The Tarbell controller already provides this feature.

Extended BIOS function support is available via the selDsk entry point. This BIOS supports "Flush Track Buffer" (function 0). See selDsk in the ABIOS.ASM file for details.

IOBYTE Implementation

CP/M 2.2AT implements CP/M's IOBYTE feature which allows redirection of logical CP/M devices to a variety of physical devices. IOBYTE implementation is required by some programs (e.g., Kermit) for proper operation. The possible logical-to-physical device assignments are shown below:

CON device (bits 1,0): 00 - TTY on 88-SIO at 0/1 01 - CRT on 88-2SIO at 10h/11h* 10 - BAT (indirect through RDR and LST devices) 11 - UC1 on 88-2SIO at 12h/13h	**PUN** device (bits 5,4): 00 - TTY on 88-SIO at 0/1 01 - PTP on 88-2SIO at 10h/11h* 10 - UP1 on cassette port (SIO 6/7) 11 - UP2 on 88-2SIO at 12h/13h
RDR device (bits 3,2): 00 - TTY on 88-SIO at 0/1 01 - PTR on 88-2SIO at 10h/11h* 10 - UR1 on cassette port (SIO 6/7) 11 - UR2 on 88-2SIO at 12h/13h	**LST** device (bits 7,6): 00 - TTY on 88-SIO at 0/1 01 - CRT on 88-2SIO at 10h/11h* 10 - LPT on 88-LPC at 2/3 11 - UL1 on 88-2SIO at 12h/13h

* = Default IOBYTE

Patching the IOBYTE

The STAT utility in CP/M is typically used to make IOBYTE device assignments as required. These assignments are temporary and the defaults shown above are restored whenever CP/M is cold started. To permanently modify the IOBYTE, the default IOBYTE in the CP/M boot image must be patched.

To change the default IOBYTE that CP/M loads on cold start, the MOVCPMx file can be patched or memory can be modified after running MOVCPMx, but prior to writing the image to disk using SYSGEN. Patching the IOBYTE would be necessary, for example, to force CP/M to use the serial port on an SIO board as the console upon cold start instead of the 2SIO serial port.

The IOBYTE is at location 20B4h in memory after running MOVCPMx or loading MOVCPMx.COM into memory using DDT. Each of the four logical devices occupies a two bit field in the IOBYTE. Each two bit field, in turn, selects one of four physical devices for the logical device. The bit field locations and values are shown in the device mapping table above.

The default IOBYTE in binary is 01010101 (55h). Note that this is a value of 1 in each of the four logical device fields. As an example, to force the CON, RDR and PUN devices to use an SIO serial board port at I/O address 0/1 and an 88-LPC line printer controller as the LST device, the IOBYTE should be set to binary 10000000 (80h).

Updating an Altair CP/M to 2.2AT

Installing or updating to CP/M version 2.2AT is as simple as running MOVCPM8 (or MOVCPM5 for 5.25” mini disk), located in the same directory as this file, followed by SYSGEN. You can upload MOVCPMx.COM to a CP/M disk using the PCGET utility.

Additional programs in this directory include a fast format utility (AFORMAT) and a fast disk copy utility (ACOPY) for the Altair drives, and a multi-function disk utility (COPY) for the Tarbell drives. You’ll need to upload these utilities and use them for disk maintenance rather than similar utilities that might already be on your disk.

Alternatively, a complete disk image can be written using the PC2FLOP utility. The disk image file “CPM22AT11-24K.DSK” is sized for 24K of RAM. After booting the new disk, run MOVPCM8 followed by SYSGEN to size CP/M for the amount of RAM in your system. Use “CPM22AT11-24K (mini).DSK” and MOVCPM5 for the 5.25” mini disk.

Using MOVCPMx

MOVCPMx is used to create an image of CP/M configured for the amount of RAM in a system. The SYSGEN program is used to write that image to the boot sectors of a disk. Following are a few examples of using MOVCPM8:

`A>MOVCPM8`	Create CP/M sized for the amount of RAM in this system
`A>MOVCPM8 48`	Create CP/M sized for 48K of RAM
`A>MOVCPM8 56`	Create CP/M sized for 56K of RAM

When MOVCPM8 completes, the CP/M image is in memory. The very next step MUST be to write the image to the boot sectors on a disk (run SYSGEN), or to save the image as a file to disk (use the SAVE command as detailed in messages from MOVCPM8).

To write the image to the boot sectors of a disk, run the SYSGEN program. A typical SYSGEN session after executing MOVCPM8 is as follows:

```
A>SYSGEN
SYSGEN Ver 1.2
For MITS DISK CP/M

Source drive name (or RETURN to skip) – press return to use the CP/M image in memory
Destination drive name (or RETURN to reboot)– type drive letter, e.g., A (not echoed)
Destination on A, then type RETURN – press RETURN to write CP/M to the specified drive
Function Complete
Destination drive name (or RETURN to reboot)– if desired, write CP/M to another floppy or to a
                                              different drive. If finished and a disk just
                                              written is in drive A, reboot the computer from the
                                              front panel instead of pressing return.
```