CP/M for the Altair 8800 and Altair Floppy Drives

When I set out to find an original version of CP/M that ran on an Altair 8800 computer with Altair's 88-DCDD floppy drive, I didn't think it would be a difficult task. CP/M archive websites have versions of CP/M customized to run on every old platform imaginable – every platform but the Altair, that is!

The rarity of CP/M releases for the Altair platform was due, in part, to a rapid decline in the Altair's popularity at just the time CP/M sales were taking off. 1977 marked Pertec's acquisition of MITS and the release of Altair DOS, but by the summer of 1978, Pertec ceased production of the Altair computer line. Meanwhile, during 1977-78, CP/M version 1.4 became the most widely adopted version of CP/M to date. In 1979, a year after the Altair computer line was discontinued, CP/M version 2.2 – the industry standard we're most familiar with – was released.

With the large number of Altair systems already in the marketplace, it seems that ports of CP/M to the Altair platform would still have been fairly common. However, coding a BIOS to support the 88-DCDD floppy drives was non-trivial. The Altair drives were heavily dependent on software to provide even the most basic drive functions. Newer S-100 controllers and drives were much better suited to the coming age of disk-based computing.

With the help of members of the "Altair Computer Club," a Yahoo group, I was directed to the only working copy of CP/M 2.2 I have been able to find that was produced for the original Altair hardware. An Altair distributor in the Houston, Texas area, "Burcon Inc.," claimed copyright 1980 on a version of CP/M 2.2 for the Altair 8800 and 88-DCDD floppy drive. The Burcon CP/M, in turn, is a direct descendant of an Altair compatible CP/M 1.4, copyright 1979 by Lifeboat Associates. It is not clear whether either Lifeboat or Burcon was actually involved in authoring the software for these versions of CP/M.

Within the Yahoo group, Burcon CP/M existed on eight-inch floppy media, without source, and with very few copies left. I felt it was important to permanently archive this part of Altair history. I also wanted to recreate the source for the Burcon BIOS so that not only would this version of CP/M be archived, but it would also available for study and for generation of new CP/M systems for the Altair.

Tom Sanderson of the Altair Computer Club provided me with a memory dump of the BIOS while Burcon CP/M was running on his computer. I have taken this memory dump and generated a commented source of the Burcon CP/M BIOS. This source file is included in the folder along with this document. I have also created the tools necessary to "sysgen" new or modified versions of Burcon CP/M as needed.

Also included in the folder are two Burcon CP/M disk images: "cpm.dsk" is an applications disk with a number of common CP/M programs including mbasic. "sysgen.dsk" includes all the files required to generate new versions of Burcon CP/M. The image files are compatible with the Altair32 and SIMH emulators. Be sure to download and run the latest version of the emulators as a bug fix was required in the floppy emulation in order to properly execute code in Burcon CP/M. Finally, the image files can also be written directly to an eight-inch floppy and run on real Altair hardware. Feel free to contact me regarding this process.

The Burcon (MITS) BIOS for CP/M 2.2

The Altair 88-DCDD floppy controller and drives provide very little hardware support for data error checking, sector positioning or track positioning. The Altair floppy has 137 bytes per sector, leaving nine "unused" bytes assuming a typical 128 byte payload. It is up to software to effectively use the remaining bytes to provide data validation in support of the hardware. For example, it is up to software to set the MSBit of the 1st byte of each sector to act as a hardware sync bit. It is up to software to reduce head current for tracks 43-76. It is up to software to determine how data is verified and where to store the verification checksum. It is up to software to store and check track and sector number verification data in each sector (if stored or checked at all!). The MITS "standard," as used for the various versions of "Disk BASIC" and "Altair DOS," actually uses the 137 byte sector in one manner for tracks 0-5 and in a different manner for tracks 6-76.

Because the content of the entire physical sector, and how that content is used, is laid open to the application software, the Altair 88-DCDD is a difficult drive to code for and uses significant host computer resources. This is one reason the Altair computer and floppy combination was not an ideal platform for CP/M.

In the face of the obstacles mentioned above, the Burcon BIOS is a very well written piece of software. The BIOS implements a sector and track format that is essentially the same as the MITS standard mentioned above. However, since the directory and file format for CP/M is completely different than that used by MITS products, the floppies are not interchangeable between CP/M and MITS software. Extensive error checking is performed verifying track position, sector position, data checksum and sync/stop bytes.

Disk Format

The Burcon BIOS supports up to four drives. The drive are configured as follows:

- Block Size: 2,048 bytes
- Total Blocks: 150
- Directory Entries: 64
- Data/Directory Space Available: 307,200 bytes.
- Two system tracks are reserved for booting CP/M (not part of the space listed above)
- Single logical extent per physical extent (typical setting for this block size would be two logical extents per physical. Probably set this way for compatibility with CP/M 1.4).

For comparison, the Altair32 and SIMH emulators use 254 blocks x 1,024 bytes for 260,096 bytes of data and directory storage with 256 directory entries. Seven system tracks are reserved for booting CP/M.

Using a disk format very similar to MITS Disk BASIC and Altair DOS, the Burcon BIOS treats tracks 0-5 differently than tracks 6-76.This disk layout is detailed in the tables below.

Track 0 through 5

Byte	Use
0	Track number + 0x80
1-2	Sixteen bit address in memory of the end of the bootloader (0x100). This same value is set in all sectors of tracks 0-5.
3-130	128 byte data payload
131	Stop byte 0xff
132	Checksum (sum of the 128 byte payload)
133-136	Ignored

Track 6 through 76

Byte	Use
0	Track number + 0x80
1	Sector number
2-3	Not used, but in checksum
4	Checksum (bytes 2-134, not including byte 4)
5-6	Not used, but in checksum
7-134	128 byte data payload
135	Stop byte 0xff
136	Stop byte 0x00

For read operations on tracks 0-5, the BIOS verifies the track number, the stop byte at 131 and the checksum.

For read operations on tracks 6-76 , the BIOS verifies the track number, the sector number, the checksum, the 0xff stop byte at 135 and the 0x00 stop byte at 136.

Write operations put the check values listed above into the sector. The write operation also handles the head current reduction at track 43 (HCS).

CP/M provides for sector skewing to improve disk performance using the SECTRAN BIOS entry point. The Burcon BIOS uses an 8 sector skew for three sectors followed by a 10 sector skew every fourth sector. Since even skews are used, a single odd skew of 3 sectors is inserted halfway through the sector translation table. Interestingly, the BIOS does an *additional* 17 sector skew for tracks 6-76 on top of the CP/M sector skew. It seems the authors wanted to duplicate the 17 sector skew used in MITS products for tracks 6-76. However, since this skew is performed in addition to the CP/M skew, the result is not as expected. In fact, the math works out such that the 17 sector skew has almost no effect!

The head loading algorithm in the BIOS loads the head for any disk read or write (if not already loaded). The head is left loaded until console input is requested through CP/M. This keeps head load/unload activity to a minimum, yet at the same time, provides a reasonable point at which to unload the head to reduce media wear.

Additional Notes

The BIOS uses the 1st port on a 2SIO serial interface board (I/O address 0x10) as the CP/M Console, Punch and Reader. The 2nd port on the 2SIO board is initialized, but is not used. For the List device, the BIOS assumes an 88-LPC board (line printer controller) at I/O address 0x02.

To assist with code development, debug and to provide flexibility, the Burcon BIOS includes a flag byte that can be modified in the BIOS source or patched in memory. The flags provide the following options:

- On cold start, have the CCP process a stored command line on entry (set to false)
- On warm start, have the CCP process a stored command line on entry (set to false)
- Raw I/O – read/write the 137 byte sector directly from the BIOS local buffer with no additional processing (set to false)
- Enable 8080 interrupts after completion of disk I/O (set to true)
- Verify all disk writes (set to false)
- Force track number validation for all track seek operations (set to false). Note: This flag primarily affects write operations since the BIOS performs track validation for reads independent of this flag.