Miscellaneous UHCI Notes (page 3-13)

On the Intel 430TX and later motherboards, the two reserved bits in the Port register, bits 11:10, are actually used for over-current conditions. Bit 11 is used to indicate that there was a change in the over-current condition and is Read/Write Clear while bit 10 is Read-only and indicates whether there is an over-current condition. However, now for the reason why bit 10 was set in previous pages. Instead of setting a pin high, the hardware chose to set a pin low when an over current condition exists. Intel chose to have bit 10 clear to indicate normal operation while being set indicated an over-current condition, just the opposite of the voltage on the pin. Via Tech chose to have bit 10 set to indicate normal operation while being clear indicated an over-current condition, matching the hardware pin. You need to check the PCI Vendor ID to know how to decipher bit 10.

Appendix A

Location of the two bootable image files, included with each volume, are not in the `main/` folder but in their respective volume's main folder.

For example, the two image files for Volume 8, `freedos.img` and `freedos_hd.img`, are in the `main/usb` folder. Each respective volume will have similar image files in their respective folder.

These are bootable images, one as a floppy image and the other as an 80 Meg hard drive image, bootable to FreeDOS and each includes the utilities and other files ready to test on actual hardware.

The floppy image, `freedos.img`, has a minimal amount of FreeDOS installed so that there is enough room for the utilities needed. The larger image has a fully functional FreeDOS system and plenty of free space to be able to place other utilities and/or test files.

Appendix B

For some reason when the utilities are run under FreeDOS and QEMU, the combination of DJGPP (the C compiler used), FreeDOS, and CWSDPMI.EXE, you can only run each .EXE once. The next time you try, the QEMU environment freezes.

I found the alternative DPMI server called PMODE/DJ and tried it. It works just fine. I can run the .EXEs under a QEMU emulated environment just fine. Following the directions from PMODE/DJ, I have renamed its .EXE file to CWSDPMI.EXE so that the utilities will find and use the DMPI server.

Therefore, the CWSDPMI.EXE file on the image files is not the Sandmann version, it is the PMODE/DJ version.

The Sandmann version allows my .EXEs to run just fine under a Bochs emulation or real hardware. It must be something that QEMU doesn't like.

Since these utilities are intended to run on real hardware anyway, you may use either DPMI server.

The Sandmann version, the original CWSDPMI, with source, can be found at:

http://sandmann.dotster.com/cwsdpmi/

The PMODE/DJ version, with source, can be found at:
http://www.delorie.com/pub/djgpp/current/v2misc/pmode13b.zip
http://www.delorie.com/pub/djgpp/current/v2misc/pmode13s.zip

Thank you,
Ben