AHCI 1.1

Errata 001 Draft

AHCI 1_1 Errata_001.doc

Please send comments to Amber Huffman
amber.huffman@intel.com

Table of Contents

1 COMRESET Prior to Switching from AHCI to legacy mode

1.1 Description of Technical Issue

When switching from AHCI mode to legacy mode, software should issue a COMRESET to each port to ensure that any residual state is cleared and that a device signature is available.

1.2 Description of Correction to Specification

Modify section 10.2 as follows:

10.2 Hardware Prerequisites to Enable/Disable GHC.AE

When a legacy interface is supported in addition to AHCI (CAP.SAM = '0'), software may desire to switch from one mode to the other. Actively switching between AHCI and a legacy mode is strongly discouraged due to OS sensitivity to the class code used by the controller. This section lists hardware requirements only for switching between AHCI mode and a legacy mode, it does not comprehend any software considerations.

To switch from legacy mode to AHCI mode, the legacy interface shall be in an idle state and there should be no outstanding commands to any devices connected to the controller. Additionally, any transaction errors reported through the legacy task file status register (i.e. ERR == '1') shall be rectified prior to switching to AHCI mode. This can be accomplished via a software reset. At this point, software can begin the sequence for initializing the AHCI controller outlined in section 10.1.

To switch from AHCI mode to legacy mode, the AHCI controller must be brought to an idle state for all ports and there should be no outstanding commands to any devices connected to the controller. The PxCMD.ST should be cleared to '0' for all ports and software should ensure that the PxCMD.CR bit is cleared to '0' for all ports. The PxCMD.FRE bit should be cleared to '0' for all ports and software should ensure that PxCMD.FR is cleared to '0' for all ports. All outstanding errors shall be serviced by the host and all PxSERR registers shall be cleared to 0h. Then software should clear the GHC.IE bit to '0' to ensure there are no interrupts that occur on the AHCI controller. At this point, software may set the GHC.AE bit to '0'. After clearing the GHC.AE bit to '0', software should then issue a COMRESET to each SATA port to ensure that the legacy controller recognizes whether a drive is present on the port, and if so then captures its signature. Commands may be issued at this point to the legacy controller. Software shall initialize the devices after switching modes.

2 Various Typos

2.1 Description of Technical Issue

Through reviewing the material, several typos have been identified. This correction fixes several of those errors.

2.2 Description of Correction to Specification

In section 3.1.1, modify the Number of Ports entry in the HBA Capabilities register as follows:

04:00	RO	Impl. Spec.	**Number of Ports (NP):** 0's based value indicating the maximum number of ports supported by the HBA silicon. A maximum of 32 ports can be supported. A value of '0h', indicating one port, is the minimum requirement. Note that the number of ports indicated in this field may be more than the number of ports indicated in the ~~GHC.~~PI register.

Modify the first paragraph of section 5.6.2.4 as follows:

5.6.2.4 ATAPI Packet DMA Read

Software builds a command as described in section 5.5.1. The command shall have a PRD table, is ~~not~~ ATAPI, and is not queued. It is a DMA read (data to memory), therefore CH(pFreeSlot).W (Write) shall be cleared to '0', and CH(pFreeSlot).P (Prefetch) may optionally be set to '1' per the rules described in section 5.5.2.

Modify the second paragraph of section 11.7 as follows:

11.7 Example (Informative)

This section describes an example scenario of commands completing on multiple ports when command completion coalescing is used. This example is meant to illustrate core concepts of command completion coalescing.

Software selects ports 2, 4, and 5 to be in the command completion set by writing CCC_PORTS = ~~00000032h~~ 00000034h. The CCC_CTL.TV value is set by software to 5000 to ensure that a CCC interrupt is generated at least every 5 seconds to ensure that completed commands are not timed out at a higher level by the OS. The CCC_CTL.CC value is set to 5 by software to cause a CCC interrupt after 5 total commands have completed among ports 2, 4, and 5. Software has selected the value of 5 based on the current amount of traffic being seen on ports 2, 4, and 5 and based on quality of service requirements. After setting the CCC_CTL.TV and CCC_CTL.CC values appropriately, software enables CCC by setting CCC_CTL.EN to '1'.