AHCI 1.2

Errata 001

AHCI 1_2 Errata_001.doc

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

Table of Contents

1 PRD Interrupt and Avoiding Data Corruption Concerns

1.1 Description of Technical Issue

Some device implementations may return data to the host with a correct CRC on the FIS, but that has data padding. Essentially, the data is incorrect and should not be acted on by the host.

The PRD Interrupt feature described in section 5.4.2 of the AHCI specification relies on the device to return valid data in all data packets or corrupt the CRC of FISes that have invalid data. Thus, hosts should only use this feature in situations where the host knows that invalid data will not be returned with a correct CRC on the FIS.

1.2 Description of Correction to Specification

Add a new paragraph to the beginning of section 5.4.2 as shown below:

5.4.2 PRD Interrupt

Caution: This feature should only be used in situations where it is known that the device only returns valid data in FISes that have a correct CRC. If the device returns padded data with a correct CRC, there is a potential for data corruption. It is the responsibility of the host driver to ensure that this feature is only used when devices do not exhibit data padding behavior.

When a PRD entry is exhausted, the HBA may be told to generate an interrupt via the 'I' bit in the PRD entry. Note, though, that a PRD is not considered exhausted until all Data FISes that transfer data that is pointed to by that PRD entry is complete.

For example, if the Data FIS is 8 KB, and this is covered by 3 PRD entries, the data is not considered valid at the end of the first or second PRD, since CRC has not yet been checked, even though the data has been copied to memory or the device. Therefore, if the 'I' bit is set in the PRD entry, the HBA must hold onto it internally and not set PxIS.DPS until the Data FIS is complete and CRC is correct. Once correct, PxIS.DPS can be set, and if PxIE.DPE and GHC.IE are set, the HBA shall generate an interrupt.

Conversely, if the PRD entry is 16 KB and two 8 KB Data FISes are used to transfer all of the data pointed to by the PRD entry, then the PRD interrupt associated with that PRD entry shall not be signaled until after the second Data FIS transfer has completed successfully.

The PRD Interrupt is an opportunistic interrupt. The PRD Interrupt should not be used to definitively indicate the end of a transfer. Two PRD interrupts could happen close in time together such that the second interrupt is missed when the first PRD interrupt is being cleared.

2 Data Underflow Clarification

2.1 Description of Technical Issue

The HBA is only required to handle data underflow conditions gracefully in situations where the device is allowed by the ATA/ATAPI specification to create such a condition. For example, underflow is valid for a READ STREAM command but not for READ DMA.

2.2 Description of Correction to Specification

Modify section 6.1.6 as shown:

6.1.6 Command List Underflow

Command list underflow is defined as software building a command table that has more total bytes than the transaction given to the device.

These requirements only apply for commands where underflow is a valid condition in a successful operation (e.g. READ STREAM). For data writes, both PIO and DMA, the device shall detect an error and end the transfer. These errors are most likely going to be fatal errors that will cause the port to be restarted. For data reads, the HBA shall update its PRD byte count with the total number of bytes received from the last FIS, and may be able to continue normally, but is not required to.

The HBA is not required to detect underflow conditions for native command queuing commands.

3 Port Multiplier and Asynchronous Notification

3.1 Description of Technical Issue

In the AHCI 1.1 revision, an asynchronous notification SDB FIS is not posted to system memory. However, in the updated state machine it implies that an asynchronous notification SDB FIS is always posted. This erratum clarifies that the asynchronous notification SDB FIS is only posted if FIS-based switching is enabled or in command-based switching if the PM Port field matches the port currently being accessed.

3.2 Description of Correction to Specification

Modify section 5.3.5.3 as shown:

5.3.5.3 NDR:Accept

NDR:Accept	HBA accepts the FIS with a status of R_OK.		
	1. FIS type is D2H Register FIS or PIO Setup FIS and (pBsy[pPmpCur] = '0' and pDrq[pPmpCur] = '0')	→	P:Idle
	2. FIS Type is D2H Register and PxCMD.ST=0	→	P:RegFisUpdate
	3. PxCMD.FRE=0	→	P:NotRunning[1]
	4. FIS Type is D2H Register	→	RegFIS:Entry
	5. FIS Type is Set Device Bits and 'N' bit is set to '1' and PxFBS.EN = '0' and PM Port field does not match the port currently being accessed.	→	SDB:Notification
	6. FIS Type is Set Device Bits	→	SDB:Entry
	7. FIS Type is DMA Activate	→	DX:Entry
	8. FIS Type is DMA Setup	→	DmaSet:Entry
	9. FIS Type is BIST Activate and far-end loopback is enabled in the FIS	→	BIST:FarEndLoopback[2]
	10. FIS Type is BIST Activate and far-end loopback is not enabled in the FIS	→	BIST:TestOngoing[2]
	11. FIS Type is PIO Setup	→	PIO:Entry
	12. Else (FIS type is unknown)	→	UFIS:Entry
	NOTE: 1. Software shall set PxCMD.FRE to '1' prior to setting PxCMD.ST to '1'. Setting PxCMD.ST to '1' when PxCMD.FRE is cleared to '0' will cause indeterminate results. 2. Handling of the BIST Activate FIS when using FIS-based switching is vendor specific. It is recommended that BIST procedures with a connected Port Multiplier be performed with command-based switching. Note that BIST support is not standardized for Port Multipliers.		

In this state, the non-Data FIS received has been verified to be valid. If the FIS is a Register FIS or PIO Setup FIS and BSY and DRQ are cleared for that device, the HBA accepts the FIS with R_OK but does not perform any updates based on it; see section 6.1.2. In this condition, the device does not own the taskfile, and thus the received FIS is ignored.