AHCI 1.0

Erratum 003

AHCI 1_0 Erratum_003.doc

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

Table of Contents

1 Consistent references to FPDMA Queued Commands

1.1 Description of Technical Flaw

FPDMA Queued commands are not referred to consistently throughout the AHCI spec.

1.2 Description of Correction to Specification

Make the following changes, primarily in section 5.5.4:

All instances of “queued DMA read” shall be replaced with “READ FPDMA QUEUED”. All instances of “queued DMA write” shall be replaced with “WRITE FPDMA QUEUED”. All instances of “a queued DMA command” shall be replaced with “a native queued command”. Change title of section 5.5.4 from “HBA Assisted Queued DMA Transfers” to “Native Queued Command Transfers”.

2 FPDMA Queued example has incorrect PxSACT value

2.1 Description of Technical Flaw

In section 5.5.4.2, there is an example of an FPDMA Queued command sequence. In the step marked "Device sends SDB FIS to clear slots 2 and 5", the updated value for the PxSACT register is incorrect. After commands with tags 2 and 5 are complete, commands with tags 0 and 8 are still pending. Thus the PxSACT value should be 00000101h, rather than 00001001h which means commands with tags 0 and 12 are outstanding.

2.2 Description of Correction to Specification

Make the following changes in section 5.5.4.2:

Device sends SDB FIS to clear slots 2 and 5

At this point, the device sends an SDB FIS to indicate slots 2 and 5 are complete. The HBA shall accept this FIS by traversing the **Exam:AcceptNonData** macro state, and then traverses the SDB:Entry, and, since the received FIS had its I bit set, the SDB:SetIntr → SDB:SetIS → SDB:GenIntr states, and returns to H:Idle. The PxSACT register is now equal to ~~"00001001h"~~ "00000101h".

3 PxCMD.CLO bit should be marked as RW1, not RW

3.1 Description of Technical Flaw

In section 3.3.7, the command list override bit (bit 3) should be marked as RW1, not RW. This bit can only be set to '1' by software, writing a '0' has no effect.

3.2 Description of Correction to Specification

Make the following changes in section 3.3.7:

03	~~RW~~ RW1	0	**Command List Override (CLO):** Setting this bit to '1' causes PxTFD.STS.BSY and PxTFD.STS.DRQ to be cleared to '0'. This allows a software reset to be transmitted to the device regardless of whether the BSY and DRQ bits are still set in the PxTFD.STS register. The HBA sets this bit to '0' when PxTFD.STS.BSY and PxTFD.STS.DRQ have been cleared to '0'. A write to this register with a value of '0' shall have no effect. This bit shall only be set to '1' immediately prior to setting the PxCMD.ST bit to '1' from a previous value of '0'. Setting this bit to '1' at any other time is not supported and will result in indeterminate behavior.

4 Interrupt for PIO Setup FISes

4.1 Description of Technical Flaw

In section 5.2.8, the state machine specifies how to generate an interrupt if the 'I' bit is set in a PIO Setup FIS and appropriate enables are set. The section is missing a state that should set the IS.IPS[x] bit and check that the GHC.IE (global interrupt enable) bit is set prior to generating the HBA interrupt.

4.2 Description of Correction to Specification

Make the following changes in section 5.2.8.4 and add a new section immediately following section 5.2.8.4:

5.2.8.4 PIO:SetIntr

PIO:SetIntr	Set PxIS.PSS to '1'		
	1. PxIE.PSE = '1'	→	~~PIO:GenIntr~~ PIO:SetIS
	2. Else	→	AggrPM:Entry

This state is entered when the PIO Setup FIS has the interrupt 'I' bit set.

5.2.8.5 PIO:SetIS

DR:PrdSetIS	HBA sets IS.IPS(x) to '1'.		
	1. GHC.IE = '1'	→	PIO:GenIntr
	2. Else	→	AggrPM:Entry

This state is entered to set the interrupt for this port on the controller.

5 Data Transmit state has multiple names

5.1 Description of Technical Flaw

In section 5.2.9.1 and 5.2.9.2, the DX:Transmit state is referred to inconsistently as “DX:Start” and “DX:XmitStart”. All references to DX:Transmit need to be consistent.

5.2 Description of Correction to Specification

Make the following changes in sections 5.2.9.1 and 5.2.9.2:

5.2.9.1 DX:Entry

DX:Entry	The HBA constructs a Data FIS for command list entry hbaDataTag. The PMP field of the Data FIS is set to the value in PxCLB[CH(hbaDataTag)][PMP]. The HBA fetches PRDs and data from locations specified in the hbaDataTag command list entry.

1.	No data to transmit and hbaPioXfer = ‘1’	→	PIO:Update
2.	No data to transmit and hbaPioXfer = ‘0’	→	H:Idle
3.	Else	→	~~DX:Start~~ DX:Transmit

This state starts to construct a Data FIS to transmit to the device. This state is entered after a PIO Setup, DMA Activate, or DMA Setup FIS is received.

5.2.9.2 DX:Transmit

~~**DX:XmitStart**~~ **DX:Transmit**	HBA transmits the Data FIS to the device. Continue to fetch PRDs and data as necessary to complete the Data FIS. As PRDs are completed and R_OK is received for the FIS containing the data in that PRD, log the ‘I’ bit in the PRD into hbaPrdIntr.

1.	SYNC escape received for Data FIS	→	ERR:SyncEscapeRecv
2.	Transmission failed	→	ERR:Fatal
3.	End of PRD table reached, or maximum Data FIS length of 8KB has been transferred.	→	DX:UpdateByteCount

This state transmits a Data FIS to the device. The HBA will fetch PRDs and data as necessary to complete the Data FIS.

6 Completion processing for NCQ commands

6.1 Description of Technical Flaw

In section 5.4.3, the completion processing for a non-queued command sequence is described. The completion sequence for native queued commands needs to be added.

6.2 Description of Correction to Specification

Make the following changes in section 5.4.3:

5.4.3 Processing Completed Commands

Software processes the interrupt generated by the device for command completion. In the interrupt service routine, software checks IS.IPS to determine which ports have an interrupt pending.

For each port that has an interrupt pending:

1. Software determines the cause of the interrupt by reading the PxIS register. It is possible for multiple bits to be set
2. Software clears appropriate bits in the PxIS register corresponding to the cause of the interrupt.
3. Software clears the interrupt bit in IS.IPS corresponding to the port.
4. If executing non-queued commands, software reads the PxCI register and compares the current value to a previously read value. If executing native queued commands, software reads the PxSACT register and compares the current value to a previously read value. It completes with success any commands whose bit has been cleared in the respective register since the last value was read.
5. If there were errors, noted ~~either~~ in the PxIS register ~~or PxTFD.STS.ERR~~, software performs error recovery actions (see section 6.2.2).