SATA AHCI-IP reference design manual

Rev1.3 1-Mar-16

1. Overview

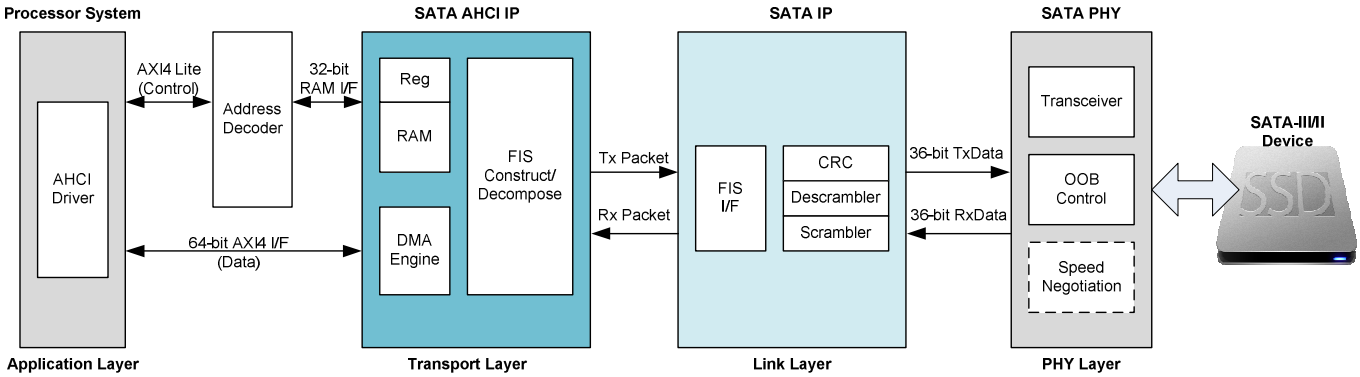

Figure 1-1 SATA AHCI IP Demo System

The operation of SATA protocol can be split into four layers. SATA AHCI IP is designed for transport layer, so it needs to connect with Processor system for application layer and SATA IP for link layer, as shown in Figure 1-1. Transport layer is responsible to construct Frame Information Structures (FISes) for transmission and to decompose received FISes. The sequence of FIS is different in each ATA command, controlled by software on Processor. More details about SATA IP and SATA PHY are described in SATA IP document.

The processor runs Linux OS and accesses SATA AHCI IP through AHCI driver. 64-bit AXI4 bus interface is used for DMA data transfer between DMA engine inside the IP and main memory while register inside the IP is controlled through RAM I/F. Address decoder is used to decode address and convert AXI4-Lite interface to be RAM I/F. More details of the reference design are described in this document.

2. Hardware

Figure 2-1 shows the hardware details in SATA AHCI IP.

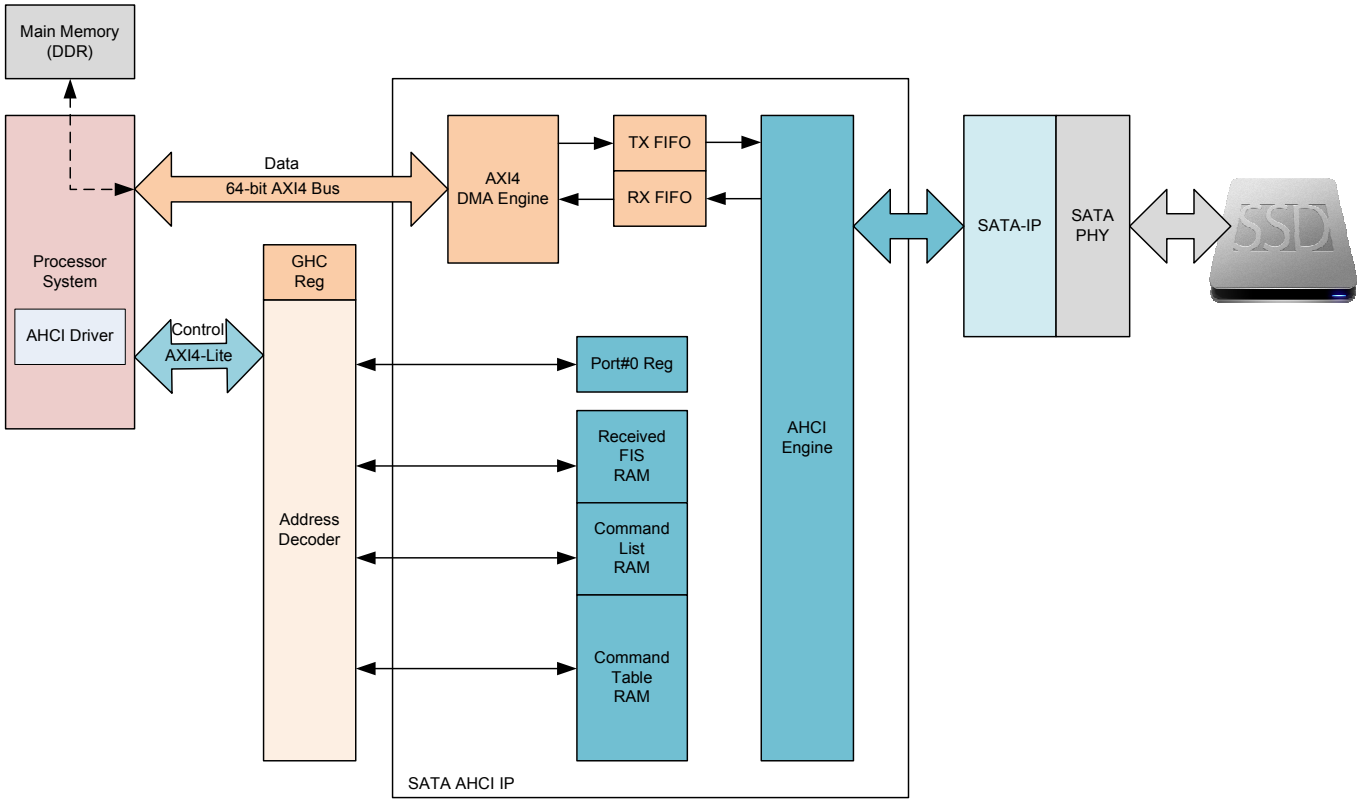

Figure 2-1 SATA AHCI IP Block diagram

To transmit FIS packet to SATA-IP, there are two FIS types for host operation, i.e. Command FIS and Data FIS. Command FIS is prepared in Command Table RAM by Processor. After processor set start signal, AHCI Engine reads 5-Dword data of Command FIS from Command Table RAM within each command slot, and transfers Command FIS to SATA-IP. Based on command queue feature, AHCI Engine can send up to 32 commands to SATA-IP and then wait the response from SATA device to select the active command by DMA Setup FIS.

To transmit Data FIS, AXI4 DMA Engine will dump data from Main memory and store to TX FIFO following the setting value in PRD. PRD is used to define transfer size and main memory address to start burst data transfer. Data within TX FIFO will be burst transferred to SATA-IP by AHCI Engine after SATA device returns DMA Activate FIS. AHCI engine will insert FIS header to data packet before sending to SATA-IP. Based on scatter-gather feature, Data in each command can split into 120 parts, following the total number of PRD slots.

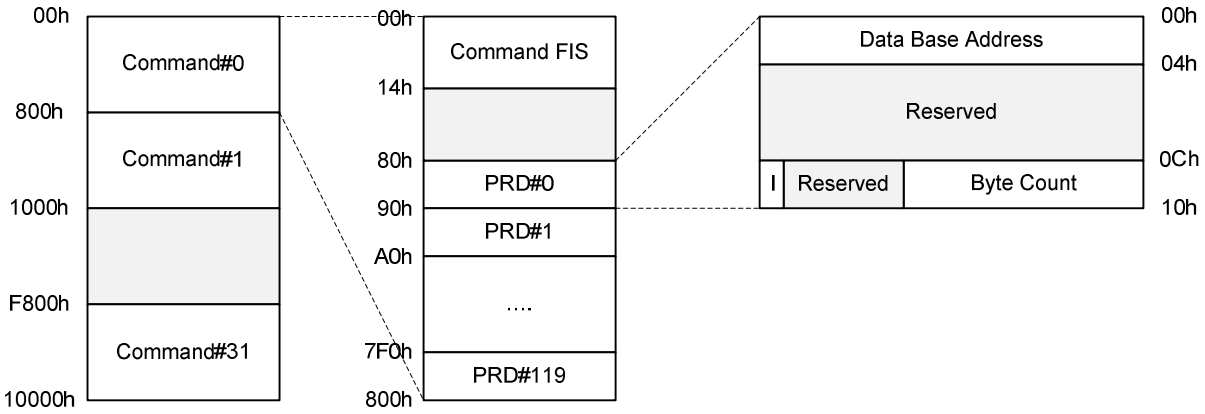

Figure 2-2 Command Table RAM

For receive operation, received FIS types can be grouped into two groups, i.e. non-Data FIS and Data FIS. AHCI Engine checks received FIS type within each SATA packet sent from SATA-IP and store to RX FIFO in case of Data FIS. Before storing the packet to RX FIFO, the FIS header will be removed to pass only the data. Similar to transmit side, AXI4 DMA engine will burst data transfer from RX FIFO to main memory following the setting in PRD. In case of non-Data FIS, received FIS is stored to Received FIS RAM in different area following FIS type number, as shown in Figure 2-3.

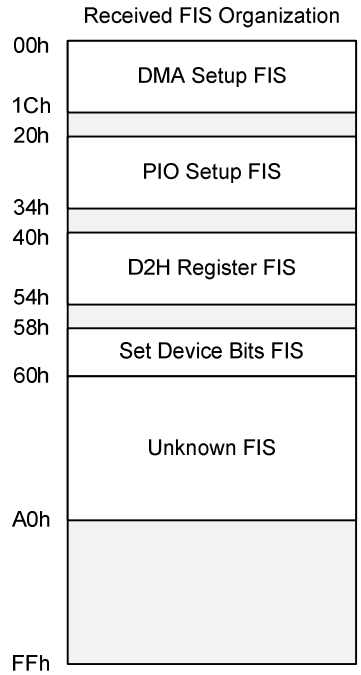

Figure 2-3 Received FIS RAM

AHCI register in the IP is designed following SATA AHCI 1.3.1 specification, which can be downloaded from following website.
http://www.intel.com/content/www/us/en/io/serial-ata/serial-ata-ahci-spec-rev1-3-1.html

In AHCI specification, Received FIS, Command List, and Command Table are stored in Main memory, but SATA AHCI IP uses BRAM to store instead. More details about memory map address are described in SATA AHCI IP datasheet.

3. Software

This topic describes basic operation in the firmware to support five ATA commands, i.e. IDENTIFY DEVICE (ECh), WRITE DMA EXT (35h), READ DMA EXT (25h), WRITE FPDMA QUEUED (61h), and READ FPDMA QUEUED (60h). For SATA device which does not support queued command, WRITE/READ DMA EXT must be used for transferring data between the host and the device. While SATA device which can support queued command, WRITE/READ FPDMA QUEUED is recommended to use for higher performance.

For Write/Read DMA command, the host can send only one command to the device, so one command slot will be used. Each command can support up to 120 PRDs for allocating data in 120 different areas.

For Write/Read Queued command, the host can send up to 32 commands to SATA device without waiting data processing. Tag number inside Command FIS, valid between 0-31, is used to refer to command slot number. The device selects the active slot by returning DMA Setup FIS to the host.

For the host, it can send the new command to the device if some commands slots are available. The command slot will be updated to available status after the device returns Set Device Bits FIS at the end of command operation. At the same time, the device will scan the new command in the queue. If new command is detected and selected, DMA Setup FIS will be sent to the host. By using command queue feature, the device can select the best command for each operation time, so it can expect the better performance than no queue.

More details about GHC and Port#0 register setting value for each command are described as follows.

3.1 INITIALIZATION

After power-on system, Global HBA Control (GHC) and Port#0 register must be initialized. Following is the example sequence to run AHCI application.

1) Reset hardware system by setting GHC.HR='1'
2) Enable to receive FIS by setting PxCMD.FRE='1'
3) Enable to process command list by setting PxCMD.ST='1'
4) Enable interrupt from received FIS, i.e.
    - PxIE.DHRE='1': Enable interrupt from D2H register FIS
    - PxIE.PSE='1': Enable interrupt from PIO Setup FIS
    - PxIE.DSE='1': Enable interrupt from DMA Setup FIS
    - PxIE.SDBE='1': Enable interrupt from Set Device Bits FIS
    - PxIE.UFE='1': Enable interrupt from Unknown FIS
    - PxIE.DPE='1': Enable interrupt from PRD with 'I' bit set operation end

After system end of initialization, SATA device will return D2H register FIS (Signature FIS) to the host.

5) Interrupt will be asserted from PxIS.DHRS flag, and PxIS.TFES will be also asserted from default value of error bit in status register.
6) The host clears interrupt by setting PxIS.DHRS='1' and also setting PxIS.TFES='1' to avoid the interrupt from Task file error.
7) Enable all interrupts by setting PxIE register = 0xFFFF_FFFF.

3.2 IDENTIFY DEVICE

The host sends IDENTIFY DEVICE command to check disk information. The example sequence for IDENTIFY DEVICE command is described as follows.

1) In this example, command slot#0 is used, so first step is setting PRDTL=1 in DW0 of Command Header0 area. Only one PRD is enough to store 512-byte Identify device data returned from SATA device.

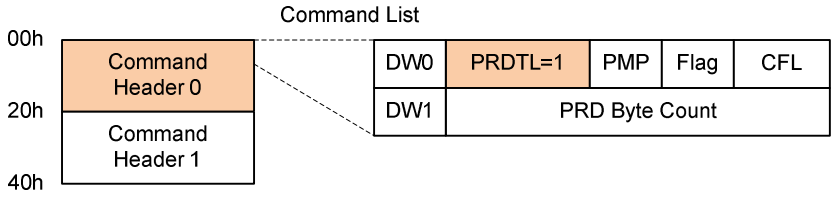

Figure 3-1 Set Command List for Identify Device Command

2) Prepare 5-Dword Command FIS in Slot#0 of Command Table RAM.

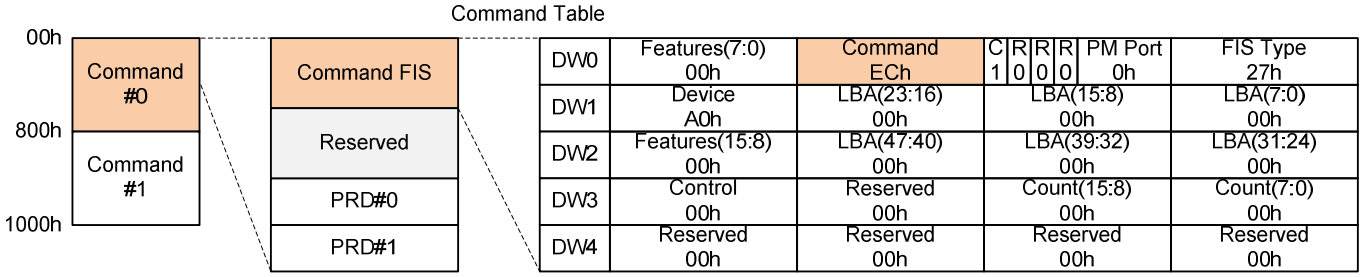

Figure 3-2 Set Command FIS for Identify Device Command

3) Set Data Base Address of available area in Main memory to store 512-byte Identify device data, set 'I' bit='1', and set Byte Count=0x1FF to PRD#0 in Slot#0 of Command Table RAM.

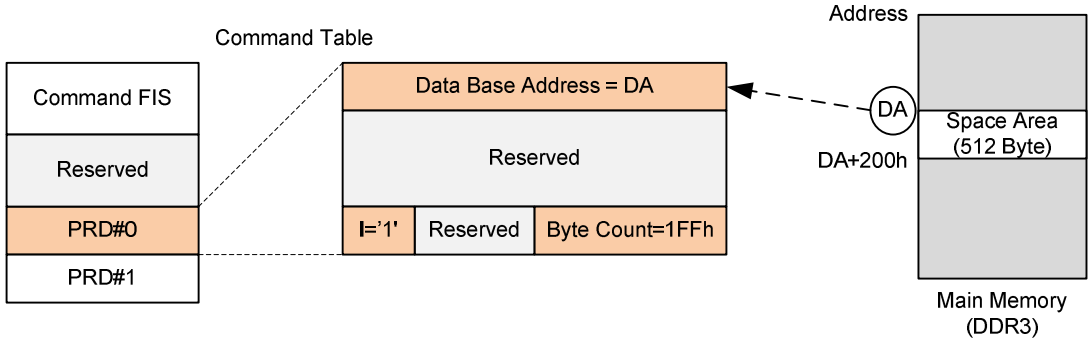

Figure 3-3 Set PRD for Identify Device Command

4) Set PxCI.b0 ='1' to send out Command FIS in slot#0.

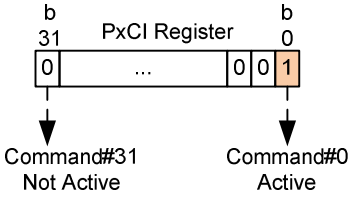

Figure 3-4 Set PxCI Register for Identify Device Command

5) Wait interrupt returned from PIO setup FIS (PxIS.PSS='1') and PxIS.DPS must be equal to '1' from returned Identify device data.

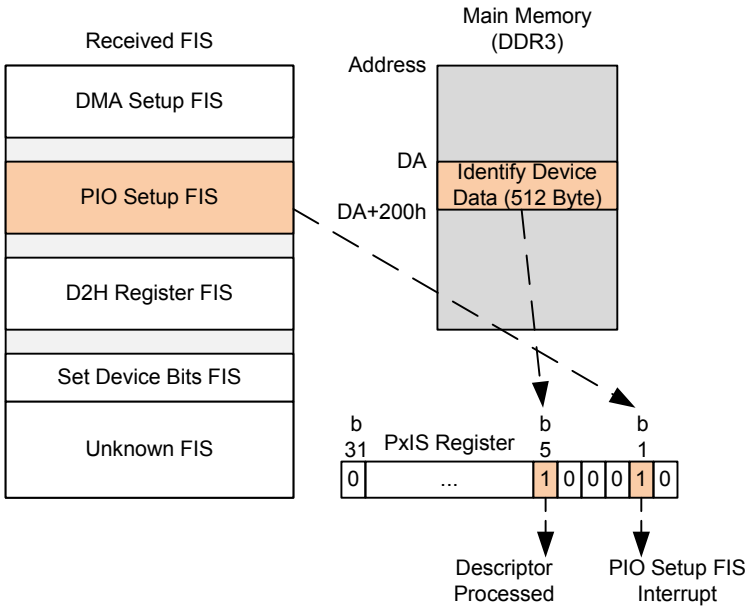

Figure 3-5 Received FIS returned from Identify Device Command

6) Set PxIS.PSS='1' and PxIS.DPS='1' to clear interrupt status.
7) Confirm that PxCI.b0='0' from command process end.
8) Identify device data storing at Data Base Address area within Main memory is ready to read.

3.3 WRITE DMA EXT

The example sequence is follows.

1) Prepare recorded data to space area in Main memory. Assumed that recorded data is spitted into n+1 segments, as shown in Figure 3-6.

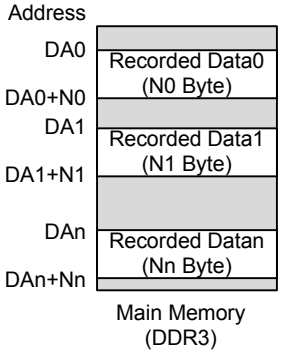

Figure 3-6 Prepare Data for Write DMA Ext command

2) In this example, command slot#0 is used. Set PRDTL = the total numbers of PRD in Command Header0 area.

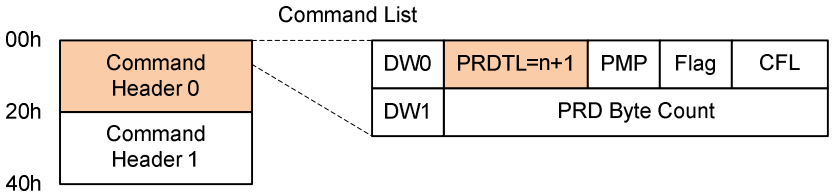

Figure 3-7 Set Command List for Write DMA Ext Command

3) Prepare 5-Dword Command FIS in Slot#0 of Command Table RAM.

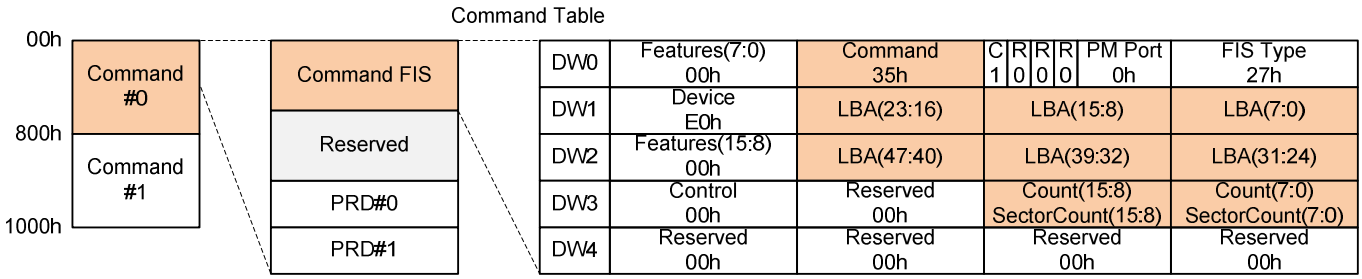

Figure 3-8 Set Command FIS for Write DMA Ext Command

4) Set Data Base Address of each PRD = Start address value of each recorded data segment in Main memory, and set Byte count = data size in each recorded data segment, as shown in Figure 3-9. 'I' bit in the last PRD must be set to '1'.

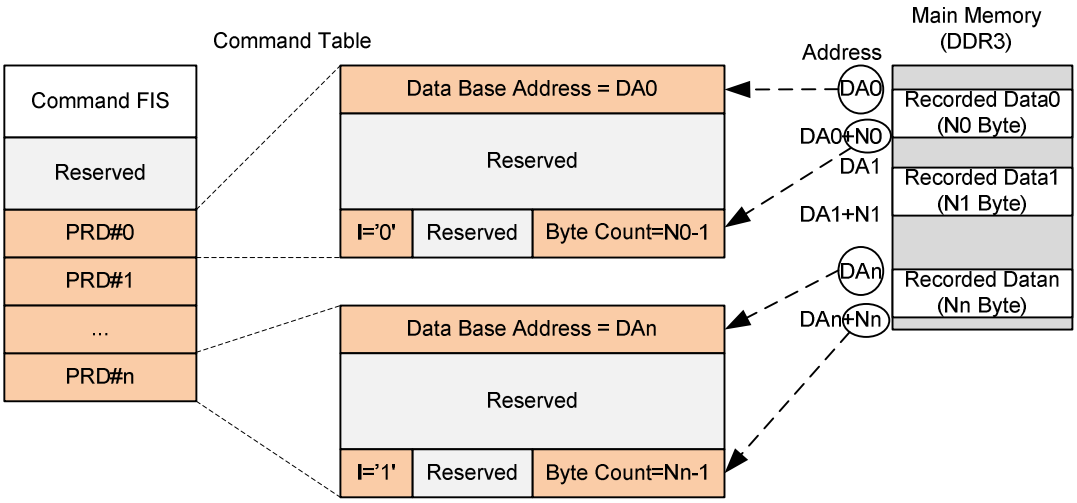

Figure 3-9 Set PRD for Write DMA Ext Command

5) Set PxCI.b0='1' to send out Command FIS in slot#0.
6) Wait interrupt returned from D2H Register FIS (PxIS.DHRS='1') and PxIS.DPS must be equal to '1' from end of transferred data in last PRD.

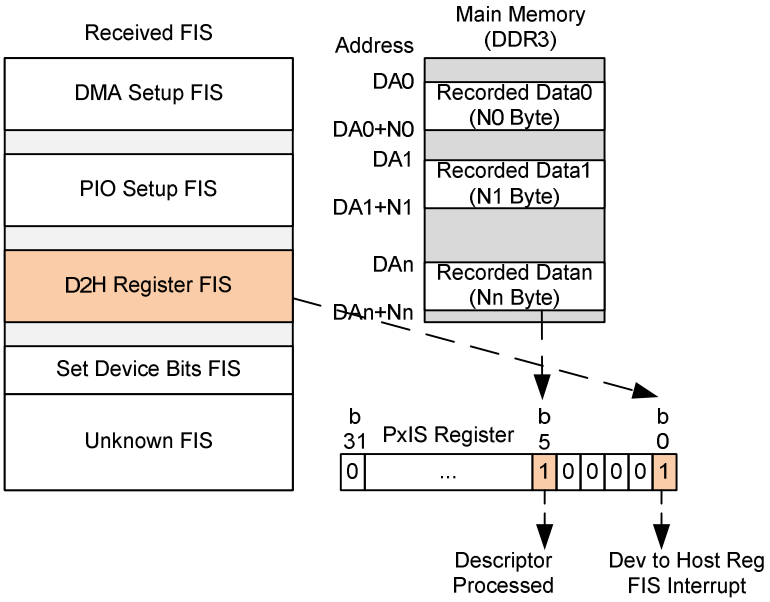

Figure 3-10 Received FIS returned from Write DMA Ext Command

7) Set PxIS.DHRS='1' and PxIS.DPS='1' to clear interrupt status
8) Confirm that PxCI.b0='0' from command process end.

3.4 READ DMA EXT

The example sequence is follows.

1) Prepare space area in Main memory to store read data from SATA device.

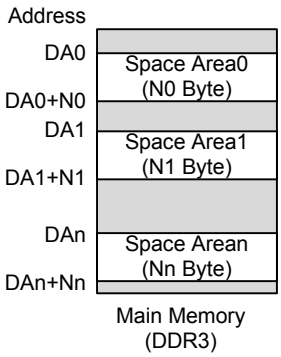

Figure 3-11 Prepare Space Area for Read DMA Ext Command

2) Follow Step 2) – 8) of Write DMA Ext Command to prepare command/data and send command. 5-Dword Command FIS for Read DMA Ext is shown in Figure 3-12.

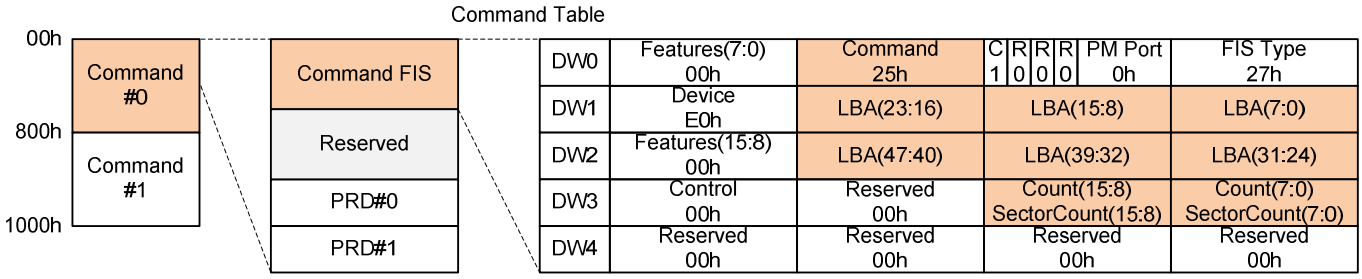

Figure 3-12 Set Command FIS for Read DMA Ext Command

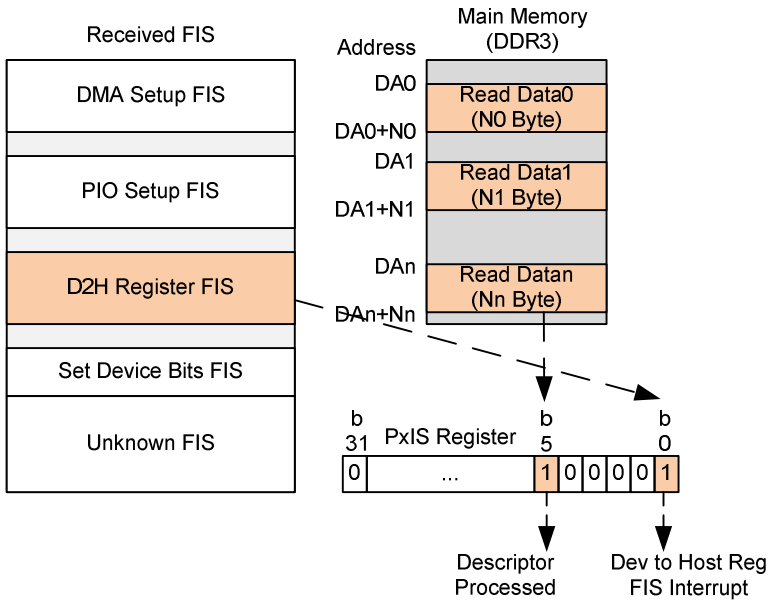

Figure 3-13 Read Data returned from Read DMA Ext Command

3) Read data in Main memory is now ready to read.

3.5 WRITE FPDMA QUEUED

For the basic sequence, this example will describe the step for one slot command firstly.

1) Read PxSACT Register to check slot available. Assumed that slot#0 is available. Then, set PxSACT.b0='1' to reserve slot#0.

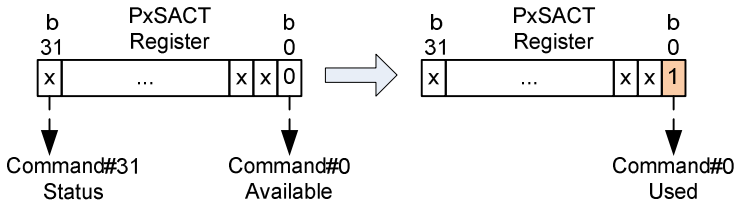

Figure 3-14 Command#0 Slot Available in PxSACT Register

2) Follow Step1) – 5) in Write DMA Ext Command to prepare command/data and send command. 5-Dword Command FIS for Write FPDMA Ext is shown in Figure 3-15.

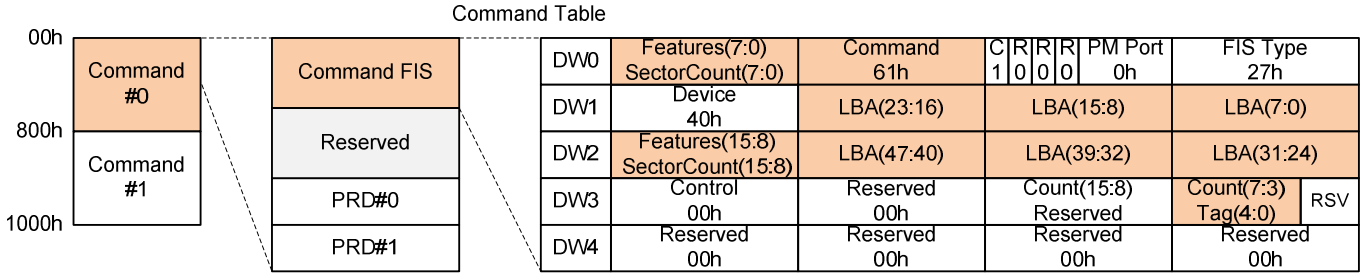

Figure 3-15 Set Command FIS for Write FPDMA Queued Command

3) Interrupt is found from D2H Register FIS (PxIS.DHRS='1') to be command acknowledgement. Set PxIS.DHRS='1' to clear interrupt.
4) Confirm that PxCI.b0='0' from command process end.
5) Interrupt from DMA Setup FIS (PxIS.DSS='1') is found to inform the active command slot to the host. In this example, active slot is slot#0. Then, DMA Engine in the IP starts to transfer data from Main memory to SATA device following PRD value. Set PxIS.DSS='1' to clear interrupt.
6) Interrupt from Descriptor Processed (PxIS.DPS='1') is found after all data in last PRD position has been transferred to SATA device. Set PxIS.DPS='1' to clear interrupt.
7) Interrupt from Set Device Bits FIS (PxIS.SDBS='1') is found to end the write operation. Set PxIS.SDBS ='1' to clear interrupt.
8) Confirm that PxSACT.b0='0' to change slot status back to available.

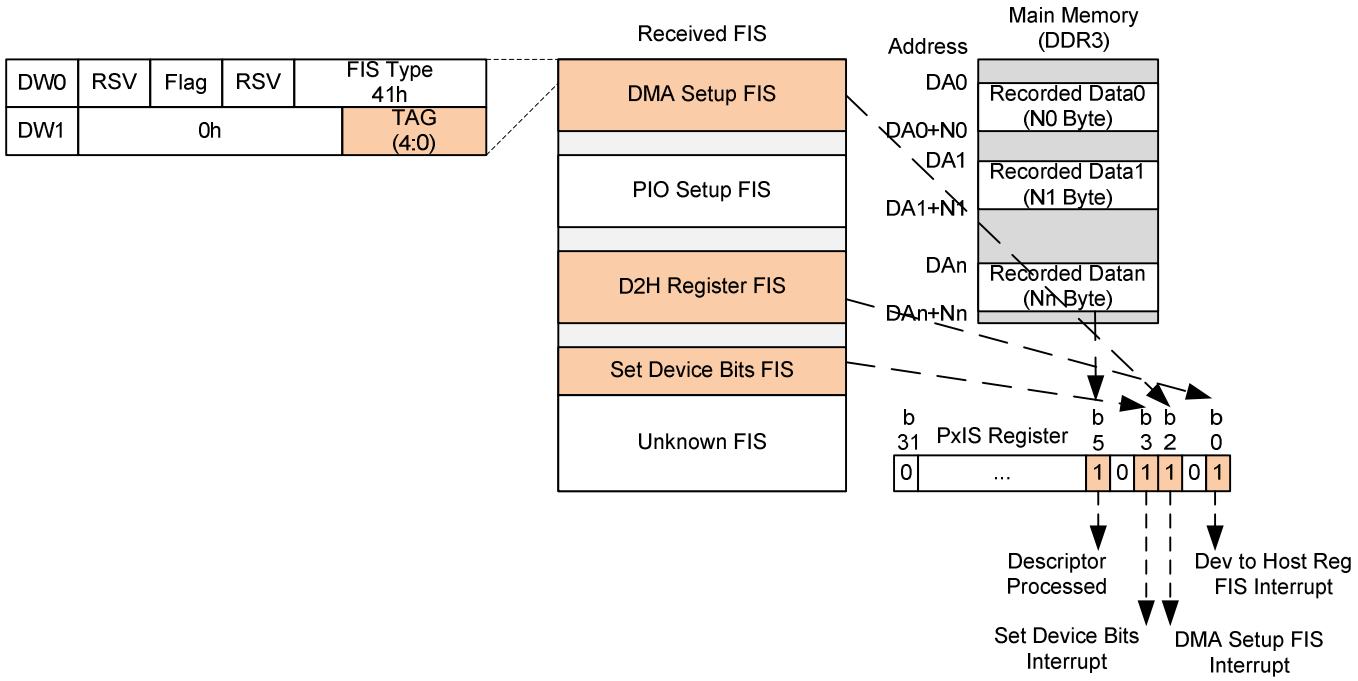

Figure 3-16 Received FIS returned from Write FPDMA Queued Command

To run 32 queued commands, all eight steps can be split into two phases running concurrently, i.e. command phase in Step 1) – 4), and data phase in Step 5) – 8). Command phase scans the available slot from PxCI register and fills new command if available slot is found. New command can be filled until all 32 slots are full. While data phase is the interrupt management to monitor hardware operation that is in normal condition.

3.6 READ FPDMA QUEUED

The sequence of this command is almost similar to Write FPDMA Queued command. The different point is that Main memory area is allocated for read data from SATA device like Read DMA Ext command. 5-Dword Command FIS for Read FPDMA Queued is shown in Figure 3-17.

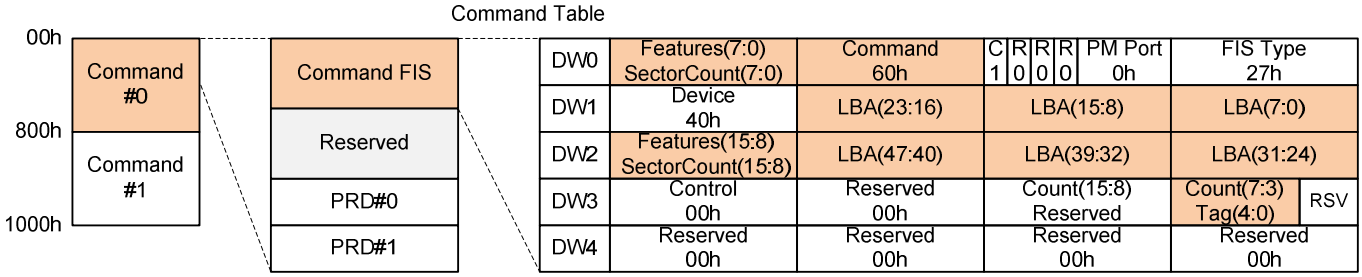

Figure 3-17 Set Command FIS for Read FPDMA Queued Command

4. Revision History

Revision	Date	Description
1.0	7-Nov-14	Initial Release
1.1	17-Jul-15	Update IP block diagram
1.2	8-Jan-16	Add SATA-II device support in Figure 1-1
1.3	1-Mar-16	Update Figure1-1 to support fixed speed mode