SATA AHCI-IP reference design manual

Rev1.3 30-Jun-23

SATA AHCI-IP reference design manual

Rev1.3 30-Jun-23

1. Overview

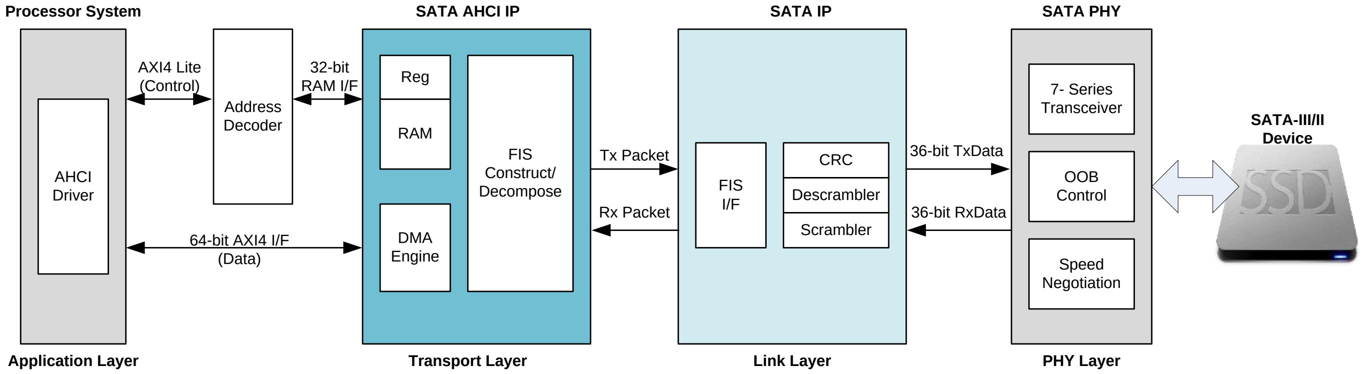


Figure 1-1 SATA AHCI IP Demo System

The operation of SATA protocol can be split into four layers. SATA AHCI IP handles 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 (FIS) for transmission and to decompose received FIS. The sequence of FIS transmission depends on the ATA command value, controlled by the software on Processor. More details about SATA IP and SATA PHY are described in SATA IP document.

The reference design provides full system to evaluate AHCI operation on Zynq platform. The processor runs PetaLinux OS and controls SATA AHCI IP through AHCI driver. 64-bit AXI4 bus interface is used for burst data transferring between DMA engine inside the IP and the 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.

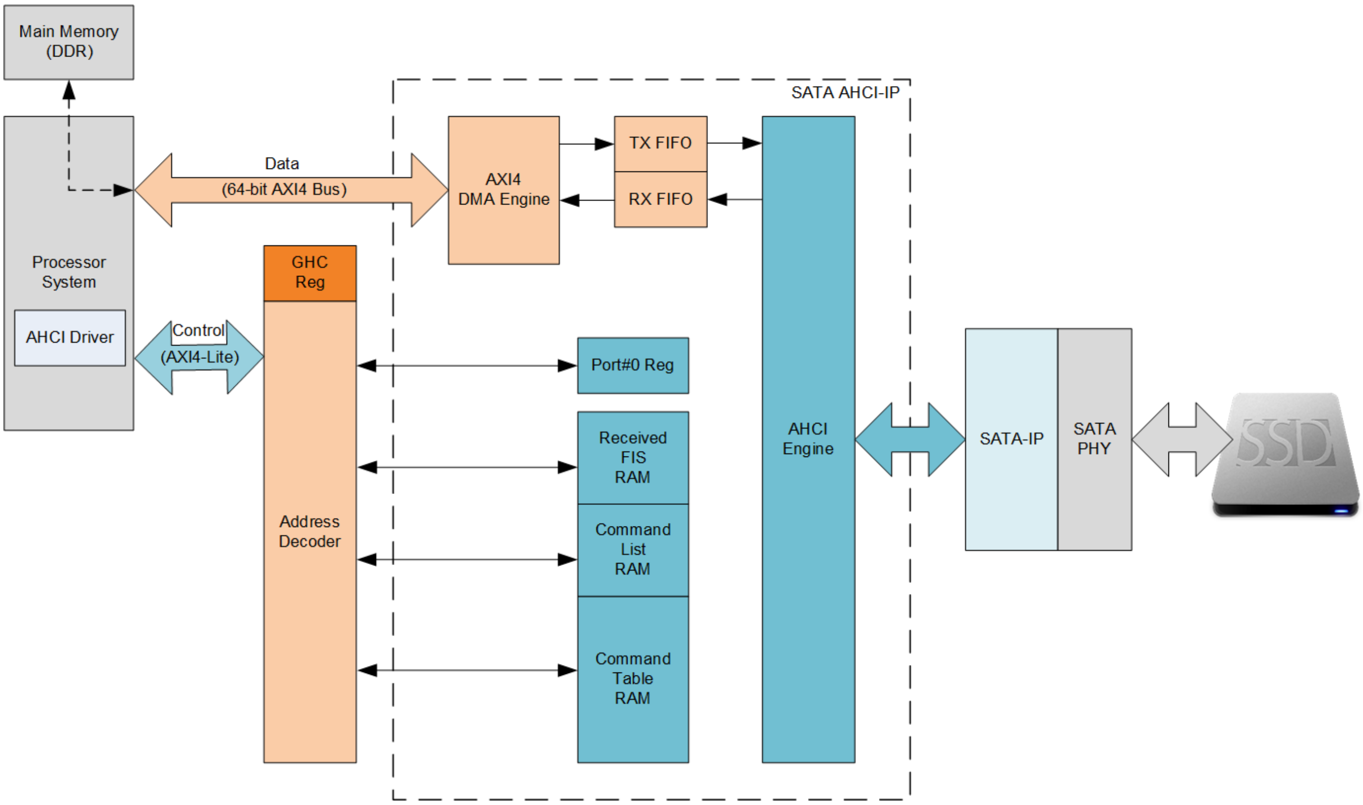


Figure 2-1 SATA AHCI IP Block diagram

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 the Processor sets start signal, AHCI Engine reads 5-Dword data of Command FIS from Command Table RAM within each command slot and then transfers Command FIS to SATA-IP. Based on command queue feature, AHCI Engine can send up to 32 commands to SATA-IP. After that, it waits the response from SATA device to select the active command by DMA Setup FIS.

To transmit Data FIS, AXI4 DMA Engine reads data from Main memory and stores to TX FIFO by using the parameters in PRD. PRD defines the transfer size and the main memory address to start transferring a burst data. Data within TX FIFO is burst transferred to SATA-IP by AHCI Engine after SATA device returns DMA Activate FIS. AHCI engine inserts FIS header to data packet before sending to SATA-IP. Based on scatter-gather feature, Data in each command can split to 120 segments, following the total number of PRD slots.

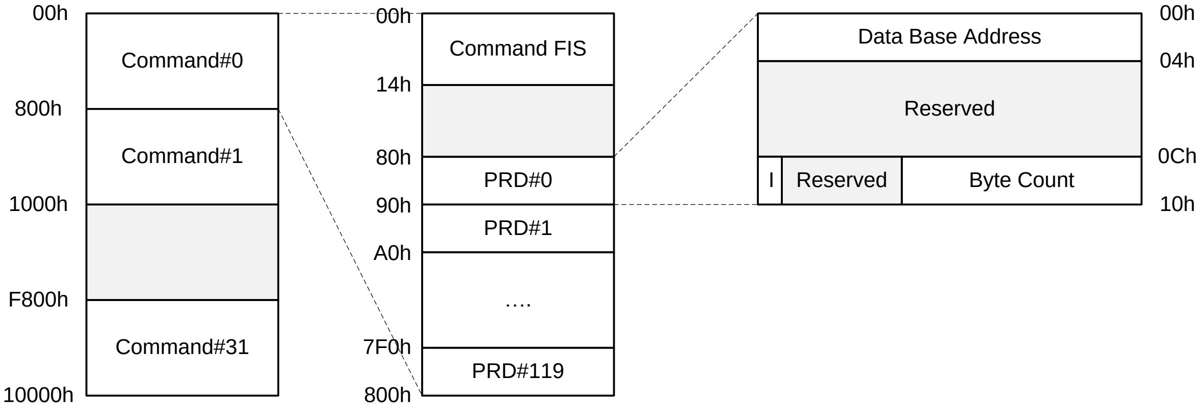


Figure 2-2 Command Table RAM

For receive operation, received FIS types can be grouped to two groups, i.e., non-Data FIS and Data FIS. AHCI Engine reads received FIS type within each SATA packet received from SATA-IP. If it is Data FIS, the FIS header is removed and then store only the data to RX FIFO. Similar to transmit side, AXI4 DMA engine transfers a burst data from RX FIFO to the main memory by using the parameters in PRD. In case of non-Data FIS, received FIS is stored to Received FIS RAM which reserves the specific area for each FIS type, as shown in Figure 2-3.

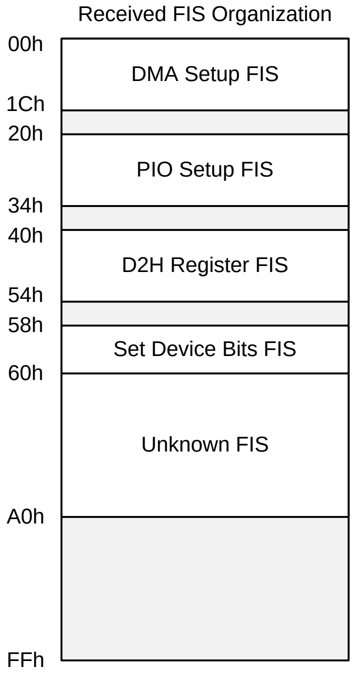


Figure 2-3 Received FIS RAM

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 the main memory. However, SATA AHCI IP moves these data to store to BRAM instead. More details about memory map address are described in SATA AHCI IP datasheet.

3. Software

This topic describes the 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 sends only one command to the device and needs to waits until the command operation is done before sending the next command. Therefore, one command slot is used. Each command supports up to 120 PRDs for splitting the data in the command to 120 segments stored at the different memory areas.

For Write/Read Queued command, the host sends up to 32 commands to SATA device without waiting until the previous command operation is done. 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 free. Each command slot is free after the device returns Set Device Bits FIS at the end of command operation. Also, the device scans the new command in the queue. If new command is detected and the device selects it for operating, DMA Setup FIS is sent to the host. Therefore, the device can re-order the command for operating to get the best performance.

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. The example sequence to run AHCI application is described as follows.

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 returns D2H register FIS (Signature FIS) to the host.

5) Interrupt is asserted from PxIS.DHRS flag. Also, PxIS.TFES is asserted by the default value of error bit in status register.
6) The host clears interrupt by setting PxIS.DHRS=’1’ and 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 the first step is setting PRDTL=1 in DW0 of Command Header0 area. One PRD is enough to store 512-byte Identify device data returned from SATA device.

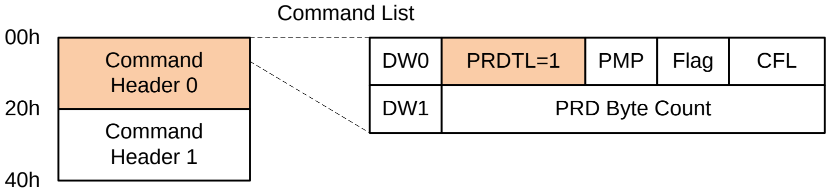
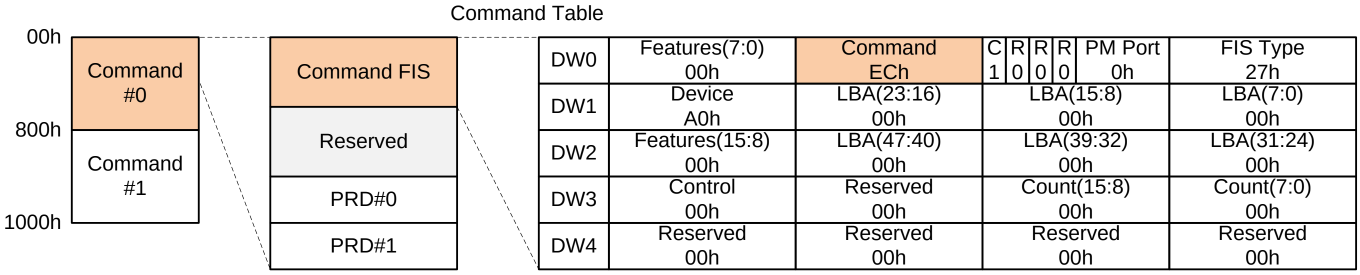


Figure 3-1 Set Command List for Identify Device Command

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



DW0	Features(7:0) 00h	Command ECh	C R R R PM Port 1 0 0 0 0h	FIS Type 27h
DW1	Device A0h	LBA(23:16) 00h	LBA(15:8) 00h	LBA(7:0) 00h
DW2	Features(15:8) 00h	LBA(47:40) 00h	LBA(39:32) 00h	LBA(31:24) 00h
DW3	Control 00h	Reserved 00h	Count(15:8) 00h	Count(7:0) 00h
DW4	Reserved 00h	Reserved 00h	Reserved 00h	Reserved 00h

Figure 3-2 Set Command FIS for Identify Device Command

3) Set Data Base Address of free area in the main memory to store 512-byte Identify device data. After that, set ‘I’ bit=’1’ and Byte Count=0x1FF to PRD#0 in Slot#0 of Command Table RAM.

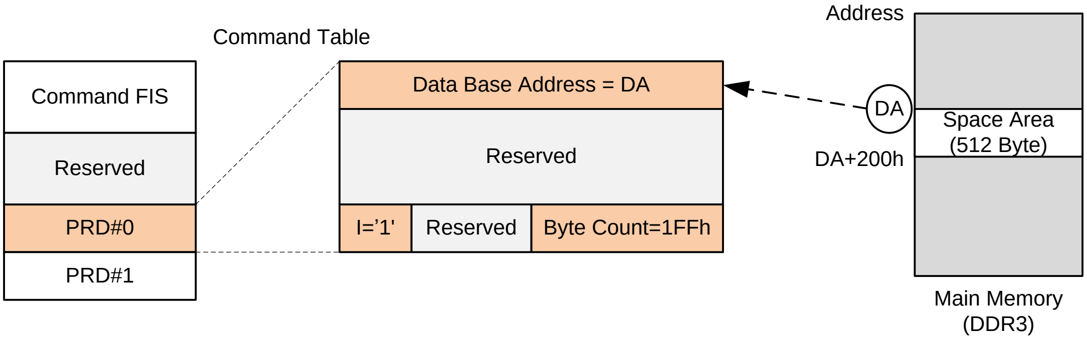


Figure 3-3 Set PRD for Identify Device Command

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

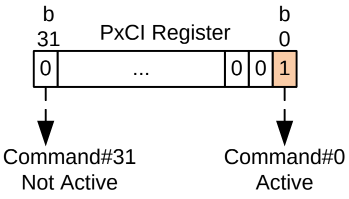


Figure 3-4 Set PxCI Register for Identify Device Command

5) Wait for interrupt asserted from PIO setup FIS (PxIS.PSS='1'). Also, PxIS.DPS must be asserted to '1' from returned Identify device data.

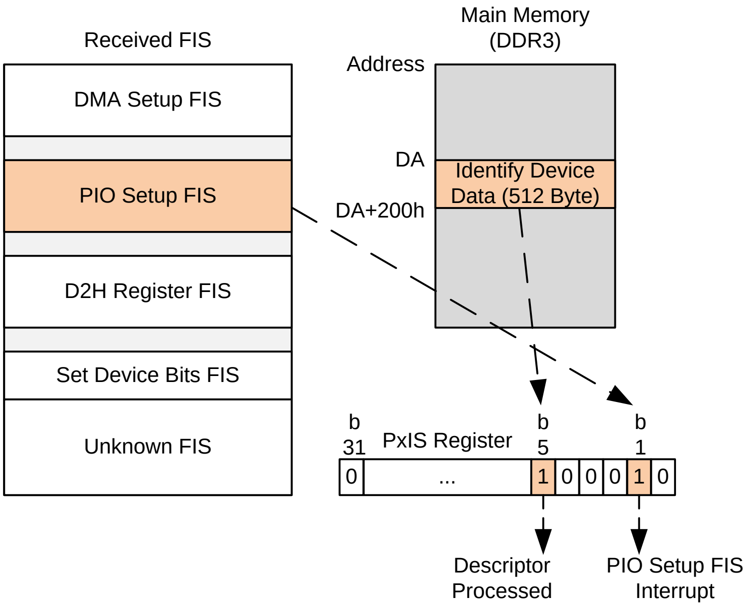


Figure 3-5 Received FIS returned from Identify Device Command

6) Set PxIS.PSS='1' and PxIS.DPS='1' to clear interrupt status.
7) Confirm that PxCI.b0='0' after the command operation is done.
8) Identify device data stored at Data Base Address area of the main memory is ready to read.

3.3 WRITE DMA EXT

The example sequence is described as follows.

1) Prepare the write data to free area in the main memory. Assumed that the write data is spitted to (n+1) segments, as shown in Figure 3-6.

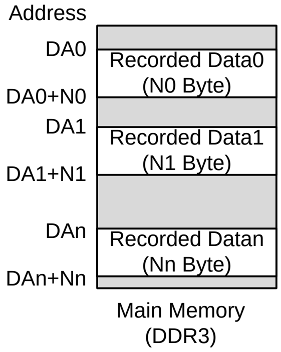
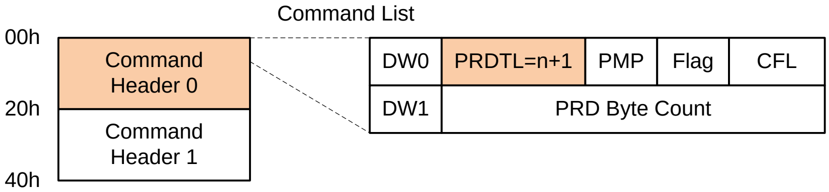


Figure 3-6 Prepare Data for Write DMA Ext command

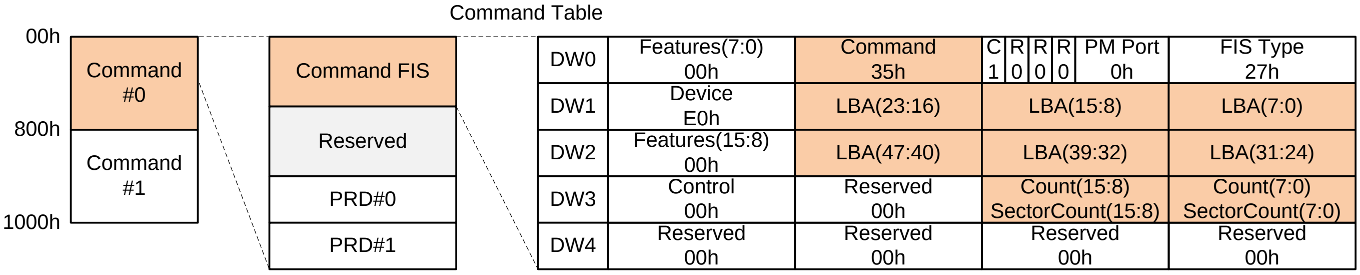
2) In this example, command slot#0 is used. Set PRDTL = the total number of PRDs in Command Header0 area.



DW0	PRDTL=n+1	PMP	Flag	CFL
DW1	PRD Byte Count			

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

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



DW0	Features(7:0) 00h	Command 35h	C 1, R 0, R 0, R 0, PM Port 0h	FIS Type 27h
DW1	Device E0h	LBA(23:16)	LBA(15:8)	LBA(7:0)
DW2	Features(15:8) 00h	LBA(47:40)	LBA(39:32)	LBA(31:24)

DW3	Control
	00h
		Reserved
		00h
			Count(15:8)
			SectorCount(15:8)
				Count(7:0)
				SectorCount(7:0)

DW4	Reserved 00h	Reserved 00h	Reserved 00h	Reserved 00h

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

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

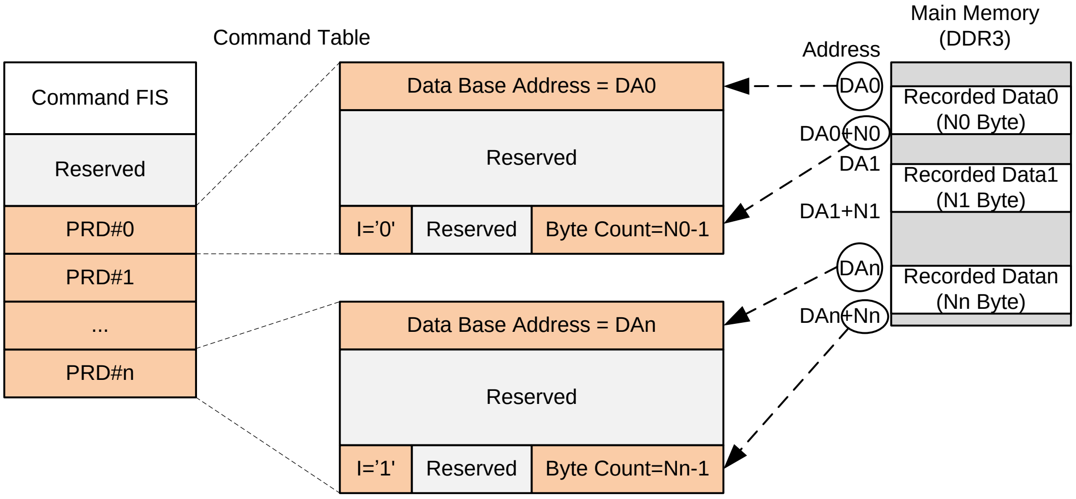


Figure 3-9 Set PRD for Write DMA Ext Command

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

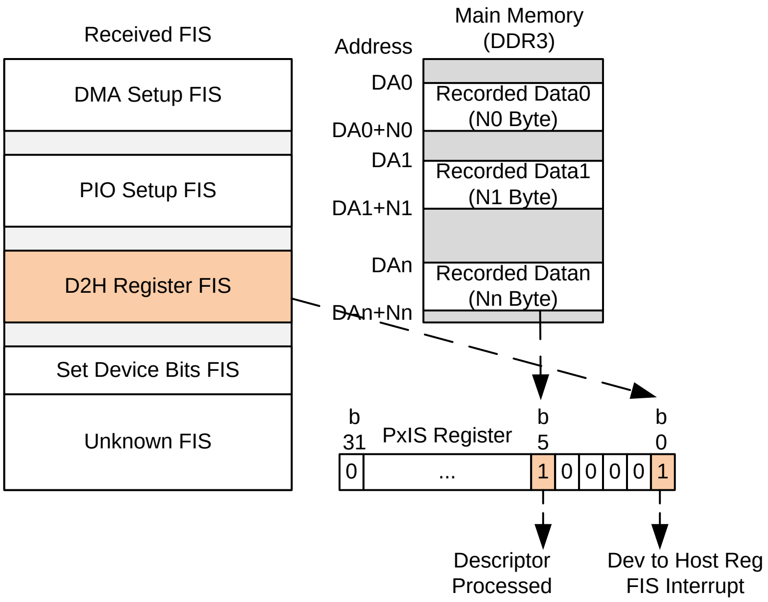


Figure 3-10 Received FIS returned from Write DMA Ext Command

7) Set PxIS.DHRS='1' and PxIS.DPS='1' to clear interrupt status
8) Confirm that PxCI.b0='0' after the command operation is done.

3.4 READ DMA EXT

The example sequence is described as follows.

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

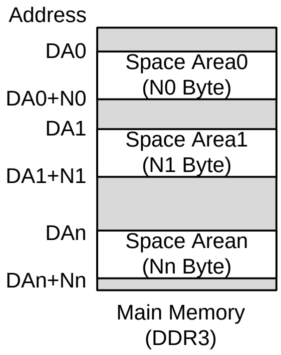
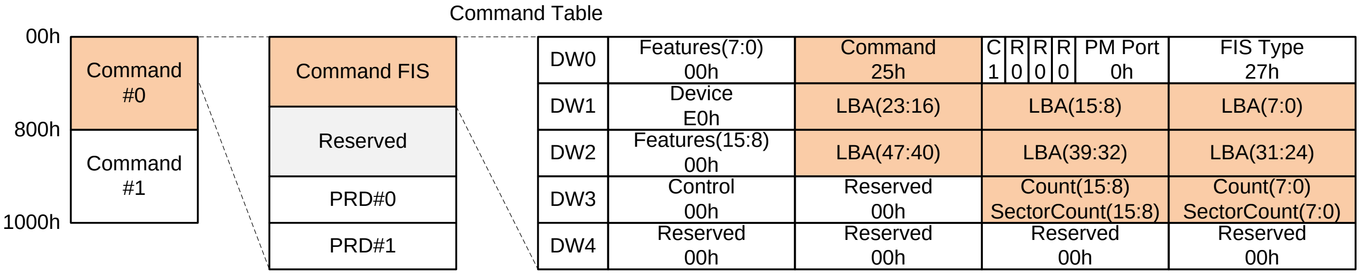


Figure 3-11 Prepare Free space for Read DMA Ext Command

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



DW0	Features(7:0) 00h	Command 25h	C R R R PM Port 1 0 0 0 0h	FIS Type 27h
DW1	Device E0h	LBA(23:16)	LBA(15:8)	LBA(7:0)
DW2	Features(15:8) 00h	LBA(47:40)	LBA(39:32)	LBA(31:24)

DW3	Control
	00h
		Reserved
		00h
			Count(15:8)
			SectorCount(15:8)
				Count(7:0)
				SectorCount(7:0)

DW4	Reserved 00h	Reserved 00h	Reserved 00h	Reserved 00h

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

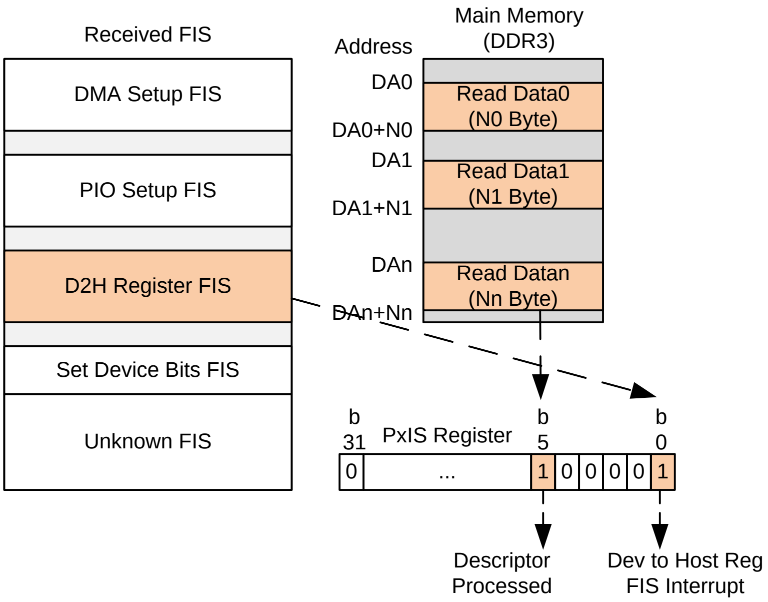


Figure 3-13 Read Data returned from Read DMA Ext Command

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

3.5 WRITE FPDMA QUEUED

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

1) Read PxSACT Register to check the free slot. Assumed that slot#0 is available. Set PxSACT.b0='1' to reserve slot#0.

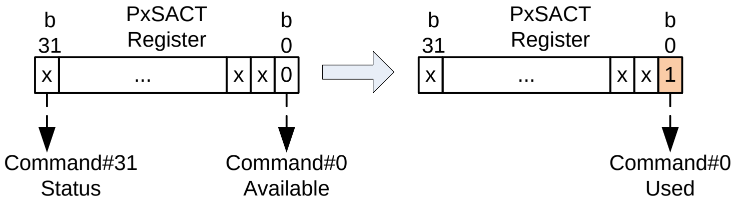
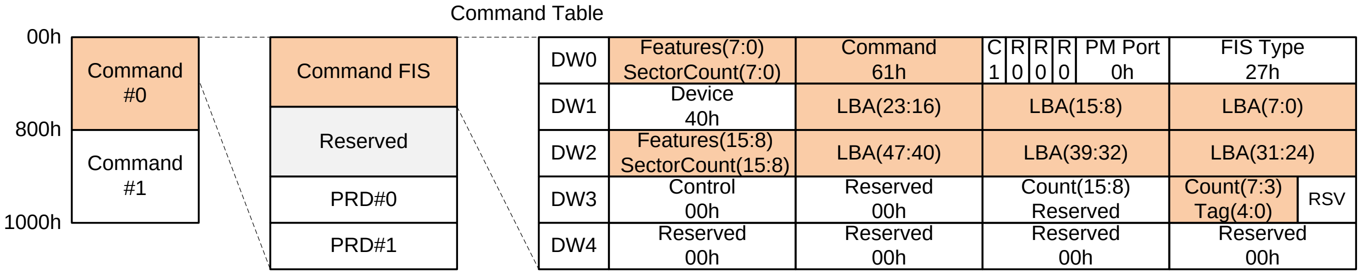


Figure 3-14 Command#0 Slot Available in PxSACT Register

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



DW0	Features(7:0)
	SectorCount(7:0)
		Command
		61h
			C
			1
				R
				0
					R
					0
						R
						0
							PM Port
							0h
								FIS Type
								27h	

DW1	Device 40h	LBA(23:16)	LBA(15:8)					LBA(7:0)	

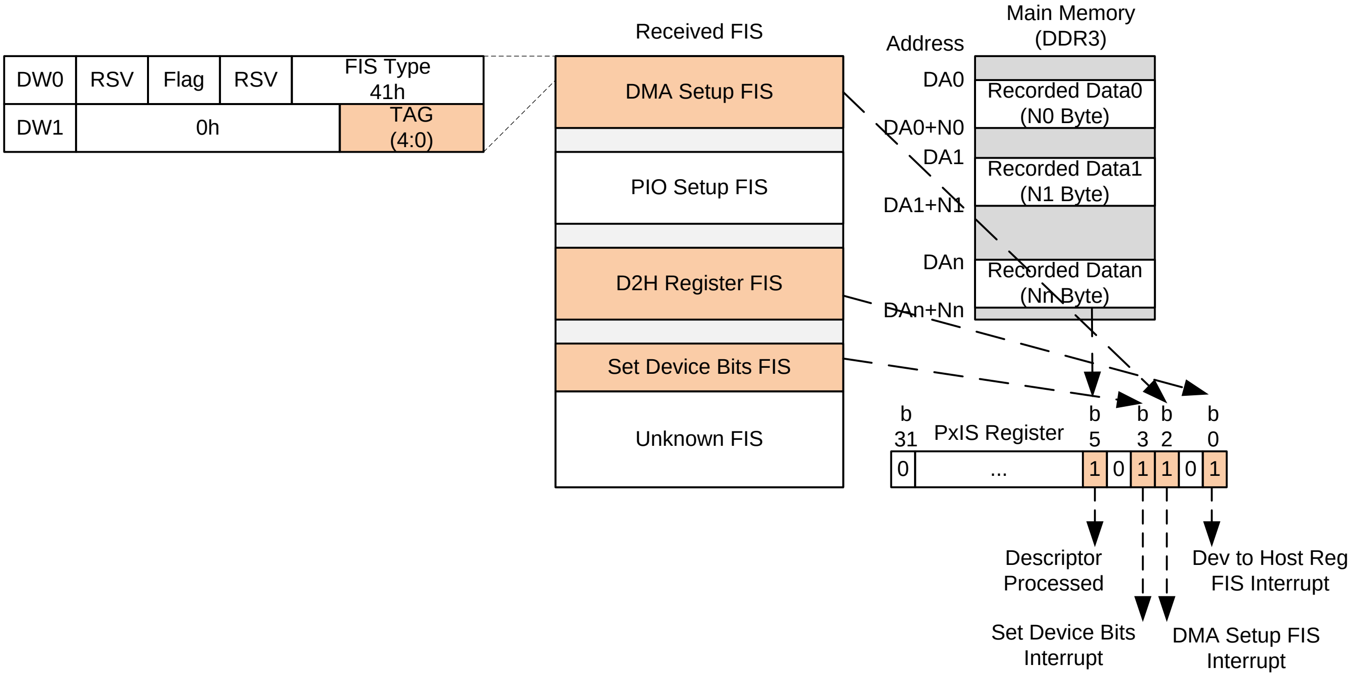
DW2	Features(15:8)
	SectorCount(15:8)
		LBA(47:40)	LBA(39:32)					LBA(31:24)	

DW3	Control
	00h
		Reserved
		00h
			Count(15:8)
			Reserved
								Count(7:3)
								Tag(4:0)
									RSV

DW4	Reserved 00h	Reserved 00h	Reserved 00h					Reserved 00h	

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. After that, set PxIS.DHRS='1' to clear interrupt.
4) Confirm that PxCI.b0='0' after the command operation is done.
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. DMA Engine in the IP starts transferring the data from the main memory to SATA device by using the parameters from PRD. After that, set PxIS.DSS='1' to clear interrupt.
6) Interrupt from Descriptor Processed (PxIS.DPS='1') is asserted after all data in the last PRD position is completely transferred to SATA device. After that, set PxIS.DPS='1' to clear interrupt.
7) Interrupt from Set Device Bits FIS (PxIS.SDBS='1') is asserted at the end of the write operation. After that, set PxIS.SDBS ='1' to clear interrupt.
8) Confirm that PxSACT.b0='0' to free the slot.



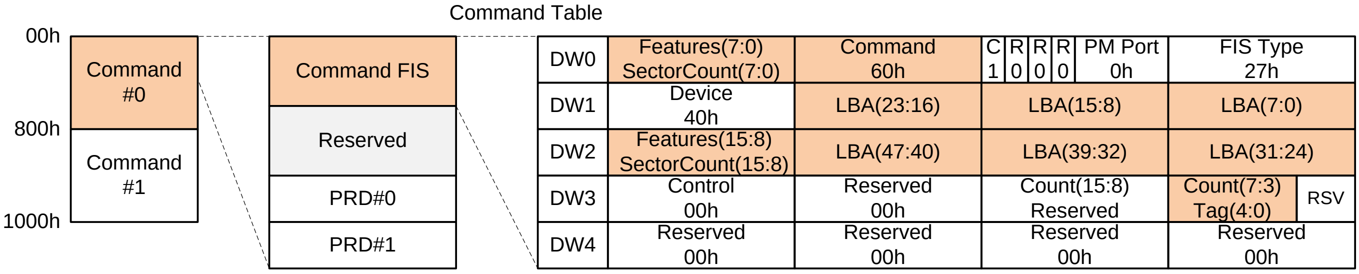
DW0	RSV	Flag	RSV	FIS Type 41h
DW1	0h			TAG (4:0)

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 parallelly, i.e., command phase in Step 1) – 4) and data phase in Step 5) – 8). Command phase scans the free slot from PxCI register and fills the new command if free slot is found. Up to 32 commands can be filled. While data phase is the interrupt management to monitor the error condition while the hardware is operating.

3.6 READ FPDMA QUEUED

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



DW0	Features(7:0)
	SectorCount(7:0)
		Command
		60h
			C
			1
				R
				0
					R
					0
						R
						0
							PM Port
							0h
								FIS Type
								27h

DW1	Device 40h	LBA(23:16)	LBA(15:8)					LBA(7:0)

DW2	Features(15:8)
	SectorCount(15:8)
		LBA(47:40)	LBA(39:32)					LBA(31:24)

DW3	Control 00h	Reserved 00h	Count(15:8) Reserved					Count(7:3) Tag(4:0) / RSV
DW4	Reserved 00h	Reserved 00h	Reserved 00h					Reserved 00h

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

4. Revision History

Revision	Date	Description
1.3	9-Nov-22	Remove board name
1.2	8-Jan-16	Add SATA-II device support in Figure 1-1
1.1	17-Jul-15	Update IP block diagram
1.0	7-Nov-14	Initial Release