

SATA AHCI IP by Baremetal Demo Instruction

Rev1.1 10-Nov-16

This document describes step-by-step to run AHCI-IP demo by using baremetal OS on FPGA board and AB09-FMCRAID board. The host can transfer with SATA-II/III device which supports NCQ command only.

1 Environment Setup

To demo AHCI-IP baremetal OS design, please prepare following hardware.

- 1) Supported FPGA development board: ZC706/Zynq-Mini-ITX(Z100 model)
- 2) PC with Xilinx programmer software (iMPACT/Vivado) and Serial console software
- 3) SATA cable for Zynq Mini-ITX (Z100) or AB09-FMCRAID board for other boards.
- 4) SATA-II/III device
- 5) Xilinx Power adapter for Xilinx board or ATX power supply for Zynq Mini-ITX board
- 6) micro USB cable for programming FPGA between FPGA Development board and PC
- 7) mini/micro USB cable for Serial console connecting between FPGA board and PC



Figure 1-1 SATA AHCI-IP baremetal OS demo environment on ZC706 board





Figure 1-2 SATA AHCI-IP baremetal OS demo environment on Zynq Mini-ITX(Z100) board



2 Demo setup

- 1) Check all system is power off.
- 2) Set up board option
 - a) For ZC706 board only,
 - i. Set SW11="00000" to configure PS from JTAG, as shown in Figure 2-1.
 - ii. Det SW4="01" to connect JTAG with USB-to-JTAG interface, as shown in Figure 2-2.



Figure 2-1 SW11 setting to configure PS from JTAG on ZC706 board



Figure 2-2 SW4 setting to use USB-to-JTAG on ZC706 board

- b) For Zynq Mini-ITX board only,
 - i. Set SW7="00000" to configure PS from JTAG, as shown in Figure 2-3.
 - ii. As shown in Figure 2-4, install a jumper on JP1 pins 1-2 to enable JTAG chain, install the power module onto the board via J8, J9, J10 connectors, and connect ATX power cable to FPGA board via P2 connector.



Figure 2-3 SW7 setting to configure PS from JTAG on Zyng Mini-ITX





3) Connect the SATA-II/III device to the board by the following steps.

a) For Xilinx board,

- i. Connect AB09-FMCRAID board to FMC(1)-HPC connector on Xilinx development board.
- ii. Connect SATA-II/III device to CN0 on FMCRAID board.
- iii. Connect power to power connector on FMCRAID board.

The connections are shown in Figure 2-5.





- b) For Zynq Mini ITX board,
 - i. Connect the device to the SATA connector (J12) on the board using SATA cable.
 - ii. Connect SATA Power cable to the device.

The connections are shown in Figure 2-6.



Figure 2-6 SATA-II/III device connection for Zynq Mini ITX

4) Set DIPSW bit 1-2 to select SATA speed mode.





Figure 2-7 DIPSW to select SATA speed mode

DIPSW[2]	DIPSW[1]	Description		
'1'	'1'	Fixed-speed at SATA3 (6.0 Gbps)		
'1'	'0'	Fixed-speed at SATA2 (3.0 Gbps)		
'0'	'X'	Auto-speed negotiation mode		
Table 2-1 Descripiton of DIPSW for SATA speed				



- 5) Connect micro USB cable from FPGA development board to PC for JTAG programming
- 6) Connect mini/micro USB cable from FPGA board to PC for Serial console.



Mini USB for Serial console Figure 2-8 USB cable connection

- 7) Power on FPGA development board and power supply for SATA device.
- 8) Open Serial console such as TeraTerm, HyperTerminal and set Buad rate=115,200 Data=8 bit Non-Parity Stop=1.
- 9) For Zynq Mini-ITX (Z100)/ZC706 board, open ISE command prompt or Vivado TCL shell, change current directory to baremetalOS folder, and run MiniITX_7z100/ zc706_sata_ahci.bat, as shown in Figure 2-9 and Figure 2-10.



Figure 2-9 Command script for download demo file by ISE tool





10) Check LED status on FPGA development board. The description of LED is follows.

LED	ON	OFF		
LEDL/D4	OK	150 MHz of SATA clock cannot lock. Please check 150 MHz clock source for SATA.		
LEDC/D5	OK	SATA-IP cannot detect SATA device. Please check SATA device and the connection.		
LEDR/D6	SATA-III	SATA-II		
LED0/D7	Always C)FF		
Table 2-2 LED Status of reference design on FPGA board				

11) After programming completely, LEDL/D4 and LEDC/D5 will be ON after complete SATA initialization process. LEDR/D6 shows SATA speed status, as shown in Figure 2-11.



Figure 2-11 LED status after system set up complete on SATA-3/2 speed

12)Main menu will be displayed on Serial console as shown in Figure 2-12. Then, user can execute each command operation. Please check serial cable connection if this menu is not displayed on the console.

COM4 - PuTTY	x
Start SATA AHCI design [Ver = 1.2 Waiting device ready Reset in loop.	1
SATA AHCI menu [Ver = 1.2] [0]or[X] : AHCI RESET [1]or[I] : IDENTIFY DEVICE [2]or[W] : WRITE FPDMA QUEUED [3]or[R] : READ FPDMA QUEUED [4]or[D] : DUMP DATA IN DDR	
	٣



3 Main Menu

3.1 AHCI RESET

Select '0' or 'X' for sending hardware reset signal to AHCI-IP. "AHCI RESET selected" is displayed before reset IP asserted. After complete reset sequence, main menu will be displayed, as shown in Figure 3-1.

+++ AHCI	RESET selected +++	1
Reset in	loop.	
SATA	AHCI menu [Ver = 1.2]	-
[0]or[X]	: AHCI RESET	10
[1]or[]]	: IDENTIFY DEVICE	
[2]or[W]	: WRITE FPDMA QUEUED	=
[3]or[R]	: READ FPDMA QUEUED	
[dlow[D]	: DUMP DATA IN DDR	

Figure 3-1 AHCI Reset Output

3.2 IDENTIFY DEVICE

Select '1' or 'l' for sending "IDENTIFY DEVICE" command to HDD/SSD. Disk information (Model name, disk capacity) will be displayed by using this menu. In the last line, it will show that the device can or cannot support Native Command Queuing feature. This demo can run only with SATA device which can support NCQ.





3.3 WRITE FPDMA QUEUED

Select '2' or 'W' for sending "WRITE FPDMA QUEUED" command to HDD/SSD. Three inputs are required for this menu, i.e.

- Start LBA: this value is the start sector number of HDD/SSD to write data.
- Sector Count: this value is the total transfer size in sector unit (512 byte) for writing HDD/SSD. This size is the data size for CPU to fill to write buffer. If the input is more than 65536 (maximum size for one SATA command), only 65536 sector data is filled and the next command will use same data area with the first command.
- Write Pattern: this value is used for selecting test pattern to write to buffer and then forward to HDD/SSD. There are six test patterns in this demo, i.e. 32-bit increment [0], 32-bit decrement [1], 0000000H [2], FFFFFFFH [3], current data in read buffer [4], and LFSR counter [5].

After software receives all inputs correctly,

- "Prepare data" will be displayed during CPU writing test pattern data to write buffer.
- "Execute Write" will be displayed during CPU sending WRITE FPDMA QUEUED command and transferring data from write buffer to HDD/SSD.
- Transfer speed will be displayed after write operation complete.

Figure 3-3 shows the example of test result when operation complete. Write operation will be cancelled if receiving error input as shown in Figure 3-4.

P COM4 - PuTTY	
+++ WRITE FPDMA QUEUED selected +++ Enter Start LBA : 0 - 500118191 (0x1DCF32AF) => 0 Enter Sector Count : 1 - 500118192 (0x1DCF32B0) => 0x1000000 Write Pattern ? : [0]Inc32 [1]Dec32 [2]A11_0 [3]A11_1 [4]Rdbuf Prepare Data Data ready Execute Write 12345678	(5)1LFSR => 5
SATA AHCI menu [Ver = 1.2] [Ø]or[X] : AHCI RESET [1]or[I] : IDENTIFY DEVICE [2]or[W] : WRITE FPDMA QUEUED [3]or[R] : READ FPDMA QUEUED [4]or[D] : DUMP DATA IN DDR	- 111 - 1

Figure 3-3 WRITE FPDMA QUEUED command input and output

COM4 - PuTTY	
+++ WRITE FPDMA QUEUED selected +++ Enter Start LBA : 0 - 500118191 (0x1DCF32AF) => 0 Enter Sector Count : 1 - 500118192 (0x1DCF32B0) =>	50018193
SATA AHCI menu [Ver = 1.2] [Ø]or[X] : AHCI RESET [1]or[I] : IDENTIFY DEVICE [2]or[V] : WRITE FPDMA QUEUED [3]or[R] : READ FPDMA QUEUED [4]or[D] : DUMP DATA IN DDR	

Figure 3-4 Write Operation cancelled from error input



3.4 READ FPDMA QUEUED

Select '3' or 'R' for sending "READ FPDMA QUEUED" command to HDD/SSD. Two or three inputs are required for this menu, i.e.

- Start LBA: same description with Start LBA in WRITE FPDMA QUEUED menu.
- Sector Count: same description with Sector Count in WRITE FPDMA QUEUED menu. If this input is not more than 65536, the third input will be displayed for selecting verification pattern. If input is more than 65536, the third input will not be displayed to skip data verification process for checking performance only, as shown in Figure 3-5.
- Verify Pattern (Optional): this value is used for selecting verification pattern. This input should be matched with the pattern in WRITE FPDMA QUEUED menu. Six verification patterns can be selected, similar to write pattern. "Verify Data ... Success" is displayed for success case, and "Data Mismatch with failure value" is displayed for failure case, as shown in Figure 3-6.

Similar to WRITE FPDMA QUEUED menu, Read operation will be cancelled if receiving error input, as shown in Figure 3-7.

P COM4 - PuTTY		3
+++ READ FPDMA QUEUED selected +++ Enter Start LBA : 0 - 500118191 (0x1DCF32AF) => 0 Enter Sector Count : 1 - 500118192 (0x1DCF32B0) => 1 12345678 Total = 8[GB] , Time = 15[s] , Transfer speed = 566	0×1000000) [MB/s]	•
SATA AHCI menu [Ver = 1.2] [0]or[X] : AHCI RESET [1]or[I] : IDENTIFY DEVICE [2]or[V] : WRITE FPDMA QUEUED [3]or[R] : READ FPDMA QUEUED [4]or[D] : DUMP DATA IN DDR		

Figure 3-5 READ FPDMA QUEUED command without verify

🔮 COM4 - PuTTY 📼	COM4 - PuTTY	
<pre>+++ READ FPDMA QUEUED selected +++ Enter Start LBA : 0 - 500118191 (0x1DCF32AF) => 0 Enter Sector Count : 1 - 500118192 (0x1DCF32B0) => 65536 Total = 32(MB) , Time = 61(ms] , Transfer speed = 549(MB/s] Verify Pattern ? : (01Inc32 [1]Dec32 [2]All_0 [3]All_1 [4]Wrbuf [5]LFS Verify DataStart check Success SATA AHCI menu [Ver = 1.2] (01or(X] : AHCI RESET (11or(I) : IDENTIFY DEVICE (21or(W] : WRITE FPDMA QUEUED (31or(R] : READ FPDMA QUEUED (31or(R] : READ FPDMA QUEUED (41or(D) : DUMP DATA IN DDR</pre>	<pre>+++ READ FPDMA QUEUED selected +++ Enter Start LBA : 0 - 500118191 (0x1DCF32AF) => 0 Enter Sector Count : 1 - 500118192 (0x1DCF32B0) => 65536 Total = 32[MB], Time = 61[ms], Transfer speed = 549[MB/s] Verify Pattern ? : [0]Inc32 [1]Dec32 [2]All_0 [3]All_1 [4]Vrbuf Verify DataStart check Data Misnatch ADDR[0x20000000]=> T[0x00000000] F[0x00000001] SATA AHCI menu [Ver = 1.2] [0]or[X] : AHCI RESET [1]or[I] : IDENTIFY DEVICE [2]or[V] : WRITE FPDMA QUEUED [3]or[R] : READ FPDMA QUEUED [4]or[D] : DUMP DATA IN DDR</pre>	• [5]LFSR =>8

Figure 3-6 READ FPDMA QUEUED with verify process



COM4 - PuTTY	
*** READ FPDMA QUEUED selected *** Enter Start LBA : 0 - 500118191 (0x1DCF32AF) => 0 Enter Sector Count : 1 - 500118192 (0x1DCF32B0) =>	500118193
SATA AHCI menu [Ver = 1.2] [0]or[X] : AHCI RESET	
[1]OF[I]] : IDENTIFY DEUTCE [2]OF[W] : WRITE FPDMA QUEUED [3]OF[R] : READ FPDMA QUEUED	
[4]or[D] : DUMP DATA IN DDR	

Figure 3-7 Read Operation cancelled from error input



3.5 DUMP DATA IN DDR

Select '4' or 'D' to dump data from buffer to display on Serial Console. In this demo, DDR is mapped to address = 0000_0000h - 3FFF_FFFh. Six submenus can be selected, i.e.

- 'G': this submenu is used to select the address to read, as shown in Figure 3-8. The address can be input to be hex value by adding prefix "0x", so normally input will be received in decimal value.

[G]oto [N]ext	[P]rev [W]r	buf [R]dbuf	[C]learbuf	? g	
0x20000100]	866666666666666666666666666666666666666	00000041	00008842	00000043	
8x288881181	00000044	000000045	000000046	00000847	
8x200001201	00000048	000000849	0000004A	0000004B	
0x200001301	0000004C	0000004D	0000084E	0000004F	
0x200001401	00000050	00000051	00000052	00000053	
0x200001501	00000054	00000055	00000856	00000057	
0x200001601	00000058	00000059	0000005A	0000005B	
8x200001701	0000005C	0000005D	0000005E	0000005F	
0x200001801	00000060	00000061	00000862	00000863	
0x200001901	00000064	00000065	00000066	00000067	
0x200001A01	00000068	00000069	0000066A	0000086B	
0x200001801	0000006C	0000006D	0000006E	0000006F	
0x200001C01	00000070	00000071	00009872	00000873	
0x200001001	00000074	00000075	00000076	00000077	
0x200001E01	00000078	00000079	0000987A	0000007B	1
8x200001F01	0000007C	0000087D	0000007E	0000007F	Ļ
Gloto [N]ext	[P]rev [W]r	buf [R]dbuf	[C]learbuf	?	

- 'N': this submenu is used to read next 256 byte data in buffer, as shown in Figure 3-9.
- 'P': this submenu is used to read previous 256 byte data in buffer, as shown in Figure 3-9.



🛃 COM4 - PuTTY	🐉 COM4 - PUTTY
[8x28988188] 889888848 88888841 88988842 88988843	[0x20008200] 00000088 0000081 0000082 0000083
[8x20000110] 00000044 00000045 00000046 00000047	[0x20000210] 00000084 00000085 00000086 00000087
[0x20000120] 00000048 00000049 0000004A 0000004B	[0x20008220] 00000088 0000089 000008A 000008B
[0x20000130] 000004C 0000004D 000004E 000004F	[0x20008238] 0000008C 000008D 000008E 000008F
[0x20000140] 00000050 00000051 00000052 00000053	[0x20000240] 00000090 00000091 00000092 00000093
[8x20000150] 00000054 00000055 00000056 00000057	[0x20008256] 00000094 00008095 00000096 00000097
[0x20000160] 00000058 00000059 0000005A 0000005B	[0x20000260] 00000098 00000099 0000009A 0000009B
[0x20000170] 0000005C 0000005D 0000005E 0000005F	[8x28888278] 8888899C 8888899D 8888899E 8888899F
[0x20000180] 0000060 00000061 0000062 0000063	[0x20000280] 000000A0 000000A1 000000A2 000000A3
[9x20000190] 00000064 00000065 00000066 00000067	[0x20008298] 000000A4 000000A5 000000A6 000000A7
[8x200001A0] 00000068 00000069 0000006A 0000006B	[0x200002A0] 000000A8 000000A9 000000AA 000000AB
[9x200001B0] 000006C 0000006D 000006E 000006F	[0x200002B0] 000000AC 000000AD 000000AE 000000AF
[8x200001C0] 0000070 00000071 00000072 00000073	L0x200002C01 000000B0 000000B1 000000B2 000000B3
[0x200001D0] 00000074 00000075 00000076 00000077	[0x200002D0] 000000B4 000000B5 000000B6 000000B7
Lex200001E0 J 00000078 00000079 0000007A 0000007B	LUX200002E01 000000B8 000000B9 000000BA 000000BB
LOX200001F01 0000007C 0000007D 0000007E 0000007F	TAXTAGASTAT AGAGAGEC GAGAGABD AGAGAARE AAAAAARE
Gloto INJext LPirev LWirbut LKidbut LCilearbut (n)	Gooden and Constant Constant (Constant)
	LUX20000160 J 00000040 00000041 00000042 00000043
LEX200002101 00000054 00000005 00000005 00000005 (L0X200001101 00000011 00000015 00000016 00000017
LEX200002201 00000008 00000005 0000000H 00000005	L0X200001201 00000048 00000047 0000004H 0000004B
L0X200002301 00000000 00000000 00000000 000000000	[0,20000130] 00000010 00000010 00000012 00000011 [0,20000130] 00000010 00000010 00000012 00000012
[0,20000210] 00000070 00000071 00000072 00000073	[073000140] 00000020 00000021 00000027 00000023
L0X200002301 00000071 00000073 00000070 00000077	[0*38884181 88888624 8888860 88888629 88888629 88888629
19,200002001 00000076 00000077 00000078 00000075	[8^38884.461] 88888826, 88888821, 8888882, 8888822, 888822, 888822, 888822, 888822, 888822, 888822, 8888822, 8888822, 8888822, 8888822, 8888822, 8888822, 88822, 888622, 888622, 888822,
[0,20000270] 0000070 0000070 0000071 00000071	[0x200001703 00000000 000000000 000000000000
[9-20000209] 00000004 0000001 00000001 00000001	[8-2000103] 0000000 0000001 00000002 00000003
[9-20000270] 00000011 00000000 0000000 00000000	[Av20000175] 00000001 00000003 0000000 00000001
[Av28888288] BBBBBBBC BBBBBBBD BBBBBBBD BBBBBBBB	[Av20000118] 0000000 0000007 00000001 00000000
[Av200002001] COCOCOLO COCOCOLO COCOCOLO COCOCOLO COCOCOLO COCOCOCO	[Av289881(3)] 88888828 88888821 88888822 88888823
[8x289982081 88888884 88888885 88888886 88888887	[8x288881081 081 08888824 08888875 08888876 08888877
[0x200002E0] 000000B8 000000B9 000000BA 000008BB	[8x289881E9] 88888878 88888879 88888878 8888878
[0x200002F0] 000000BC 000000BD 00000BE 00000BF	[8x299991F9] 0000007C 0000007D 0000007E 0000007F
[Gloto [N]ext [P]rev [W]rbuf [R]dbuf [C]learbuf ?	[G]oto [N]ext [P]rev [W]rbuf [R]dbuf [C]learbuf ?

Figure 3-9 Read Next/Previous 256 byte data in buffer

- 'W': this submenu is used to read 256 byte data at top of write buffer, as shown in Figure 3-10.
- 'R': this submenu is used to read 256 byte data at top of read buffer, as shown in Figure 3-10.

B COM4 - PuTTY	0	COM4 - PuTTY	
[G]oto [N]ext [P]rev [W]rbuf [R]dbuf [0x2800000] 0000000 0000000 0000000 [0x28000010] 00000004 00000005 [0x28000020] 00000004 00000005 [0x28000020] 00000000 00000001 [0x28000040] 00000000 00000001 [0x28000040] 00000010 00000011 [0x28000060] 00000014 00000015 [0x28000060] 00000010 00000010 [0x28000060] 00000010 00000011 [0x28000060] 00000010 00000010 [0x28000080] 00000024 00000025 [0x28000080] 00000024 00000021 [0x280000800] 000000024 0	[C]learbuf ? . 00000002 0000003 00000006 00000007 00000006 00000008 00000002 00000008 00000012 00000013 00000012 00000017 00000016 00000018 00000012 00000018 00000012 00000017 00000022 00000023 00000026 00000027 00000026 00000027 00000028 00000027 000000202 00000027 00000020 00000027 00000020 00000027 00000020 00000033 00000030 00000033 00000030 00000033 00000030 00000033 00000030 00000035 00000031 00000035	[G]oto [N]ext [P]rev [W]rbuf [R]dbuf [Gx2200000] 0000000 0000000 00000001 [0x22000020] 00000004 00000005 [Gx22000020] 00000004 00000005 [Gx22000020] 00000002 00000005 [Gx22000020] 00000002 00000005 [Gx22000050] 00000018 00000015 [Gx22000050] 00000014 00000015 [Gx22000050] 00000014 00000015 [Gx22000050] 00000014 00000015 [Gx22000050] 00000012 00000011 [Gx22000050] 00000024 00000021 [Gx22000008] 00000022 00000025 [Gx22000008] 00000032 00000021 [Gx22000008] 00000030 00000021 [Gx22000008] 00000022 00000020 [Gx22000008] 00000034 00000035 [Gx22000008] 00000034 00000035 [Gx22000008] 00000038 00000037 [Gx22000008] 00000032 <th>IC31earbuf? P 00000002 00000003 00000006 00000007 00000006 00000007 00000002 00000007 00000002 00000007 00000012 00000017 00000016 00000017 00000016 00000017 00000012 00000017 00000022 00000023 00000022 00000027 00000022 00000027 00000022 00000027 00000022 00000027 00000022 00000027 00000032 00000037 00000034 00000037 00000035 00000037 00000036 00000037 00000037 00000037 00000037 00000037 00000037 00000037</th>	IC31earbuf? P 00000002 00000003 00000006 00000007 00000006 00000007 00000002 00000007 00000002 00000007 00000012 00000017 00000016 00000017 00000016 00000017 00000012 00000017 00000022 00000023 00000022 00000027 00000022 00000027 00000022 00000027 00000022 00000027 00000022 00000027 00000032 00000037 00000034 00000037 00000035 00000037 00000036 00000037 00000037 00000037 00000037 00000037 00000037 00000037

Figure 3-10 Read 256 byte data at top of write/read buffer



- 'C': this submenu is used to clear data in write/read buffer to be zero value. Select 'Y' to confirm for clear write/read buffer, but user can select 'N' to not clear the current buffer.

Putty				(E	
[G]oto [N]ext Clear Write Bu Clear Read But SATA AHCI [Ø]or[X] : AHC [1]or[I] : IDI [2]or[W] : WR [3]or[R] : REA [4]or[D] : DU	IP]rev [W]r Iffer ? [Y/N] nenu [Ver = CI RESET ENTIFY DEVIC ITE FPDMA QUE ID FPDMA QUE IP DATA IN D	buf [R]dbuf] => [y] : Cle => [y] : Cle 1.2T E EUED UED DR	[C]learbuf ear Write Bu ar Read Buffe	fer er	*
[0x20000000] [0x20000020] [0x20000020] [0x20000020] [0x20000040] [0x20000040] [0x20000060] [0x20000080] [0x20000080] [0x20000080] [0x20000080] [0x20000080] [0x20000080] [0x20000080] [0x20000080] [0x20000080] [0x20000080] [0x20000080] [0x20000080] [0x20000080] [0x20000080] [0x20000080] [0x20000080] [0x20000080] [0x20000080]	00000000 00000000 00000000 00000000 0000				

Figure 3-11 Clear buffer to be zero

User can exit this menu by input other key, such as 'x'.

[G]oto	[N]ext	[P]rev	[W]rbuf	[R]dbuf	[C]learbuf	? X	1
SA1	A AHCI	nenu []	ler = 1.3	2]			
[1]or[]	1] : HH] : ID	ENTIFY I	DEVICE				
[2]or[/] : VR	ITE FPD	IA QUEUE	D			
[3]0r[]	8] : RE	AD FPDM	QUEUED				ï

Figure 3-12 Exit dump menu



dg_ahciip_baremetal_instruction_en.doc **4 Revision History**

Revision	Date	Description
1.0	9-Mar-16	Initial version release
1.1	10-Nov-16	Support Zyng Mini-ITX