AHCI PCIe SSD-IP (APS-IP) reference design manual

Rev1.1 14-Jul-16

1. Overview

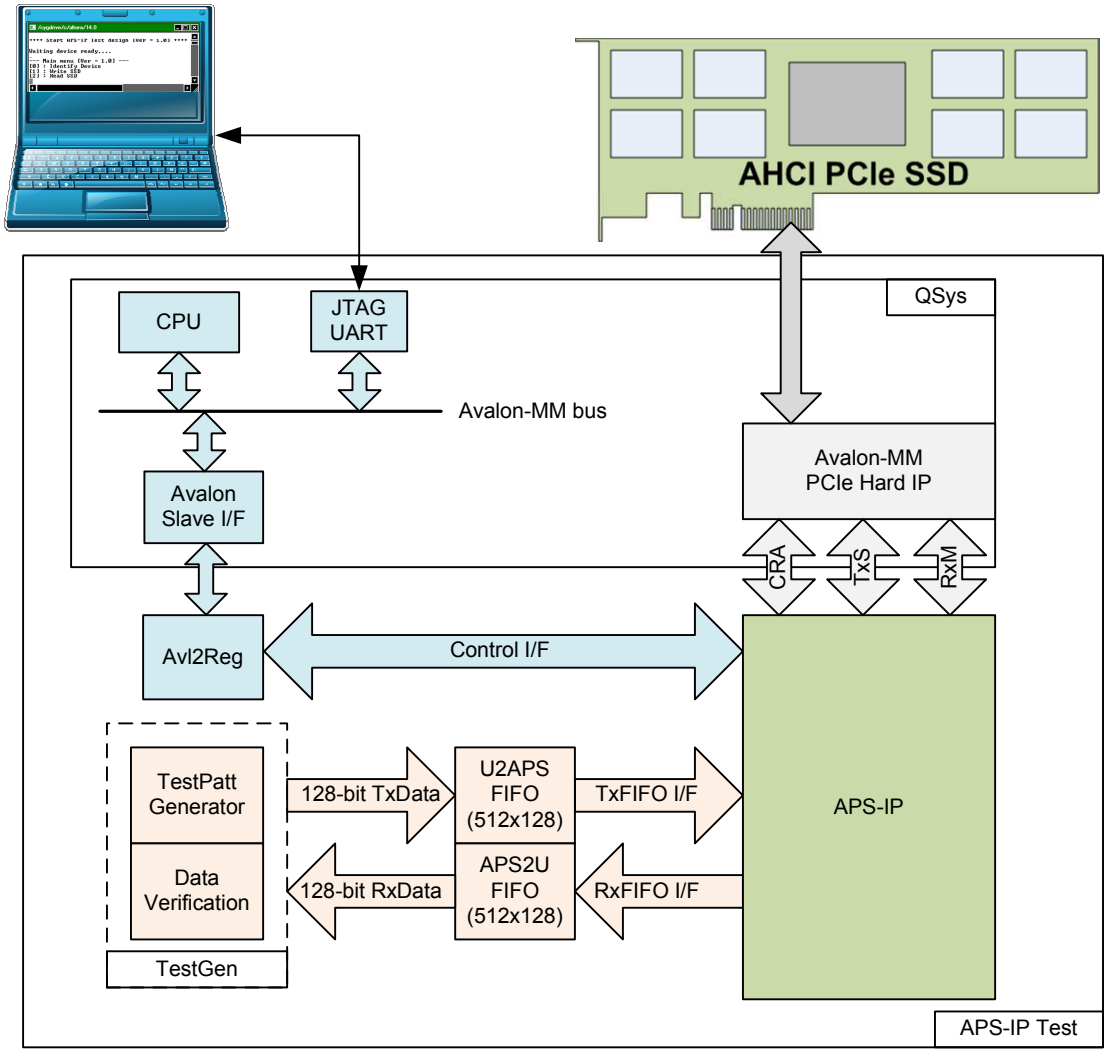

Figure 1 APS-IP Demo System

The reference design integrates APS-IP with simple logic to write and read data with AHCI PCIe SSD at high-speed rate. CPU is additional designed for user interface through NiosII command shell. For simple test, user can input the parameters such as start address, transfer size, and command from keyboards, and the logic will convert all inputs to be signal input for APS-IP. When the operation is completed, CPU will check time usage and then calculate to be write/read performance for that SSD. To interface with CPU bus, Avl2Reg module is used to decode the address and data from the bus to be control/status signal for APS-IP. Data ports of APS-IP are connected to external FIFO. Test data for write test or data verification are generated by TestGen module. All modules run in the same clock domain which is source from Avalon-MM PCIe Hard IP. This clock is equal to 250/125 MHz when interfacing with 4-lane AHCI PCIe Gen3/Gen2 SSD.

User can download APS-IP datasheet and send request to evaluate the IP from our website, http://www.dgway.com/APS-IP_A_E.html.
The real transfer performance in the demo depends on each AHCI PCIe SSD specification.

2. CPU and Peripheral

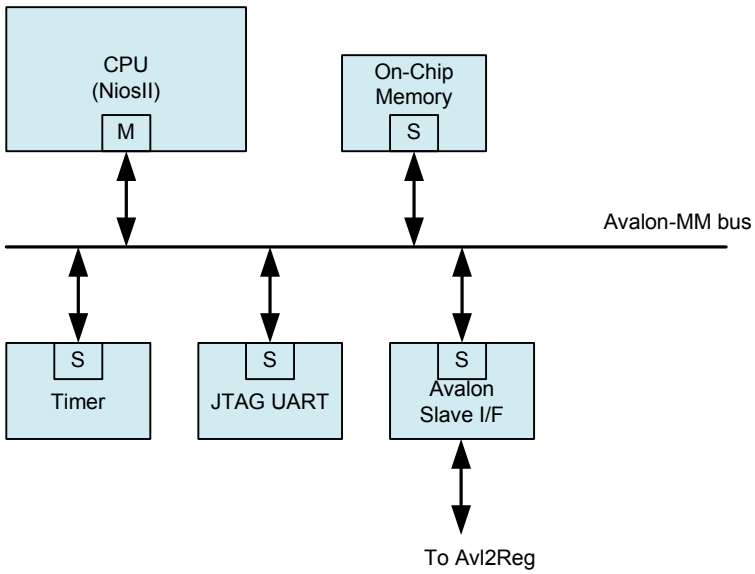

Figure 2 CPU system in reference design

In reference design, CPU peripherals consist of JTAG UART for user interface, Timer for performance measurement, and Memory for CPU firmware. Avalon Slave I/F is connected to Avalon-MM bus for CPU controlling/monitoring APS-IP. More details about memory map for CPU to access Avalon-MM bus are follows.

Table 1 Register Map

Address Rd/Wr	Register Name (Label in the “apsiptest.c”)	Description
BA+0x00 Wr	User Address (Low) Reg (USRADRL_REG)	[31:0]: Input to be start sector address (UserAddr[31:0] for APS-IP)
BA+0x04 Wr	User Address (High) Reg (USRADRH_REG)	[15:0]: Input to be start sector address (UserAddr[47:32] for APS-IP)
BA+0x08 Wr	User Length (Low) Reg (USRLENL_REG)	[31:0]: Input to be transfer length in sector unit (UserLen[31:0] for APS-IP)
BA+0x0C Wr	User Length (High) Reg (USRLENH_REG)	[15:0]: Input to be transfer length in sector unit (UserLen[47:32] for APS-IP)
BA+0x10 Wr	User Command Reg (USRCMD_REG)	[1:0]: Input to be user command (UserCmd for APS-IP) “00”-Identify device, “10”-Write SSD, “11”-Read SSD When this register is written, the design will generate command request to APS-IP to start new command operation.
BA+0x14 Wr	Test Pattern Reg (PATTSEL_REG)	[1:0]: Test pattern select “00”-Increment, “01”-Decrement, “10”-All 0, “11”-All 1

Address Rd/Wr	Register Name (Label in the “apsiptest.c”)	Description
BA+0x100 Rd	User Status Reg (USRSTS_REG)	[0]: APS-IP busy flag (‘0’: Idle, ‘1’: Busy) [1]: Error output from APS-IP (‘0’: Normal, ‘1’: Error) [2]: Data verification fail (‘0’: Normal, ‘1’: Error) [4:3]: PCIe speed from IP (“00”: No linkup, “01”: PCIe Gen1, “10”: PCIe Gen2, “11”: PCIe Gen3)
BA+0x104 Rd	Total disk size (Low) Reg (LBASIZEL_REG)	[31:0]: Total capacity of SSD in sector unit (LBASize[31:0] from APS-IP)
BA+0x108 Rd	Total disk size (High) Reg (LBASIZEH_REG)	[15:0]: Total capacity of SSD in sector unit (LBASize[47:32] from APS-IP)
BA+0x10C Rd	User Error Type Reg (USRERRTYPE_REG)	[31:0]: User error status (UserErrorType[31:0] from APS-IP)
BA+0x114 Rd	Port Interrupt Status Reg (PORTINTSTS_REG)	[31:0]: Port interrupt status (PortIntStatus[31:0] from APS-IP)
BA+0x120 Rd	Data Failure Address (Low) Reg (RDFAILNOL_REG)	[31:0]: Latch value of failure address[31:0] in byte unit from read command
BA+0x124 Rd	Data Failure Address (High) Reg (RDFAILNOH_REG)	[24:0]: Latch value of failure address [56:32] in byte unit from read command
BA+0x130 Rd	Expected value Word0 Reg (EXPPATW0_REG)	[31:0]: Latch value of expected data [31:0] from read command
BA+0x134 Rd	Expected value Word1 Reg (EXPPATW1_REG)	[31:0]: Latch value of expected data [63:32] from read command
BA+0x138 Rd	Expected value Word2 Reg (EXPPATW2_REG)	[31:0]: Latch value of expected data [95:64] from read command
BA+0x13C Rd	Expected value Word3 Reg (EXPPATW3_REG)	[31:0]: Latch value of expected data [127:96] from read command
BA+0x140 Rd	Read value Word0 Reg (RDPATW0_REG)	[31:0]: Latch value of read data [31:0] from read command
BA+0x144 Rd	Read value Word1 Reg (RDPATW1_REG)	[31:0]: Latch value of read data [63:32] from read command
BA+0x148 Rd	Read value Word2 Reg (RDPATW2_REG)	[31:0]: Latch value of read data [95:64] from read command
BA+0x14C Rd	Read value Word3 Reg (RDPATW3_REG)	[31:0]: Latch value of read data [127:96] from read command

CPU firmware sequence in the demo is follows.

- Receive user command from NiosII command shell which is Identify device or write/read command.

For Identify device command,

1) Set USRCMD_REG=”00”. Test logic will generate command and request to APS-IP. Busy flag (USRSTS_REG[0]) will change from ‘0’ to ‘1’.
2) CPU will wait until command complete or any error found by monitoring USRSTS_REG value. Bit[0] will be cleared to ‘0’ when command is completed. Bit[1] will be asserted to ‘1’ when any error is detected. If any error is detected, error message will be displayed.
3) To be test result, SSD capacity will be displayed by reading from LBASIZEL/H_REG.

For write/read command,

1) Receive start address, transfer length, and test pattern value from user through command shell. If any input is invalid, the operation will be cancelled.
2) Get all inputs and set the value to USRADRL/H_REG, USRLENL/H_REG, and USRCMD_REG (USRCMD_REG=”10” for write transfer, and “11” for read transfer).
3) Similar to step 2) in Identify device command. But USRSTS_REG[2] will be also monitored for read command to confirm that read data is correct.
4) During running command, dummy message will be printed in every second to show that the system will be alive. Finally, test performance will be displayed on the command shell when command is completed.

3. Avl2Reg

Input/Output signals of APS-IP and the parameter for TestGen module are mapped to register address of CPU bus within this module. Write data and address from Avalon-MM bus will be decoded and converted to be input parameters in the system. Status signals from APS-IP and TestGen will be mapped to data multiplexer within the module to return to Avalon-MM with valid signal.
For Arria10 board which supports PCIe Gen3 speed, asynchronous logic is included within Avl2Reg module to support different clock domain between CPU and APS-IP system.

4. TestGen

In this module, there are two operations, i.e. generating test data to WrFf port when user selects write command, or verifying received data from RdFf port when user selects read command. The details of logic design inside this module are displayed in Figure 3.

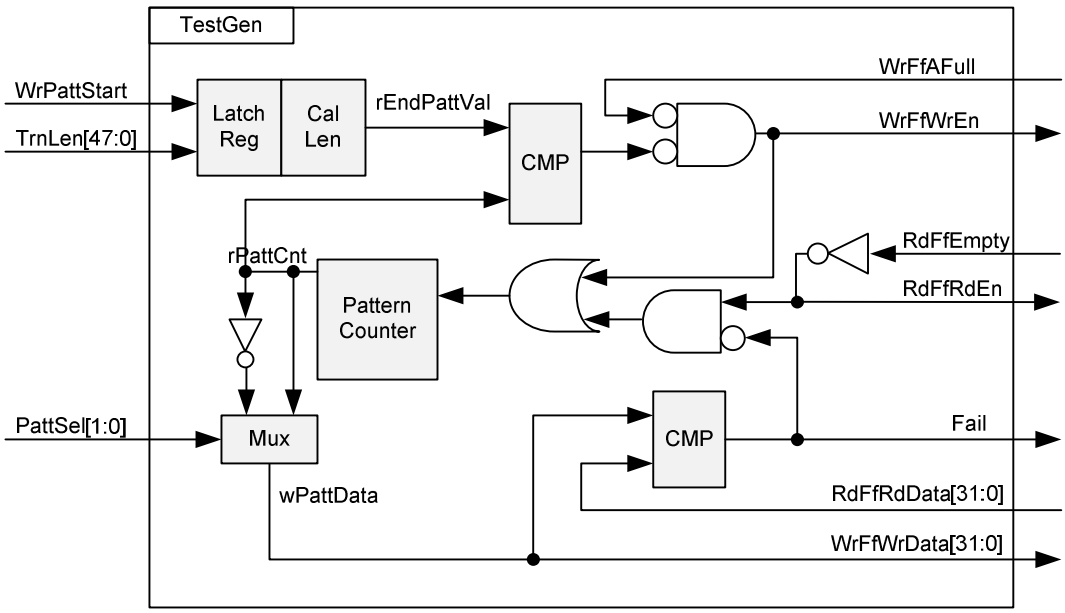

Figure 3 Logic design in TestGen

For write transfer, test pattern will be generated from Pattern counter module after WrPattStart is asserted. WrFfAFull is monitored to confirm that WrFf still have free space to store new test data. Test data pattern will be fed to WrFf when FIFO is available and stopped when total transfer size is equal to set value from user. TrnLen is the input to show total transfer size in sector unit and used to calculate the end value of test data pattern for stopping data generating. Four test patterns can be selected through PattSel input.

For read transfer, read enable of RdFf is asserted when FIFO has available data, monitored by RdFfEmpty signal. Test data generator is used to generate expected data to compare and verify RdFfRdData value. If data is mismatched, Fail flag will be asserted.

5. Revision History

Revision	Date	Description
1.0	29-Jan-16	Initial Release
1.1	14-Jul-16	Support PCIe Gen3 speed