SATA Host-IP reference design manual

Rev1.1 24-Jan-17

1. Overview

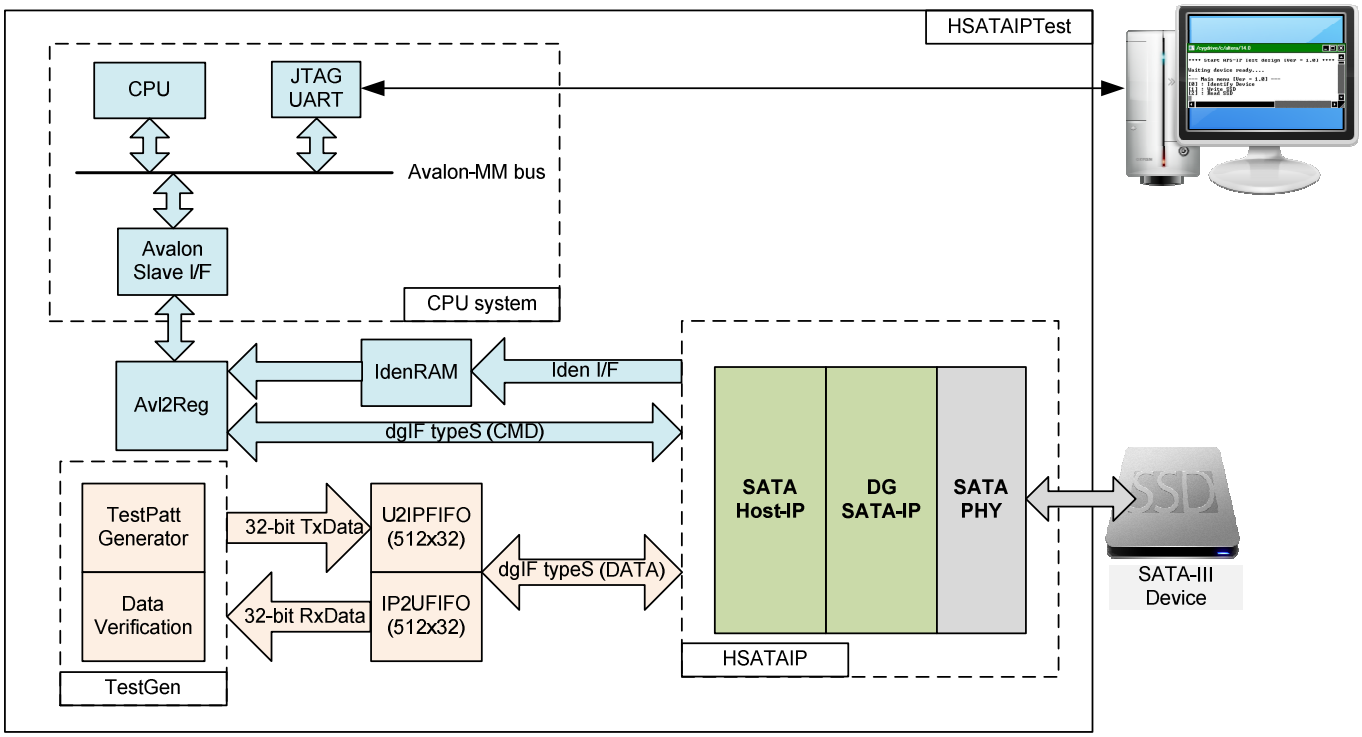

Figure 1 SATA Host IP Demo System

The reference design integrates SATA Host-IP, DG SATA-IP, and SATA-PHY to be SATA host standard platform. Simple logic is designed to show write/read data with SATA-III device at high-speed rate. CPU is additional designed from user interface through JTAG UART.

For simple test, user can input the parameters such as start address, transfer size, and command to NiosII command shell. The logic will decode all inputs and convert to be input value for SATA Host-IP. When the command is completed, CPU will check time usage and then calculate write/read performance of SATA-III device to be user output. To interface with CPU bus, Avl2Reg module is used to decode the address and data from CPU bus to be command interface of dgIF typeS for connecting with SATA Host-IP. Two external FIFOs are used to be buffer between SATA Host-IP and TestGen module.

User clock frequency in the reference design is 166 MHz, but CPU System within the reference designs runs at 50 MHz clock. So, Avl2Reg module will include asynchronous logic to support clock crossing domain between CPU system and user logic.

User can download SATA Host-IP datasheet and send the request to evaluate our IP from the website: http://www.dgway.com/SATA-IP_A_E.html .
The real transfer performance in the demo depends on SATA device specification.

2. CPU

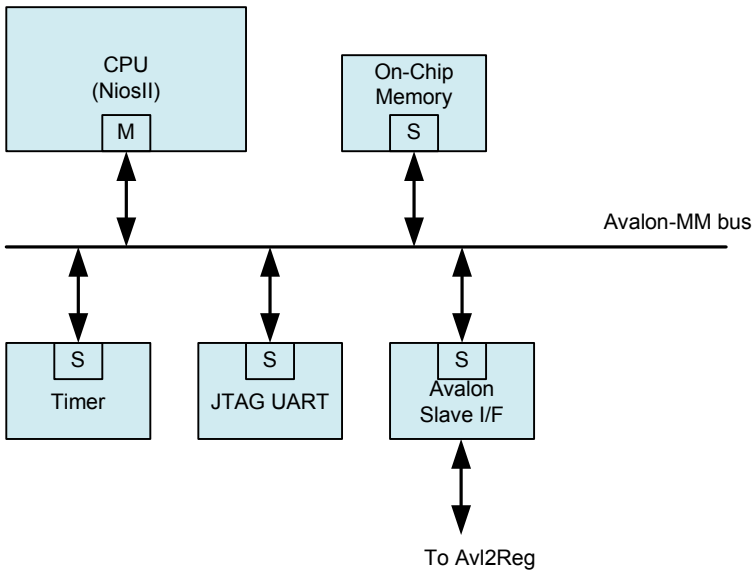

Figure 2 CPU system in reference design

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 test system. 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 “hsataiptest.c”)	Description
BA+0x00 Wr	User Address (Low) Reg (USRADRL_REG)	[31:0]: Input to be start sector address (UserAddr[31:0] of dgIF typeS)
BA+0x04 Wr	User Address (High) Reg (USRADRH_REG)	[15:0]: Input to be start sector address (UserAddr[47:32] of dgIF typeS)
BA+0x08 Wr	User Length (Low) Reg (USRLENL_REG)	[31:0]: Input to be transfer length in sector unit (UserLen[31:0] of dgIF typeS)
BA+0x0C Wr	User Length (High) Reg (USRLENH_REG)	[15:0]: Input to be transfer length in sector unit (UserLen[47:32] of dgIF typeS)
BA+0x10 Wr	User Command Reg (USRCMD_REG)	[1:0]: Input to be user command (UserCmd of dgIF typeS) “00”-Identify device, “01”-Security Erase, “10”-Write SSD, “11”-Read SSD When this register is written, the design will generate command request to SATA Host-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
BA+0x18 Wr	User Reset Reg (USRRST_REG)	[0]: ‘1’-Reset test system, ‘0’-Release reset

Address Rd/Wr	Register Name (Label in the “hsataiptest.c”)	Description
BA+0x100 Rd	User Status Reg (USRSTS_REG)	[0]: UserBusy of dgIF typeS (‘0’: Idle, ‘1’: Busy) [1]: UserError of dgIF typeS (‘0’: Normal, ‘1’: Error) [2]: Data verification fail (‘0’: Normal, ‘1’: Error) [4:3]: SATA speed from IP “00”: No linkup, “01”: SATA Gen1 (Not supported), “10”: SATA Gen2 (Not supported), “11”: SATA Gen3
BA+0x104 Rd	Total disk size (Low) Reg (LBASIZEL_REG)	[31:0]: Total capacity of SATA device in sector unit (LBASize[31:0] of dgIF typeS)
BA+0x108 Rd	Total disk size (High) Reg (LBASIZEH_REG)	[15:0]: Total capacity of SATA device in sector unit (LBASize[47:32] of dgIF typeS)
BA+0x10C Rd	User Error Type Reg (USRERRTYPE_REG)	[31:0]: User error status (UserErrorType[31:0] of dgIF typeS)
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+0x140 Rd	Read value Word0 Reg (RDPATW0_REG)	[31:0]: Latch value of read data [31:0] from read command
BA+0x150 Rd	Current test byte (Low) Reg (CURTESTSIZEL_REG)	[31:0]: Current test data size of TestGen module in byte unit (bit[31:0])
BA+0x154 Rd	Current test byte (High) Reg (CURTESTSIZEH_REG)	[24:0]: Current test data size of TestGen module in byte unit (bit[56:32])
BA+0x2000 – 0x21FF	Identify Device Command Data (IDENCTRL_REG)	512-byte Identify Device Data

After initialization complete, CPU firmware in the demo will be in idle state to wait user command input through Serial console. The command can be Identify device, Security erase, Write, or Read command. The sequence of each command is follows.

For Identify device command,

1) Set USRCMD_REG=”00”. Test logic will generate command and request to SATA Host-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, SATA device model name, security feature set supported, and erase time value decoded from IDENCTRL_REG are displayed to the command shell. Also, SATA device capacity read from LBASIZEL/H_REG are displayed.

For Security erase command,

1) Set USRCMD_REG=”01”. Test logic will generate command and request to SATA Host-IP. Busy flag (USRSTS_REG[0]) will change from ‘0’ to ‘1’.
2) CPU will wait until command complete by monitoring USRSTS_REG value. Bit[0] will be cleared to ‘0’ when command is completed. This operation may use long time to operate, so there is dummy message displayed on the console every second to show system alive status. Finally, total time usage will be printed on the command shell when command is completed.

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, current transfer size from CURTESTSIZE_REG will be printed to console during operating. Finally, test performance will be displayed on the command shell when command is completed.

3. Avl2Reg

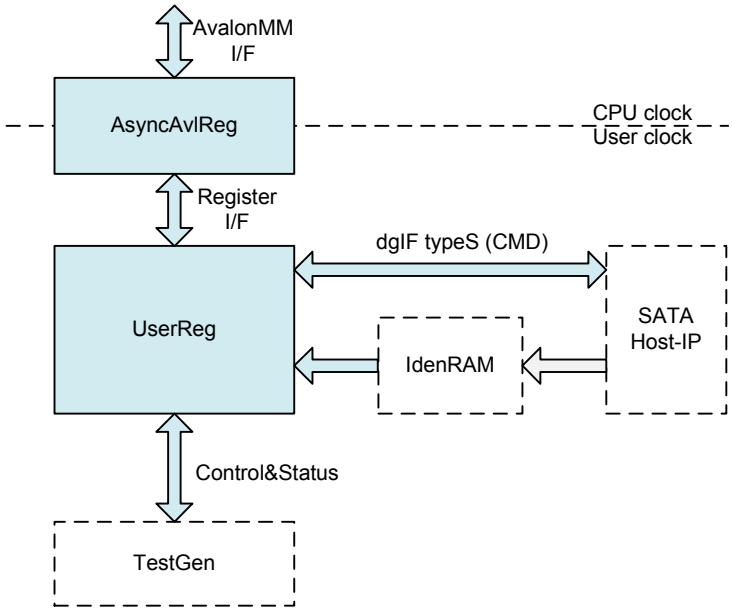

Figure 3 Avl2Reg interface

This module consists of two submodules, i.e. AsyncAvlReg and UserReg. AsyncAvlReg is designed to convert Avalon interface to be register interface and convert clock domain from CPU clock to user clock system. UserReg module is designed to decode write/read address which is mapped following Table 1. Transfer parameters such as transfer direction, size, and address from user will be converted to command interface of dgIF typeS for SATA Host-IP and control signal for TestGen module. During transferring, CPU read the register to check IP status, TestGen result, and Identify device data.

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 4.

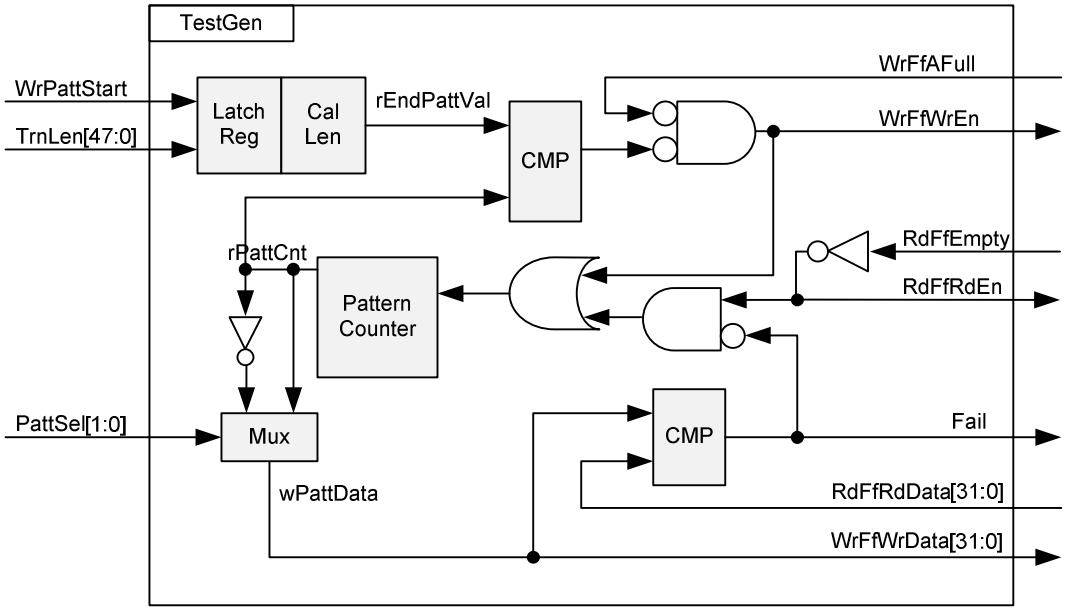

Figure 4 Logic design in TestGen

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 set 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, i.e. increment, decrement, all 0, and all 1 value.

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	28-Oct-16	Initial Release
1.1	24-Jan-17	Update signal to dgIF typeS