# SATA Host-IP reference design manual

Rev1.2 14-Nov-17

## 1. Overview

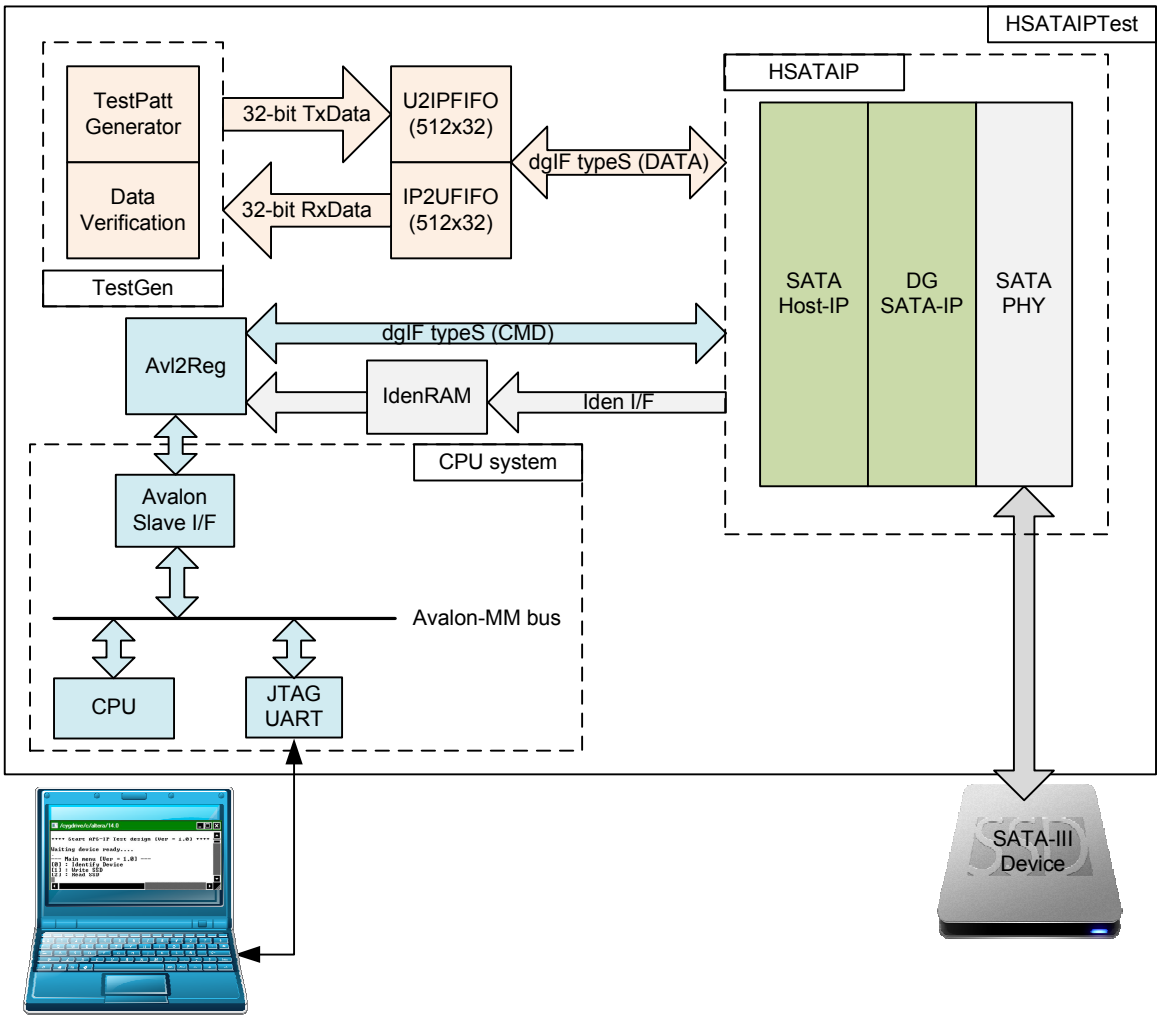


Figure 1-1 SATA Host IP Demo System

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 for user interface through JTAG UART.

For simple test, user inputs the parameters such as start address, transfer size, and command to NiosII command shell. After that, the logic processes all inputs and converts to be SATA Host-IP input. After the operation is completed, CPU checks time usage and displays write/read performance of SATA-III device. To interface with CPU bus, Avl2Reg module is used to decode the address and data from Avalon bus and converts to command interface of dgIF types. Data interface of dgIF types is connected to external FIFOs and transferred to SATA-IP. TestGen module includes test pattern generator to generate Test data. The test data is transferred to SATA device in Write Test, and used to be expected value for data verification in Read Test. Identify data is transferred to IdenRAM. Identify data is used by CPU to decode SATA model name to display on NiosII command shell.

User clock frequency to run SATA Host-IP in the reference design is 166 MHz. but CPU System runs at 100 MHz. Asynchronous circuit is required within Avl2Reg.

User can download SATA Host-IP datasheet and send the request to evaluate the 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 and Peripherals

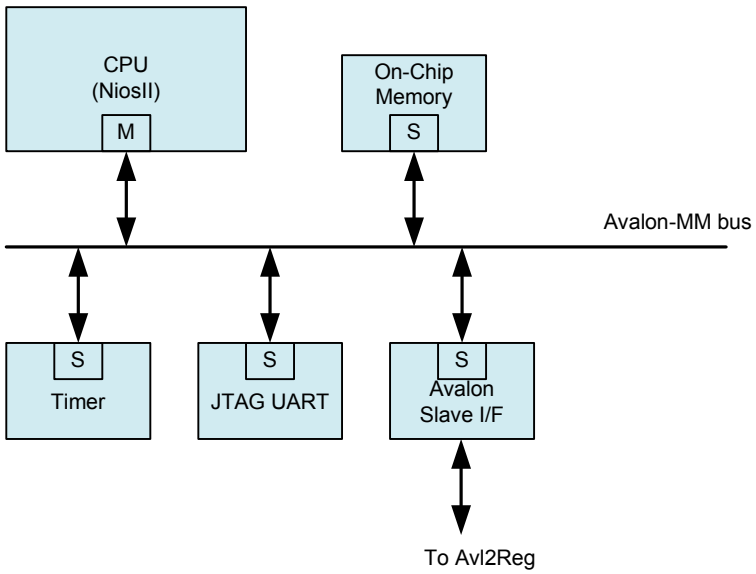


Figure 2-1 CPU system in reference design

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 and monitoring test system. More details about memory map for CPU to access Avalon-MM bus are shown as follows.

Table 1 Register Map

| Address<br>Rd/Wr | Register Name<br>(Label in the “hsataiptest.c”) | Description |
|---|---|---|
| BA+0x00<br>Wr | User Address (Low) Reg<br>(USRADRL_REG) | [31:0]: Input to be start sector address (UserAddr[31:0] of dgIF typeS) |
| BA+0x04<br>Wr | User Address (High) Reg<br>(USRADRH_REG) | [15:0]: Input to be start sector address (UserAddr[47:32] of dgIF typeS) |
| BA+0x08<br>Wr | User Length (Low) Reg<br>(USRLENL_REG) | [31:0]: Input to be transfer length in sector unit (UserLen[31:0] of dgIF typeS) |
| BA+0x0C<br>Wr | User Length (High) Reg<br>(USRLENH_REG) | [15:0]: Input to be transfer length in sector unit (UserLen[47:32] of dgIF typeS) |
| BA+0x10<br>Wr | User Command Reg<br>(USRCMD_REG) | [1:0]: Input to be user command (UserCmd of dgIF typeS)<br>“00”-Identify device, “01”-Security Erase, “10”-Write SSD, “11”-Read SSD<br>When this register is written, the design will generate command request to SATA Host-IP to start new command operation. |
| BA+0x14<br>Wr | Test Pattern Reg<br>(PATTSEL_REG) | [2:0]: Test pattern select<br>“000”-Increment, “001”-Decrement, “010”-All 0, “011”-All 1, “100”-LFSR |
| BA+0x18<br>Wr | User Reset Reg<br>(USRRST_REG) | [0]: ‘1’-Reset test system, ‘0’-Release reset |

| Address<br>Rd/Wr | Register Name<br>(Label in the “hsataiptest.c”) | Description |
|---|---|---|
| BA+0x100<br>Rd | User Status Reg<br>(USRSTS_REG) | [0]: UserBusy of dgIF typeS (‘0’: Idle, ‘1’: Busy)<br>[1]: UserError of dgIF typeS (‘0’: Normal, ‘1’: Error)<br>[2]: Data verification fail (‘0’: Normal, ‘1’: Error)<br>[4:3]: SATA speed from IP<br>“00”: No linkup, “01”: SATA Gen1 (Not supported),<br>“10”: SATA Gen2 (Not supported), “11”: SATA Gen3 |
| BA+0x104<br>Rd | Total disk size (Low) Reg<br>(LBASIZEL_REG) | [31:0]: Total capacity of SATA device in sector unit<br>(LBASize[31:0] of dgIF typeS) |
| BA+0x108<br>Rd | Total disk size (High) Reg<br>(LBASIZEH_REG) | [15:0]: Total capacity of SATA device in sector unit<br>(LBASize[47:32] of dgIF typeS) |
| BA+0x10C<br>Rd | User Error Type Reg<br>(USRERRTYPE_REG) | [31:0]: User error status (UserErrorType[31:0] of dgIF typeS) |
| BA+0x11C<br>Rd | SATA Host IP Test pin Reg<br>(TESTPIN_REG) | [31:0]: TestPin[31:0] output from SATA Host-IP |
| BA+0x120<br>Rd | Data Failure Address (Low) Reg<br>(RDFAILNOL_REG) | [31:0]: Latch value of failure address[31:0] in byte unit from read command |
| BA+0x124<br>Rd | Data Failure Address (High) Reg<br>(RDFAILNOH_REG) | [24:0]: Latch value of failure address [56:32] in byte unit from read command |
| BA+0x130<br>Rd | Expected value Word0 Reg<br>(EXPPATW0_REG) | [31:0]: Latch value of expected data [31:0] from read command |
| BA+0x140<br>Rd | Read value Word0 Reg<br>(RDPATW0_REG) | [31:0]: Latch value of read data [31:0] from read command |
| BA+0x150<br>Rd | Current test byte (Low) Reg<br>(CURTESTSIZEL_REG) | [31:0]: Current test data size of TestGen module in byte unit (bit[31:0]) |
| BA+0x154<br>Rd | Current test byte (High) Reg<br>(CURTESTSIZEH_REG) | [24:0]: Current test data size of TestGen module in byte unit (bit[56:32]) |
| BA+0x2000<br>– 0x21FF | Identify Device Command Data<br>(IDENCTRL_REG) | 512-byte Identify Device Data |

After initialization complete, CPU firmware in the demo is in idle state to wait user command input through NiosII command shell. The user command is Identify device, Security erase, Write, or Read command. The sequence of each command is as follows.

For Identify device command,

1) Set USRCMD_REG=”00”. Next, Test logic generates command and sends request to SATA Host-IP. After that, busy flag (USRSTS_REG[0]) changes from ‘0’ to ‘1’.
2) CPU waits until command complete or error is found by monitoring USRSTS_REG value. Bit[0] is cleared to ‘0’ when command is completed. Bit[1] is asserted to ‘1’ when some errors are detected. In case of error situation, error message is 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 NiosII command shell. Also, SATA device capacity read from LBASIZEL/H_REG is displayed.

For Security erase command,

1) Set USRCMD_REG=”01”. Test logic generates command and sends request to SATA Host-IP. Next, Busy flag (USRSTS_REG[0]) changes from ‘0’ to ‘1’.
2) CPU waits until command complete by monitoring USRSTS_REG value. Bit[0] is cleared to ‘0’ when command is completed. This operation may use long time to operate, so there is dummy message displayed on NiosII command shell every second to show system still alive. Finally, total time usage is displayed on NiosII command shell after command is completed.

For Write/Read command,

1) Receive start address, transfer length, and test pattern value from user through NiosII command shell. If some inputs are 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] is also monitored for read command to confirm that read data is correct.
4) During running command, current transfer size from CURTESTSIZE_REG is displayed on NiosII command shell every second. Finally, test performance is displayed on NiosII command shell when command is completed.

# 3. Avl2Reg

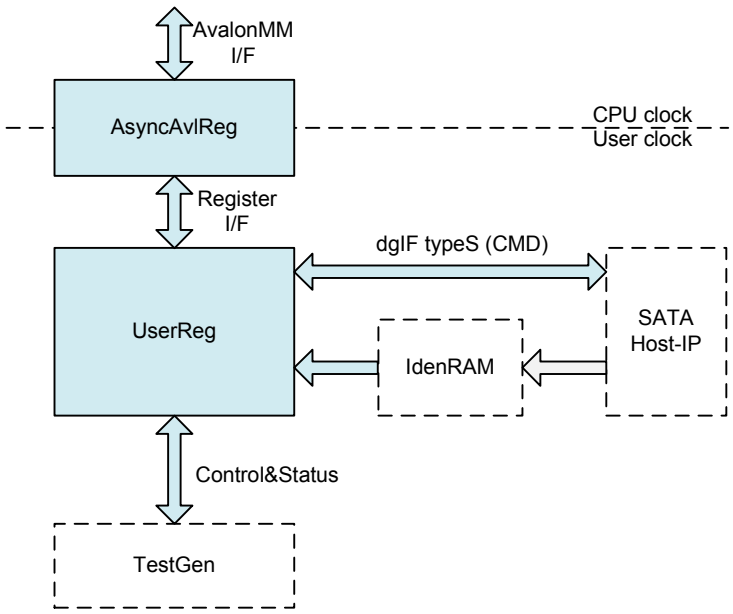


Figure 3-1 Avl2Reg interface

This module consists of two submodules, i.e. AsyncAvlReg and UserReg. AsyncAvlReg is designed to convert Avalon interface to be register interface and to convert clock domain from CPU clock to user clock system. UserReg module includes the logic to decode Write/Read address to select the register for current access. The address is decoded following Table 1. Transfer parameters such as transfer direction, size, and address from user are converted to be command interface of dgIF typeS for SATA Host-IP and converted to be control signal for TestGen module. During transferring, CPU reads the register to check SATA Host-IP status, TestGen result, or Identify device data.

# 4. TestGen

In this module, there are two modes. For Write command, it generates test data to WrFf port, while it verifies received data from RdFf port with expected value for Read command. The details of logic design inside this module are displayed in Figure 4-1.

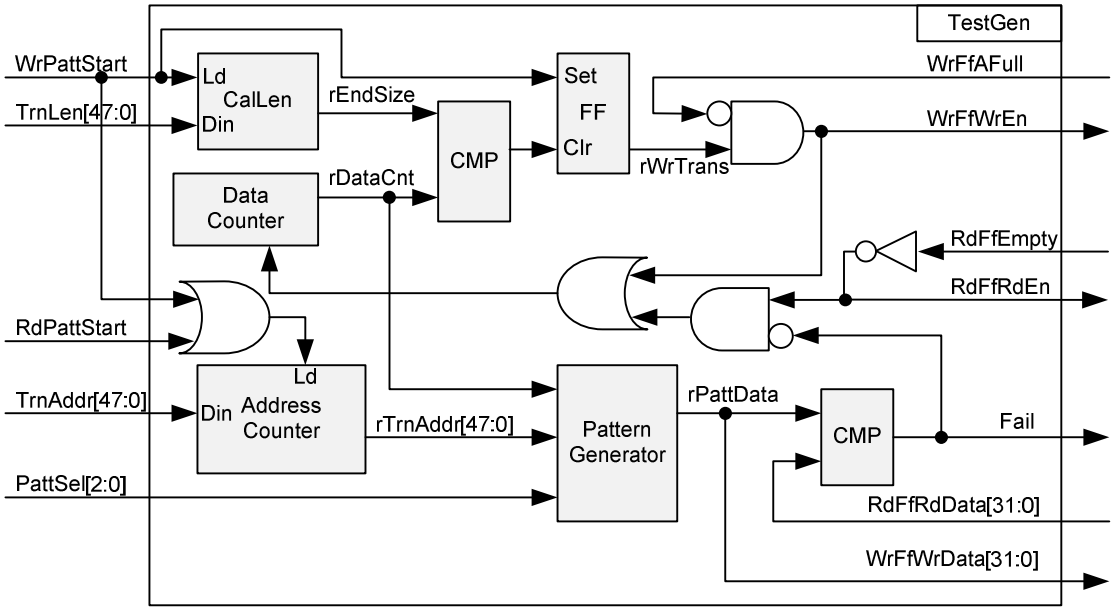


Figure 4-1 Logic design in TestGen

To start write transfer, WrPattStart is asserted to '1' with valid TrnLen and TrnAddr. TrnLen is loaded to calculate the end address of write transfer. rWrTrans is the control signal to generate write enable signal for WrFf port. rWrTrans is asserted to '1' by WrPattStart and de-asserted to '0' at the end transfer, monitored by rEndSize. WrFfAFull is monitor to pause data write transfer when WrFfAFull is asserted to '1'.

Two counters are designed in this module. The first is data counter, used to count total transfer size for both write and read command. The counter enable is controlled by write enable of WrFf or read enable of RdFf. This counter is monitored to de-assert rWrTrans to '0' after complete write command. The second is address counter which is increased when 512-byte data is transferred to/from WrFf/RdFf. The start value of address counter is loaded from TrnAddr signal, so the counter shows the sector address of current data transfer.

The sector address is used by Pattern Generator to create 64-bit unique header for each sector. Also, pattern generator uses the address to be the initial value before calculating the next value following test pattern mode. rPattData is used to be WrFfWrData for write command and applied to be expected value for read command.

For read transfer, RdPattStart is used to load the address counter only. RdFfRdEn is created by using not logic of RdFfEmpty signal to read data when RdFf is available. Fail flag will be asserted to '1' if RdFfRdData from RdFf port is not equal to test pattern.

## 5. Example Test Result

The example test result when running demo system by using 256 GB Samsung 850 Pro is shown in Figure 5-1.

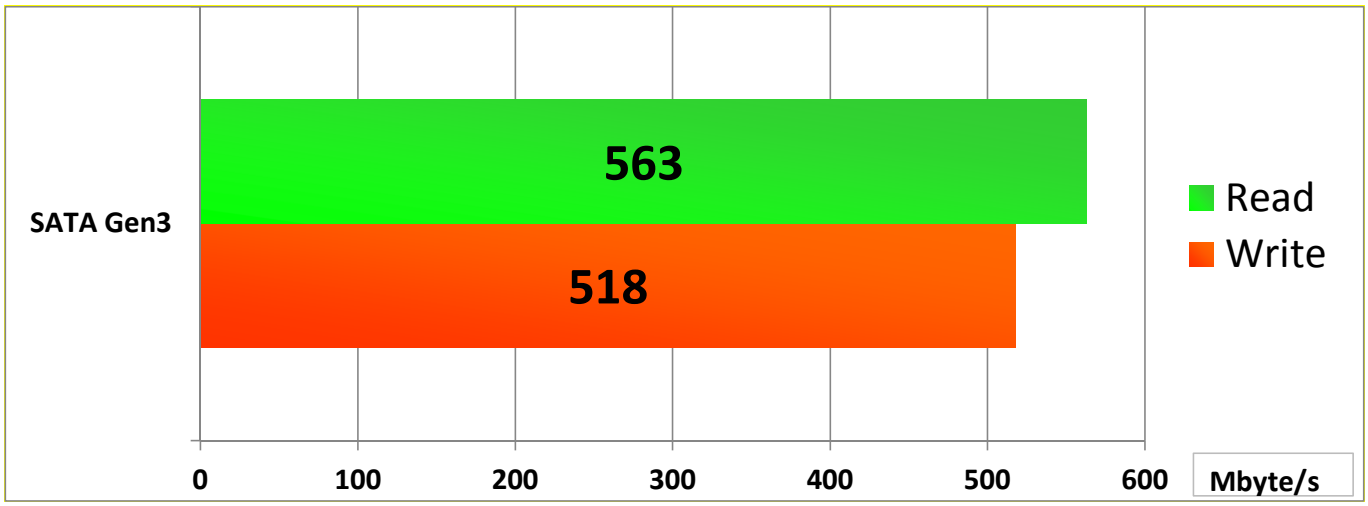


Figure 5-1 Test Performance of SATA Host IP demo by using Samsung 850 Pro SSD

By using SATA Gen3 design with Samsung 850 Pro SSD, write performance is about 518 Mbyte/sec and read performance is about 563 Mbyte/sec.

## 6. Revision History

| Revision | Date | Description |
|---|---|---|
| 1.0 | 28-Oct-16 | Initial Release |
| 1.1 | 24-Jan-17 | Update signal to dgIF typeS |
| 1.2 | 14-Nov-17 | Add LFSR pattern |