HCTL-IP RAID0x8 reference design manual

Rev1.0 22-Aug-23

HCTL-IP RAID0x8 reference design manual

Rev1.0 22-Aug-23

1 Introduction

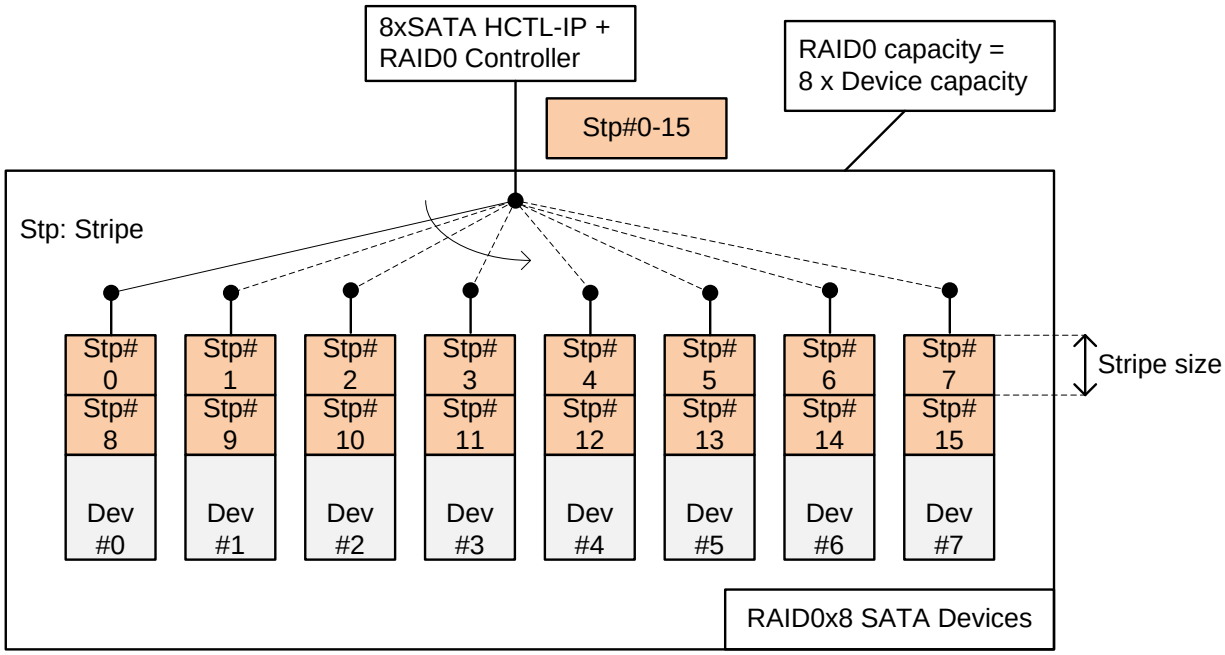

Figure 1-1 RAID0 by 8 SATA devices data format

RAID0 system uses multiple storages to extend total storage capacity and to increase write/read performance. Assumed that total number of device is N, total storage capacity is equal to N multiply by amount of storage. Write and read speed are almost equal to N multiply by speed of one device.

Data format of RAID0 is shown in Figure 1-1. Data stream of the host side are split into a small stripe and transfer to one device at a time. Stripe size is the data size to store in one device before switching to other devices.

In the reference design, eight devices are applied to run RAID0 system. Stripe size is equal to 512 byte (one sector). All devices connecting in the system should be same model to match characteristic and get the best performance and capacity. By using RAID0 reference design, the total capacity is equal to eight times of one device capacity and the performance result for write and read are almost eight times of one device performance.

The demo does not include DDR and uses only FIFO implemented by BlockRAM to be the buffer. Test performance in the demo is average speed. The system is suitable for high-speed data recording application which has flow control to pause data transferring when the device is not ready to transfer data.

Before running the reference design, it is recommended to study SATA HCTL-IP datasheet and single channel demo firstly from following link.
http://www.dgway.com/products/IP/SATA-IP/dg_sata_host_ip_data_sheet_en.pdf
http://www.dgway.com/products/IP/SATA-IP/dg_satahostip_refdesign_en.pdf
http://www.dgway.com/products/IP/SATA-IP/dg_satahostip_instruction_en.pdf

2 Hardware overview

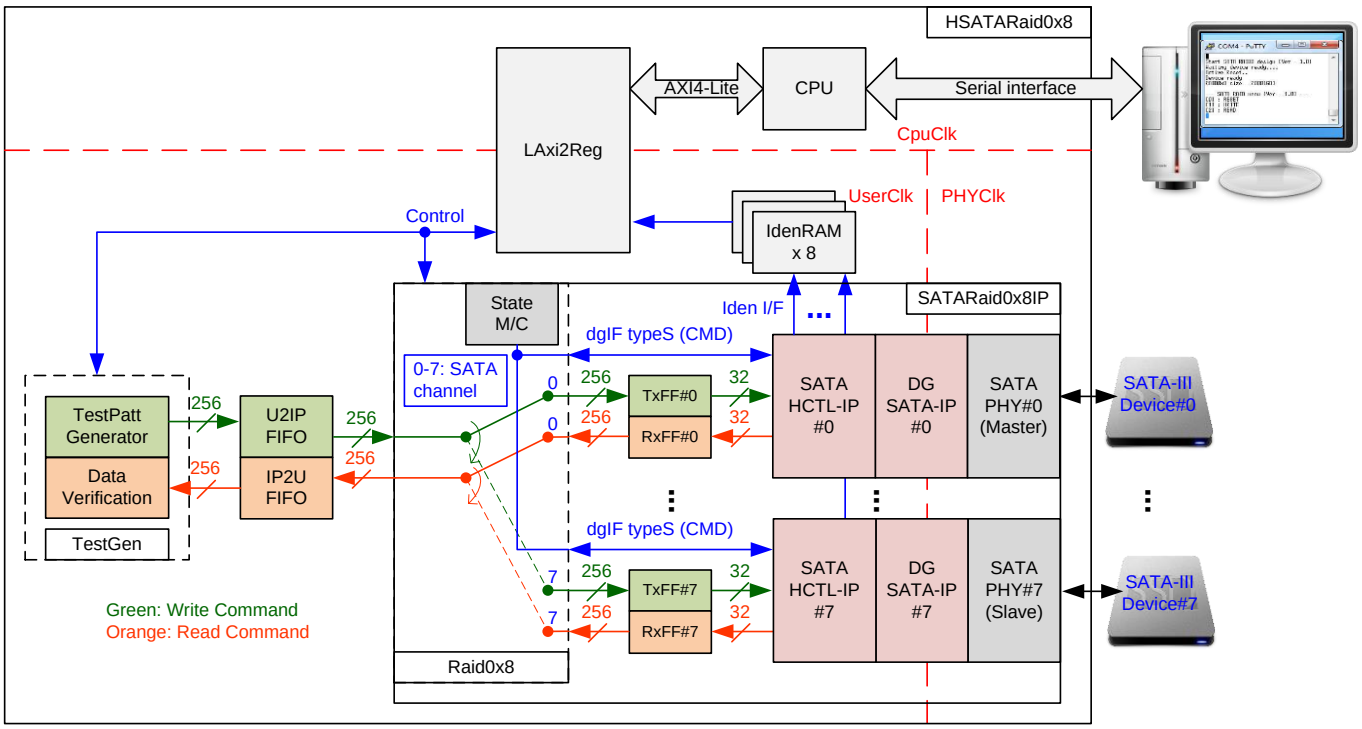

Figure 2-1 HCTL-IP RAID0x8 demo hardware

The hardware system can be split into three groups following the interface i.e.

1) TestGen: The example of user logic to write and read data in this reference design is TestGen module. TestGen module generates test data to U2IPFIFO at the highest speed with flow control in Write command. For Read command, TestGen reads and verifies test data from IP2UFIFO at the highest speed with flow control. TestGen uses 256-bit data bus and runs in UserClk domain which is equal to 200 MHz, so maximum bandwidth is 6400 MB/s which is more than maximum performance of eight SATA devices.
2) RAID: Eight SATA HCTL-IPs are applied to control eight SATA-III devices. Raid0x8 arranges data stored in eight devices as RAID0 format. Raid0x8 decodes the current address to select one SATA device to transfer data with TestGen. Data bus size of RAID0 is 256-bit which is eight times of SATA HCTL-IP to match data bandwidth with eight SATA devices. Two FIFOs (TxFF and RxFF) are connected between Raid0x8 and SATA HCTL-IP to convert data bus size.
3) CPU: Test operation in the demo is controlled by user through Serial console. CPU firmware is designed to receive test parameters and the command from user. CPU sets parameters to the hardware through AXI4-Lite bus. LAxi2Reg has the register sets of test parameters which are mapped to different address of CPU. LAxi2Reg decodes the address of AXI4-Lite bus to select the active parameter. For write access, Write data from AXI4-Lite bus is set to the selected parameter following the address. For read access, Read data from selected parameter is returned to AXI4-Lite bus. Read access is applied for CPU monitoring and displaying the hardware status to the user through Serial console.

   More details of the hardware are described as follows.

2.1 TestGen

This module is designed to generate Test pattern to WrFf in Write command or reads data from RdFf to verify in Read command at the fastest speed to check system performance. The details of hardware inside TestGen are shown in Figure 2-2.

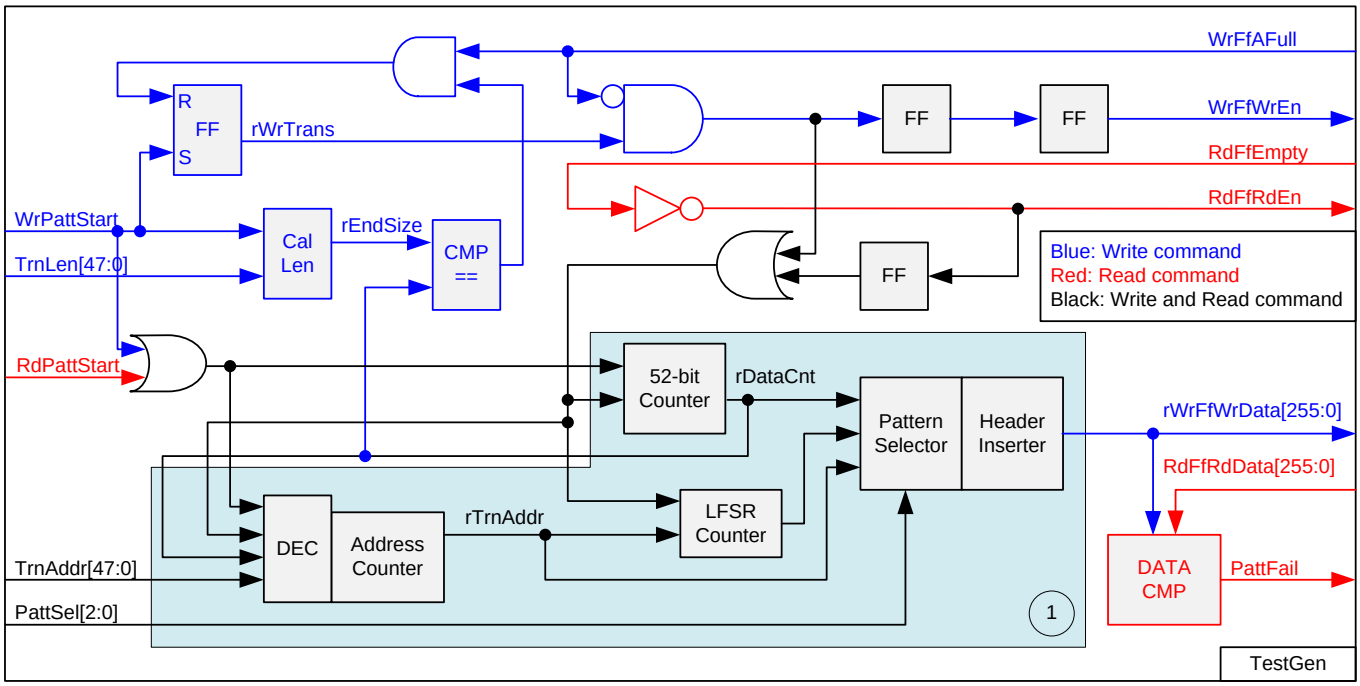

Figure 2-2 TestGen hardware

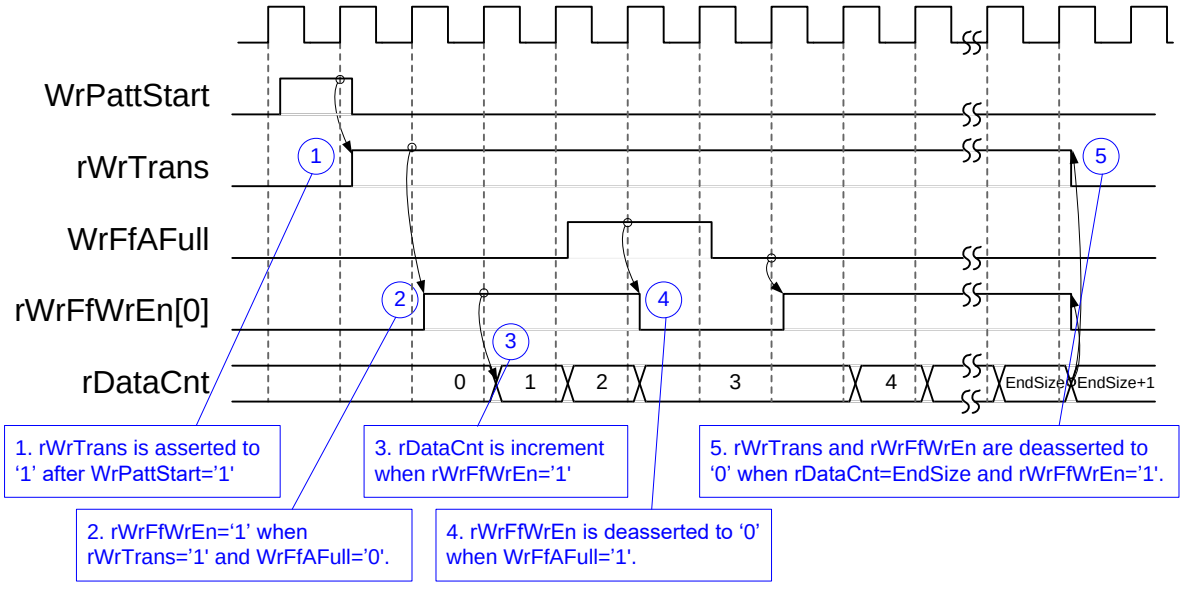

Figure 2-3 Timing diagram of Write operation in TestGen

rWrTrans signal is asserted to '1' during write operation. It is latch to '1' when WrPattStart from LAxi2Reg is asserted to '1' for one clock cycle. rWrFfWrEn[0] is asserted to '1' to generate test data to WrFf when rWrTrans='1' and WrFfAFull='0'. If WrFfAFull='1', rWrFfWrEn[0] will be de-asserted to '0' to pause data transferring. rDataCnt is data counter to check total transfer size, increased by rWrFfWrEn[0]. When total data are transferred completely (rDataCnt=EndSize), rWrTrans and rWrFfWrEn[0] are de-asserted to '0' to stop data transferring.

For read operation, RdFfRdEn signal is designed by using NOT logic to RdFfEmpty. RdFfRdEn is also used to be counter enable for rDataCnt to generate test pattern to verify with the received data (RdFfRdData).

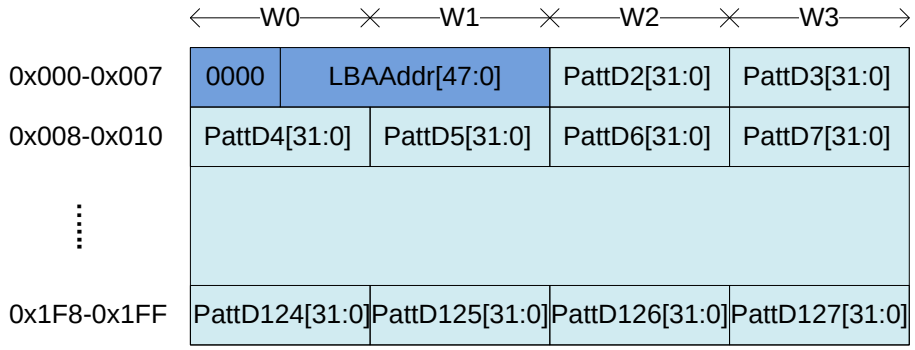

Figure 2-4 Test pattern format in each sector

Block no.1 in lower side of Figure 2-2 shows the logic for generating test pattern in TestGen module. To create unique test data for each sector, test pattern in this demo is designed as shown in Figure 2-4.

Test pattern consists of two parts, i.e. 64-bit header in word#0 and word#1 of each sector and test data in word#2 – word#127. 64-bit header is created by using LBA address value of the data (LBA address is the address in sector unit). As shown in Figure 2-2, rTrnAddr is loaded to TestGen module to create 64-bit header value for the 1st sector data. rTrnAddr is increased after completing to transferring one sector data. So, the logic to create rTrnAddr needs to monitor rDataCnt and write/read enable signal to know the end of one sector transfer.

TestGen supports to generate five patterns, i.e. 32-bit increment, 32-bit decrement, all 0, all 1, and 32-bit LFSR. 32-bit increment is generated by using lower-bit of rTrnAddr and rDataCnt. Decrement pattern is designed by using NOT logic with increment data. To create 32-bit LFSR counter, 256-bit data is designed by using two sets of 32-bit LFSR pattern as shown in Figure 2-5.

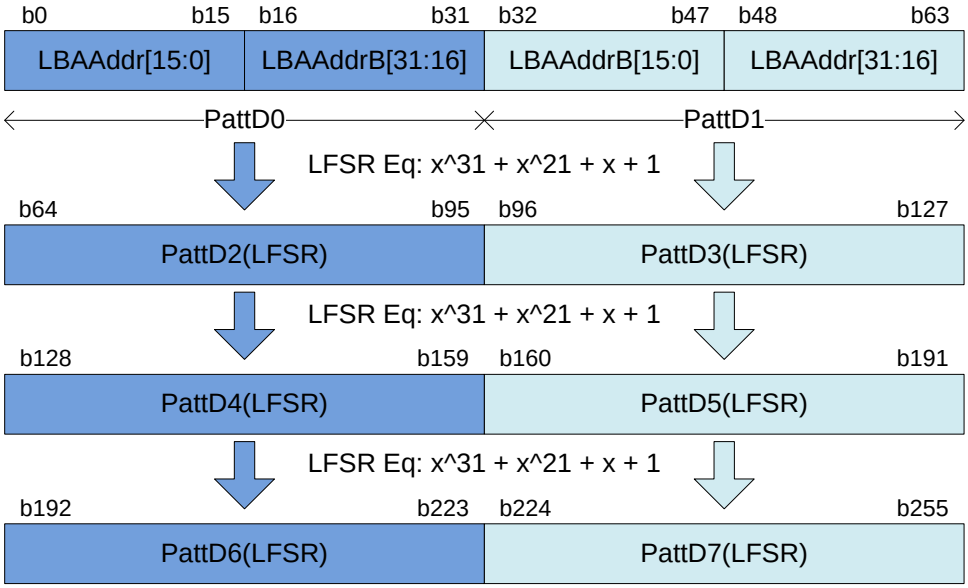

Figure 2-5 LFSR pattern in TestGen

Start value of LFSR is designed by using combination signal of 32-bit LBA address and NOT logic of LBA address (LBAAddrB means NOT logic of LBAAddr signal). PattD0/D2/D4/D6 are the sequence of LFSR counter set#0 which using LBAAddr[15:0] and NOT LBAAddr[31:16] to be start value. PattD1/D3/D5/D7 are the sequence of LFSR counter set#1 which using NOT LBAAddr[15:0] and LBAAddr[31:16] to be start value. 256-bit data is generated within one clock, so PattD2-D7 must design the logic to calculate 32-bit LFSR pattern as look-ahead style.

3-bit PattSel signal are used to select one of five test patterns. Header Inserter logic inserts 64-bit header to be the 1st data of each sector. After that, test data from pattern counter is transferred to rWrFfWrData. In Read command, rWrFfWrData is used to be expected value to compare with read data from FIFO (RdFfRdData). PattFail is asserted to '1' when data verification is failed.

2.2 RAID

User interface of RAID is designed by using dgIF typeS format. CMD interface is connected to LAxi2Reg to receive the parameter from user through Serial console. Data bus size of RAID block is 256-bit which is connected with U2IPFIFO and IP2UFIFO. Another side of RAID is connected to eight SATA devices. The system includes eight IdenRAMs for receiving Iden I/F of SATA HCTL-IP#0 - #7. The hardware details of RAID are shown in Figure 2-6.

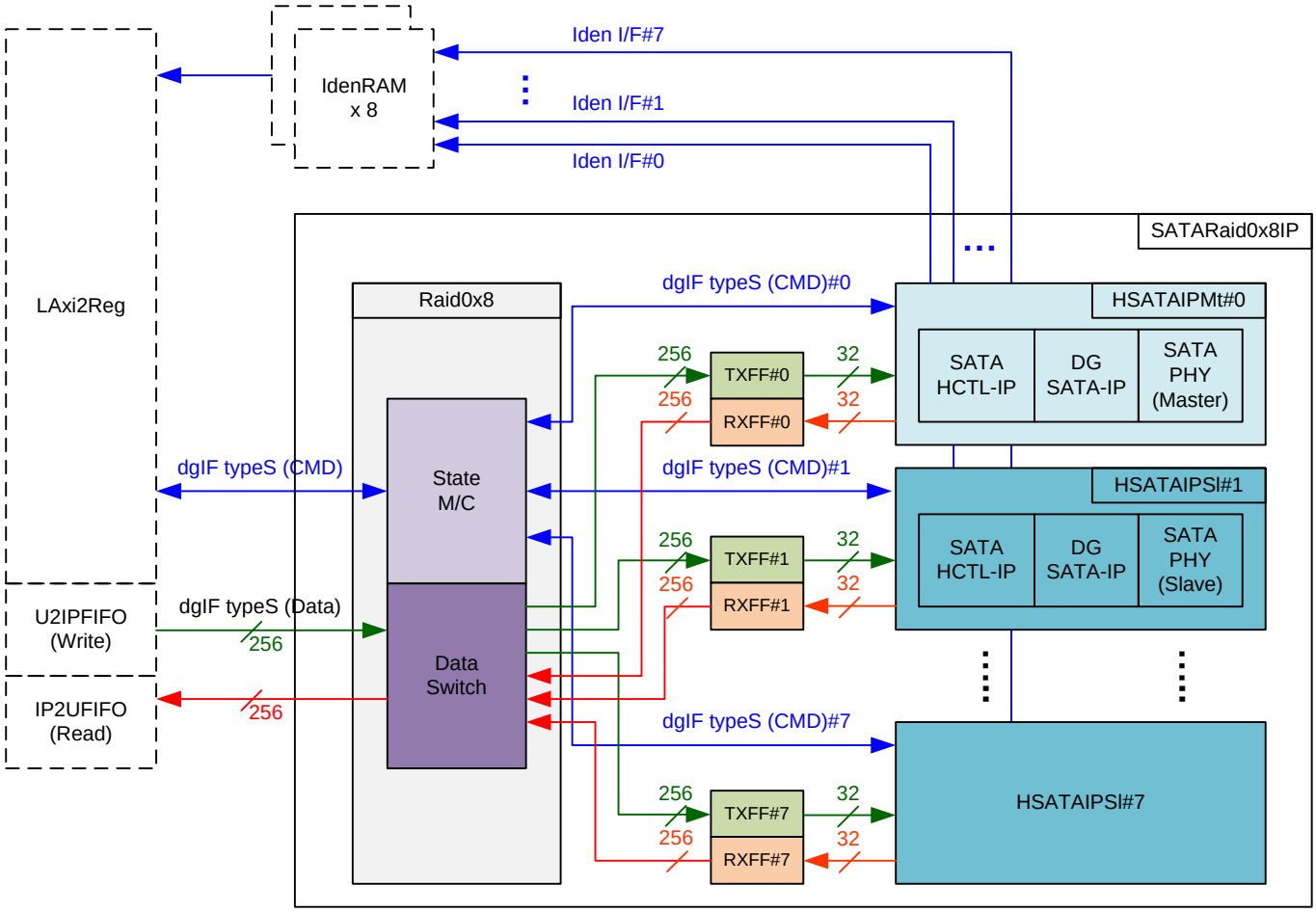

Figure 2-6 RAID hardware

The hardware inside RAID consists of three modules, i.e. Raid0x8, eight sets of FIFO, and eight sets of HSATAIP.

2.2.1 HSATAIP

HSATAIP is the group of complete HDL for SATA application. It consists of SATA HCTL-IP, SATA-IP, and SATAPHY. More details of SATA HCTL-IP are described in datasheet. http://www.dgway.com/products/IP/SATA-IP/dg_sata_host_ip_data_sheet_en.pdf

HSATAIP has two HDL codes because of different SATA PHY. There are two types of SATA PHY in RAID design, i.e. Master and Slave. All SATA PHYs use same clock source for transceiver operation. Clock source is created in Master PHY and then forwarded to all Slave PHYs.

2.2.2 FIFO

Two FIFOs are applied for one SATA channel. TxFIFO is designed to convert 256-bit data which is Raid0x8 bus size to 32-bit data which is HSATAIP bus size. FIFO size is 16 kByte, so it can store data up to 31 sectors. RxFIFO size is same as TxFIFO, but it converts data from 32-bit to 256-bit.

2.2.3 Raid0x8

Raid0x8 consists of state machine for controlling command interface and Data Switch to controlling data interface.

After receiving new command request from LAxi2Reg, state machine decodes the parameters and then calculates the address and transfer length of each HSATAIP. After that, state machine generates command request with the valid address and length to all HSATAIPs.

Next, state machine is responsible to control Data Switch for selecting the active SATA channel. State machine needs to calculate the current sector address by loading the start value from LAxi2Reg and then increases after completing to transfer one sector data with U2IPFIFO/IP2UFIFO. Three lower bit of current sector are used to select SATA channel, i.e. “000” for HSATAIP#0, “001” for HSATAIP#1, …, “111” for HSATAIP#7.

There are many pipeline registers to switch data bus of each SATA channel. The overhead time from pipeline register and control signal is about 6 clock cycles for transferring 512-byte data (16x256-bit). To compensate this overhead time, clock domain to run Raid0x8 must be more than 150 MHz which is SATA PHY clock. It is recommended to set clock frequency more than 190 MHz to compensate 27% overhead time (6 / (16+6) = 27%). In the reference design, Raid0x8 clock frequency is set to 200 MHz.

The user interface of Raid0x8 is shown in Table 2-1. Command and data interface are designed as dgIF typeS format. The status and Iden port of all SATA HCTL-IP are mapped to user interface directly to monitor each SATA device status independently.

Table 2-1 Signal description of Raid0x8 (only user interface)

Signal	Dir	Description
User Interface		
RstB	In	Reset signal. Active low. Please use same reset signal as SATA HCTL-IP.
Clk	In	User clock. Must use the same clock as SATA HCTL-IP.
dgIF typeS		
UserCmd[1:0]	In	User Command. “00”: Identify device command, “01”: Security erase “10”: Write device, “11”: Read device.
UserAddr[47:0]	In	Start address of device to write/read in sector unit (512 byte).
UserLen[47:0]	In	Total transfer size in the request in sector unit (512 byte). Valid from 1 to (LBASize-UserAddr).
UserReq	In	Request the new command. Can be asserted only when the IP is Idle (UserBusy=’0’). Asserted with valid value on UserCmd/UserAddr/UserLen signals.
UserBusy	Out	IP Busy status. New request will not be allowed if this signal is asserted to ‘1’.
LBASize[47:0]	Out	Total device capacity in sector unit (512 byte). Default value is 0. This value is equal to eight times of LBASize value output from IP#0.
UserError	Out	Error flag. Assert when UserErrorType is not equal to 0. The flag can be reset only by de-asserting RstB to ‘0’.
UserErrorType[63:0]	Out	Error status which are mapped from status in each SATA HCTL-IP. One byte is mapped to UserErrorType[7:0] from one IP. IP#0 is mapped to the lowest bye while IP#7 is mapped to the highest byte. [7:0] – Mapped from UserErrorType[7:0] of IP#0. [15:8] – Mapped from UserErrorType[7:0] of IP#1. … [63:56] – Mapped from UserErrorType[7:0] of IP#7.
UserFifoWrCnt[15:0]	In	Write data counter of User received FIFO. Used to check FIFO space size. If total size is less than 16-bit, please fill ‘1’ to upper bit. UserFifoWrEn can be asserted when UserFifoWrCnt[15:5] is not equal to all 1.
UserFifoWrEn	Out	Write data valid of User received FIFO
UserFifoWrData[255:0]	Out	Write data bus of User received FIFO. Synchronous to UserFifoWrEn.
UserFifoRdCnt[15:0]	In	Read data counter of User transmit FIFO. Used to check data available size in FIFO. If total FIFO size is less than 16-bit, please fill ‘0’ to upper bit. UserFifoRdEn can be asserted when UserFifoRdCnt[15:4] is not equal to 0.
UserFifoEmpty	In	FIFO empty flag of User transmit FIFO. This signal is unused in the design.
UserFifoRdEn	Out	Read valid of User transmit FIFO
UserFifoRdData[255:0]	In	Read data returned from User transmit FIFO. Valid after UserFifoRdEn asserted one clock cycle.
Other Interface		
TestPin[0-7][31:0]	Out	Direct mapped from TestPin in each SATA HCTL-IP. Index 0-7 is referred to IP number. [0]-IP#0, [1]-IP#1, … , [7]-IP#7.
TimeOutSet[31:0]	Out	Timeout value of all SATA HCTL-IPs. Time unit is following Clk frequency value.
TrnLinkup[7:0]	Out	Linkup signal from trn_link_up in each SATA-IP. Index 0-7 is referred to IP number. [0]-IP#0, [1]-IP#1, … , [7]-IP#7.
IdenWrAddr[0-7][6:0]	Out	Direct mapped from IdenWrAddr in each SATA HCTL-IP. Index 0-7 is referred to IP number. [0]-IP#0, [1]-IP#1, … , [7]-IP#7.
IdenWrEn[7:0]	Out	Direct mapped from IdenWrEn in each SATA HCTL-IP. Index 0-7 is referred to IP number. [0]-IP#0, [1]-IP#1, … , [7]-IP#7.
IdenWrData[0-7][31:0]	Out	Direct mapped from IdenWrData in each SATA HCTL-IP. Index 0-7 is referred to IP number. [0]-IP#0, [1]-IP#1, … , [7]-IP#7.

Timing diagram of user interface and Identify device interface are similar to SATA HCTL-IP. Please see more details from SATA HCTL-IP datasheet. Data interface for RAID operation are shown as follows.

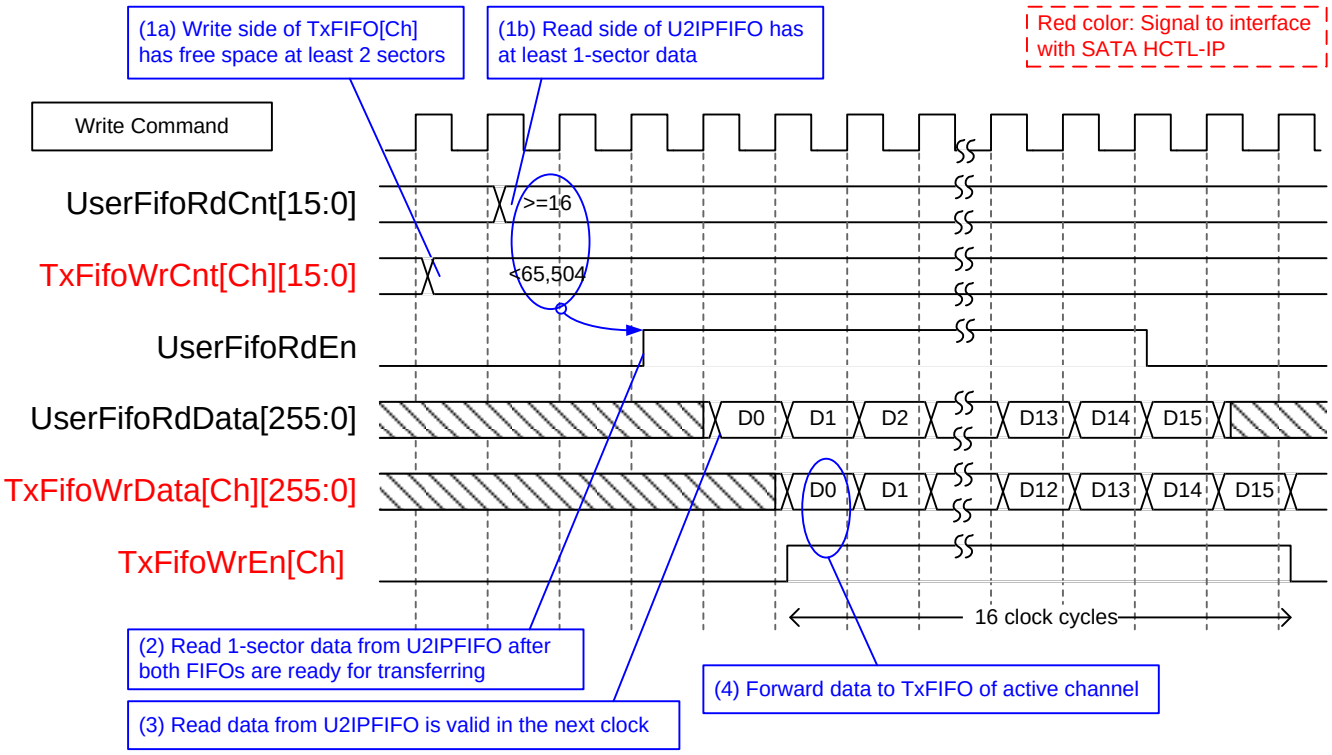

Figure 2-7 Raid0x8 data timing diagram of Write command

When user sends Write command to RAID, data is forwarded from U2IPFIFO to TxFIFO[0]-[7]. One TxFIFO is active to transfer one sector data and the active SATA channel is switched to next channel for transferring next sector following RAID0 behavior. Before forwarding data, UserFifoRdCnt and TxFifoWrCnt of active channel are monitored to confirm that at least 1 sector data is stored in U2IPFIFO and at least 2-sector free space is available in TxFIFO of active channel. UserFifoRdEn is asserted for 16 clock cycles to transfer 512-byte data continuously.

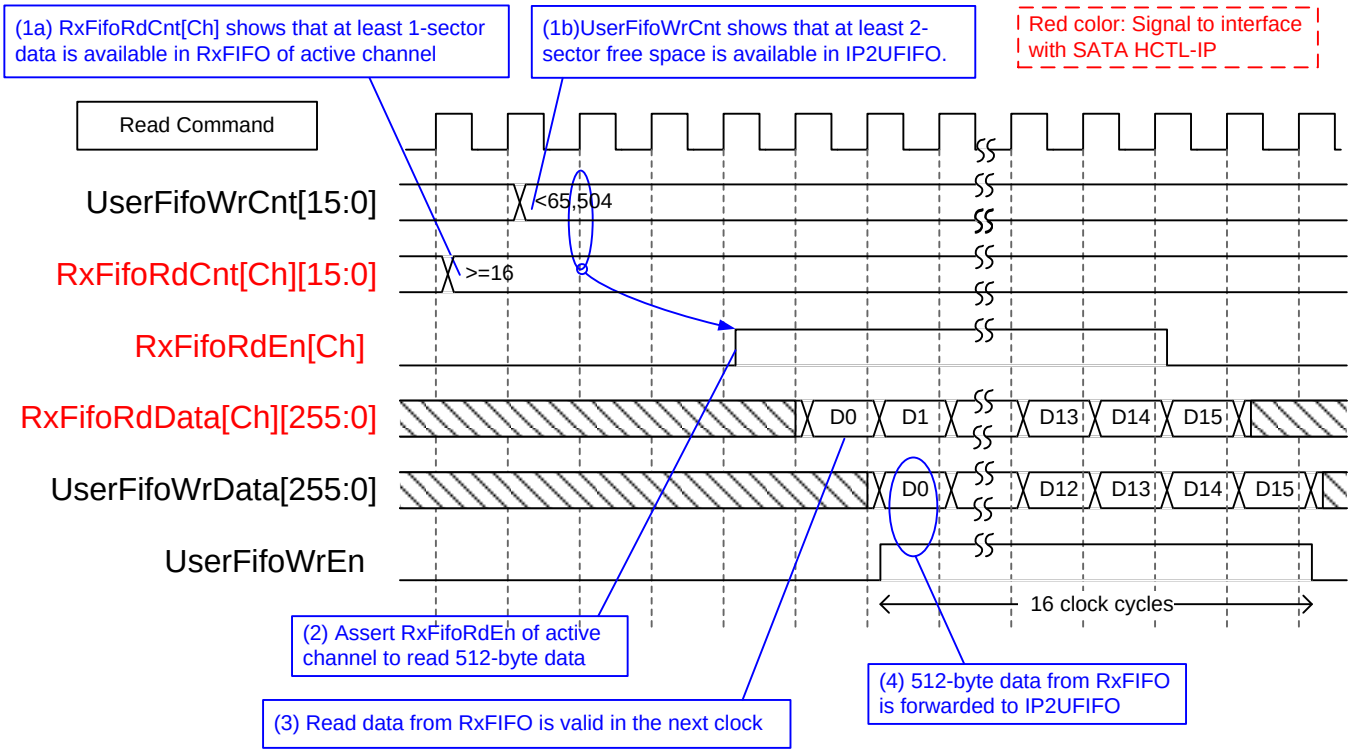

Figure 2-8 Raid0x8 data timing diagram of Read command

When user sends Read command to RAID, data is forwarded from RxFIFO[0]-[7] to IP2UFIFO, as shown in Figure 2-8. Similar to Write command, one RxFIFO is active for each 512-byte transfer, and then active SATA channel is switched to the next channel for next sector transfer. Before forwarding data, UserFifoWrCnt and RxFifoRdCnt of active channel are monitored to confirm that at least 1 sector data is available in RxFIFO of active channel and at least 2-sector free space is available in IP2UFIFO. UserFifoWrEn is asserted for 16 clock cycles to transfer 512-byte data continuously.

2.3 CPU and Peripherals

The hardware is connected to CPU through AXI4-Lite bus, similar to other CPU peripherals. The hardware registers are mapped to CPU memory address, as shown in Table 2-2. The control and status registers for CPU access are designed in LAxi2Reg.

LAxi2Reg connects to many hardwares in the system such as TestGen, Raid0x8 and IdenRAM to get the control and status signals of each module. As shown in Figure 2-9, there are two clock domains applied in this block, i.e. CpuClk which is used to interface with AXI4-Lite bus of CPU and UserClk which is user clock domain for TestGen and RAID0 block.

AsyncAxiReg includes asynchronous circuit between CpuClk and UserClk. More details of each hardware are described as follows.

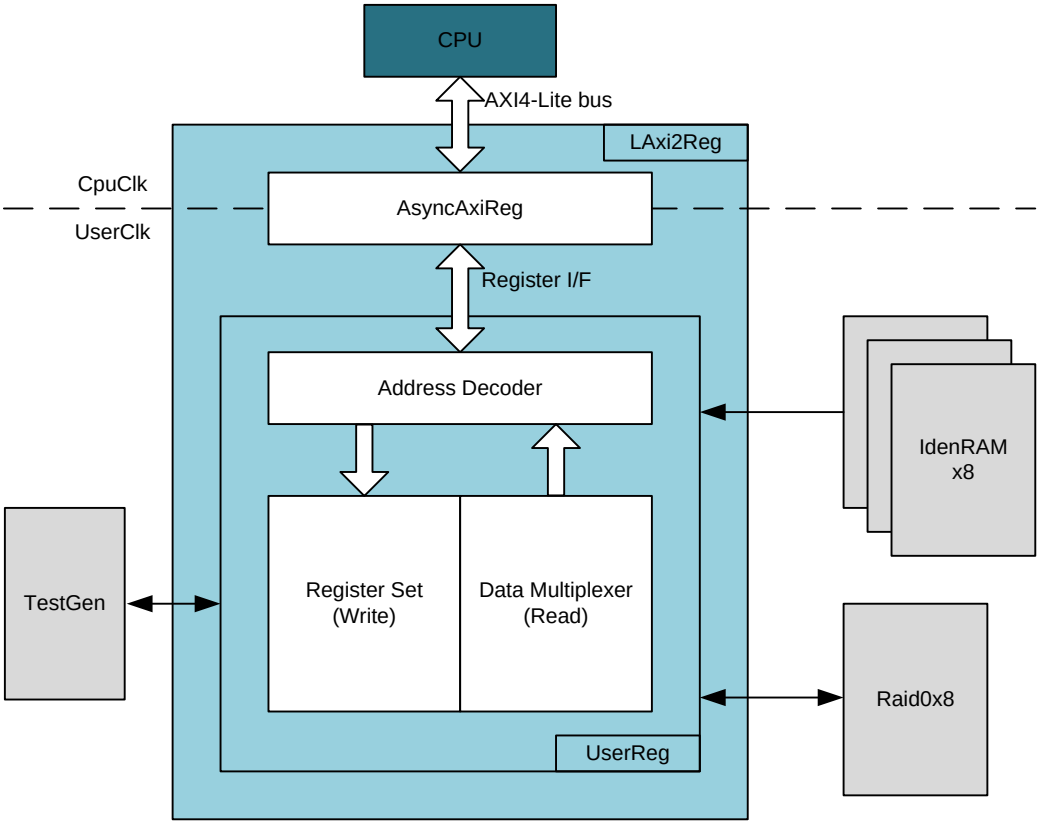

Figure 2-9 CPU and peripherals hardware

2.3.1 AsyncAxiReg

This module is designed to convert the signal interface of AXI4-Lite to be register interface. Also, it supports to convert clock domain from CpuClk to be UserClk domain. Timing diagram of register interface is shown in Figure 2-10.

To write register, timing diagram is same as RAM interface. RegWrEn is asserted to '1' with the valid signal of RegAddr (Register address in 32-bit unit), RegWrData (write data of the register), and RegWrByteEn (the byte enable of this access: bit[0] is write enable for RegWrData[7:0], bit[1] is used for RegWrData[15:8], …, and bit[3] is used for RegWrData[31:24]).

To read register, AsyncAxiReg asserts RegRdReq='1' with the valid value of RegAddr (the register address in 32-bit unit). After that, the module waits until RegRdValid is asserted to '1' to get the read data through RegRdData signal.

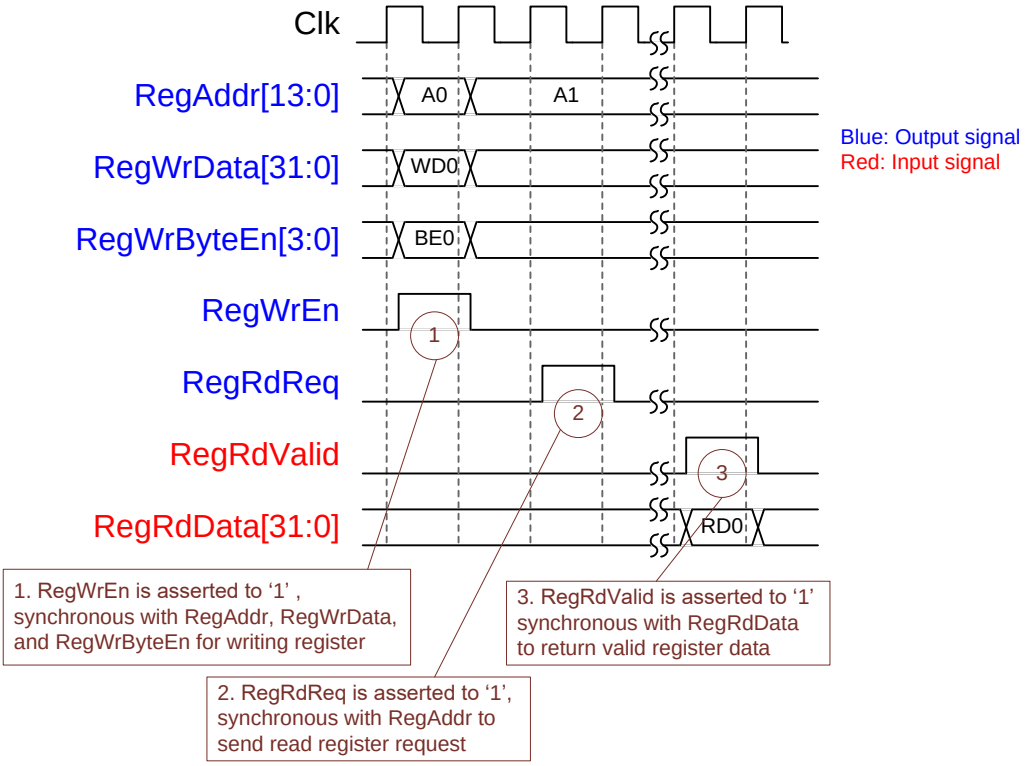

Figure 2-10 Register interface timing diagram

2.3.2 UserReg

As shown in Figure 2-9, after RegWrEn or RegRdReq is asserted to '1' to request write or read register access, RegAddr is loaded to Address decoder to select the active register. For write register, RegWrData signal is latched to be the new value for the request register. In this module, RegWrByteEn is not used, so CPU firmware needs to access the hardware register by using 32-bit pointer only.

For read request, there are many status signals for CPU access from TestGen, Raid0x8, and IdenRAM. So, data multiplexer includes many pipelines register to select the read data to return to CPU. RegRdValid is designed by using two D Flip-flops, input by RegRdReq signal. So, the read access has two clock cycles latency after detecting RegRdReq signal.

Memory map of control and status signals inside UserReg module is shown in Table 2-2.

Table 2-2 Register Map

Address Rd/Wr	Register Name (Label in the “hsataraidtest.c”)	Description
BA+0x000 Wr	User Address (Low) Reg (USRADRL_REG)	[31:0]: Input to be start sector address (UserAddr[31:0] of dgIF typeS for RAID0)
BA+0x004 Wr	User Address (High) Reg (USRADRH_REG)	[15:0]: Input to be start sector address (UserAddr[47:32] of dgIF typeS for RAID0)
BA+0x008 Wr	User Length (Low) Reg (USRLENL_REG)	[31:0]: Input to be transfer length in sector unit (UserLen[31:0] of dgIF typeS for RAID0)
BA+0x00C Wr	User Length (High) Reg (USRLENH_REG)	[15:0]: Input to be transfer length in sector unit (UserLen[47:32] of dgIF typeS for RAID0)
BA+0x010 Wr	User Command Reg (USRCMD_REG)	[1:0]: Input to be user command (UserCmd of dgIF typeS for RAID0) “00”-Identify device, “01”-Security erase, “10”-Write Dev, “11”-Read Dev. When this register is written, the design generates command request to RAID0 to start new command operation.
BA+0x014 Wr	Test Pattern Reg (PATTSEL_REG)	[2:0]: Test pattern select “000”-Increment, “001”-Decrement, “010”-All 0, “011”-All 1, “100”-LFSR
BA+0x100 Rd	User Status Reg (USRSTS_REG)	[0]: UserBusy of dgIF typeS for RAID0 (‘0’: Idle, ‘1’: Busy) [1]: UserError of dgIF typeS for RAID0 (‘0’: Normal, ‘1’: Error) [2]: Data verification fail (‘0’: Normal, ‘1’: Error) [4:3]: SATA speed from IP “00”: No linkup, “11”: SATA Gen3, Others: Not supported [15:8] Linkup flag, directly mapped from each SATA-IP (bit[8]-IP#0, [9]-IP#1, … , [15]-IP#7).
BA+0x104 Rd	Total device size (Low) Reg (LBASIZEL_REG)	[31:0]: Total capacity of RAID0 in sector unit (LBASize[31:0] of dgIF typeS for RAID0)
BA+0x108 Rd	Total device size (High) Reg (LBASIZEH_REG)	[15:0]: Total capacity of RAID0 in sector unit (LBASize[47:32] of dgIF typeS for RAID0)
BA+0x180 Rd	User Error Type 0-3 Reg (USRERRTYPE0_REG)	[31:0]: User error status, directly mapped from UserErrorType[7:0] of HCTL-IP#0 - #3 (bit[7:0]: IP#0, bit[15:8]: IP#1, …, bit[31:24]: IP#3).
BA+0x184 Rd	User Error Type 4-7 Reg (USRERRTYPE1_REG)	[31:0]: User error status, directly mapped from UserErrorType[7:0] of HCTL-IP#4 - #7 (bit[7:0]: IP#4, bit[15:8]: IP#5, …, bit[31:24]: IP#7).
BA+0x1C0 Rd	Test pin of SATA HCTL-IP#0 Reg (TESTPIN0_REG)	[31:0]: TestPin[0][31:0] of RAID0x8 which is direct mapped from TestPin of SATA HCTL-IP#0
BA+0x1C4 Rd	Test pin of SATA HCTL-IP#1 Reg (TESTPIN1_REG)	[31:0]: TestPin[1][31:0] of RAID0x8 which is direct mapped from TestPin of SATA HCTL-IP#1
BA+0x1C8 Rd	Test pin of SATA HCTL-IP#2 Reg (TESTPIN2_REG)	[31:0]: TestPin[2][31:0] of RAID0x8 which is direct mapped from TestPin of SATA HCTL-IP#2
BA+0x1CC Rd	Test pin of SATA HCTL-IP#3 Reg (TESTPIN3_REG)	[31:0]: TestPin[3][31:0] of RAID0x8 which is direct mapped from TestPin of SATA HCTL-IP#3
BA+0x1D0 Rd	Test pin of SATA HCTL-IP#4 Reg (TESTPIN4_REG)	[31:0]: TestPin[4][31:0] of RAID0x8 which is direct mapped from TestPin of SATA HCTL-IP#4
BA+0x1D4 Rd	Test pin of SATA HCTL-IP#5 Reg (TESTPIN5_REG)	[31:0]: TestPin[5][31:0] of RAID0x8 which is direct mapped from TestPin of SATA HCTL-IP#5
BA+0x1D8 Rd	Test pin of SATA HCTL-IP#6 Reg (TESTPIN6_REG)	[31:0]: TestPin[6][31:0] of RAID0x8 which is direct mapped from TestPin of SATA HCTL-IP#6
BA+0x1DC Rd	Test pin of SATA HCTL-IP#7 Reg (TESTPIN7_REG)	[31:0]: TestPin[7][31:0] of RAID0x8 which is direct mapped from TestPin of SATA HCTL-IP#7

Address Rd/Wr	Register Name (Label in the “hsataraidtest.c”)	Description
BA+0x200 Rd	Data Failure Address (Low) Reg (RDFAILNOL_REG)	[31:0]: Latch value of failure address [31:0] in byte unit from Read command
BA+0x204 Rd	Data Failure Address (High) Reg (RDFAILNOH_REG)	[24:0]: Latch value of failure address [56:32] in byte unit from Read command
BA+0x240 Rd	Expected value Word0 Reg (EXPPATW0_REG)	[31:0]: Latch value of expected data [31:0] in TestGen during Read command
BA+0x244 Rd	Expected value Word1 Reg (EXPPATW1_REG)	[31:0]: Latch value of expected data [63:32] in TestGen during Read command
BA+0x248 Rd	Expected value Word2 Reg (EXPPATW2_REG)	[31:0]: Latch value of expected data [95:64] in TestGen during Read command
BA+0x24C Rd	Expected value Word3 Reg (EXPPATW3_REG)	[31:0]: Latch value of expected data [127:96] in TestGen during Read command
BA+0x250 Rd	Expected value Word4 Reg (EXPPATW4_REG)	[31:0]: Latch value of expected data [159:128] in TestGen during Read command
BA+0x254 Rd	Expected value Word5 Reg (EXPPATW5_REG)	[31:0]: Latch value of expected data [191:160] in TestGen during Read command
BA+0x258 Rd	Expected value Word6 Reg (EXPPATW6_REG)	[31:0]: Latch value of expected data [223:192] in TestGen during Read command
BA+0x25C Rd	Expected value Word7 Reg (EXPPATW7_REG)	[31:0]: Latch value of expected data [255:224] in TestGen during Read command
BA+0x280 Rd	Read value Word0 Reg (RDPATW0_REG)	[31:0]: Latch value of read data [31:0] in TestGen during Read command
BA+0x284 Rd	Read value Word1 Reg (RDPATW1_REG)	[31:0]: Latch value of read data [63:32] in TestGen during Read command
BA+0x288 Rd	Read value Word2 Reg (RDPATW2_REG)	[31:0]: Latch value of read data [95:64] in TestGen during Read command
BA+0x28C Rd	Read value Word3 Reg (RDPATW3_REG)	[31:0]: Latch value of read data [127:96] in TestGen during Read command
BA+0x290 Rd	Read value Word4 Reg (RDPATW4_REG)	[31:0]: Latch value of read data [159:128] in TestGen during Read command
BA+0x294 Rd	Read value Word5 Reg (RDPATW5_REG)	[31:0]: Latch value of read data [191:160] in TestGen during Read command
BA+0x298 Rd	Read value Word6 Reg (RDPATW6_REG)	[31:0]: Latch value of read data [223:192] in TestGen during Read command
BA+0x29C Rd	Read value Word7 Reg (RDPATW7_REG)	[31:0]: Latch value of read data [255:224] in TestGen during Read command
BA+0x2C0 Rd	Current test byte (Low) Reg (CURTESTSIZEL_REG)	[31:0]: Current test data size of TestGen module in byte unit (bit[31:0])
BA+0x2C4 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 – 0x2FFF Rd	Identify Device Command Data (IDENCTRL_REG)	[511:0]: 512-byte Identify device data from SATA CH#0 [1023:512]: 512-byte Identify device data from SATA CH#1 [1535:1024]: 512-byte Identify device data from SATA CH#2 [2047:1536]: 512-byte Identify device data from SATA CH#3 [2559:2048]: 512-byte Identify device data from SATA CH#4 [3071:2560]: 512-byte Identify device data from SATA CH#5 [3583:3072]: 512-byte Identify device data from SATA CH#6 [4095:3584]: 512-byte Identify device data from SATA CH#7

3 CPU Firmware

After system boot-up, CPU initializes its peripherals such as UART and Timer. Next, CPU monitors link up signal of all SATA devices by comparing USRSTS_REG[15:8]=0xFF. Main menu is displayed after all SATA devices link up.

CPU firmware supports four commands, following USRCMD_REG value, i.e. “00” for Identify device, “01” for Security erase, “10” for Write, and “11” for Read. More details of the sequence in each command are described as follows.

3.1 Identify device command

The sequence of the firmware when user selects Identify device command is below.

1) Set USRCMD_REG=”00”. Next, RAID0 sends Identify device command to all SATA HCTL-IPs. RAID0 busy flag (USRSTS_REG[0]) changes from ‘0’ to ‘1’.
2) CPU waits until the operation is completed or some errors are found by monitoring USRSTS_REG value. Bit[0] is de-asserted to ‘0’ when command is completed. Bit[1] is asserted to ‘1’ when some errors are detected. In case of error condition, there is error message displayed on the console. If the command is completed, the data from Identify device command of all SATA devices will be stored in IdenRAMs.
3) CPU reads Identify device data from IdenRAMs which are mapped to IDENCTRL_REG address. Then, CPU displays the information such as SATA device model name, security feature set supported, and erase time value on Serial console. Also, RAID0 device capacity (LBASIZEL/H_REG) is displayed in GB unit on the console.

3.2 Write/Read command

The sequence of the firmware when user selects Write/Read command is below.

1) Receive start address, transfer length, test pattern through Serial console. 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, PATTSEL_REG, and USRCMD_REG (USRCMD_REG=”10” for write transfer, and “11” for read transfer).
3) CPU waits until the operation is completed or some errors (except verification error) are found by monitoring USRSTS_REG[2:0].
   If USRSTS_REG[2] is asserted to ‘1’, verification error message will be displayed. After that, CPU still runs until end of operation or user inputs any key to cancel operation.
4) During running command, current transfer size reading from CURTESTSIZE_REG is displayed every second. Finally, test performance is displayed on Serial console when command is completed.

3.3 Security erase command

The sequence of the firmware when user selects Security erase command is below.

1) Set USRCMD_REG=”01”. Next, RAID0 sends Security erase command to all SATA HCTL-IPs. RAID0 busy flag (USRSTS_REG[0]) changes from ‘0’ to ‘1’.
2) CPU waits until command complete by monitoring USRSTS_REG value. Bit[0] is de-asserted 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 is displayed on Serial console when command is completed.

4 Example Test Result

The example test result when running demo system by using 8x1 TB Samsung 850 Pro is shown in Figure 4-1.

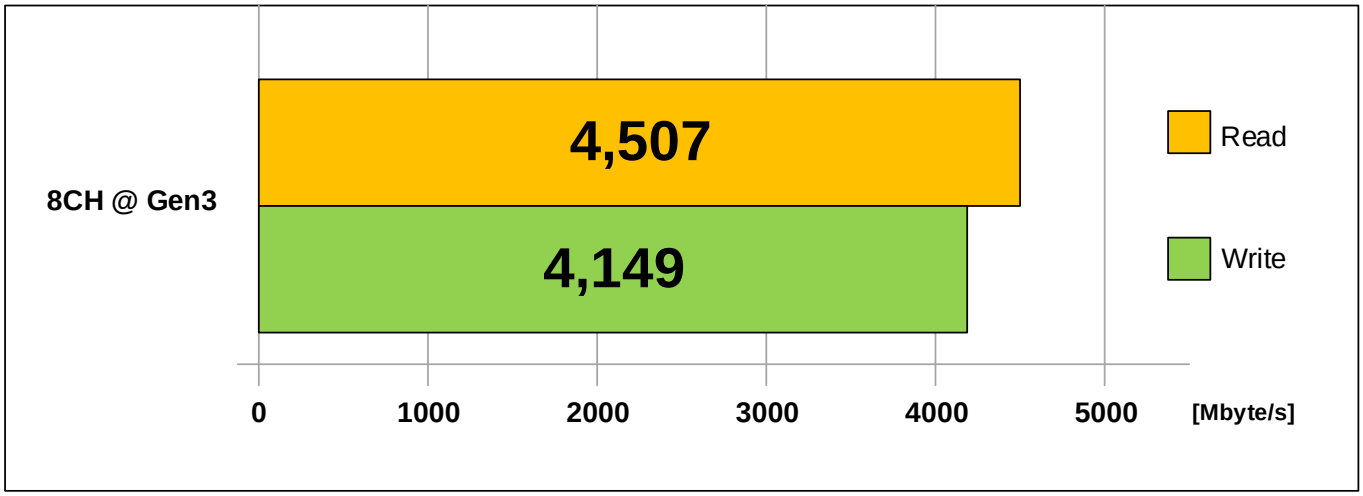

Figure 4-1 Test Performance of RAID0x8 IP demo by using Samsung 850 Pro SSD

When running 8-ch RAID0 with 8 SATA Gen3 SSDs on ZCU102 board, write performance is about 4100 Mbyte/sec and read performance is about 4500 Mbyte/sec.

5 Revision History

Revision	Date	Description
1.0	13-Mar-18	Initial version release