

SATA-IP Host reference design manual

Rev1.4 02-Jun-09

1. Introduction

Serial ATA (SATA) is an evolutionary replacement for the Parallel ATA (PATA) physical storage interface. SATA interface increases speed transfer to be 1.5 Gbps for SATA-I and 3.0 Gbps for SATA-II. To communication by SATA protocol, there are four layers in its architecture, i.e. Application, Transport, Link, and Phy.

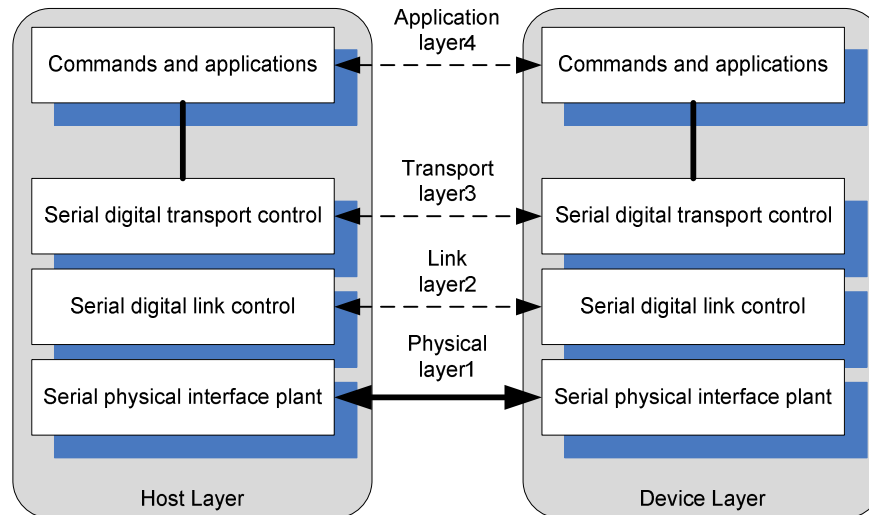


Figure1 SATA Communication Layer

The Application layer is responsible for overall ATA command execution, including controlling Command Block Register accessed. The Transport layer is responsible for placing control information and data to be transferred between the host and device in a packet/frame, known as a Frame Information Structure (FIS). The Link layer is responsible for taking data from the constructed frames, encoding or decoding each byte using 8b/10b, and inserting control characters such that the 10-bit stream of data may be decoded correctly. The Physical layer is responsible for transmitting and receiving the encoded information as a serial data stream on the wire.

This reference design provides evaluation system which implements all SATA communication layers for Host side to transfer high speed data with SATA Hard Disk Drive. The SATA-IP core is designed to operate with GTP transceiver of the Virtex-5 platform and this SATA-IP reference design is implemented on ML506/505 Evaluation board. More details are described as follows.

2. Environment

This reference design is based on the following environment as shown in Figure2.

- ML506 Platform (For ML505, changing only target device)
- ISE 10.1.03 / EDK 10.1.03
- SATA Peripheral (SATA HDD) connect SATA cable to J40 on ML506
- Serial (RS232C) communication, connect RS232C cable to P3 on ML506 (Set baud rate=115,200 / data=8bit / Non-Parity / Stop=1bit)

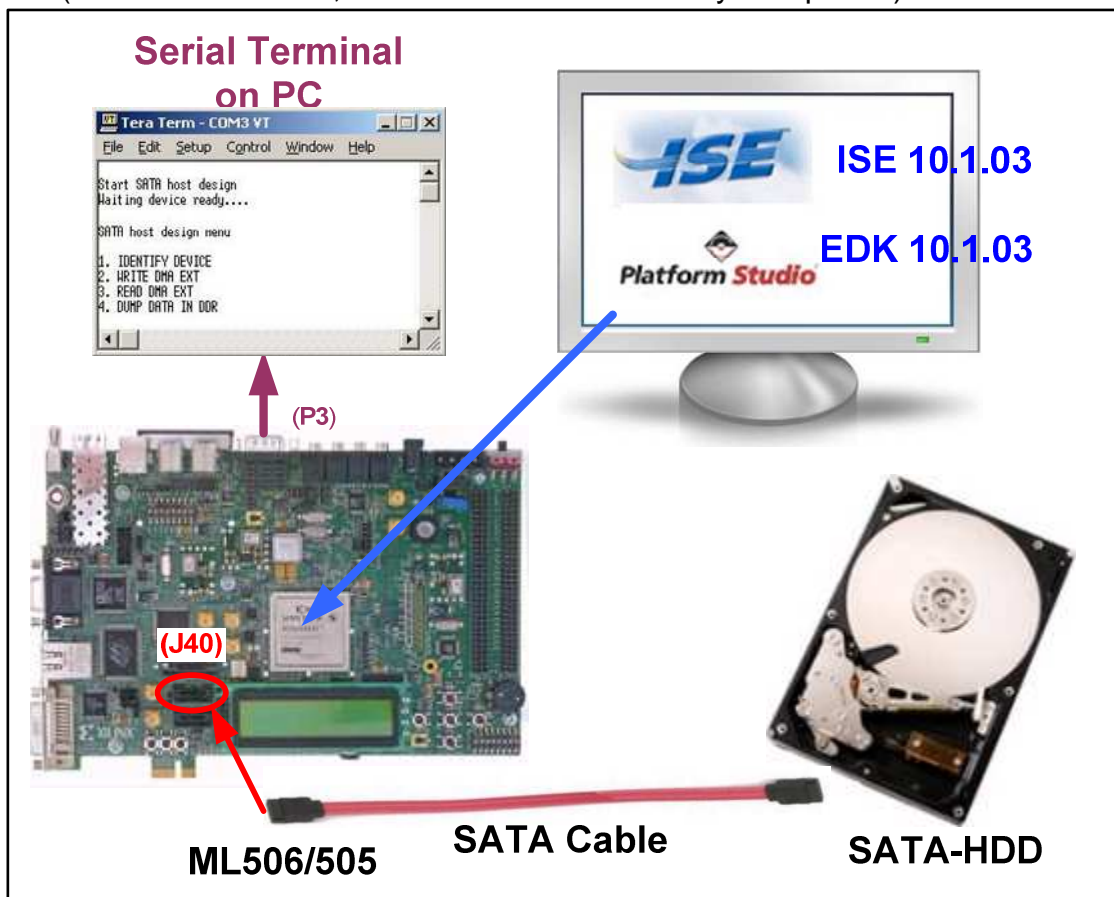


Figure2 Reference design environment

Refer to “SATA_IP Host Demo Instruction” for operation procedure of this reference design. For evaluation version, IP-Core has 1-hour time limitation to use. After 1-hour use, IP-core will stop any data transfer.

3. Hardware description

- SATA IP Host design implementation on Virtex5 FPGA

As shown in Figure3, SATA IP consists of only Link Layer and some part of Transport Layer, so that users need to prepare other Layer such as PHY Layer and Transport Layer by themselves. This reference design describes Transport Layer and PHY Layer example based on ML506 Evaluation board from Xilinx.

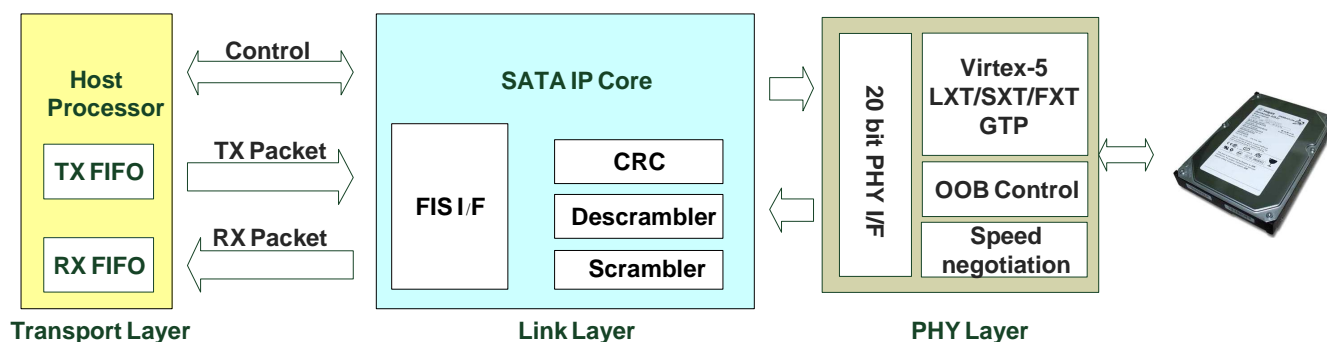


Figure3 Connection between Link Layer and Transport/PHY Layer

- PHY Layer

Virtex-5 LXT/SXT/FXT has a built-in high-speed serial circuit in its GTP block, and PHY Layer of SATA will be implemented with this GTP block. PHY Layer includes OOB (Out-of-Band) Control, Automatic Speed Negotiation logic to support SATAII/I, and 20bit PHY interface for Link Layer communication.

This reference design is modified from Xilinx application note (XAPP870), available on Xilinx website, so that PHY Layer is also optimized with GTP resource usage of Virtex5. SATA characteristic report of rpt087 is also available on Xilinx website.

Before building user board, user must read carefully and must follow design guide line described in UG196 (Virtex-5 FPGA RocketIO GTP Transceiver User Guide).

PHY Layer circuit source code in this reference design is stored in “sata2phy_ml505.v” that also includes “oob_control.v” and “speed_neg_control.v” module. “sata2phy_ml505.v” is modified to support three modes, i.e. Fixed-SATAI, Fixed-SATAII, and Auto-negotiation. One of three modes is selecting by define one of three parameters in line 30-32 of “sata2phy_ml505.v” file, as shown in Figure4. Default mode in reference design is Auto-negotiation.

```

24:
25: // Select
26: // - FIXI    for Fixed-SATAI speed PHY design
27: // - FIXII   for Fixed-SATAII speed PHY design
28: // - AUTONEG for Auto speed negotiation PHY design
29:
30: //`define FIXI
31: //`define FIXII
32: `define AUTONEG // Auto speed negotiation
33:
34: module sata2phy_ml505 #
35: (

```

Figure4 Header in “sata2phy_ml505” for selecting SATA PHY function

dg_sata_ip_refdesign_host_en.doc

SATA connector (J40) with auto-negotiation function can be connected both SATA-I and SATA-II HDD, but only SATA-I/SATA-II HDD can be connected for Fixed-SATAI/Fixed-SATAII function. Auto-negotiation mode uses shared PLL reset pin in Virtex-5 GTP which will be effect to another HDD connected at SATA connector (J41), so one HDD in each GTP_Tile of Virtex-5 is recommended for user design instead of two HDDs. Shared PLL reset is not required in Fixed-SATAII or Fixed-SATAI mode, so two HDDs can be connected with two SATA connectors (J40/J41) in each GTP_Tile, especially in RAID application. Another advantage for Fixed SATA-I and Fixed SATA-II mode is only one DCM required in PHY design, but two DCMs are required for auto-negotiation mode.

Note: Changing PHY parameter is required to re-implement reference design hardware, so this features is permitted only production version.

- Transport Layer

User need to build Transport Layer by themselves because this Layer structure depends on hardware architecture and user application. Transport Layer in this reference design uses NPI interface to operate with MicroBlaze. Similar to typical SATA controller, this reference design will build FIS data on the main memory and communicate with Link Layer by DMA mechanism.

Transport Layer circuit source code in this reference design is stored in “npi_sata.vhd” that also includes SATA IP Core and PHY Layer instance.

- MPMC interface for flexible and RAID design

This reference design uses MicroBlaze as a host processor, and uses MPMC (Multi-port memory controller) as main memory controller. MPMC can support up to 8 channels for memory access port, and each port can be set to PLB (Processor Local Bus) for PowerPC/MicroBlaze connection or NPI (Native Port Interface) for user logic connection. For more detail of MicroBlaze, NPI, or MPMC, please refer to Xilinx technical document.

This reference design directly connects SATA IP core with NPI port of MPMC to execute DMA transfer to/from the main memory. Since this design has already built as a peripheral IP Core of EDK (Embedded Development Kit), it is easily change the number of SATA channel by setting the number of NPI port on EDK to support RAID system. For evaluation version, hardware design cannot re-implement and only software on EDK can be modified. So, additional design for multiple SATA channel application such as RAID system is permitted to implement by Xilinx software for production version.

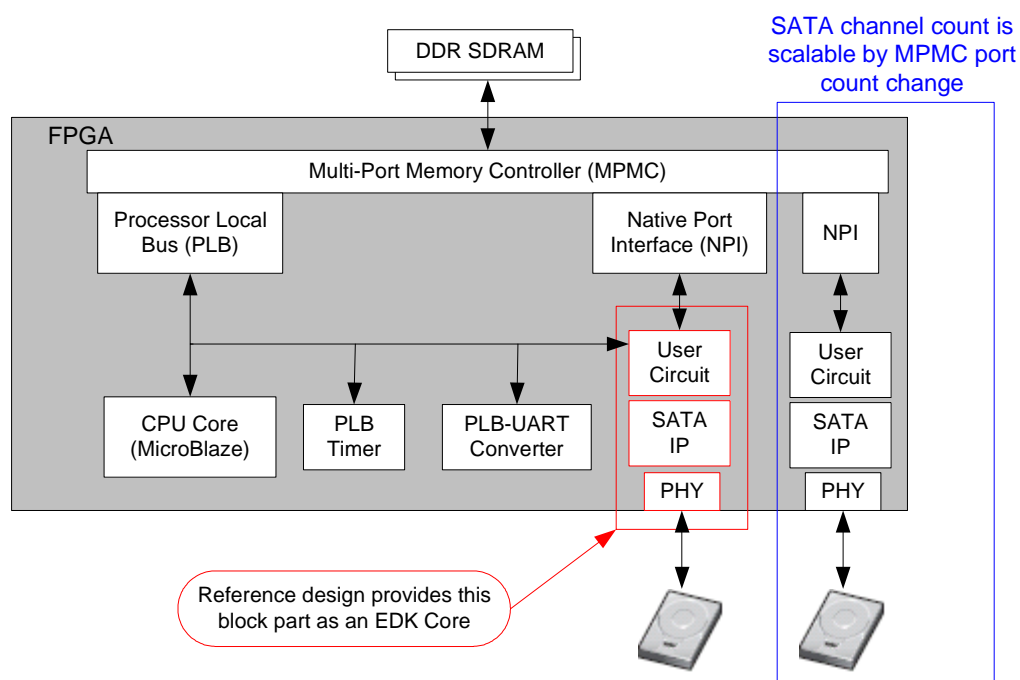


Figure5 Block diagram of the reference design

● Build connection circuit between Memory controller and IP Core

Interface signal of SATA IP core is shown in Table 1. Transport Layer interface signals are separated into two groups, i.e. transmit and receive. Transport signal waveform of data transmit transaction and data receive transaction is shown in Figure6 and 7 respectively. Figure8 shows detailed block diagram of logic connection. Table 2 is a register mapping of MicroBlaze side.

Software on MicroBlaze along with DMA mechanism will transmit FIS data on the main memory to the Link Layer of SATA IP, or will received data from the Link Layer to the main memory. During receiving data from the Link Layer, control logic circuit will transfer Data FIS to the different address space for different FIS type by checking Data FIS header information. Also, during transmitting data, control logic circuit will automatically add Data FIS header at the top of the data packet. With this mechanism, user logic do not need to mind about header information control for data management.

Signal	Signal Direction	Description
Common Interface Signal		
trn_reset	In	Reset SATA IP core. Active high.
trn_link_up	Out	Transaction link up is asserted when the core establish the communication with SATA PHY.
trn_clk	In	Clock which is synchronized with trn_XXX signal for interface with the Host. There is no global clock buffer inside SATA IP core for this signal, so external global clock buffer should be inserted. This clock frequency is required to be higher than core_clk frequency.
core_clk	In	IP Core operating frequency output (37.50MHz for SATA-I, 75.00MHz for SATA-II). This clock is generated from SATA PHY.
dev_host_n	In	Device or Host design assignment. '0': ATA Host IP Core, '1': ATA Device IP Core (Use '0' for the host reference design)
Transmit Transaction Interface		
trn_tsof_n	In	Transmit Start-Of-Frame (SOF): Indicate start each SATA FIS packet. Active low.
Trn_teof_n	In	Transmit End-Of-Frame (EOF): Indicate end each SATA FIS packet. Active low.
Trn_td[31:0]	In	Transmit Data: SATA FIS packet data to be transmitted.
Trn_tsrc_rdy_n	In	Transmit Source Ready: Indicates that trn_td[31:0] from the Host is valid. Active low.
Trn_tdst_rdy_n	Out	Transmit Destination Ready: Indicate that the core is ready to accept data on trn_td[31:0]. Active low. - trn_tsrc_rdy_n must be de-asserted within 4 period of trn_clk after trn_tdst_rdy_n is de-asserted. So the core can accept 4 DWORD of trn_td[31:0] after trn_tdst_rdy_n is de-asserted.
Trn_tsrc_dsc_n	In	Transmit Source Abort: Indicates that the Host cancel the current SATA FIS packet. May be asserted any time between SOF and EOF. Active low. - Once asserted, the Core will send out SYNC primitive to abort the current transfer
trn_tdst_dsc_n	Out	Transmit Destination Abort: Indicates that the core is aborting the current SATA FIS packet. Asserted when the physical link is going into reset. Active low.

Table1 SATA IP interface signal definition

Signal	Signal Direction	Description
Receive Transaction Interface		
trn_rsof_n	Out	Receive Start-Of-Frame (SOF): Indicate start each SATA FIS packet. Active low.
Trn_reof_n	Out	Receive End-Of-Frame (EOF): Indicate end each SATA FIS packet. Active low.
Trn_rd[31:0]	Out	Receive Data: SATA FIS packet data to be transmitted.
Trn_rsrc_rdy_n	Out	Receive Source Ready: Indicates that trn_rd[31:0] from the core is valid. Active low.
Trn_rdst_rdy_n	In	Receive Destination Ready: Indicate that the Host is ready to accept data on trn_rd[31:0]. Active low. - trn_rsrc_rdy_n will be de-asserted within 4 period of trn_clk after trn_rdst_rdy_n is de-asserted. So Host should be supported to accept 4 DWORD of trn_rd[31:0] after trn_rdst_rdy_n is de-asserted.
Trn_rsrc_dsc_n	Out	Receive Source Abort: Indicates that the core cancel the current SATA FIS packet. May be asserted any time between SOF and EOF. Active low.
Trn_rdst_dsc_n	In	Receive Destination Abort: Indicates that the Host cancel the current SATA FIS packet. Active low. - Once asserted, the core will send out SYNC primitive to abort the current transfer
SATA PHY Interface for Virtex5 GTP		
PHYRESET	In	SATA PHY reset. Active high. This signal is used to reset data buffer interface with SATA PHY when PHY is reset.
PHYCLK	In	Reference Clock for 16-bit SATA PHY (Virtex5 GTP) - 75MHz for SATA-I - 150MHz for SATA-II This clock is generated from DCM inside SATA PHY. It's used for both both TX and RX data by elastic buffer in GTP of SATA PHY.
TXDATA[15:0]	Out	16-bit transmit data from the core to the GTP
TXDATAK[1:0]	Out	2-bit Data/Control for the symbols of transmitted data. ("00": data byte, "01": control byte, "1X": undefined).
RXDATA[15:0]	In	16-bit receive data from the GTP to the core.
RXDATAK[1:0]	In	2-bit Data/Control for the symbols of received data. ("00": data byte, "01": control byte, "1X": undefined)
RXDATAVALID	In	Indicates PHY lock and valid data on RXDATA and RXDATAK.
LINKUP	In	Indicates that SATA link communication is established. Active high.
PLLLOCK	In	Indicates that DCM of GTP is locked. Active high.

Table1 SATA IP interface signal definition (Cont'd)

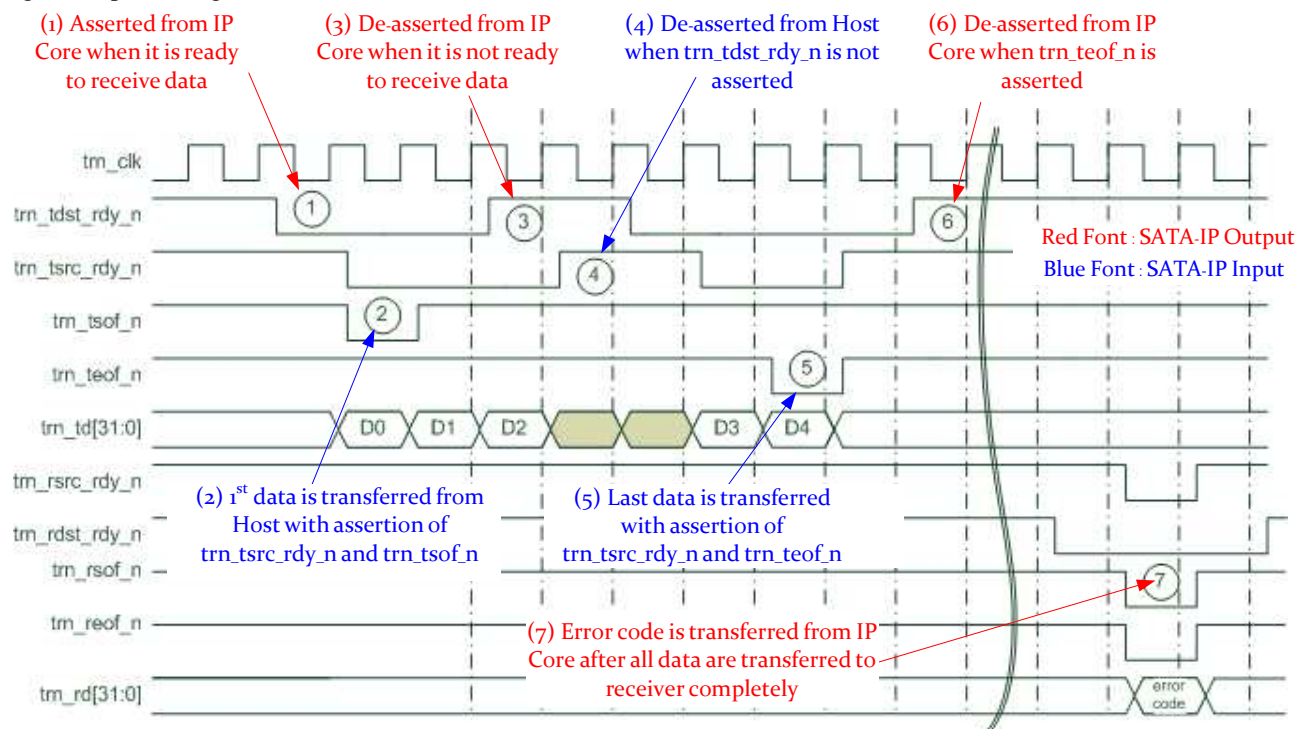


Figure6 Waveform of data transmit transaction

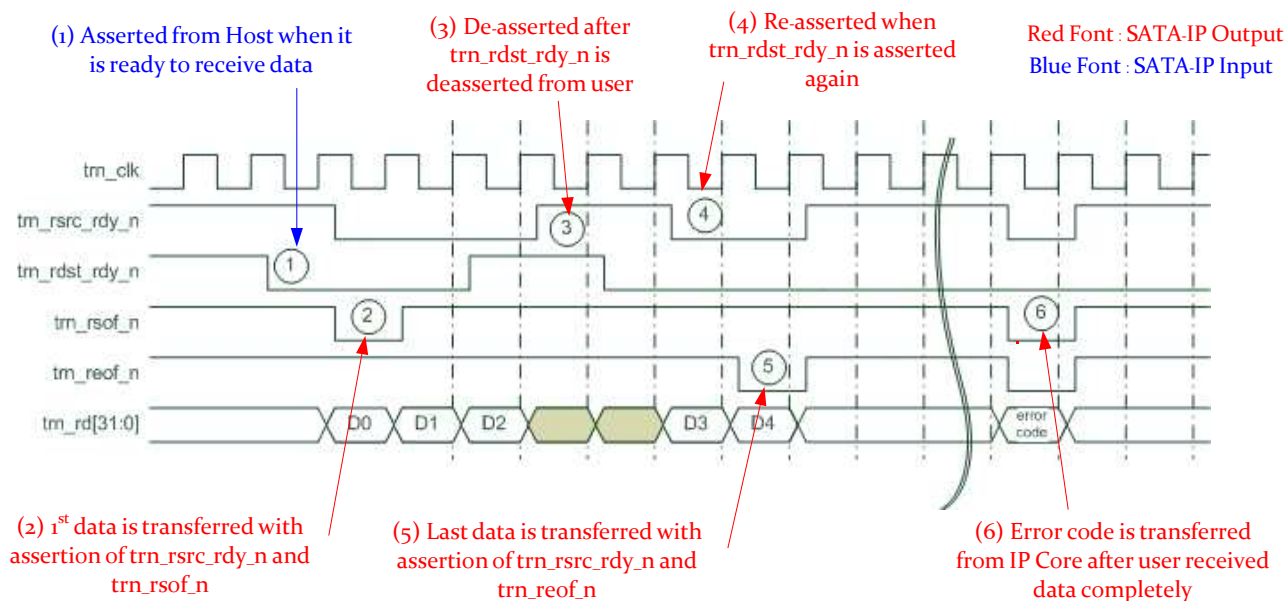


Figure7 Waveform of data receive transaction

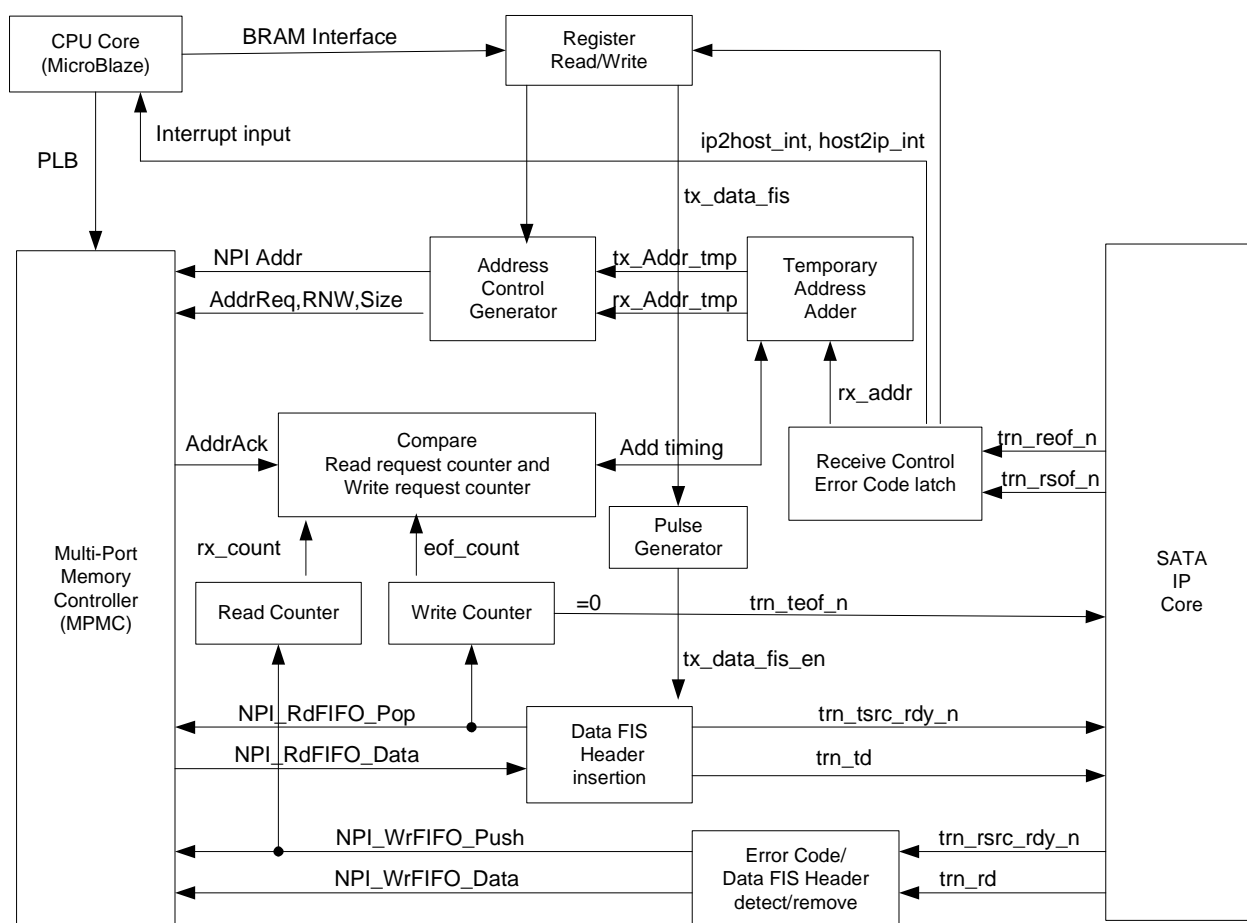


Figure8 Block diagram of logic connection

Address Rd/Wr	Register Name (Label in the "sata_host.c")	Description (Bit order is little endian)
BA+0x00 Rd	Status Reg. (STATUS)	[8]MPMC Ready [7:4]OOB status data [3]GTP PLL lock status [1]Gen2 Link OK [0]SATA IP Link OK
BA+0x04 Rd	Error Code Reg. (ERROR_CODE)	SATA IP Error code. Set when transmit/receive completion. User can detect CRC or FIS error.
BA+0x08 Rd	Interrupt Clear Reg. (INT_CLEAR)	Clear interrupt by read operation at this register
BA+0x0C Rd	Receive Word Count Reg. (RX_COUNT)	Received word count, the sum of all FIS data count until clear.
BA+0x00 Wr	Transmit Data Address Reg. (TX_ADDR)	Set top address of transmit data storage area.
BA+0x04 Wr	Received Data Address1 Reg. (RX_ADDR)	Set top address of received data area except Data FIS. Top address[7:0] needs to be equal to 0.
BA+0x08 Wr	Control Reg. (CONTROL)	[31]SATA Reset [30]Transmit Request [29]Send Data FIS [15:0]Transmit word count. Write operation will reset RX_COUNT register
BA+0x0C Wr	Received Data Address2 Reg. (RX2_ADDR)	Set top address of received Data FIS.

(BA : Base Address)

Table2 Register mapping from CPU side

4. Software description

- SATA Device access via FIS

Communication between the Host and the Device via SATA is done by FIS (Frame Information Structure) data structure. MicroBlaze in the Host design will build FIS data structure on its main memory space, and will send it to the Device by DMA controller that operates bus master. And FIS data sent from the Device is also transferred on the main memory by DMA controller.

Thus, MicroBlaze will execute access to the SATA Device by the following sequence.

- 1- Build FIS Data structure (First FIS command should be RegH2D FIS)
- (2) Transmit FIS Data
- (3) Wait to receive FIS Data
- (4) Read received FIS Data
- (5) Additional FIS data transmit/receive if necessary.

FIS transmit and receive counts are different according to the protocol type, but the brief sequence should be as above.

- Software of reference design

Software of this reference design implements three popular commands, i.e. IDENTIFY DEVICE, DMA READ EXT, and DMA WRITE EXT. Target SATA HDD device for this reference design should support 48bit LBA (Logical Block Address) mode.

When SATA Device is powered on, it always sends Register – Device to Host FIS, so Host must wait this FIS from SATA Device before issue first command.

- IDENTIFY DEVICE

Table3 shows FIS structure of IDENTIFY COMMAND that gets device information from SATA Device. This command requires parameter settings for its Command Opcode (ECh) and device number that is typically set to zero in SATA device. Device register value will be A0H because obsolete bit#7 and bit#5 are recommended to set. “C” bit should be set whenever SATA Host sends command to the SATA Device, and it is also the same for all other commands issue.

After finish parameter settings to Register – Host to Device FIS, Host sends it to Link Layer. SATA Device will firstly send PIO Setup FIS, and then send Data FIS that includes device information.

For detailed device information, please refer to the ATA Standard document that can be obtained from <http://www.t13.org/>. This reference design only shows device model number, 48bit LBA support information, and disk capacity information.

0	Features 00h	command ECh	C R R R PM Port 1 0 0 0 0h	FIS Type (27h)
1	Device A0h	LBA High 00h	LBA Mid 00h	LBA Low 00h
2	Features(exp) 00h	LBA High(exp) 00h	LBA Mid(exp) 00h	LBA Low(exp) 00h
3	Control 00h	Reserved(0)	sector Count(exp) 00h	Sector Count 00h
4	Reserved(0)	Reserved(0)	Reserved(0)	Reserved(0)

Table3 IDENTIFY COMMAND FIS structure

● **DMA READ EXT**

Table 4 shows FIS structure of DMA READ EXT that reads data from SATA Device. There are two data transfer types, i.e. PIO and DMA, but their difference is insignificant for SATA case. In SATA Device, speed performance of PIO transfer and DMA transfer are also not so different. Because DMA Read process is easier than PIO, this reference design selects DMA transfer.

Host will set Opcode to 25H, LBA bit (bit#6) of Device register, LBA address, and read sector count to the Register – Host to Device FIS, and then transmit to SATA Device. Device will send read data equal with read sector count setting in Data FIS, and then send Register – Device to Host FIS to finish this command.

0	Features 00h	command 25h	C R R R PM Port 1 0 0 0 0h	FIS Type (27h)
1	Device E0h	LBA High LBA[23:16]	LBA Mid LBA[15:8]	LBA Low LBA[7:0]
2	Features(exp) 00h	LBA High(exp) LBA[47:40]	LBA Mid(exp) LBA[39:32]	LBA Low(exp) LBA[31:24]
3	Control 00h	Reserved(0)	sector Count(exp) sector_count[15:8]	Sector Count sector_count[7:0]
4	Reserved(0)	Reserved(0)	Reserved(0)	Reserved(0)

Table4 DMA READ EXT FIS structure

● **DMA WRITE EXT**

Table 5 shows FIS structure of DMA WRITE EXT that writes data to SATA Device. FIS structure is almost identical to that of DMA READ EXT. Host will set Opcode to 35H, LBA bit, LBA address, write sector count. After sending this Host to Device FIS, Device will send DMA Activate FIS to the Host. Then, Host sends first Data FIS to the Device.

Host will repeat sending Data FIS to the Device until all the data transfer is completed. Finally, Device sends Register- Device to Host FIS to finish this command.

0	Features 00h	command 35h	C R R R PM Port 1 0 0 0 0h	FIS Type (27h)
1	Device E0h	LBA High LBA[23:16]	LBA Mid LBA[15:8]	LBA Low LBA[7:0]
2	Features(exp) 00h	LBA High(exp) LBA[47:40]	LBA Mid(exp) LBA[39:32]	LBA Low(exp) LBA[31:24]
3	Control 00h	Reserved(0)	sector Count(exp) sector_count[15:8]	Sector Count sector_count[7:0]
4	Reserved(0)	Reserved(0)	Reserved(0)	Reserved(0)

Table5 DMA WRITE EXT FIS structure

- Necessary consideration

Host software source code of this design is stored in "sata_host.c". Note that this reference design does not include error check or recovery from illegal/unexpected behavior. So user needs to add such consideration that software should check status or error check when Register – Device to Host FIS is received from the Device.

Figure9 shows reference design operation result on serial terminal screen.

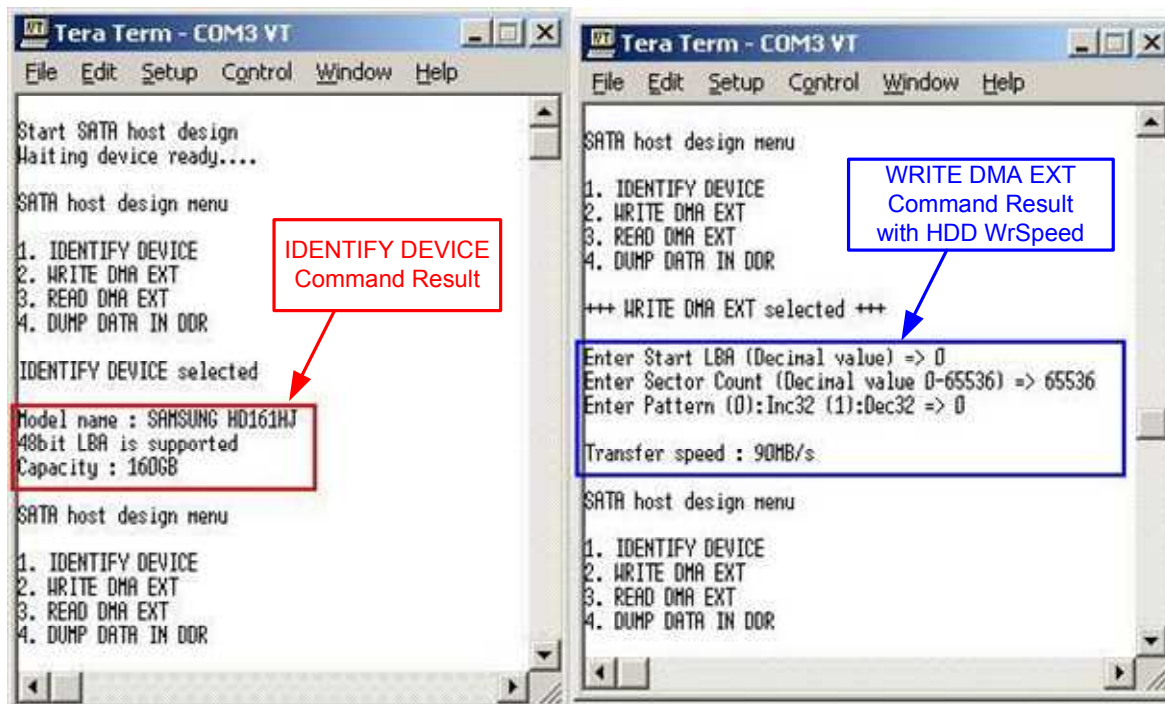


Figure9 Operation result sample screen

5. Revision History

Revision	Date	Description
0.1	22-Sep-08	Initial draft
1.0	29-Oct-08	Initial release including evaluation version description
1.1	10-Nov-08	Added SATA technology introduction description
1.2	14-Nov-08	Auto-negotiation support by using parameter setting
1.3	12-Dec-08	Add dev_host_n signal and update test application menu
1.4	02-Jun-09	Add trn_clk description

Copyright: 2008 Design Gateway Co.,Ltd.