January 16, 2012 Product Specification Rev1.0

Design Gateway Co.,Ltd
54 BB Building 13th Fl., Room No.1302 Sukhumvit 21 Rd. (Asoke), Klongtoey-Nua, Wattana, Bangkok 10110
Phone: (+66) 02-261-2277
Fax: (+66) 02-261-2290
E-mail: sales@design-gateway.com
URL: www.design-gateway.com

Core Facts

Provided with Core	
Documentation	User Guide, Design Guide
Design File Formats	Encrypted hdl File
Verification	Test Bench, Simulation Library
Instantiation Templates	VHDL
Reference Designs & Application Notes	NIOS2 Project, See Reference Design Manual
Additional Items	Demo on Stratix IV GX Development Kit
Simulation Tool Used	
ModelSim-Altera 6.6d	
Support	
Support Provided by Design Gateway Co., Ltd.	

Features

- Compliant with the Serial ATA specification revision 3.0
- Support both of SATA Host and SATA Device (Applicable to SATA Peripheral development)
- Simple transaction interface with Host processor or DMA Engine
- 32-bit internal data path
- 4KB FIFO implemented by Memory block in transmit and receive paths
- RX Elastic buffer to interface PHY
- Support both SATA-III and SATA II Speed
- Low frequency operation
  - IP Core clock 75 MHz for SATA-II and 150 MHz for SATA-III
- CONT primitive support for continue primitive suppression to reduce EMI
- Support 40bit width PHY implemented by Transceiver Block on Startix IV GX device
- Host SATA Reference design available on Stratix IV GX Development Kit (with HSMC SATA board)

Table 1: Example Implementation Statistics

Family	Example Device	Fmax (MHz)	Combinational ALUTs[1]/ Logic Elements	Registers[1]	Pin[2]	Block Memory bit	GXB[3]	Design Tools
Stratix IV GX	EP4SGX230KF40C2	322	1,436	2,310	194	33,792	1	QuartusII 10.1

Notes:
1) Actual logic resource dependent on percentage of unrelated logic
2) Assuming all core I/Os and clocks are routed off-chip
3) GXB is not used in SATA IP core, but they are used in SATA PHY design.

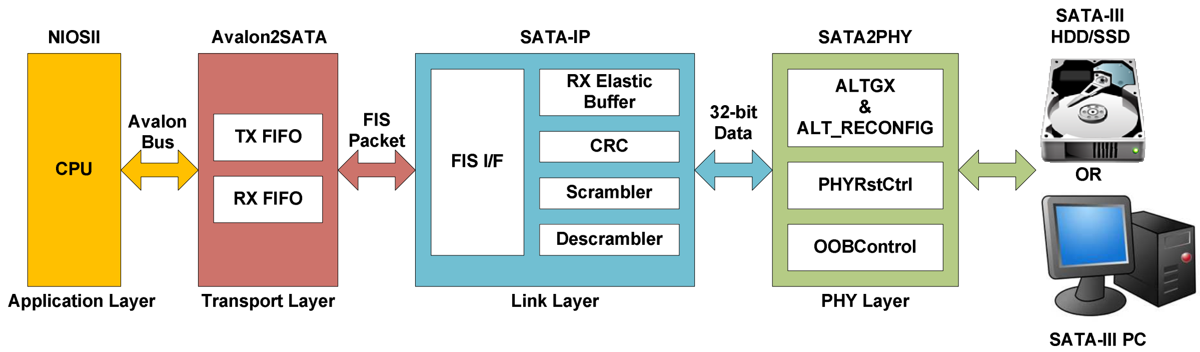

Figure 1: SATA IP Block Diagram

Applications

SATA IP Core is ideal for use in a variety of storage application which require high speed data transfer, cost, scalability and features extensibility such as embedded storage system, RAID controller and High speed and large capacity data acquisition system.

Moreover, the IP also supports SATA Device operation so that SATA Peripherals or SATA Bridge application is also possible.

General Description

The SATA IP Core implements the link layer and some parts of transport layer for communication between upper protocol layer managed by Host processor and PHY layer implemented by GXB Transceiver. For Host interface, the IP provides a simple TX and RX transaction interface to transfer 32-bit data between transport layer and Host processor at low frequency (at least 150 MHz for SATA-III and 75 MHz for SATA-II) which are easy to interface with an embedded processor on FPGA (NIOSII). For PHY interface, the IP is designed to support 40-bit PHY interface with 150MHz reference clock for SATA-III 6.0Gbps operation and 75 MHz for SATA-II 3.0Gbps operation.

The SATA IP Core evaluation on Stratix IV GX development kit is possible before IP purchase by using free demonstration sof file. Host reference design to show how to use IP Cores communicating with transport layer and PHY layer for HDD/SSD is also shipped with IP Core, so that user can easilly and immediately start logic design with the SATA IP Core.

Functional Description

The SATA IP Core is designed to operate under control of a system controller to transfer SATA FIS packet from/to system memory consisted of the following components.

Link Layer

The Link layer transmits and receives frames, transmits primitives based on control signals from the transport layer, and receives primitives from SATA PHY which are converted to control signals to the transport layer.

- **RX Elastic Buffer**

  This buffer is used to change clock domain of received data from recovery clock (RECCLK) to synchronous with core_clk.

- **CRC**

  The CRC of a frame is a Dword (32-bit) field that shall follow the last Dword of the contents of a FIS and precede EOF primitive.

- **Scramble**

  The content of a frame is scrambled before transmission by SATA PHY. Scrambling is performed on Dword quantities by XORing the data to be transmitted with output of a linear feedback shift register (LFSR) by SATA-IP Core.

- **Descramble**

  The content of a frame from SATA PHY is descrambled before transmission to transport layer. Descrambling is performed the same ways as scrambling to get FIS.

Transport Layer

The Transport layer constructs frame information structure (FIS) for transmission and decomposes received frame information structures. It also notifies the link layer of the required data flow control, generate status signal for upper layer.

- **FIS Interface**

  Provides the interface and data flow control for transmits and receive a transferred transaction with Host.

System Controller

The system controller is typically a host processor that executes application software to communicate with SATA IP Core and handle an upper layer SATA protocol. The system controller may consist of host processor, DMA Engine, TX FIFO and RX FIFO.

SATA PHY

SATA PHY design is designed by using Transceiver block (GXB) in Stratix IV GX and this module has been proved on Stratix IV GX development kit at SATA-III speed (6.0 Gbps) and SATA-II speed (3.0 Gbps). The operation on user board or user design is not guaranteed. Since SATA PHY source code will be provided to customer after purchasing, user can modify source code and any parameter in SATA PHY by own.

Core I/O Signals

Descriptions of all signal I/O are provided in Table 2.

Table 2: Core I/O Signals.

Signal	Dir	Clk	Description
Common Interface Signal			
trn_reset	In	trn_clk	Reset SATA IP core. Active high. Assert at least 4 clock period of core_clk for reset SATA-IP.
trn_link_up	Out	trn_clk	Transaction link up is asserted when the core establish the communication with SATA PHY.
trn_clk	In		Clock signal for interface with the Host. This clock frequency is required to be higher than core_clk frequency.
core_clk	In		IP Core operating frequency output (150 MHz for SATA-III and 75 MHz for SATA-II).
dev_host_n	In	trn_clk	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	trn_clk	Not used now.
trn_teof_n	In	trn_clk	Transmit End-Of-Frame (EOF): Indicate end each SATA FIS packet. Active low.
trn_td[31:0]	In	trn_clk	Transmit Data: SATA FIS packet data to be transmitted.
trn_tsrc_rdy_n	In	trn_clk	Transmit Source Ready: Indicate that trn_td[31:0] from the Host is valid. Active low.
trn_tdst_rdy_n	Out	trn_clk	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	trn_clk	Transmit Source Abort: Assert 1 clock period of trn_clk during operation (between tsof and teof) when the Host requires to cancel current write operation. Active low. After asserted, the Core will send SYNC primitive to SATA-PHY for abort the current transfer. The Host needs to wait until trn_tdst_rdy_n ready again before sending next packet. See Figure 4 for more details.
trn_tdst_dsc_n	Out	trn_clk	Transmit Destination Abort: Assert 1 clock period of trn_clk from the Core to cancel current write operation when SYNC primitive is received during data write operation. Active low. See Figure 6 for more details.

Table 1 SATA-IP Interface Signal Description

Signal	Dir	Clk	Description
Receive Transaction Interface			
trn_rsof_n	Out	trn_clk	Receive Start-Of-Frame (SOF): Indicate start each SATA FIS packet. Active low.
trn_reof_n	Out	trn_clk	Receive End-Of-Frame (EOF): Indicate end each SATA FIS packet. Active low.
trn_rd[31:0]	Out	trn_clk	Receive Data: SATA FIS packet data to be transmitted.
trn_rsrc_rdy_n	Out	trn_clk	Receive Source Ready: Indicate that trn_rd[31:0] from the core is valid. Active low.
trn_rdst_rdy_n	In	trn_clk	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	trn_clk	Receive Source Abort: Assert 1 clock period of trn_clk from the Core to cancel current read operation when SYNC primitive is received during data read operation. Active low. See Figure 7 for more details.
trn_rdst_dsc_n	In	trn_clk	Receive Destination Abort: Assert 1 clock period of trn_clk during read operation (between rsof and reof) when the Host requires to cancel current read operation. Active low. After asserted, the core will send SYNC primitive to SATA-PHY for abort the current transfer. The Host needs to wait until trn_tdst_rdy_n ready again before sending next packet. See Figure 5 for more details.
SATA PHY Interface			
LINKUP	In	core_clk	Indicate that SATA link communication is established. Active high.
PLLLOCK	In	core_clk	Indicate that PLL of SATA PHY is locked. Active high.
TXDATA[31:0]	Out	core_clk	32-bit transmit data from the core to SATA PHY
TXDATAK[3:0]	Out	core_clk	4-bit Data/Control for the symbols of transmitted data. (“0000”: data byte, “0001”: control byte, others: undefined).
RECCLK	In		Clock Recovery to synchronous with received data from SATA PHY
RXDATA[31:0]	In	RECCLK	32-bit receive data from the SATA PHY to the core.
RXDATAK[3:0]	In	RECCLK	4-bit Data/Control for the symbols of received data. (“0000”: data byte, “0001”: control byte, others: undefined)
RXDATAVALID	In	RECCLK	Indicate that RXDATA from SATA PHY is valid.
RXDATAOUT[31:0]	Out	core_clk	RXDATA signal after Elastic buffer and sychronous with core_clk
RXDATAKOUT[3:0]	Out	core_clk	RXDATAK signal after Elastic buffer and sychronous with core_clk
RXDATAVALIDOUT	Out	core_clk	Indicate that RXDATAOUT is valid.

Table1 SATA-IP Interface Signal Description (Cont’d)

Timing Diagram

As shown in Figure 2, first data will be transferred with asserting trn_tsof_n and trn_tsrc_rdy_n after the core is ready by monitoring trn_tdst_rdy_n signal. The core can receive at most 4 data from the host after deasserted trn_tdst_rdy_n. trn_td and trn_tsrc_rdy_n are connected to internal FIFO. trn_teof_n with trn_tsrc_rdy_n are asserted when final data is transferred. After packet is transferred from the Host to the core, the Host will wait to receive error code packet data returned from device to check that all data are received without any error.

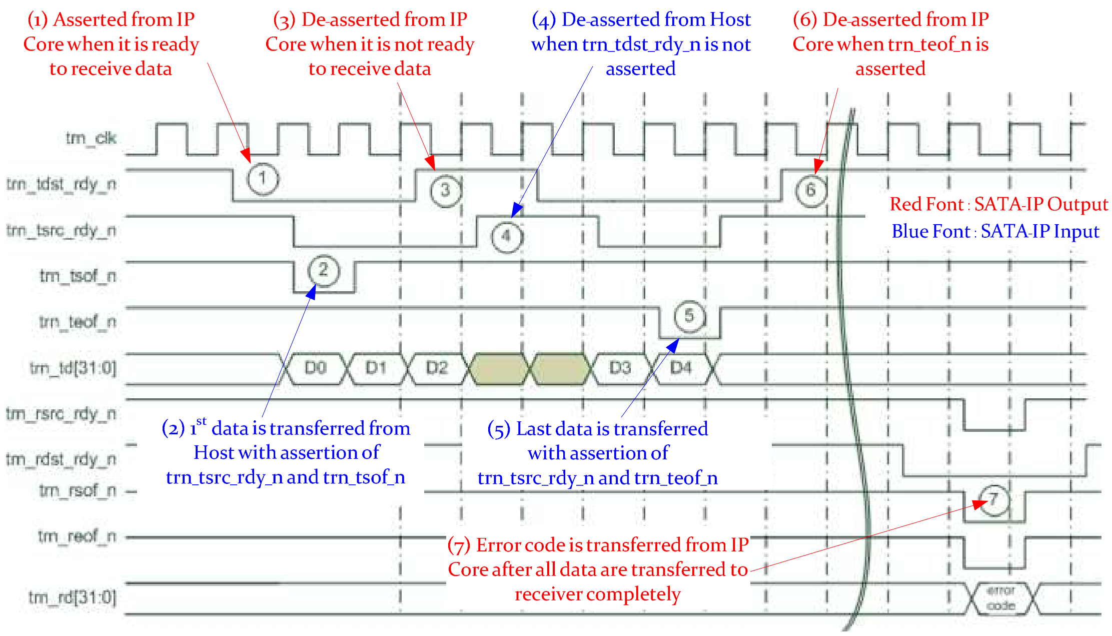

Figure 2: Transmit Transaction Interface Timing

Similar to Figure 2, first data will be transferred from the core after trn_rdst_rdy_n signal is asserted. trn_rdst_rdy_n signal must be deasserted before data buffer inside the Host is full at least 4 clock period. After packet is transferred from the core to the Host, the Host will wait to receive error code packet data returned from device.

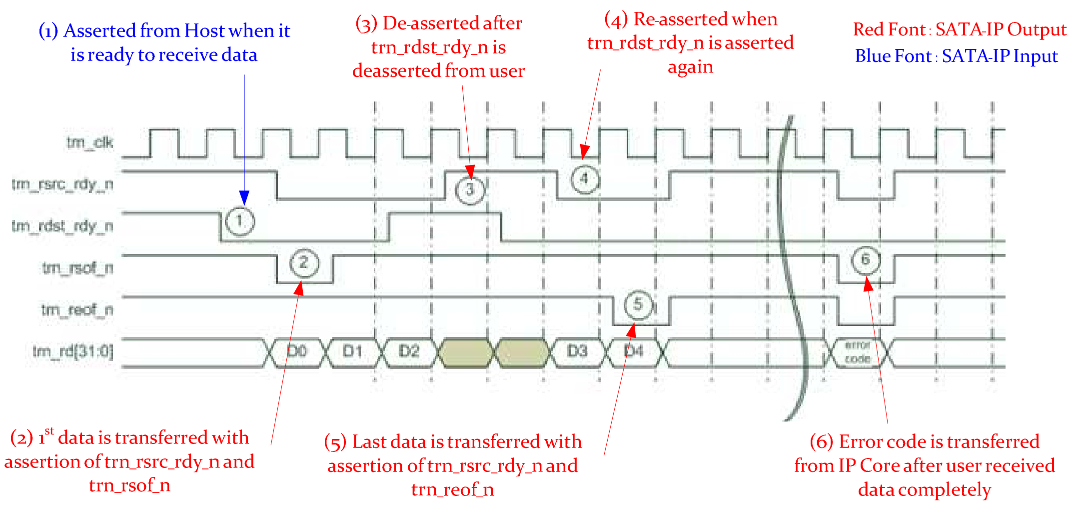

Figure 3: Receive Transaction Interface Timing

Error code which is shown in timing diagram is designed for the Host to check that current data packet can be transferred completely or not. So, the Host should be checked error code value after end transfer. The detail of error code is shown in Table3.

Table 3: Error code description.

Bit	Signal Name	Description
[31:27]	Reserved	Always zero
[26]	Dir	Current transfer direction flag. ‘0’: From the Host to SATA IP, ‘1’: From SATA IP to the Host
[25:24]	Error	Error code flag. “00”: No error “01”: Bad/Unknown SATA FIS packet. WTRM primitive is received during read operation or R_ERR primitive is received at the end of write operation. Please check data packet is correct format or not when this error detected. “10”: CRC error. Please check SATA signal quality when this error detected. “11”: Reserved
[23:8]	Reserved	Always zero
[7:0]	FIS Type	This byte indicates the header of error code packet. “0xEF” is defined to be different from other SATA FIS.

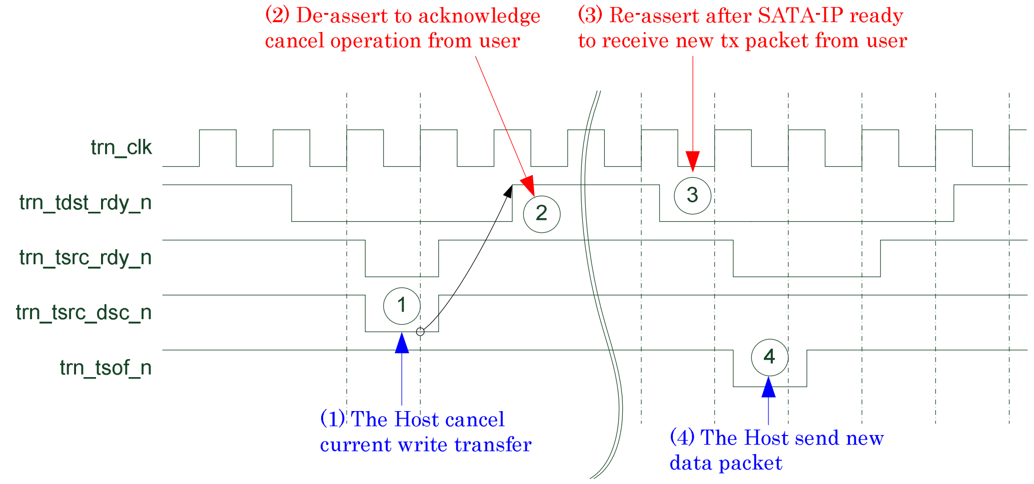

Figure 4: trn_tsrc_dsc_n timing diagram

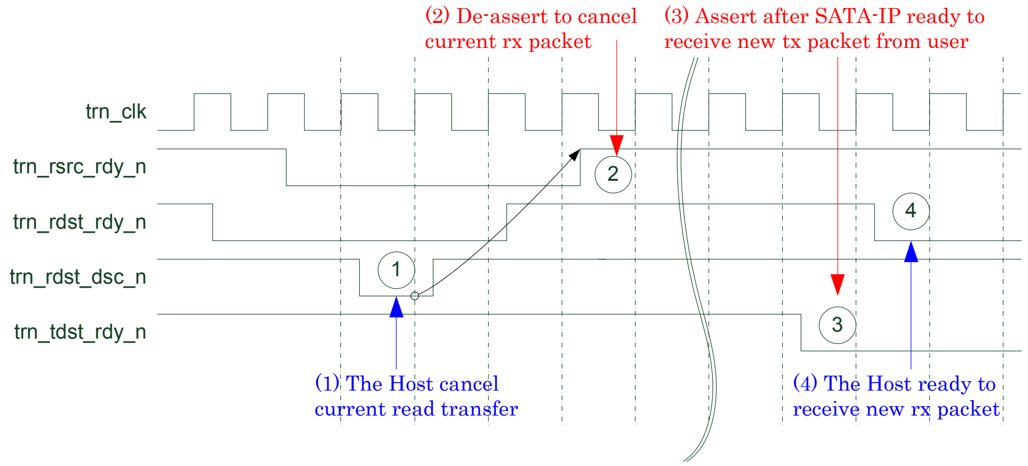

Figure 5: trn_rdst_dsc_n timing diagram

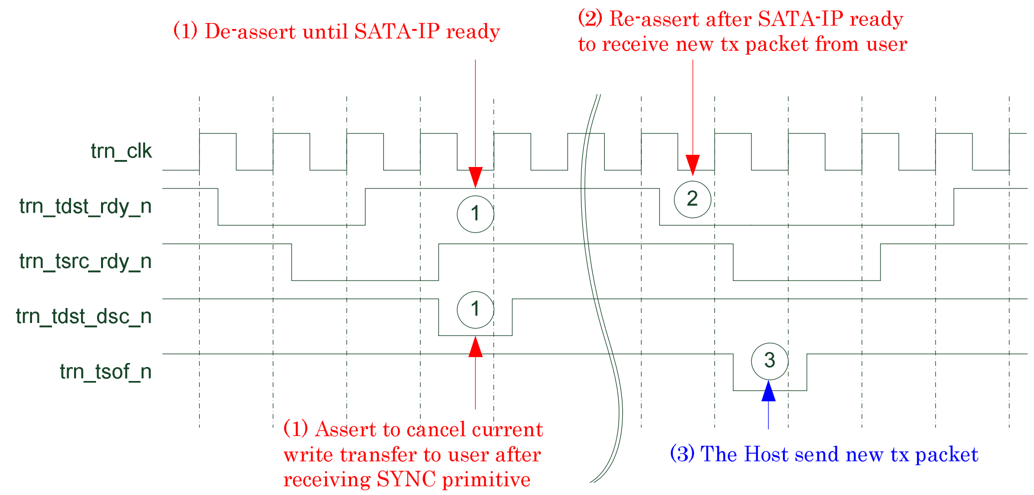

Figure 6: trn_tdst_dsc_n timing diagram

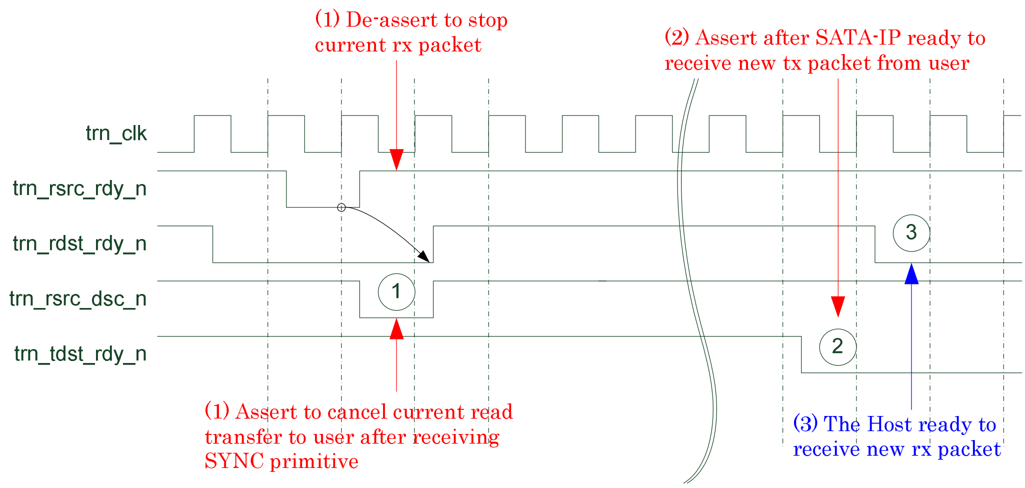

Figure 7: trn_rsrc_dsc_n timing diagram

Verification Methods

The SATA IP Core functionality was verified by simulation and also proved on real board design by using Stratix IV GX development board.

Recommended Design Experience

Experience design engineers with a knowledge of Gigabit Transceiver block and NIOS II EDS should easily integrate this IP into their design. For user board development, compliance with Transceiver Architecture section in Stratix IV Device Handbook is strongly recommended.

Ordering Information

This product is available directly from Design Gateway Co., Ltd. Please contact Design Gatway Co., Ltd. for pricing and additional information about this product using the contact information on the front page of this datasheet.

Revision History

Revision	Date	Description
1.0	Jan-16-2012	New release