Design Gateway Co.,Ltd
54 BB Building 14th Fl., Room No.1402 Sukhumvit 21 Rd. (Asoke), Klongtoey-Nua, Wattana, Bangkok 10110
Phone: (+66) 02-261-2277
Fax: (+66) 02-261-2290
E-mail: ip-sales@design-gateway.com
URL: www.design-gateway.com

Core Facts

Provided with Core	
Documentation	User Guide, Design Guide
Design File Formats	Encrypted Netlist File
Constraints Files	User constraint file
Verification	Test Bench, Simulation Library
Instantiation Templates	VHDL
Reference Designs & Application Notes	Vivado Project, See Reference Design Manual
Additional Items	Demo on AC701/KC705/ ZC706/VC707/VC709/KCU105
Simulation Tool Used	
Vivado Simulator	
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 BlockRAM in transmit and receive paths
- Support SATA III/II Speed
- Support NCQ command
- Low frequency operation
  - IP Core clock 150 MHz for SATA-III
  - IP Core clock 75 MHz for SATA-II
- CONT primitive support for continue primitive suppression to reduce EMI
- Support 40bit width PHY implemented by GTP/GTX/GTH
- Many reference designs on Xilinx evaluation board running with AB09-FMCRAID adapter board from Design Gateway
  - 1-ch SATA host demo reference design on AC701/KC705/ZC706/VC707/VC709/KCU105 board
  - 4-ch SATA RAID0 demo reference design on KC705/ZC706/VC707/VC709/KCU105 board
  - 8-ch SATA RAID0 demo reference design on VC709 board
  - 1-ch SATA host with exFAT support design on KC705/ZC706 board
  - SATA device demo reference design on AC701/KC705/ZC706 board
  - SATA bridge demo reference design on AC701/KC705 board
  - SATA AHCI IP demo reference design on ZC706 board
  - PCIe SATA AHCI demo reference design on KC705/VC707 board
  - 1-ch SATA host demo by pure-hardware logic on AC701/KC705/ZC706/VC707 board
  - 4-ch SATA RAID0 demo by pure-hardware logic on KC705/ZC706/VC707 board

Table 1: Example Implementation Statistics for 7-Series device

Family	Example Device	Fmax (MHz)	Slice Regs	Slice LUTs	Slices [1]	IOB[2]	BUFG	RAMB18	PLL	GTP/GTX	Design Tools
Artix-7	XC7A200TFBG676-2	222	863	1022	448	154	3	2	1	1	Vivado2013.2
Kintex-7	XC7K325TFFG900-2	285	863	1023	475	154	3	2	1	1	Vivado2013.2
Zynq-7000	XC7Z045FFG900-2	285	863	1028	482	154	3	2	1	1	Vivado2013.2
Virtex-7	XC7VX485TFFG1761-2	333	863	1026	444	154	3	2	1	1	Vivado2013.2
Virtex-7	XC7VX690TFFG1761-2	333	863	1024	476	154	3	2	1	1	Vivado2013.2

Table 2: Example Implementation Statistics for Ultrascale device

Family	Example Device	Fmax (MHz)	LUT FF	LUT Logic	CLB	IOB[2]	BUFG	RAMB18	PLL	GTH	Design Tools
Kintex-Ultrascale	XCKU040FFVA1156-2E	433	1168	1015	224	154	-	2	-	1	Vivado2014.4

Notes:

1) Actual slice count dependent on percentage of unrelated logic. The example is the report from utilization_placed.rpt file

2) Assuming all core I/Os and clocks are routed off-chip

3) BUFG, PLL, and GTP/GTX/GTH resource is not used in SATA IP core, but they are used in SATA PHY design.

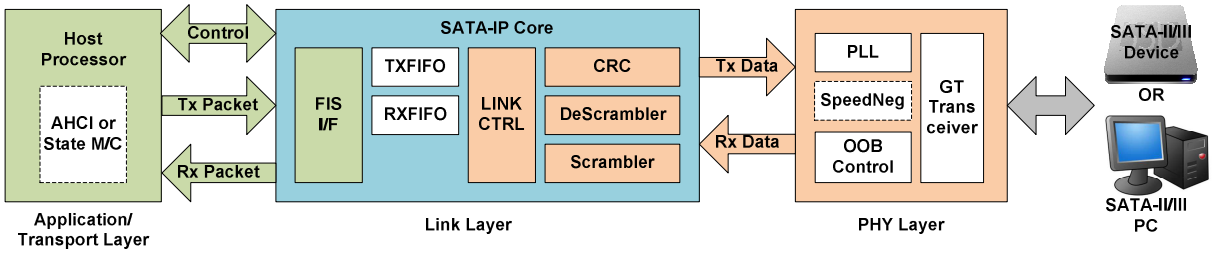

Figure 1: SATA IP Block Diagram

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 GTP/GTX/GTH 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) which are easy to interface with an embedded processor on FPGA (ARM/Microblaze) or interface with pure-hardware logic. For PHY interface, the IP is designed to support 40-bit PHY interface with 150MHz reference clock for SATA-III 6.0Gbps and 75MHZ for SATA-II 3.0Gbps operation.

The SATA IP Core evaluation on many Xilinx evaluation boards are provided before IP purchase by using free demonstration bit file. Several reference design are provided, such as 1-ch Host reference design, 4-ch Host RAID reference design, 8-ch Host RAID reference design. 1-ch Host design includes speed auto-negotiation to interface with both SATA-III and SATA-II device, while 4-ch/8-ch Host RAID design shows high performance transfer by interface with 4/8 fixed SATA-III device as RAID0 system. Final performance for RAID0 is about 4/8 times of one SATA Device. The reference designs are provided to customer to reduce development time for your own system.

Functional Description

The SATA IP Core is designed to operate under control of a system controller to transfer SATA FIS packet between system memory such as DDR, BlockRAM and 40-bit interface at transceiver of PHY layer. The SATA IP Core consists of the following components.

Link Layer

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

- **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

The example of SATA PHY reference design source code is also provided after purchasing. PHY source code in RAID reference design is different from 1-ch host design. In RAID design, PHY can support only SATA-III speed (6.0Gbps) to reduce clock resource within PHY layer and for easily reset sequence design. There are two source codes included in RAID demo, the master channel includes PLL and clock buffer while the others (slave channel) do not include. This solution can reduce clock resource to share PLL and BUFG for all 4/8 SATA channels in one/two QUAD.

In 1-ch host design on AC701/KC705/ZC706/VC707, PHY example includes “speed_neg_control” module to control SATA speed negotiation function. This module makes PHY on 1-ch design support both SATA-II and SATA-III devices.

Core I/O Signals

Descriptions of all signal I/O are provided in Table 3.

Table 3: 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 (150MHz for SATA-III, 75MHz for SATA-II). This clock is generated from SATA PHY.
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: Indicates 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 3 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: Indicates 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	Indicates that SATA link communication is established. Active high.
PLLLOCK	In	core_clk	Indicates that PLL of SATA PHY is locked. Active high.
TXDATA[31:0]	Out	core_clk	32-bit transmit data from the core to the 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).
RXDATA[31:0]	In	core_clk	32-bit receive data from the SATA PHY to the core.
RXDATAK[3:0]	In	core_clk	4-bit Data/Control for the symbols of received data. (“0000”: data byte, “0001”: control byte, others: undefined)

Table 3 SATA-IP Interface Signal Description (Cont’d)

Timing Diagram

As shown in Figure 2, data will be transferred with asserting 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.

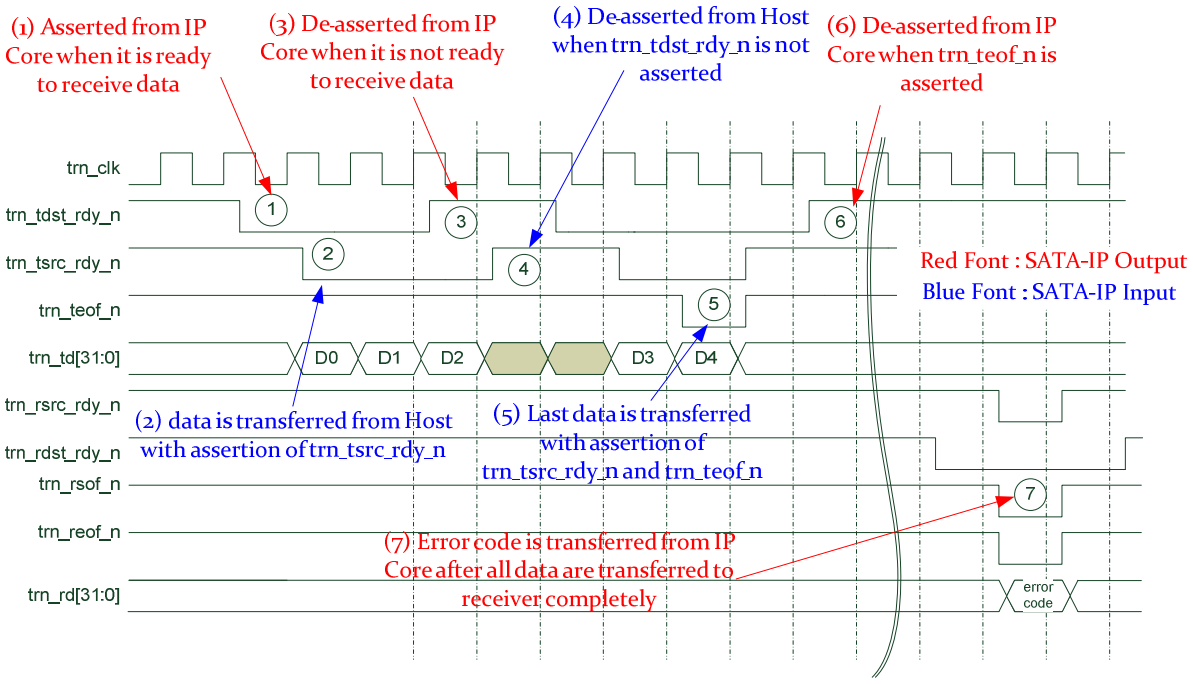

Figure 2: Transmit Transaction Interface Timing

Similar to Figure 2, the 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.

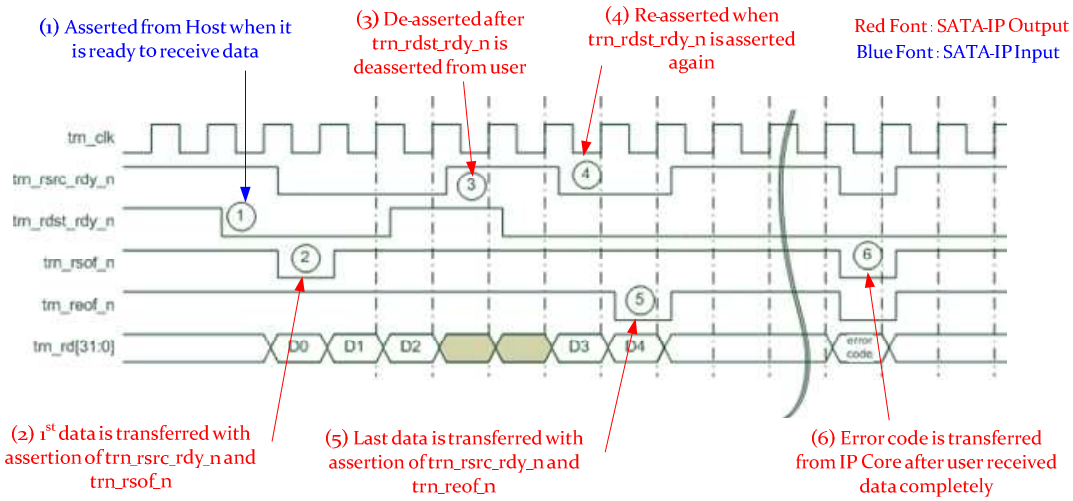

Figure 3: Receive Transaction Interface Timing

Error code 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 check error code value after end transfer. The details of error code is shown in Table 4.

Table 4: 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.

User can cancel current transaction by asserting disconnect signal to SATA-IP, i.e. trn_tsrc_dsc_n for transmit side and trn_rdst_dsc_n for received side. For transmit side, after user cancel transaction, trn_tdst_rdy_n status must be monitored to check IP acknowledge, as shown in Figure 4.

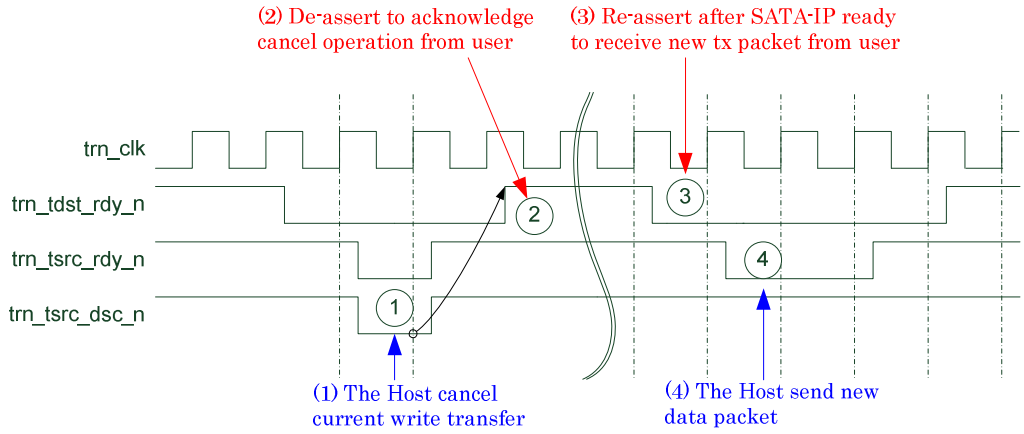

Figure 4: trn_tsrc_dsc_n timing diagram

For received side, after asserting trn_rdst_dsc_n signal, trn_rsrc_rdy_n will be deasserted to stop current received transfer, as shown in Figure 5.

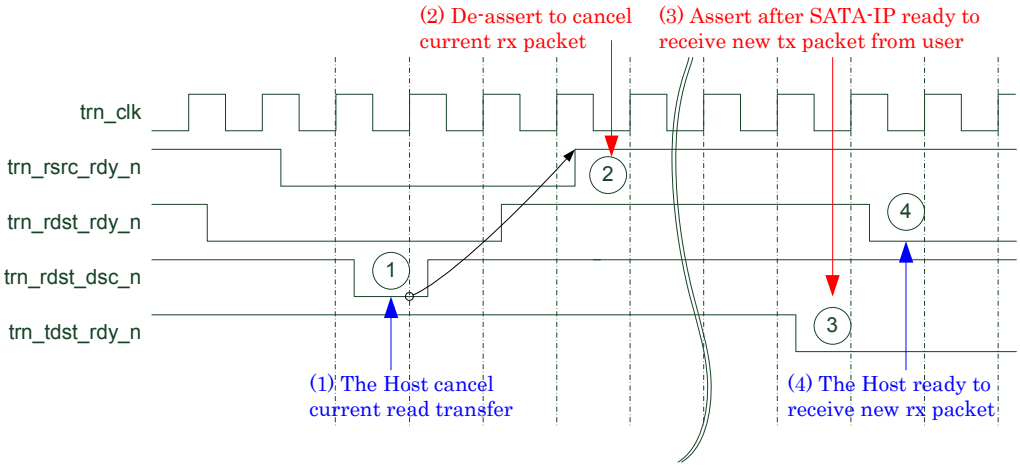

Figure 5: trn_rdst_dsc_n timing diagram

If the target sends SYNC primitives to cancel transmit operation or data collision is detected, trn_tdst_dsc_n will be asserted, as shown in Figure 6. If transmit packet is short, trn_tdst_dsc_n may be asserted after end of packet but before error code from IP arrived. User can re-send the packet after trn_tdst_rdy_n is asserted. If data collision is detected, IP will send received packet after deassert trn_tdst_rdy_n. So, user needs to process received packet completely before re-send the previous transmit packet.

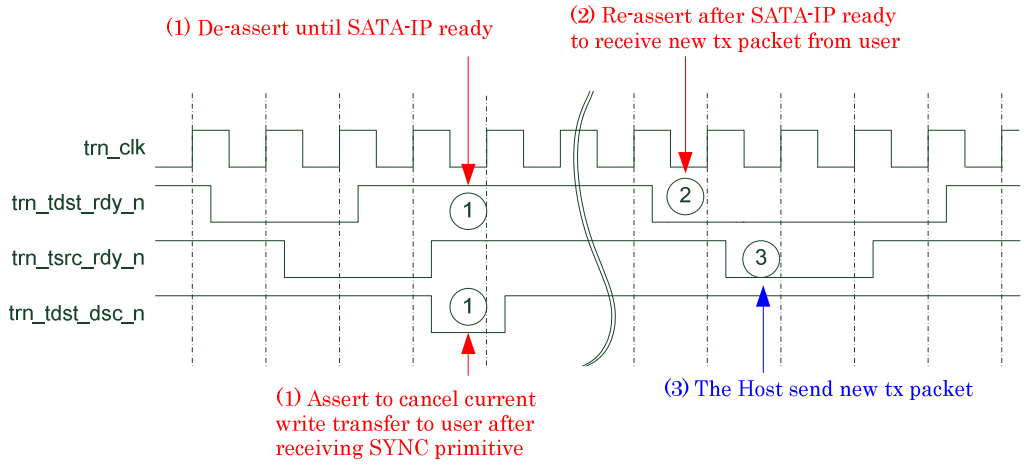

Figure 6: trn_tdst_dsc_n timing diagram

If the target cancels the current received packet to user, trn_rsrc_dsc_n will be asserted. trn_rsrc_rdy_n status will be changed back to '1' to stop current transfer, as shown in Figure 7.

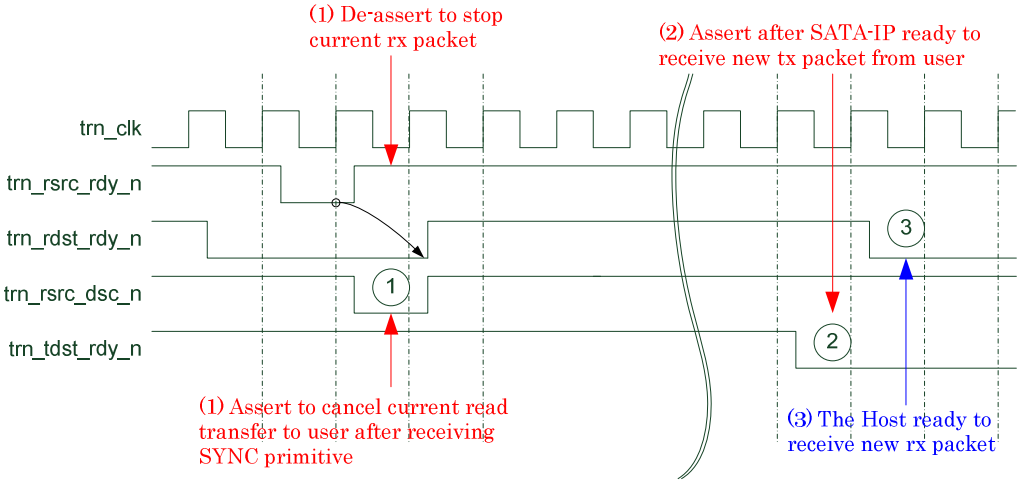

Figure 7: trn_rsrc_dsc_n timing diagram

Verification Methods

The SATA IP Core functionality was verified by simulation and also proved on real board design by using AC701/KC705/ZC706/VC707/VC709/KCU105 evaluation board.

Recommended Design Experience

Experience design engineers with a knowledge of RocketIO and Vivado Tools should easily integrate this IP into their design. For user board development, compliance with design guideline described in UG476 (7 Series FPGAs GTX/GTH Transceivers User Guide), UG482 (7 Series FPGAs GTP Transceivers User Guide), or UG576 (Ultrascale GTH Transceivers User Guide) 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
2.0	Oct-8-2014	Support NCQ command
2.1	Jan-21-2016	Support KCU105 board