June 17, 2021 | Product Specification | Rev2.4

Design Gateway Co.,Ltd
E-mail: ip-sales@design-gateway.com
URL: design-gateway.com

Core Facts

Provided with Core	
Documentation	Reference design manual, Demo instruction manual
Design File Formats	Encrypted Netlist File
Instantiation Templates	VHDL
Reference Designs & Application Notes	Vivado Project, See Reference Design Manual
Additional Items	Demo on AC701/KC705/ ZC706/VC707/VC709/KCU105/ Zynq Mini-ITX/ZCU102/VCU118
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
- Simple user interface and 32-bit data bus
- Include two 4KB FIFOs to be data buffer
- Support SATA III/II Speed and NCQ command
- Require low user clock frequency (at least 150 MHz for SATA-III or 75 MHz for SATA-II)
- CONT primitive support for continue primitive suppression to reduce EMI
- Provide SATA PHY including Xilinx transceiver as HDL code in reference design
- Many IP options for SATA application - HCTL IP, AHCI IP, FAT32 IP, and exFAT IP
- Many reference designs on FPGA evaluation board (Most boards require AB09-FMCRAID adapter)
  - 1-ch SATA host design
  - 4-ch SATA RAID0 design
  - 1-ch SATA host design with exFAT support by firmware
  - SATA device design
  - SATA bridge design
  - SATA AHCI IP design
  - PCIe SATA AHCI IP design
  - 1-ch HCTL IP design
  - 4-ch HCTL IP design
  - 8-ch HCTL IP with/without DDR design
  - SATA FAT32 IP design
  - SATA exFAT design

Table 1: Example Implementation Statistics for 7-Series device

Family	Example Device	Fmax (MHz)	Slice Regs	Slice LUTs	Slices[1]	IOB	BUFG	BRAMTile	PLL	GTP/GTX	Design Tools
Artix-7	XC7A200TFBG676-2	222	912	963	328	-	3	1	1	1	Vivado2019.1
Kintex-7	XC7K325TFFG900-2	285	912	963	341	-	3	1	1	1	Vivado2019.1
Zynq-7000	XC7Z045FFG900-2	285	912	963	346	-	3	1	1	1	Vivado2019.1
Virtex-7	XC7VX485TFFG1761-2	333	912	963	341	-	3	1	1	1	Vivado2019.1
Virtex-7	XC7VX690TFFG1761-2	333	912	963	339	-	3	1	1	1	Vivado2019.1

Table 2: Example Implementation Statistics for Ultrascale device

Family	Example Device	Fmax (MHz)	CLB Regs	CLB LUTs	CLB	IOB	BUFG	BRAMTile	PLL	GTH/GTY	Design Tools
Kintex-Ultrascale	XCKU040FFVA1156-2E	433	912	960	183	-	-	1	-	1	Vivado2019.1
Zynq-Ultrascale+	XCZU9EG-FFVB1156-2-I	>500	912	973	205	-	-	1	-	1	Vivado2019.1
Virtex-Ultrascale+	XCVU9P-FLGA2104-2L-E	>500	912	958	197	-	-	1	-	1	Vivado2019.1

Notes:

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

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

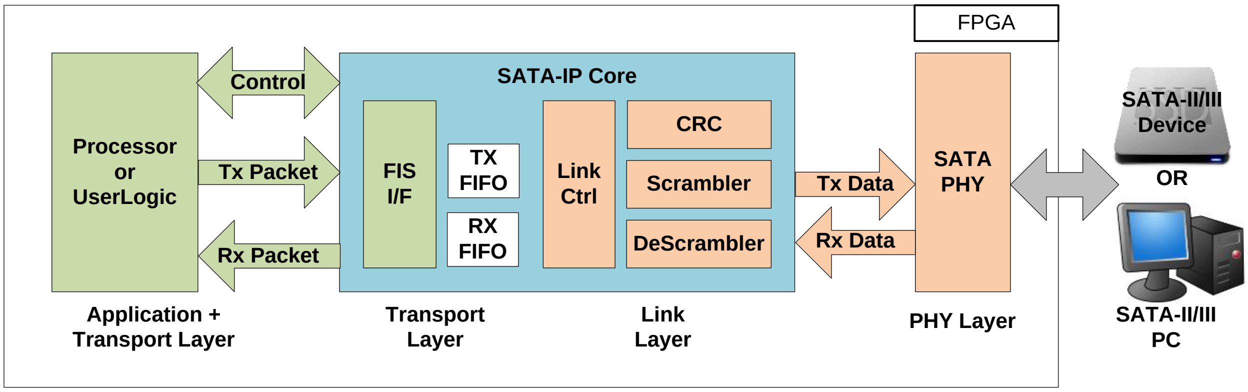

Figure 1: SATA IP Block Diagram

Applications

SATA IP is ideal for use in a variety of storage application such as embedded storage system and High speed with large capacity data acquisition system. System performance, device capacity, and data reliability are increased by using multiple SATA IPs as RAID0 operation. SATA IP is the solution which achieves high speed performance, scalability, and features extensibility.

The IP supports both Host and Device mode. so it supports the applications such as Secure storage.

General Description

SATA IP implements link layer and a part of transport layer for communication between the upper layer managed by Processor/UserLogic and PHY layer implemented by Xilinx Transceiver. For upper layer interface, the IP provides a simple TX and RX transaction interface to transfer 32-bit data between transport layer and processor at low frequency (at least 150 MHz for SATA-III). The interface is easy to connect with an embedded processor on FPGA (ARM/Microblaze) or connect with pure-hardware logic. For PHY interface, the IP supports 32-bit PHY interface with 150MHz reference clock for SATA-III 6.0Gbps and 75MHZ for SATA-II 3.0Gbps operation.

Free demo bit file to evaluate SATA IP on Xilinx evaluation boards are provided on the website. Besides, many reference designs are provided for various SATA applications such as 1-ch Host design, 4-ch RAID0 design, and exFAT support design. RAID0 is the solution to increase transfer performance and device capacity by connecting multiple SATA devices to one Host.

Furthermore, four optional IPs are provided to complete the design of all SATA protocol layers, i.e., HCTL IP, AHCI IP, FAT32 IP, and exFAT IP.

Functional Description

SATA IP converts SATA FIS packet of Processor/UserLogic interface to be data stream for SATA PHY layer. SATA IP has the logic implementing Link layer and Transport layer. Two asychronous FIFOs are the data buffer for transferring data packet between Transport layer logic and Link layer logic which run in different clock domain. Also, FIFO is applied to control data flow in SATA IP.

Link Layer

Link layer transmits primitives based on control signals from Transport layer. Also, it receives primitives from SATA PHY which are converted to control signals for Transport layer.

- **CRC**

  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 forwarding to 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.

- **Descramble**

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

Transport Layer

Transport layer constructs frame information structure (FIS) for transmission. On the other hand, it decomposes received frame information structures. It also notifies Link layer of the required data flow control and generates status signals for upper layer.

- **FIS Interface**

  Provide the interface and data flow control to transmit and receive a transferred transaction with Application layer.

Processor/UserLogic

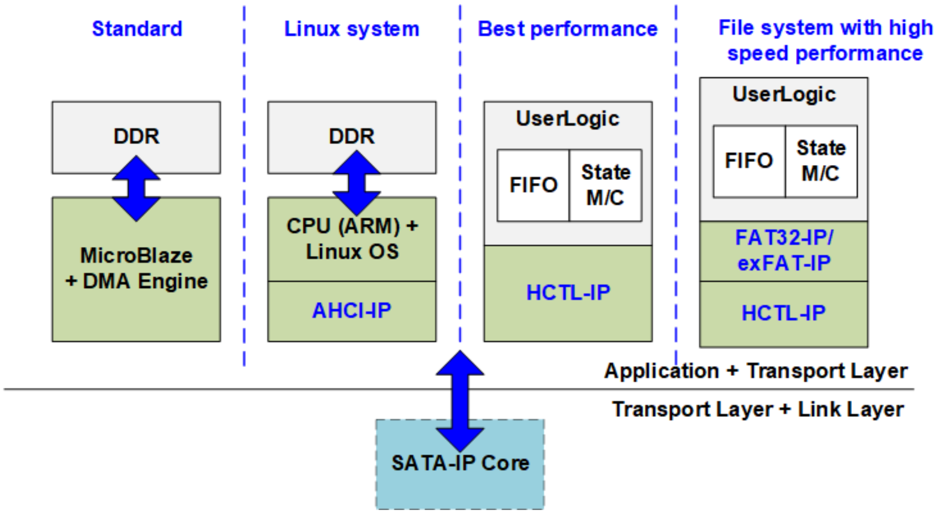

Figure 2: Processor/UserLogic implementation

In standard system, SATA IP operates with the host processor that runs the firmware for handling the packet with SATA IP. The SATA IP and the host proecessor include DMA Engine to handle FIS packet through main memory (DDR). This system is flexible to support various ATA commands and easy to upgrade the system specification. However, the system consumes large FPGA resources for building processor system and DDR controller.

As optional, AHCI IP, HCTL IP, FAT32 IP, and exFAT IP are purposed to complete Application layer of SATA protocol for many applications. AHCI IP is the option to integrate SATA IP to the processor system which runs OS and needs to access SATA IP by using standard driver, AHCI driver. While remaining IP cores are designed to minimize FPGA resource with high transfer performance achievement. They completes SATA host solutions without using the processor and DDR. The user interface of the IPs is very simple, so the user logic can be designed by simple state machine with some registers. HCTL IP implements SATA application layer which can fit to data acquisition system. FAT32 IP and exFAT IP are the additional solutions of HCTL IP to transfer f data with SATA device by FAT32 and exFAT file system, not raw data format. Please see more details from our website.

http://www.dgway.com/SATA-IP_X_E.html

SATA PHY

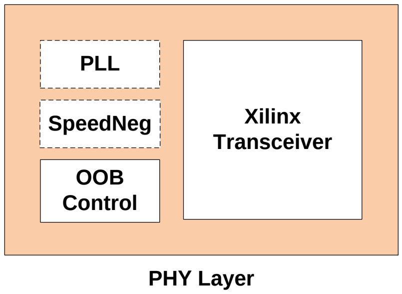

Figure 3: Hardware in SATA PHY example

The example HDL code of SATA PHY is provided in SATA IP reference design after purchasing. SATA PHY consists of at least two parts - OOB Control and Xilinx transceiver. OOB Control includes state machine for SATA initialization from system boot to link up status. Transceiver is the hardware inside Xilinx FPGA and the characteristic is different for each FPGA model. The different parameters are assigned for different FPGA model.

SpeedNeg and PLL are included in the design which supports both SATA-II and SATA-III speed. In RAID0 and SATA device application, SpeedNeg is not included. The designs are run as fixed speed. Also, PHY in RAID0 design shares some clock resources from one channel (master channel) to other channels (slave channel) for reducing the resource.

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			
IPVersion[31:0]	Out		IP version number.
trn_reset	In	trn_clk	Reset SATA IP which is synchronous reset and active high. Assert at least 4 clock cycles of core_clk to reset SATA-IP.
trn_link_up	Out	trn_clk	Transaction link up. Asserted when the core establishes the communication with SATA PHY.
trn_clk	In		Clock signal for interface with the user. This clock frequency must be higher than or equal to core_clk frequency.
core_clk	In		SATA IP operating frequency output (150MHz for SATA-III, 75MHz for SATA-II). This clock is generated by SATA PHY.
dev_host_n	In	trn_clk	Device or Host mode assignment. '0': Host IP, '1': Device IP. (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 of 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 to '1' within 4 clock cycles of trn_clk after trn_tdst_rdy_n is de-asserted to '1'. Therefore, the core can accept 4 DWORDs of trn_td[31:0] after trn_tdst_rdy_n is de-asserted to '1'.
trn_tsrc_dsc_n	In	trn_clk	Transmit Source Abort: Assert for 1 clock cycle of trn_clk during operation (between tsof and teof) for user cancelling current write operation. Active low. After asserting this signal to '0', the core sends SYNC primitive to SATA PHY to abort the current transfer. The user needs to wait until trn_tdst_rdy_n ready again before sending next packet. See Figure 6 for more details.
trn_tdst_dsc_n	Out	trn_clk	Transmit Destination Abort: Asserted for 1 clock cycle of trn_clk by the core to cancel current write operation when SYNC primitive is received. Active low. See Figure 8 for more details.
Receive Transaction Interface			
trn_rsof_n	Out	trn_clk	Receive Start-Of-Frame (SOF): Indicate start of SATA FIS packet. Active low.
trn_reof_n	Out	trn_clk	Receive End-Of-Frame (EOF): Indicate end of 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 user is ready to accept data on trn_rd[31:0]. Active low. trn_rsrc_rdy_n is de-asserted to '1' within 4 clock cycles of trn_clk after trn_rdst_rdy_n is de-asserted to '1'. Therefore, the user logic should support to accept 4 DWORD of trn_rd[31:0] after trn_rdst_rdy_n is de-asserted to '1'.
trn_rsrc_dsc_n	Out	trn_clk	Receive Source Abort: Asserted for 1 clock cycle of trn_clk by the core to cancel current read operation when SYNC primitive is received. Active low. See Figure 9 for more details.
trn_rdst_dsc_n	In	trn_clk	Receive Destination Abort: Assert 1 clock cycle of trn_clk during read operation (between rsof and reof) when the user cancels current read operation. Active low. After asserting this signal to '0', the core sends SYNC primitive to SATA-PHY to abort the current transfer. The user needs to wait until trn_tdst_rdy_n ready again before sending the next packet. See Figure 7 for more details.

Signal	Dir	Clk	Description
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 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)

Timing Diagram

As shown in Figure 4, data is transferred with asserting trn_tsrc_rdy_n to ‘0’ after the core is ready (monitored by trn_tdst_rdy_n=’0’). The core receives at most 4 data after trn_tdst_rdy_n is de-asserted to ‘1’. trn_td and trn_tsrc_rdy_n are the write data and write data valid signals for storing to FIFO inside SATA IP. trn_teof_n and trn_tsrc_rdy_n are asserted to ‘0’ when transferring the last data of the packet. After the packet is completely transferred to the core, the user waits until error code packet, generated by SATA IP, is received to show transfer status of a packet transaction.

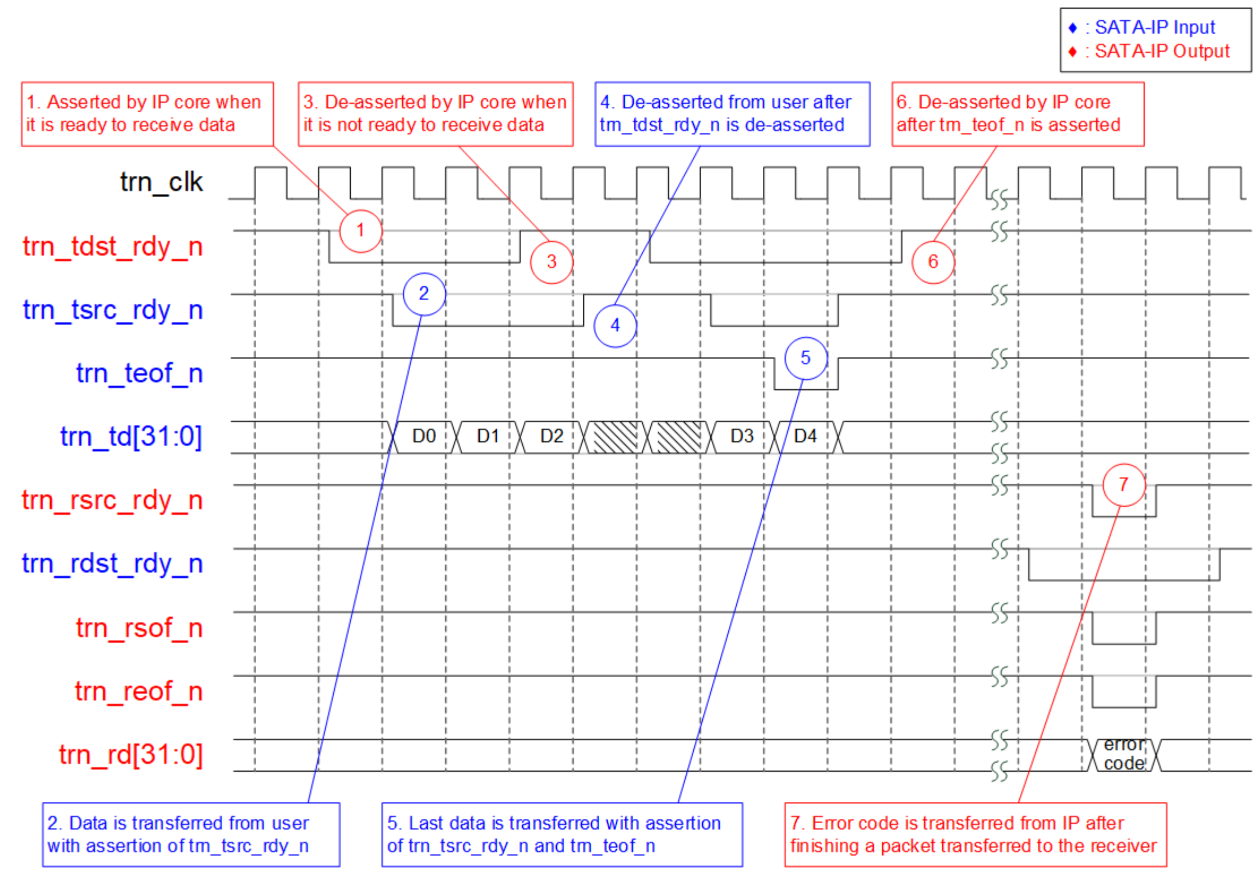

Figure 4: Transmit Transaction Interface Timing

Similar to Figure 4, the first data is transferred by the core after trn_rdst_rdy_n signal is asserted to '0'. trn_rdst_rdy_n signal must be deasserted to '1' when free space of data buffer inside the user is less than 5 (Up to four data are transferred after deasserting ready signal). After packet is transferred from the core to the user, the user waits until error code packet is returned from SATA IP.

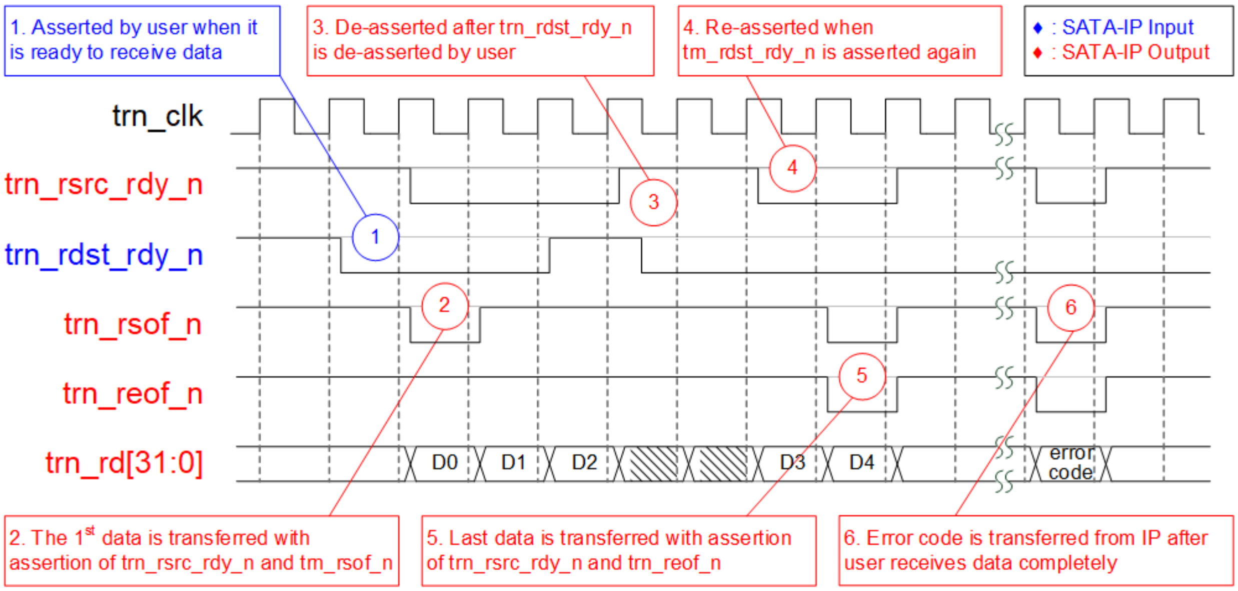

Figure 5: Receive Transaction Interface Timing

As shown in Figure 4 and Figure 5, Error code is designed for the user to check that SATA packet is transferred completely or some errors are found. Therefore, the user should check error code value after finishing transferring each packet. 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 user to SATA IP, '1': From SATA IP to the user
[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 is 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.

To cancel current transaction by user, two signals are designed to be SATA IP input - trn_tsrc_dsc_n and trn_rdst_dsc_n. trn_tsrc_dsc_n is applied to cancel current write operation while trn_rdst_dsc_n is applied to cancel current read operation.

After cancelling write operation, trn_tdst_rdy_n status must be monitored to check IP acknowledge, as shown in Figure 6. trn_tdst_rdy_n is de-asserted to '1' after operation is cancelled. The new packet could be transmitted when trn_tdst_rdy_n changes to '0' again.

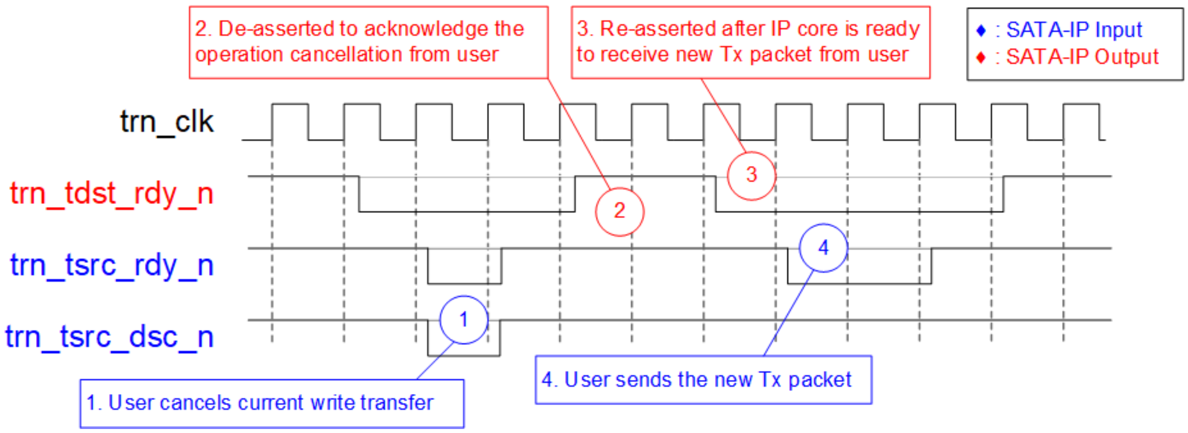

Figure 6: trn_tsrc_dsc_n timing diagram

After cancelling read operation, trn_rsrc_rdy_n is deasserted to '1', as shown in Figure 7.

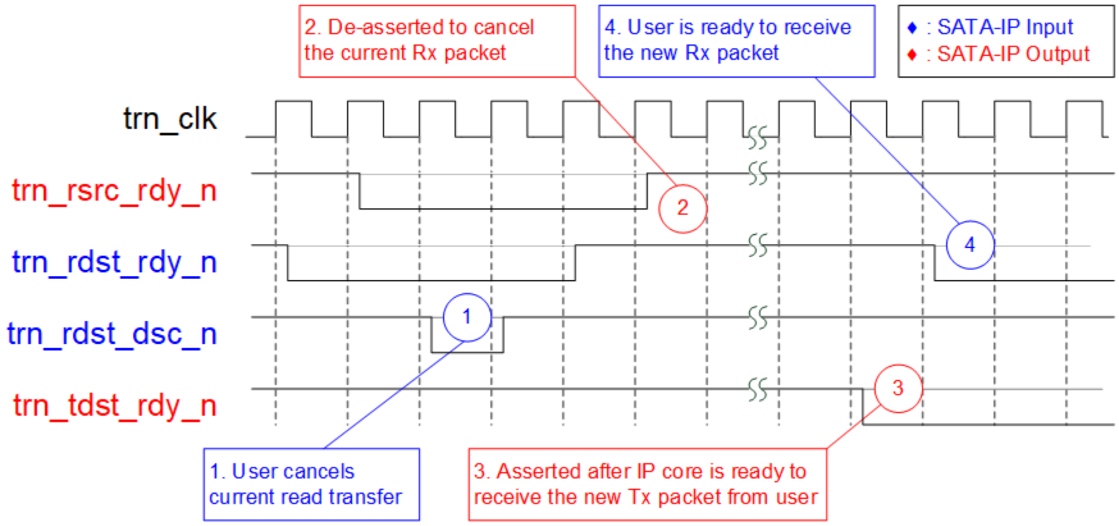

Figure 7: trn_rdst_dsc_n timing diagram

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 8. In case of short packet, trn_tdst_dsc_n may be asserted between end of packet and error code.

To re-send the packet after data collision, user needs to wait until trn_tdst_rdy_n is asserted to '0' and the received packet is processed completely.

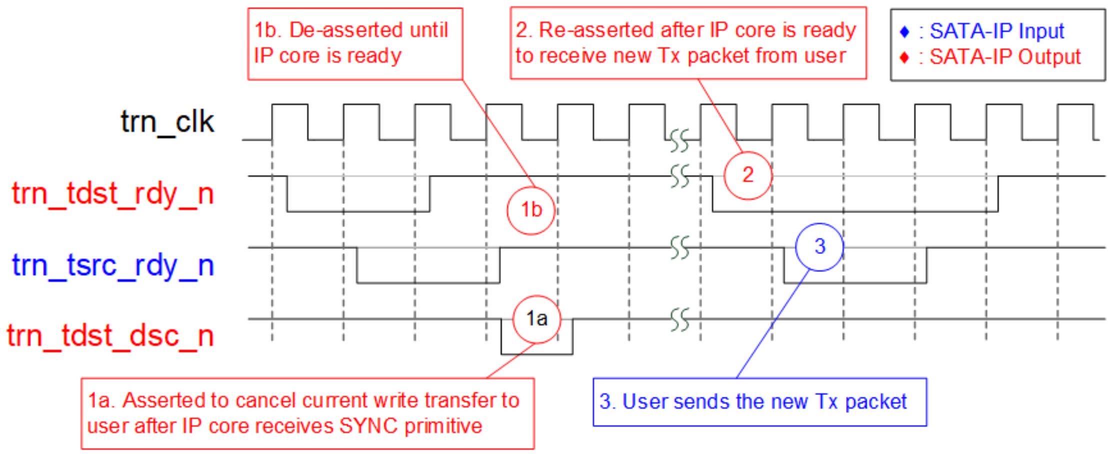

Figure 8: trn_tdst_dsc_n timing diagram

If the target cancels to send the current packet, trn_rsrc_dsc_n will be asserted to '0'. trn_rsrc_rdy_n status changes to '1' to stop current transfer, as shown in Figure 9.

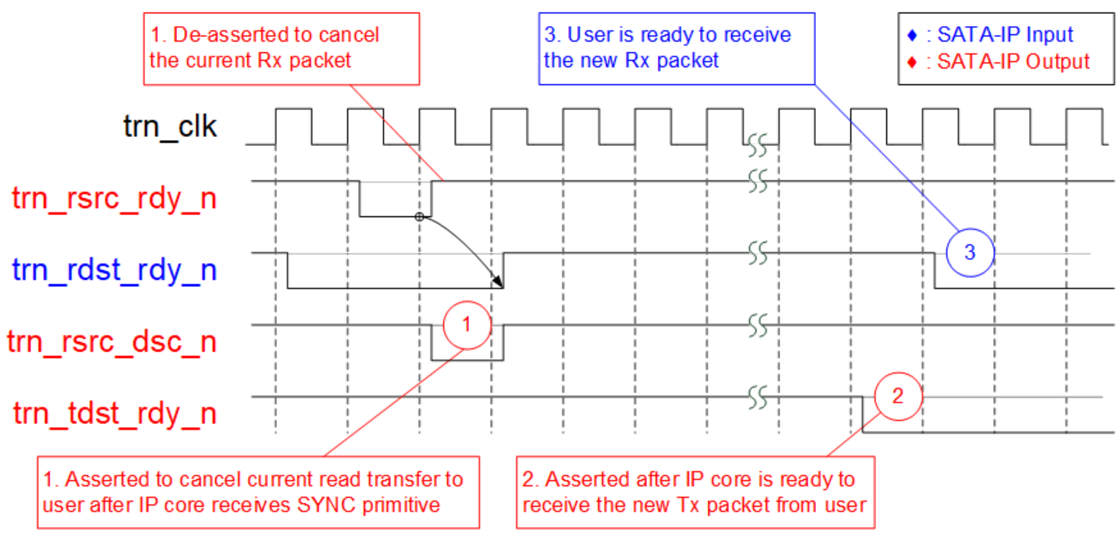

Figure 9: trn_rsrc_dsc_n timing diagram

Verification Methods

SATA IP functionality was verified by simulation and also proved on real board design by using AC701/KC705/ZC706/VC707/VC709/KCU105/ZCU102/Zynq Mini-ITX/VCU118 evaluation board.

Recommended Design Experience

Experience design engineers with a knowledge of Transciever 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), UG576 (Ultrascale GTH Transceivers User Guide), or UG578 (UltraScale GTY Transceivers User Guide) is strongly recommended.

Ordering Information

This product is available directly from Design Gateway Co., Ltd. Please contact Design Gateway 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	8-Oct-14	Support NCQ command
2.1	21-Jan-16	Support KCU105 board
2.2	18-Jan-18	Support ZCU102 board
2.3	7-May-18	Support VCU118 board
2.4	17-Jun-21	Update resource utilization and add IPVersion