SATA-III Host Controller Core

Contents

List of Figures

List of Tables

Chapter 1

SATA-III DATA SHEET

The so_ip_sata3_hctrl is a soft core implementation of SATA host controller as defined in the SATA Specification 3.2.

So_ip_sata3_hctrl soft core is fully compliant with the SATA 3.2 specification, and supports both 1.5 Gbit/s, 3.0 Gbit/s and 6.0 Gbit/s data transfer rates.

SATA Host Controller core implements physical, link and transport layers defined in the SATA 3.2 specification. It can use both RocketIO GTP and GTX transceivers to implement required physical signaling. For the interface with the host processor IP core uses standard PATA interface, and for the interface with the DMA engine standard AXI-4 Streaming TX and RX transaction interface.

So_ip_sata3_hctrl core is delivered with fully automated testbench and a compete set of tests allowing easy package validation at each stage of SoC design flow.

SATA Host Controller core uses strictly synchronous design with positive-edge clocking, no internal tri-states and a synchronous reset. It operates at 37.5 MHz system clock frequency in case of SATA-I mode (1.5 Gbit/s data transfer rate), 75 MHz in case of SATA-II mode (3.0 Gbit/s data transfer rate) and at 150 MHz in case of SATA- III mode (6.0 Gbit/s data transfer rate).

SATA Host Controller core can be evaluated using Xilinx Evaluation Platforms before actual purchase. This is achieved by using a time- limited demonstration bit files for AC701 and KC705 Xilinx evaluation platforms that allows the user to connect it's HDD to the SATA Host Controller core and evaluate system performance under different transfer scenarios. Evaluation license is also available enabling SATA Host Controller core evaluation in the user defined applications.

1.1 FEATURES

- Supports Xilinx Virtex6, Artix7, Kintex7, Virtex7 and Zynq FPGAs
- Fully compliant with the Serial ATA specification revision 3.2
- Simple transaction interface with Host processor and DMA Engine
- 32-bit internal data path
- 8KB FIFO implemented by BlockRAM in both transmit and receive paths
- Low frequency operation
  - IP Core system clock at 37.5MHz and PHY clock at 75MHz for SATA-I
  - IP Core system clock at 75.0MHz and PHY clock at 150MHz for SATA-II
  - IP Core system clock at 150.0MHz and PHY clock at 300MHz for SATA-III
- Supports 1.5 Gbit/s, 3.0 Gbit/s and 6.0 Gbit/s data transfer rates
- Supports DMA and PIO commands
- Hardware support for
  - Speed auto negotiation forSATA I/II/III

  - 48-bit address set
  - Detection of OOB, COMWAKE, K28.5, etc.
  - 8b/10b coding and decoding
  - CRC generation and checking
  - Auto insertion of HOLD primitives
  - Native Command Queuing (NCQ)
  - Port Multiplier, Port Selector
  - First Party DMA (FPDMA)
- CONT primitive support for primitive suppression to reduce EMI
- Implements the shadow register block and the serial ATA status and control registers
- Supports both Xilinx GTP and GTX RocketIO Transceivers
- Reference design available for ML605, AC701 and KC705 Xilinx Evaluation Platforms

1.2 APPLICATIONS

- Hard Disk Drives (HDD)
- Solid State Drives (SDD)
- RAID controllers
- Data transfer and storage systems

1.3 DELIVERABLES

- Source code:
  - VHDL Source Code
- VHDL test bench environment
  - Tests with reference responses
- Technical documentation
  - Datasheet
  - Installation notes
  - User manual
- Instantiation templates
- Example design
- Technical Support
  - IP Core implementation support
- Variable length maintenance
  - Delivery of IP Core updates, minor and major changes
  - Delivery of documentation updates
  - Telephone & email support

1.4 LICENSING

Netlist License

- Post-synthesis netlist
- Implementation scripts
- Constraints
- Instantiation template
- Documentation

VHDL Source License

- VHDL RTL source code
- Complete verification plan together with the functional verification environment to verify the correct operation of the core
- Vectors for testing the functionality of the core
- Simulation & implementation scripts
- Documentation

1.5 SYMBOL

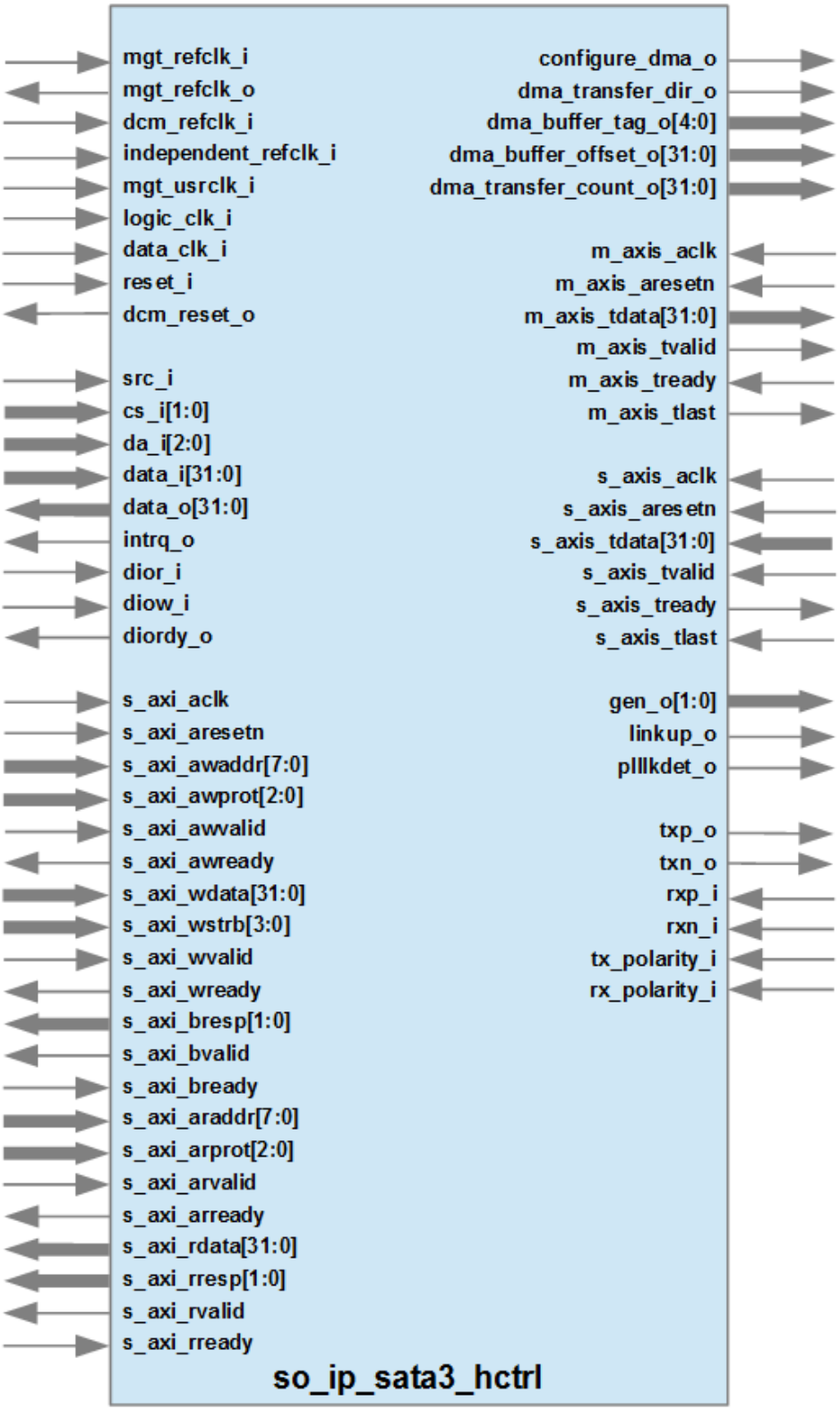

Figure 1.1: Symbol

1.6 PIN DESCRIPTION

Name	Signal Direction	Description
Global Clocks and Reset Ports		
mgt_refclk_i	Input	MGT reference clock input
mgt_refclk_o	Output	MGT reference clock output
dcm_refclk_i	Input	DCM reference clock input
independent_refclk_i	Input	Independent reference clock input used for GTX reset FSMs
mgt_userclk_i	Input	MGT user clock used in RX and TX modules
logic_clk_i	Input	Logic clock used in RX and TX modules
data_clk_i	Input	Data clock used in Physical, Link and Transport modules
reset_i	Input	Main reset
dcm_reset_o	Output	DCM reset signal
Analog Front End Interface		
txp_o	Output	The outbound high speed differential positive signal that is connected to the serial ATA cable
txn_o	Output	The outbound high speed differential negative signal that is connected to the serial ATA cable
rxp_i	Input	The inbound high speed differential positive signal that is connected to the serial ATA cable
rxn_i	Input	The inbound high speed differential negative signal that is connected to the serial ATA cable
tx_polarity_i	Input	MGT port used to invert the polarity of outgoing data.
rx_polarity_i	Input	MGT port used to invert the polarity of incoming data.
Legacy ATA Interface		
scr_i	Input	Indication that the shadow registers or SCRs are being accessed
cs_i[1:0]	Input	CS0 and CS1 chip select signals as defined in the ATA/ATAPI Specification
da_i[2:0]	Input	Device address bus as defined in the ATA/ATAPI Specification
data_i[31:0]	Input	Input data bus as defined in the ATA/ATAPI Specification
data_o[31:0]	Output	Output data bus as defined in the ATA/ATAPI Specification
intrq_o	Output	Interrupt request signal as defined in the ATA/ATAPI Specification
dior_i	Input	Read signal for shadow registers or the Data port
diow_i	Input	Write signal for shadow registers or the Data port

diordy_o	Output	Signal asserted by the SATA Host controller to indicate that data has either been captured or sourced to the data bus
AXI-Lite Interface		
s_axi_aclk	Input	Global clock signal for the interface
s_axi_aresetn	Input	Global reset signal. This signal is active LOW.
s_axi_awaddr	Input	Write address (issued by master, accepted by slave)
s_axi_awprot	Input	Write channel protection type. This port indicates the privilege and security level of the transaction and whether the transaction is a data access or an instruction access.
s_axi_awvalid	Input	Write address valid. This port indicates that the master signaling valid write address and control.
s_axi_awready	Output	Write address ready. This signal indicates that the slave is ready to accept an address and associated control signals.
s_axi_wdata	Input	Write data (issued by master, accepted by slave)
s_axi_wstrb	Input	Write strobes. This port indicates which byte lanes hold valid data. There is one write strobe bit for each eight bits of the write data bus.
s_axi_wvalid	Input	Write valid. This port indicates that valid write data and strobes are available.
s_axi_wready	Output	Write ready. This port indicates that the slave can accept the write data.
s_axi_bresp	Output	Write response. This port indicates the status of the write transaction.
s_axi_bvalid	Output	Write response valid. This port indicates that the channel is signaling a valid write response.
s_axi_bready	Input	Response ready. This port indicates that the master can accept a write response.
s_axi_araddr	Input	Read address (issued by master, accepted by slave)
s_axi_arprot	Input	Protection type. This port indicates the privilege and security level of the transaction, and whether the transaction is a data access or an instruction access.
s_axi_arvalid	Input	Read address valid. This port indicates that the channel is signaling valid read address and control information.

s_axi_arready	Output	Read address ready. This port indicates that the slave is ready to accept an address and associated control signals.
s_axi_rdata	Output	Read data (issued by slave)
s_axi_rresp	Output	Read response. This port indicates the status of the read transfer.
s_axi_rvalid	Output	Read valid. This port indicates that the channel is signaling the required read data.
s_axi_rready	Input	Read ready. This port indicates that the master can accept the read data and response information.
DMA Configuration Interface (Used with NCQ)		
configure_dma_o	Output	Indication that the DMA controller should be configured with new configuration data
dma_transfer_dir_o	Output	Indication of the direction of the upcoming DMA transfer
dma_buffer_tag_o[4:0]	Output	DMA buffer region in host memory that is selected for the data transfer
dma_buffer_offset_o[31:0]	Output	This is the byte offset into the buffer. Bits [1:0] shall be always set to zero.
dma_transfer_count_o[31:0]	Output	This is the number of bytes to be read or written. Bit zero shall be always set to zero.
DMA Data Interface		
RX DMA Interface		
m_axis_aclk	Input	AXI4-Stream clock signal
m_axis_aresetn	Input	AXI4-Stream reset signal
m_axis_tdata[31:0]	Output	Received data that should be sent to the DMA controller
m_axis_tvalid	Output	Data valid indicator
m_axis_tready	Input	Destination ready indicator
m_axis_tlast	Output	End-of-Frame indicator
TX DMA Interface		
s_axis_aclk	Input	AXI4-Stream clock signal
s_axis_aresetn	Input	AXI4-Stream reset signal
s_axis_tdata[31:0]		Data from the DMA controller that should be transmitted
s_axis_tvalid	Input	Data valid indicator
s_axis_tready	Input	Destination ready indicator
s_axis_tlast	Output	End-of-Frame indicator
SATA Controller Status Signals		
gen_o[1:0]	Output	Indication about the communication speed established between the SATA Host controller core and attached device.
linkup_o	Output	Indication that the link has been established successfully
plllkdet_o	Output	PLL lock detected.

1.7 BLOCK DIAGRAM

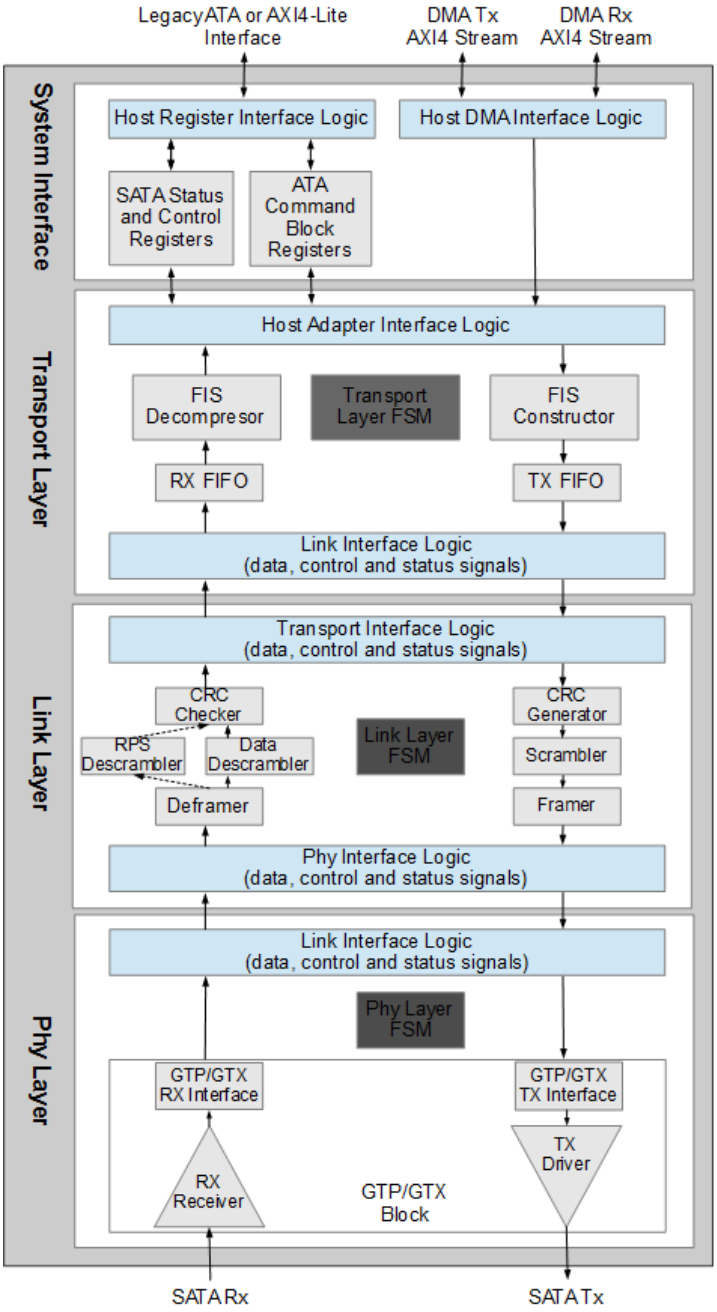

Figure 1.2: Block diagram

1.8 FUNCTIONAL DESCRIPTION

The previous diagram shows all major modules of the SATA Host Controller core that is described here in more detail.

System Interface Module

System interface module is not defined in the SATA specification. Its main purpose is to provide the easy access to the SATA Host Controller and to initiate the required SATA operations. This module is composed from the following sub-modules:

- ATA command block registers (CBRs); also know as shadow registers, are interface registers used for delivering commands to the device or posting status from the device.
- SATA status and control registers (SCRs); used for reporting additional status and error information and to allow

control of capabilities unique to Serial ATA.

- DMA interface logic; enabling the easy interface with the DMA controllers. This interface is based on the Xilinx's LocalLink Interface specification.
- Host register interface; enabling the access to the SATA host controller CBRs and SCRs.

Transport Layer Module

Transport layer module implements the functionality defined by the transport layer in the SATA specification. It is composed from the following sub-modules:

- FIS constructor; gathers FIS content based on the type of the requested FIS and places FIS into TX FIFO.
- FIS decomposer; determines the FIS type stored in the Rx FIFO, and distributes the FIS content into appropriate shadow registers or SATA status and control registers.
- RX FIFO and TX FIFO; buffers used to store the current FIS-es that are being processed.
- Transport layer FSM; control unit that implements all of the functionality as defined in SATA Specification, Chapter 10. Apart from this, transport layer control unit is also responsible for the following:
  - It notifies the Link layer of required frame transmission and passed FIS content to the Link layer.
  - Manages FIFO flow, notifies the Link layer of required flow control.
  - Receives the frame receipt acknowledge from the Link layer.
  - Reports good transmission/reception or errors to System Interface module.

Link Layer Module

Link layer module implements most of the functionality defined by the link layer in the SATA specification. This module doesn't implement 8b/10b coding and decoding, although it is defined in the link layer. These features are implemented in the Phy layer module. Link layer module is composed from the following sub-modules:

- CRC generator and checker; used for generating and checking of CRC values for outgoing and incoming frames.
- Data scrambler and descrambler as well as descrambler for repeated primitive suppression (RPS); these modules enable the EMI suppression features as defined in the SATA 3.2 specification.
- Framer; used to insert the frame envelope around Transport layer data (i.e. SOFP, CRC, EOFP, etc.).
- Deframer; used to remove framing envelope (i.e. SOFP, CRC, EOFP, etc.).
- Link layer FSM; control unit that implements all of the functionality as defined in Chapter 9 of the SATA Specification. In addition to this, link layer control unit is also responsible for the following:
  - Receives frame receipt acknowledge from peer Link layer.
  - Reports good transmission or Link/Phy layer errors to Transport layer.
  - Reports good reception or Link/Phy layer errors to Transport layer and the peer Link layer.

Phy Layer Module

Phy layer module implements the functionality defined by the Phy layer in the SATA Specification. Phy layer module is mostly implemented using the Xilinx's GTP/GTX block. Following functionality is implemented using the GTP/GTX block:

- 8b/10b encoding and decoding. These blocks implement functionality originally defined in the Link layer, but since they are part of the Xilinx GTP/GTX block, they are implemented inside the Phy layer module.
- Serialization and de-serialization of encoded data.
- Transmission and reception of 1.5 Gbps, 3.0 Gbps or 6.0 Gpbs differential NRZ serial stream at specified voltage levels.
- Extraction of data from the serial stream.
- Detection of comma character and bit and word alignment.

- OOB signaling detection and transmission.
- Provision of device status to Link layer.

Phy layer FSM is implemented outside the GTP/GTX block. This FSM implements the functionality defined in the Chapter 7 of the SATA Specification. Beside this it also implements the speed negotiation protocol between the host and device, as defined in Chapter 8 of the SATA Specification and transmits Phy status and errors to the Link layer.

1.9 VERIFICATION METHODS

SATA Host Controller core was tested both using sophisticated functional verification environment and in dedicated hardware platform. Verification environment, together with the developed verification plan, was used to extensively verify the SATA Host Controller core operation is in accordance with the SATA Specification 3.2. After reaching all functional verification goals, IP core was tested using dedicated hardware platforms, namely Xilinx's ML605, AC701 and KC705 Evaluation Platforms. Using these platforms SATA Host Controller core was implemented in FPGA and connected to various HDDs to test its operation in a real application and to estimate the performance of the core. The details about the verification methodology that was used and performance results during hardware testing can be obtained from So-Logic upon request.

1.10 &_performance DEVICE UTILIZATION & PERFORMANCE

Supported Family	Device	Slices	Slice LUTs	Slice FFs	IOs	BRAMs	MGTs
ISE Implementation Results							
Artix-7	XC7A200T-2	998	3468	2267	237	5	1
Kintex-7	XC7K325T-2	990	3512	2294	237	5	1
Virtex-7	XC7VX485-T-2	934	3490	2293	237	5	1
Zynq	XC7Z045-2	1002	3467	2293	237	5	1
Vivado Implementation Results							
Artix-7	XC7A200T-2	1365	3783	2303	237	5	1
Kintex-7	XC7K325T-2	1350	3871	2330	237	5	1
Virtex-7	XC7VX485-T-2	1470	3864	2329	237	5	1
Zynq	XC7Z045-2	1358	3864	2329	237	5	1

Notes:

1. All core I/O signals are routed off chip
2. Results were obtained using Xilinx ISE 14.7 and Vivado 2014.2 versions of software with defaults settings
3. The synthesis results provided are for reference only. Please contact So-Logic for estimates for your particular application. contact So-Logic for estimates for your particular application.

1.11 CONTACT INFORMATION

So-Logic

Lustkandlgasse 52/22

A-1090 Vienna

Austria/Europe

Phone: +43-1-3157777-11

Fax: +43-1-3157777-44

E-Mail: ip_sata3_hctrl@so-logic.net

URL: http://www.so-logic.net

1.12 REVISION HISTORY

The following table shows the revision history for this document.

Date	Version	Revision
01/09/14	1.0	Initial release.