

Features

- Lane Width of x1 Configuration
- Effective Raw Data Rate of 2.5Gbps/Lane/Direction (with Target Device Support)
- 8b/10b Encoding/Decoding for Symbols and Special Symbols (with Target Device Support)
- Data Scrambling/De-scrambling
- Link Initialization and Training
- Flow Control Initialization
- Data Integrity Checking for Both Data Link Layer Packets and Transaction Layer Packets
- Data Link Layer Retry Mechanism for Transmitted Transaction Layer Packets
- Acknowledgement and Timeout Replay Mechanisms
- All Error Statuses are Reported in the Backend User Interface

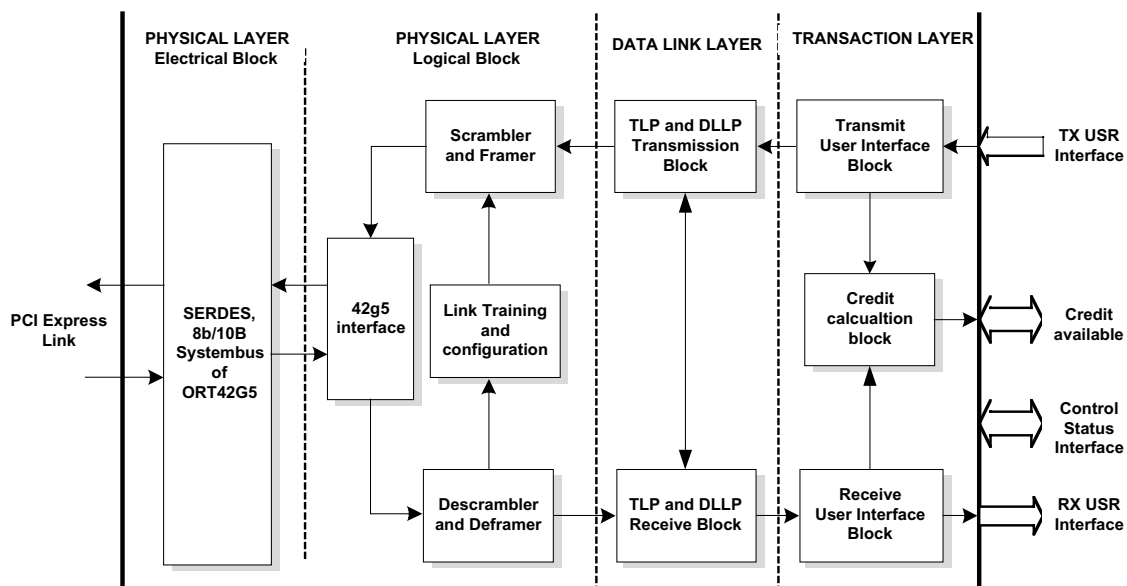
- Credit Availability Calculation and Reporting
- Documentation and Files
 - Data Sheet
 - User’s Guide
 - Lattice gate level netlist
 - Model for simulation
 - Core instantiation template
 - Testbench and testbench coding template

General Description

PCI Express is a high performance, general purpose Serial I/O Interconnect defined for a wide variety of future computing and communication platforms. The basic premise of PCI Express is that the host PCI software remains compatible with an endpoint device without new drivers or operating-system software. Salient PCI attributes, such as its usage model, load-store architecture, and software interfaces, are maintained, whereas its bandwidth-limiting and parallel bus implementation is replaced by a highly scalable, fully serial interface. PCI Express takes advantage of recent advances in point-to-point interconnects, switch-based technology, and packetized protocol to deliver new levels of performance and features.

Block Diagram

Figure 1. PCI Express Block Diagram



© 2003 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal. All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice. The product described herein is subject to continuing development, and applicable specifications and information are subject to change without notice. Such specifications and information are provided in good faith; actual performance is not guaranteed, as it is dependent on many factors, including the user's system design.

Lattice's PCI Express IP Core is an endpoint device supporting a x1 link. It consists of three layers of the endpoint device namely the Physical, Data Link and Transaction layers. This IP core targets the programmable array of the ORCA Series 4 ORT42G5 FPSC. The complete solution supports up to 2.5Gbps data rate as specified in *PCI Express Specification 1.0a*. For more information on this and other Lattice products, refer to the Lattice web site at www.latticesemi.com.

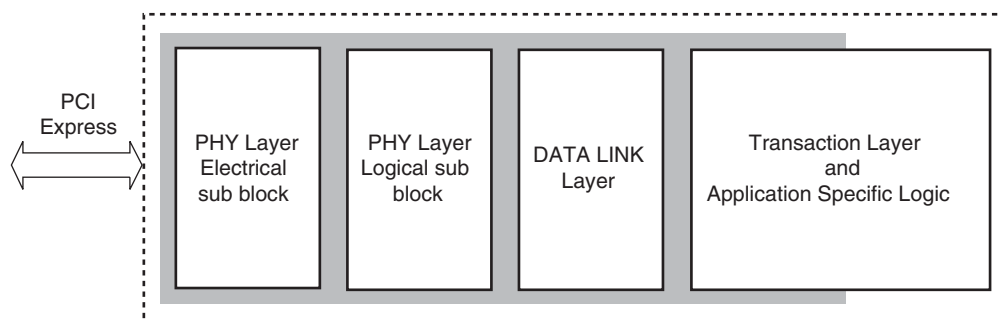
Functional Description

The Lattice PCI Express core is an implementation of the logical sub-block of Physical layer, Data Link Layer and part of the Transaction layer as specified in the PCI Express Base specification revision 1.0a.

- All configuration registers are left out of the core and all required status signals are made available at the User Interface side in order to facilitate an integrated easy implementation of configuration registers in the User Logic.
- The Lattice PCI Express core does not currently support L0s, L1, L2 and External Loop back states

The following figure gives a graphical indication of the layers implemented (shaded blocks) in this core.

Figure 2. PCI Express Layers Implemented



A typical implementation of these layers is better illustrated in the block diagram in Figure 1.

Physical Layer Implementation

Electrical Sub Block

The Electrical sub block of the Physical Layer is implemented in the Embedded logic of the FPSC. It includes one channel of SERDES and also 8b/10b logic.

Logical Sub Block

The Logical sub block of the Physical Layer is implemented in the programmable logic portion of the FPSC. This includes the Scrambler/De-scrambler, Framers / De-Framer, LTSSM block and LWLSN block. These blocks are described below:

Scrambler/De-Scrambler

The Scrambler/De-scrambler function is implemented using Linear Feedback Shift Registers. This is performed by serially XORing the 8-bit character with the 16-bit output of the LFSR. The final output stage of the LFSR is XORed with the lower bit of the data character and then both the LFSR and data register are serially advanced.

The LFSR implements the polynomial:

$$G(S) = X^{16} + X^5 + X^4 = X^3 + 1$$

Framer/De-Framer

The Framer/De-Framer implements the mechanism which uses special symbols like K28.2 (SDP) to start a DLLP, K27.7 (STP) to start a TLP, and K29.7 (END) to mark the end of either a TLP or a DLLP. When no packet informa-

tion or special ordered-sets are being transmitted, the transmitter will be sending idle data. The idle data consists of the data byte 0 (00h). During transmission of idle data the skip ordered set will continue to be transmitted.

TLPs are framed by placing a STP Symbol at the start of the TLP and an END Symbol or EDB Symbol at the end of the TLP. DLLPs are framed by placing an SDP Symbol at the start of the DLLP and an END Symbol at the end of the DLLP.

Link Training and Status State Machine (LTSSM)

The LTSSM is a set of interacting state machines implemented as per PCI Express base specification 1.0a section 4.2.5.

Link Width and Lane Sequence Negotiation (LWLSN)

The link width and lane sequence negotiation logic helps the LTSSM block to negotiate and finalize the link and lane number of the core as part of the training process. The core follows the steps as specified in the PCI Express spec 1.0a for an end point device.

Data Link Layer Implementation

The Data Link Layer tracks the state of the link. It communicates link status with the Transaction and Physical Layers, and performs Link Management through the Physical Layer.

Before starting normal operation following power-up or interconnect reset it is necessary to initialize flow control for the virtual channel. The TLP traffic starts only after this flow control initialization. This flow control initialization is implemented in the Data Link layer according to *PCI Express Specification 1.0a*.

Transmit Path

Transmit TLP Block

The Transmit TLP processing block implements the Data Link Layer requirements for TLPs; sequence number generation mechanism and LCRC calculation. It appends the information to the TLP that is getting transmitted. Also it implements the mechanism to resend the same TLP if a retry is requested from the receiver component. To support this mechanism it has to store the whole TLP that is being transmitted until this component receives an ACK for that particular TLP or until time-out indication occurs.

Transmit DLLP Block

The Transmit DLLP processing block, generates ACK or NAK DLLPS based on the instructions from the Receive TLP processing block. This block also generates InitFC DLLPs corresponding to FCI requirements. Appropriate power management DLLPs are generated as instructed by external power management logic. This block consists of a 16-bit CRC generation logic that adds the calculated CRC to the tail of all transmit DLLPs to meet data integrity requirements.

Receive Path

Receive TLP Block

Receive TLP processing block implements sequence number checking, data integrity checking, and through LCRC checking for received TLPs. It includes an appropriate retry mechanism to respond to any errors detected in received TLPs. This block is provided with a buffering mechanism to hold the whole TLP until its data integrity is checked and an appropriate instruction is generated to the Transmit DLLP block to send ACK and NAK DLLPs. Associated retry logic implements the time-out mechanism as specified in the PCI Express specification. After all these Data Link Layer requirements for a TLP are met the TLPs are sent out over Receive TLP interface to the transaction layer.

Receive DLLP Block

The main function implemented in this block is related to data integrity checking, detection of different types of DLLPs, extraction of different fields of DLLPs, ACK and NAK processing.

In case of CRC error the DLLP is discarded and the error is transmitted to the user logic.

Transaction Layer Implementation

Transmit Path

The Transmit Path acts as a bridge between the Data Link Layer and User Interface in the transmit path. This module provides handshake between the core and user logic request to transmit TLPs and PM DLLPs.

In addition, this module performs “Credit Available” calculation by monitoring the types of TLPs transmitted and the credit values received from the other node through Update FC DLLPs. It accumulates these credit values separately for PH, NPH, PD and NPD types and makes them available to the User Interface side based on the selected packet type as indicated in the input signal `pkt_type`. Refer to the Credit Available Calculation section for more details.

Receive Path

This module consists of all logic to provide handshake between the core and user interface for received TLPs. It checks for errors in the received TLPs and notifies the user interface side in case of errors like Poisoned TLP, Unsupported TLP type and Malformed TLP

Credit Available Calculation

The core maintains the value of Available Credit for each packet type. User logic can fetch this Credit Available value from the core by appropriately selecting the packet type in the input signal `pkt_type`. The core outputs the Available Credit in the output signal `crdt_avail` for the selected packet type. Please refer to the Transmit User Interface section under Timing Diagrams for more detail.

Error Conditions

Receive logic of the core checks the received DLLP and TLPs for various error conditions. In case of error, associated signals are asserted to the user logic and the DLLP or TLP is processed as per the specification requirement. Details of the error conditions are described below:

Unsupported Request

If the core receives any type of unsupported TLP request, an error is notified.

Poisoned TLP

There is no support to generate Data Poisoned TLP for transmission. But the core supports Data poisoning checks in the received TLP.

Malformed TLP

Malformed TLP detection is notified and the core transmits the error in the user interface signal.

Bad TLP

This error signal is generated to indicate that the received TLP is erroneous when there is no `rx_err` signal from the Electrical Sub Block of the Physical Layer.

Bad DLLP

This error signal is generated to indicate that the received DLLP is erroneous when there is no `rx_err` signal from the Electrical Sub Block of the Physical Layer.

Data Link Layer Protocol Error

The Data Link Layer Protocol Error is generated for the following conditions in a received DLLP:

- If $((\text{NEXT_TRANSMIT_SEQ} - 1) - \text{AckNak_Seq_Num}) \bmod 4096 > 2048$, or

- If the above error condition is not noticed, but $(ACK_{Nak_Seq_Num} - ACKD_SEQ) \bmod 4096 \Rightarrow 2048$

Replay Number Rollover

The Replay Number Rollover status signal is a pulse of one clock duration and is generated when the internal replay number changes from 3 to 4.

Replay Timeout

The Replay time is the time between transmission of the last DWORD of a TLP and reception of the first DWORD of an ACK/NACK DLLP for that TLP.

Register Description

PCI Express configuration space is not implemented in this core. But the core checks all TLPs related to the configuration access and appropriate action, as specified, is carried out in case of errors. It is left to the user logic to implement the necessary PCI Express registers as part of the design. All status and error signals are provided at the user interface to connect them to proper registers in the user logic.

User Configurable Parameters

The PCI Express core has all the three layers of the Endpoint device that is implemented in one configuration. Therefore there are no other configurations of this core.

Signal Descriptions

Table 1. PCI Express Core Signal Descriptions

Port Name	Active State	I/O	Signal Description
rst_n	Low	I	Asynchronous System Reset
div2_pclk	—	I	62.5MHz System Clock
PCI Express Line Interface (Inputs)			
REFCLKN_A	—	I	156MHz Differential Reference Clock for SERDES Block A
REFCLKP_A	—	I	156MHz Differential Reference Clock for SERDES Block A
REFCLKN_B	—	I	156MHz Differential Reference Clock for SERDES Block B
REFCLKP_B	—	I	156MHz Differential Reference Clock for SERDES Block B
USR_CLK	—	I	50 MHz clock for system bus
HDINN_AC	—	I	Rx Data input – SERDES Quad A, Channel C
HDINP_AC	—	I	Rx Data input – SERDES Quad A, Channel C
HDINN_BC	—	I	Rx Data input – SERDES Quad B, Channel C
HDINP_BC	—	I	Rx Data input – SERDES Quad B, Channel C
PASB_PDN	—	I	Active Low Power Down Input
PASB_RESETN	—	I	Active Low Reset for the Embedded Core
PASB_TESTCLK	—	I	Clock Input for BIST and Loopback Test
PASB_TRISTN	—	I	Active low 3-state for Embedded Core Output Buffers
PBIST_TEST_ENN	—	I	Selection of PASB_TESTCLK Input for BIST Test
PLOOP_TEST_ENN	—	I	Selection of PASB_TESTCLK Input for Loopback Test
PMP_TESTCLK	—	I	Clock Input for Microprocessor in Test Mode
PMP_TESTCLK_ENN	—	I	Selection of PMP_TESTCLK in Test Mode
PSYS_DOBISTN	—	I	Input to Start BIST Test
PCI Express Line Interface (Outputs)			
HDOUTN_AC	—	O	Tx Data input - SERDES Quad A, Channel C
HDOUTP_AC	—	O	Tx Data input - SERDES Quad A, Channel C
HDOUTN_BC	—	O	Tx Data input - SERDES Quad B, Channel C

Table 1. PCI Express Core Signal Descriptions (Continued)

Port Name	Active State	I/O	Signal Description
HDOUTP_BC	—	O	Tx Data input - SERDES Quad B, Channel C
REXT_A	—	O	Reference Resistor – SERDES Quad A
REXT_B	—	O	Reference Resistor – SERDES Quad B
PSYS_RSSIG_ALL	—	O	Output result of BIST Test
External Circuit Interface			
start_rx_detect	—	O	Poll rx_detect. Equivalent to TxDetectRx/Loopback in PIPE Spec
rx_detect	High	I	Indicates PCI_Express Link presence is Detected. Equivalent to RxStatus=011b in PIPE Spec
RX_USR Interface			
rxtlpu_data[31:0]	—	O	Received TLP Data to Transaction Layer
rxtlpu_st	High	O	Start of Received TLP on the “rxtlpu_data”
rxtlpu_sd	High	O	Start of Data in the Received TLP
rxtlpu_end	High	O	End of TLP on the “rxtlpu_data”
rxpm_dllp_type	—	O	Type of Received PM DLLP
rxpm_dllp_val	High	O	Valid Signal to Sample “rxpm_dllp_type”
pois_tlp	High	O	Poisoned TLP received indication
un_sup_req	High	O	Unsupported Request Received Indication
malf_tlp	High	O	Malformed TLP Received Indication
bad_tlp	High	O	Bad TLP Received Indication
bad_dllp	High	O	Bad DLLP Received Indication
dll_perr	High	O	Data Link Layer Protocol Error Indication
rnum_rlor	High	O	REPLY_NUM Rollover Indication
rply_tout	High	O	Replay Timer Timeout Indication
TX_USR Interface			
txtlpu_req	High	I	User Interface Request to Send TLPs
txtlpu_data[31:0]	—	I	TLP Data to be Transmitted on PCI Express Link
txtlpu_st	High	I	Start of TLP Signal that Indicates Start of New TLP on the txtlpu_data Bus
txtlpu_end	High	I	End of TLP Signal that Indicates End of TLP on the txtlpu_data Bus
txtlpu_nlfy	High	I	User Interface Request to Generate a Nullified TLP This Signal is to be Asserted Along With End of TLP
no_pcie_training	High	I	No LTSSM Training to be performed
pkt_type	—	I	Packet type for credit info
txpm_dllp_type	—	I	PM packet type to transmit
tx_pm	High	I	Send PM DLLP, with PM data in “tx_data”
l_retrain_link	High	I	Initiate Link Retraining
l_ext_sync	High	I	Extended syn
loc_lnk_cntl[3:0]	High	I	Link Control bits from User Interface Bit 0 – Set Hot_Reset bit in TS1/TS2 Will switch the Core to Reset state if current state is Recovery Bit 1 – Set Disable_Link bit in TS1/TS2 Will switch the Core to Disable state if current state is Recovery Bit 2 – Set Loopback bit in TS1/TS2 Will switch the Core to Tx_Loopback state if current state is Recovery Bit 3 – Set Disable_scramble bit in TS1/TS2
l_go_config	High	I	Direct the Core to Switch to Configuration State
txtlp_rdy	High	O	The Data Link Layer is Ready to Accept TLPs from Transaction Layer

Table 1. PCI Express Core Signal Descriptions (Continued)

Port Name	Active State	I/O	Signal Description
crdt_avail	—	O	Credit Available for the Packet Type

Note2:

1. For all PCI Express Line Interface signals (both input and output) refer to the ORT42G5 and ORT82G5 data sheet for their usage.
2. External Circuit Interface: rx_detect signal to be generated by an external circuit whenever the start_rx_detect signal is asserted. Refer to the Receiver Detection section in "PHY Interface for the PCI Express Architecture".
3. For usage of loc_lnk_cntl and l_go_config signals refer to the Link Training and Status State Machine section of the PCI Express specifications.

PCI Express Core Design Flow

Lattice has created a detailed software IP tutorial available on the Lattice web site at www.latticesemi.com. Both a simple IP module evaluation and tutorial and a more detailed ispLeverCORE™ tutorial are available for download. Type "tutorial" in the Lattice web site search engine.

Custom Core Configurations

For PCI Express core configurations that are not available in the Evaluation Package, please contact your Lattice sales office to request a custom configuration.

Related Information

For more information regarding core usage and design verification, refer to the *PCI Express User's Guide*, available on the Lattice web site at www.latticesemi.com.

Appendix for ORCA[®] Series 4 FPSCs – ORT42G5

Table 2. Performance and Resource Utilization¹

Name of Parameter File	ORCA 4 PFUs	LUTs	Registers	EBR	PIO	f _{MAX} (MHz)
pci_exp_t42g5_1_001.lpc	937	3,406	3,078	10	116	66.7

1. Performance and utilization characteristics are generated using an ORT42G5-2BM484. When using this IP core in a different density, package, speed, or grade within the ORCA 4 family, performance and utilization may vary.

Supplied Netlist Configurations

The Ordering Part Number (OPN) for all configurations of this core is PCI-EXP-T42G5-N1. Table 2 lists the netlists available as Evaluation Packages for the ORCA Series 4 devices, which can be downloaded from the Lattice web site at www.latticesemi.com.

To load the preset parameters for this core, click on the “Load Parameters” button inside the IP Manager tool. Make sure that you are looking for a file inside of this core’s directory location. The Lattice Parameter Configuration files (.lpc) are located inside this directory.