



## LatticeECP2M™ PCI Express x1 Endpoint IP Core

---

User's Guide

## Introduction

PCI Express is a high performance, scalable, well-defined standard for a wide variety of computing and communications platforms. It has been defined to provide software compatibility with existing PCI drivers and operating systems.

Lattice's PCI Express IP core provides a x1, endpoint solution from the electrical SERDES interface to the transaction layer. The LatticeECP2M PCI Express IP core utilizes Lattice's PCS/SERDES built in the LatticeECP2M devices to support the physical layer.

## Features

The LatticeECP2M PCI Express IP core supports the following features.

### Top Level IP Support

- 100 MHz reference clock input
- 125 MHz 16-bit data path user interface
- Credit interface for transmit and receive as well as for PH, PD, NPH, NPD, CPLH, CPLD credit types
- Higher layer control of LTSSM via ports
- Access to select configuration space information via ports
- Supported in -6 and -7 LatticeECP2M speed grades
- Utilizes ~5.5K LUTs and 4 block RAMs (single VC)

### Configuration Space Support

- PCI-compatible Type00h Configuration Space registers contained inside core
- Power Management Capability Structure registers contained inside core
- MSI Capability Structure registers contained inside core
- PCI Express Capability Structure registers contained inside core
- Extended Capabilities register for virtual channel support contained inside core
- Device Serial Number register contained inside the core.

### Transaction Layer

- Supports all types of TLPs (memory, I/O, configuration, and message)
- Power management user interface to easily send power messages
- Virtual channel support of 1 to 8 channels (separate user ports per channel)
- Flow control enforcement with separate credit interface per VC
- Optional ECRC generation/checking
- Supports 512, 1K, 2K and 4K maximum payload size TLPs
- Configuration response is generated by the core.

### Data Link Layer

- Data Link Control and Management State Machine
- Flow Control Initialization
- Ack/Nak DLLP generation/termination

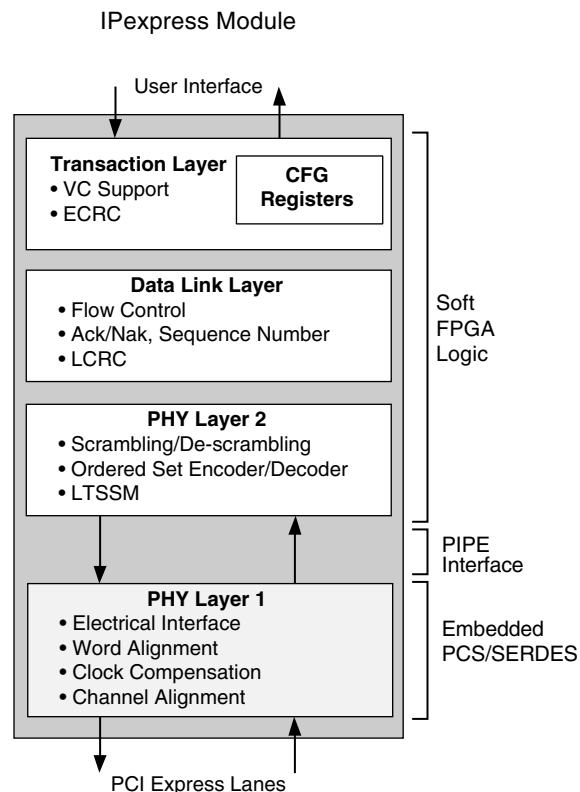
- Power Management DLLP generation/termination through simple user interface
- LCRC generation/checking
- Sequence number appending/checking/removing
- Retry buffer and retry management

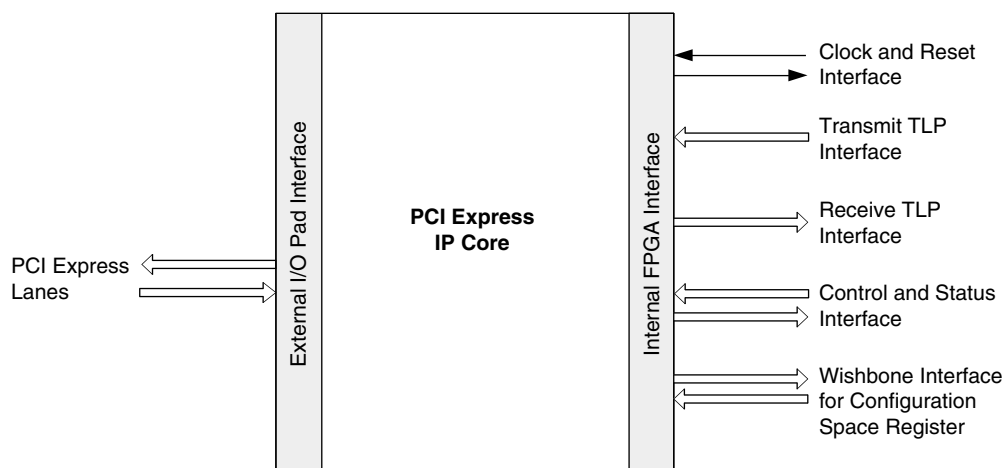
### PHY Layer Features

- 2.5Gbps CML electrical Interface
- Serialization and de-serialization
- 8b10b symbol encoding/decoding
- Link State Machine for symbol alignment
- Clock tolerance compensation supports +/- 300ppm
- Framing and application of symbols to lanes
- Data scrambling
- Link Training and Status State Machine (LTSSM)
  - Electrical idle generation
  - Receiver detection
  - TS1/TS2 generation/detection
  - Lane polarity inversion
  - Higher layer control to jump to defined states

## Block Diagrams

**Figure 1. PCI Express IP Core Technology and Functions**



**Figure 2. PCI Express Interfaces**

## Functional Description

The LatticeECP2M PCI Express x1 Endpoint IP core is implemented in different FPGA technologies. These technologies include soft FPGA fabric elements such as LUTs, registers, and embedded block RAMs (EBRs). Embedded hard elements within the LatticeECP2M PCS/SERDES are also utilized. The IP interfaces to the Embedded PCS/SERDES using the PIPE interface.

The ispLEVER® design tool IPexpress™ is used to customize and create the complete IP module for the user to instantiate in their design. Inside the module created by IPexpress are several blocks implemented in heterogeneous technologies. All of the connectivity is provided allowing the user to interact at the top level of the IP core.

Figure 1 provides a high level block diagram to illustrate the main functional blocks and the technology used to implement the PCI Express functions.

As the PCI Express core proceeds through the ispLEVER design flow specific technologies are targeted to their specific locations on the device. Figure 2 provides an implementation representation of a LatticeECP2M device with the PCI Express core.

## Parameter Descriptions

Table 1 shows the different parameters used by the IP.

**Table 1. PCI Express Parameters**

Parameter	Description	Legal Value
LANE_WIDTH	Defines the lane width of the IP. The supported value is x1.	X1
ENDPOINT_COMP	Defines the topology of the IP. The supported value is only as an endpoint.	ENDPOINT_COMP
MAX_TLP	Defines the maximum TLP supported by the IP. The supported values are 512, 1K, 2K and 4k bytes.	512,1K,2K,4K
NUM_VC	Defines the number of Virtual Channels supported by the IP. The supported values are 1, 2, 3, 4, 5, 6, 7, 8.	1, 2, 3, 4, 5, 6, 7, 8
LPEVCC	Defines the Low Priority Extended VC Count (Port VC Capability Reg1). Allowed values are = 3'b000 to `NUM_VC-1.	Max limit is NUM_VC-1
ECRC	Defines whether the ECRC generation and check logic is present or not present.	Enable/Disable
WISHBONE	Defines whether the Wishbone interface is present or not present.	Enable/Disable
EN_BAR0/1/2/3/4/5/6	Defines whether the BAR registers 0/1/2/3/4/5/6 are present or not present.	Enable/Disable
EN_ER_BAR	Defines whether the expansion ROM is present or not present.	Enable/Disable

## Signal Description

Table 2 provides the list of ports and descriptions for the PCI Express IP core. When using multiple Virtual Channels (VCs) each VC has a separate port. These ports are discussed in the Multiple VC section of this user's guide.

*Note: When generating a custom IP core via IPexpress, certain port(s) listed below may not appear in the final IP core generated. This is because the chosen IP configuration does not need the port(s).*

**Table 2. PCI Express Port List**

Port Name	Direction	Clock	Description
<b>Clock and Reset Interface</b>			
clk_100	Input		100MHz reference clock to the physical layer
sys_clk_125	Output		125MHz clock derived from refclk to be used in the user application
rst_n	Input		Active-low asynchronous reset
<b>PCI Express Lanes</b>			
hdinp0	Input		PCI Express 2.5Gbps CML inputs
hdinn0	Input		PCI Express 2.5Gbps CML inputs
hdoutp0	Output		PCI Express 2.5Gbps CML outputs
hdoutn0	Output		PCI Express 2.5Gbps CML outputs
<b>Transmit TLP Interface</b>			
tx_data_vc0[vc1..vc7][15:0]	Input	sys_clk_125	Transmit data bus [15:8] Byte N [7:0] Byte N+1
tx_req_vc0[vc1..vc7]	Input	sys_clk_125	Active high transmit request.
tx_rdy_vc0[vc1..vc7]	Output	sys_clk_125	Active high transmit ready indicator. tx_st should be provided next clock cycle after tx_rdy is high.
tx_st_vc0[vc1..vc7]	Input	sys_clk_125	Active high transmit start of TLP indicator
tx_end_vc0[vc1..vc7]	Input	sys_clk_125	Active high transmit end of TLP indicator
tx_nlfy_vc0[vc1..vc7]	Input	sys_clk_125	Active high transmit nullify TLP. Can occur anywhere during the TLP.
tx_ca_[ph,nph,cplh]_vc0 [vc1..vc7] [8:0]	Output	sys_clk_125	Transmit Interface credit available bus. Ph - Posted header Nph - Non-posted header Cplh - Completion header [7:0] - Indicates credit value. [8] - Indicates infinite credits when high
tx_ca_[pd,npd,cpld] [vc1..vc7] [12:0]	Output	sys_clk_125	Transmit interface credit available bus. [11:0] - Indicates credit value. [12] - Indicates infinite credits when high
<b>Receive TLP Interface</b>			
rx_data_vc0 [vc1..vc7] [15:0]	Output	sys_clk_125	Receive data bus [15: 8] Byte N [7: 0] Byte N+1
rx_st_vc0[vc1..vc7]	Output	sys_clk_125	Active high receive start of TLP indicator
rx_end_vc0[vc1..vc7]	Output	sys_clk_125	Active high receive end of TLP indicator
rx_ecrc_err_vc0[vc1..vc7]	Output	sys_clk_125	Active high ECRC error indicator. Indicates an ECRC error in the current TLP. Present only if ECRC is enabled.
rx_us_req_vc0[vc1..vc7]	Output	sys_clk_125	Active high unsupported request indicator. Asserted if any of the following TLP types are received: Memory Read Request-Locked, I/O Request Configuration Read/Write Type 1, the TLP is still passed to the user.

**Table 2. PCI Express Port List**

Port Name	Direction	Clock	Description
rx_malf_tlp_vc0[vc1..vc7]	Output	sys_clk_125	Active high malformed TLP indicator. Indicates a problem with the current TLPs length or format.
rx_bar_hit[6:0]	Output	sys_clk_125	Active high BAR indicator for the current TLP. If this bit is high the current TLP on the receive interface is in the address range of the defined BAR. [6] - Expansion ROM [5] - BAR5 [4] - BAR4 [3] - BAR3 [2] - BAR2 [1] - BAR1 [0] - BAR0
[ph,pd,nph,npd,cplh,cpld]_buf_status_vc0[vc1 ..vc7]	Input	sys_clk_125	Active high user buffer full status indicator.
[ph,pd,nph,npd, cplh,cpld]_processed_vc0[vc1 ..vc7]	Input	sys_clk_125	Active high indicator to inform the IP core of how many credits have been processed. Each clock cycle high counts as one credit processed.
<b>Control and Status</b>			
bus_num[7:0]	Output	sys_clk_125	Bus number supplied with configuration write
dev_num[4:0]	Output	sys_clk_125	Device number supplied with configuration write
func_num[2:0]	Output	sys_clk_125	Function number supplied with configuration write
pm_power_state[1:0]	Output	sys_clk_125	Power state bits of register at 044h
pme_en	Output	sys_clk_125	PME_En at 044h
mm_enable[2:0]	Output	sys_clk_125	Multiple message enable bits of register at 048h
msi_enable	Output	sys_clk_125	MSI enable bits of register at 048h
mes_laddr[31:0]	Output	sys_clk_125	Message lower address from register at 04Ch
mes_uaddr[31:0]	Output	sys_clk_125	Message upper address from register at 050h
mes_data[15:0]	Output	sys_clk_125	Message Data from register at 054h
cmd_reg_out[3:0]	Output	sys_clk_125	Bits 10, 8, 6, 2 from register 004h
dev_cntl_out[14:0]	Output	sys_clk_125	Device control register at 060h
lnk_cntl_out[7:0]	Output	sys_clk_125	Link control register at 068h
ftl_err_out	Output	sys_clk_125	Active high fatal error indicator.
nftl_err_out	Output	sys_clk_125	Active high non-fatal error indicator.
cor_err_out	Output	sys_clk_125	Active high correctable error indicator.
dl_inactive	Output	sys_clk_125	Data Link Control SM is in INACTIVE state
dl_init	Output	sys_clk_125	Data Link Control SM is in INIT state
dl_active	Output	sys_clk_125	Data Link Control SM is in ACTIVE state
dl_up	Output	sys_clk_125	Data Link Control SM is UP
ecrc_gen_enb	Input	Asynch	Active high ECRC generation enable. This is present if ECRC logic is enabled.
ecrc_chk_enb	Input	Asynch	Active high ECRC checking enable. This is present if ECRC logic is enabled.
sts_reg_in[15:0]	Input	sys_clk_125	Status register input bits.
cmpln_tout	Input	sys_clk_125	Completion time out.
cmpltr_abort	Input	sys_clk_125	Completer abort.
unexp_cmpln	Input	sys_clk_125	Unexpected completion.
np_req_pend	Input	sys_clk_125	Non-posted request is pending.
pme_status	Input	sys_clk_125	PME status to reg 044h.

**Table 2. PCI Express Port List**

Port Name	Direction	Clock	Description
us_req_in	Input	sys_clk_125	User Unsupported req In
phy_ltssm_state[3:0]	Output	sys_clk_125	Phy Layer LTSSM current state 0000 - Detect 0001 - Polling 0010 - Config 0011 - L0 0100 - L0s 0101 - L1 0110 - L2 0111 - Recovery 1000 - Loopback 1001 - Hot Reset 1010 - Disabled
tx_lbk_data[15:0]	Input	sys_clk_125	TX user master loopback data
tx_lbk_kcctl[1:0]	Input	sys_clk_125	TX user master loopback control
tx_lbk_rdy	Output	sys_clk_125	TX loopback is ready to accept data
rx_lbk_data[15:0]	Output	sys_clk_125	RX user master loopback data
rx_lbk_kcctl[1:0]	Output	sys_clk_125	RX user master loopback control
<b>Wishbone Interface Enabled</b>			
CLK_I	Input	CLK_I	100MHz clk to wishbone
SEL_I [3:0]	Input	CLK_I	Bank selection to wishbone
WE_I	Input	CLK_I	Write(1)/read(0) to wishbone
STB_I	Input	CLK_I	Strobe signal to wishbone
CYC_I	Input	CLK_I	Cycle signal to wishbone
DAT_I[ 31:0 ]	Input	CLK_I	Write data to wishbone
ADR_I[ 12:0 ]	Input	CLK_I	Address to wishbone
CHAIN_RDAT_in[ 31:0 ]	Input	CLK_I	Daisy chain data_in from previous slave
CHAIN_ACK_in	Output	CLK_I	Daisy chain ack_in from previous slave
ACK_O	Output	CLK_I	Ack signal from wishbone
IRQ_O	Output	CLK_I	Interrupt ack signal from wishbone
DAT_O[ 31:0 ]	Output	CLK_I	Read data from wishbone
<b>Wishbone Interface Disabled</b>			
hl_snd_beacon	Input	sys_clk_125	Active high to send beacon
hl_disable_scr	Input	sys_clk_125	Active high to set the disable scrambling bit in the TS1/TS2 sequence.
hl_gto_dis	Input	sys_clk_125	Active high request to go to Disable state when LTSSM is in Config or Recovery.
hl_gto_det	Input	sys_clk_125	Active high request to go to Detect state when LTSSM is in L2 or Disable.
hl_gto_hrst	Input	sys_clk_125	Active high request to go to Hot Reset when LTSSM is in Recovery
hl_gto_l0stx	Input	sys_clk_125	Active high request to go to L0s when LTSSM is in L0
hl_gto_l1	Input	sys_clk_125	Active high request to go to L1 when LTSSM is in L0
hl_gto_l2	Input	sys_clk_125	Active high request to go to L2 when LTSSM is in L0
hl_gto_l0stxfts	Input	sys_clk_125	Active high request to go to L0s and transmit FTS when LTSSM is in L0s
hl_gto_lbk	Input	sys_clk_125	Active high request to go to loopback when LTSSM is in Config or Recovery

**Table 2. PCI Express Port List**

Port Name	Direction	Clock	Description
hl_goto_rcvry	Input	sys_clk_125	Active high request to go to Recovery when LTSSM is in L0, L0s or L1
hl_goto_cfg	Input	sys_clk_125	Active high request to go to CFG state
no_pcie_train	Input	Async	Active high signal disables LTSSM training and forces the LTSSM to L0 as a x4 configuration. This is intended to be used in simulation only.
tx_dllp_val[1:0]	Output	sys_clk_125	Req for Sending PM/Vendor type DLLP bit [0] is for PM bit [1] is for Vendor DLLP
tx_pmtype[2:0]	Input	sys_clk_125	Type field of transmitted PM DLLP
tx_vsd_data[23:0]	Input	sys_clk_125	Vendor Type DLLP contents
tx_dllp_sent	Output	sys_clk_125	Requested PM DLLP is sent
rxdp_pmd_type[2:0]	Output	sys_clk_125	Type field of received PM DLLP
rxdp_vsd_data[23:0]	Output	sys_clk_125	Vendor-specific DLLP data.
rxdp_dllp_val[1:0]	Output	sys_clk_125	PM/vendor-specific DLLP valid bit [0] is for PM bit [1] is for Vendor DLLP
<b>Configuration Space Register Interface / Constant Values Interface</b>			
LINK_NUM[7:0]	Input	Async	Initial link number sent in TS1/TS2
N_FTS_INC[4:0]	Input	Async	Number of FTS increment
SKP_INS_CNT[9:0]	Input	Async	Skip insert count
INITFC_TIMER [11:0]	Input	Async	Init FC timer value
ACKNAK_LAT_TIME[13:0]	Input	Async	Ack/Nak latency timer control. This control sets the number of 125MHz clock cycles between Ack/Nak DLLP generation. Ack/Nak generation only occurs between TLPs so the worst case latency time would be ACKNAK_LAT_TIME + MAX TLP packet length.
INIT_[PH, NPH, CPLH]_FC[7:0]	Input	Async	Initial number of receive credits available for the credit type
INIT_[PD, NPD, CPLD]_FC[11:0]	Input	Async	Initial number of receive credits available for the credit type.
UPDATE_FREQ_[PH,NPH,CPLH][4:0]	Input	Async	Number of headers that must be processed before a UpdateFC is sent. Initial value can be 8 (0x10).
UPDATE_FREQ_[PD,NPD,CPLD][11:0]	Input	Async	Number of DWORDS of data that must be processed before a UpdateFC is sent. Initial value can be 1024 (0x400).
UPDATE_TIMER[11:0]	Input	Async	Number of 125 MHz clock cycles between UpdateFC DLLP generation. Initial value can be 4095 (0xFFFF).
INIT_REG_[000..003C][31:0]	Input	Async	Initializes PCI Base configuration space registers 0x0-0x3F
INIT_REG_[040..044][31:0]	Input	Async	Initializes PCI PM space registers 0x40-0x44
INIT_PM_DS_DATA[0..7][9:0]	Input	Async	Initializes PCI PM data registers
INIT_REG_[048..054][31:0]	Input	Async	Initializes PCI Express MSI registers
INIT_REG_[058..068][31:0]	Input	Async	Initializes PCI Express capability
INIT_REG_[104..108][31:0]	Input	Async	Initializes PCI Express device serial number registers
INT_REG_[10C..17C][31:0]	Input	Async	Initializes PCI Express virtual channel capability registers
<b>Test Ports</b>			
tlp_debug	Input	Async	Reserved for test. Connect to logic 0.



**Table 2. PCI Express Port List**

Port Name	Direction	Clock	Description
force_lsm_active	Input	Async	Reserved for test. Connect to logic 0.
force_rec_ei	Input	Async	Reserved for test. Connect to logic 0.
force_phy_status	Input	Async	Reserved for test. Connect to logic 0.
force_disable_scr	Input	Async	Reserved for test. Connect to logic 0.

## Interface Description

This section describes the data path user interfaces of the IP core. Both transmit and receive interfaces use the TLP as the data structure. The lower layers attach the start, end, sequence number, and crc.

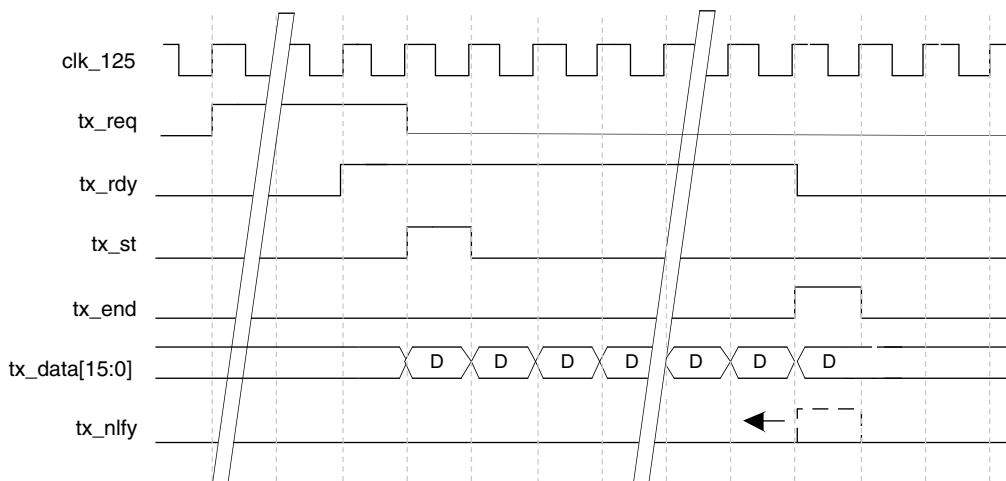
### Transmit TLP Interface

In the transmit direction the user must first check the credits available of the far end before sending the TLP. This information is found on the tx\_ca\_[ph,pd,nph,npd,cplh,cpld] bus. There must be enough credits available for the entire TLP to be sent.

The user then needs to check that the core is ready to send the TLP. This is done by asserting the tx\_req port and waiting for the assertion of tx\_rdy. When tx\_rdy is asserted the next clock cycle should provide the first 16-bit data of the TLP and assert tx\_st.

Tx\_rdy will remain high until once clock cycle before the last clock cycle of TLP data (based on the length field of the TLP). This allows the tx\_rdy to be used as the read enable of a FIFO containing the TLP(s) to be transmitted.

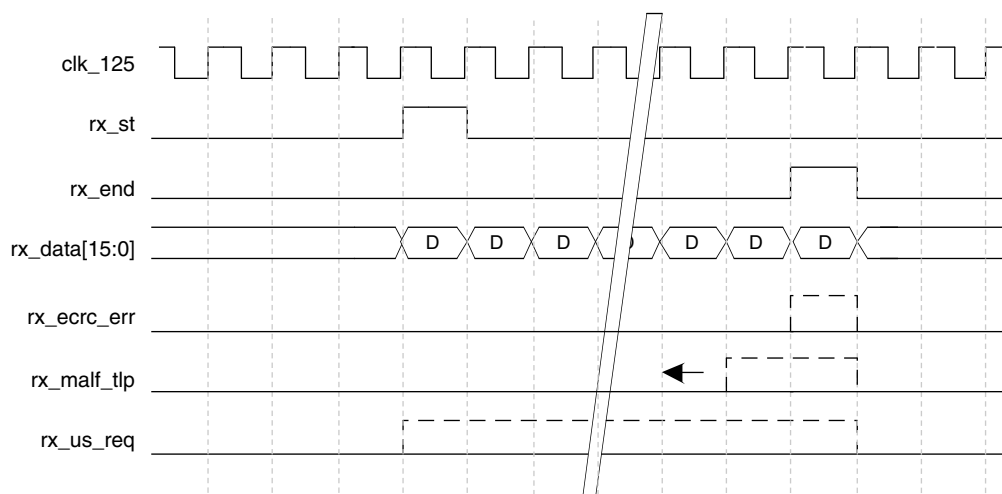
Figure 3 provides a timing diagram for the tx interface signals.

**Figure 3. Transmit Interface Timing Diagram**

### Receive TLP Interface

In the receive direction TLPs will come from the core as they are received on the PCI Express lanes. Config reads and config write TLPs will be terminated inside the core. All other TLPs will be provided to the user. Also, if the core enables any of the BARs the TLP will go through a BAR check to make sure the TLPs address is in the range of any programmed BARs before it is sent to the user. If the BAR check fails the TLP will be discarded by the core.

When a TLP is sent to the user the rx\_st signal will be asserted with the first 16-bit word of the TLP. The remaining TLP data will be provided on consecutive clock cycles until the last word with rx\_end asserted. If the TLP contains a ECRC error the rx\_ecrc\_err signal will be asserted at the end of the TLP. If the TLP has a length problem the rx\_malf\_tlp will be asserted at any time during the TLP. Figure 4 provides a timing diagram of the receive interface.

**Figure 4. Receive Interface Timing Diagram**

## Wishbone Interface

This Wishbone interface is designed according to the Wishbone Interface Specification defined in *WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores Revision B.3*, released: September 7, 2002 and connects to PCI configuration space implemented in IP and control-status registers. Using this interface, the user can modify the configuration space registers, shown in Table 3, in the IP.

## Configuration Space Interface

The PCI Express IP core includes the base configuration space registers 0x0-3F, the PCI Express Capability Structure (0x40-63), and optionally the extended VC capability. The base configuration space is the legacy PCI Type00h registers containing the Base Address Registers (BARs), command and status registers.

### Setting PCI Express Configuration Registers

The user can setup the base configuration space using the INIT\_REG\_XXX port. This port is sampled and loaded at reset.

Table 3 provides the format for the INIT\_REG\_XXX port.

Table 3. PCI Type 00h CSR Map

Device ID		Vendor ID		000h	Type 0 Reg
Status		Command		004h	
Class code			Rev ID	008h	
BIST	Header	Lat Timer	cache Ln	00Ch	
Base address register 0				010h	
Base address register 1				014h	
Base address register 2				018h	
Base address register 3				01Ch	
Base address register 4				020h	
Base address register 5				024h	
Cardbus CIS Pointer				028h	
Subsystem ID		Subsystem Vendor ID		02Ch	
Expansion ROM Base Address				030h	
Reserved			CapPtr	034h	
Reserved				038h	
Max Lat	Min Gnt	Intr Pin	Intr Line	03Ch	
PM capability		NxtCap	PM Cap	040h	PM Capability structure
Data	BSE	PMCSR		044h	
MSI Control		NxtCap	MSI Cap	048h	MSI Capability structure
Message Address (Lower)				04Ch	
Message Address (Upper)				050h	
Reserved		Message Data		054h	
PE Capability		NxtCap	PE CAP	058h	PCIe Capability structure
PCI Express Device Capability				05Ch	
Device Status		Device Control		060h	
PCI Express Link capability				064h	
Link Status		Link Control		068h	
PCI Exp. Device Serial Number Enhanced Capability Header				100h	Device Serial number capability structure
PCI Express Device Serial Number (Lower)				104h	
PCI Express Device Serial Number (Upper)				108h	
PCI Express VC Enhanced capability Register				10Ch	PCIe VC Capability Structure
Port VC Capability register 1				110h	
Port VC Capability register 2				114h	
Port VC Status Register		Port VC Control register		118h	
VC Resource Capability Register (0)				11Ch	
VC Resource Control Register (0)				120h	
VC Resource Status Register (0)		RsvdP		124h	
VC Resource Capability Register (1)				128h	
VC Resource Control Register (1)				12Ch	
VC Resource Status Register (1)		RsvdP		130h	
VC Resource Capability Register (2)				134h	
VC Resource Control Register (2)				138h	
VC Resource Status Register (2)		RsvdP		13Ch	
-----				---	
-----				---	
VC Resource Capability Register (7)				170h	
VC Resource Control Register (7)				174h	
VC Resource Status Register (7)		RsvdP		178h	

The user has control of the number of BARs to be included. If the configuration space is used then at least one BAR must be set up correctly to allow TLPs to pass through the IP core. The IP core will check that received TLP's address is in any of the enabled BARs address ranges before it is processed and passed through to the user.

Configuration requests for Type0 will be fully terminated by the IP core in the memory space of implemented registers shown in Table 3. Read/write configuration requests outside of this address will return a completion packet with "Successful Completion (SC)" as status and the data set to zeros.

### Wishbone Interface Memory Map

Table 4 provides the memory map if the user selects the wishbone interface.

**Table 4. Wishbone Interface Memory Map**

Type	Address (hex)	Bits		Default	Description
PCI Express Type00 CFG Space					
PCI Express Configuration Space	0-FFF	31:0	R/W	GUI	PCI Express Configuration Space. Includes Type0/1, New Capabilities, and extended.
IP Control and Status Registers					
Power Management	1000-1003	31	COW	0	Received Vendor type DLLP
		30			Received Power Message DLLP
		29:27			Reserved
		26:3			RX Vendor DLLP Data
		2:0			RX Power Message DLLP Type
	1004-1007	31	R/W	0	Transmit Vendor type DLLP, cleared when DLLP is sent
		30			Transmit Power Message DLLP, cleared when DLLP is sent
		29:27			Reserved
		26:3			TX Vendor DLLP Data
		2:0			TX Power Message DLLP Type
Status	1008-100B	31:24	R	0	Reserved
		23:20	R		PHY LSM Status. For X1 Core [23:21] are Reserved.
		19:16	COW		PHY Connection Status / Result of Receiver Detection. For X1 Core [19:17] are Reserved.
		15:12	COW		PHY Receive/Rx Electrical Idle. For x1 Core [15:13] are Reserved.
		11:7	R		LTSSM State *
		6:3	R		DLL/Link Control SM Status [6] - DL Inactive State [5] - DL Init State [4] - DL Active State [3] - DL Up State
		2:0	R		Reserved
	100C-100F	31:22	R/W	0	Reserved
		21:18			LTSSM goto Loopback For x1 Core [21:19] are Reserved
		17			TLP Debug Mode : TLP bypasses DLL & TRNC check.
		16			PHY/LTSSM Send Beacon
		15			Force LSM Status active
		14			Force Received Electrical Idle
		13			Force PHY Connection Status

**Table 4. Wishbone Interface Memory Map (Continued)**

Type	Address (hex)	Bits	Default	Description
		12		Force Disable Scrambler (to PCS)
		11		Disable scrambling bit in TS1/TS2
		10		LTSSM go to Disable
		9		LTSSM go to Detect
		8		LTSSM go to HotReset
		7		LTSSM go to L0s
		6		LTSSM go to L1
		5		LTSSM go to L2
		4		LTSSM go to L0s and Tx FTS
		3		Reserved
		2		LTSSM go to Recovery
		1		LTSSM go to Config
		0		LTSSM no training
	1010-1013	31:30	R/W	Reserved
		29:16		ACK/NAK Latency Timer
		15		Reserved
		14:10		Number of FTS
		9:0		SKP Insert Counter
	1014-1017	31:29	R/W	Reserved
		28:24		Update Frequency for Posted Header
		23:16		Update Frequency for Posted Data
		15:13		Reserved
		12:8		Update Frequency for Non-posted Header
		7:0		Update Frequency for Non-posted Data
	1018-101B	31:29	R/W	Reserved
		28:24		Update Frequency for Completion Headers
		23:16		Update Frequency for Completion Data
		15:12		Reserved
		11:0		Update Timer
	101C-101F	31:16	R/W	Reserved
		15:0		Link Number

\*LTSSM State Encoding :

0 - DETECT

1 - POLLING

2 - CONFIG

3 - L0

4 - L0s

5 - L1

6 - L2

7 - RECOVERY

8 - LOOPBACK

9 - HOTRST

10 - DISABLED

R - Read Only

R/W - Read and Write

COW - Clear On Write

## Memory Map Control Interfaces

The user has options for the access of the embedded memory maps of the PCS/SERDES. The SCI bus can be used to control the PCS/SERDES.

### PCS/SERDES

The PCS/SERDES contains an embedded memory map that can be accessed at run time via a SCI bus connection.

For information on SERDES/PCS registers, please refer to the LatticeECP2/M Family Data Sheet.

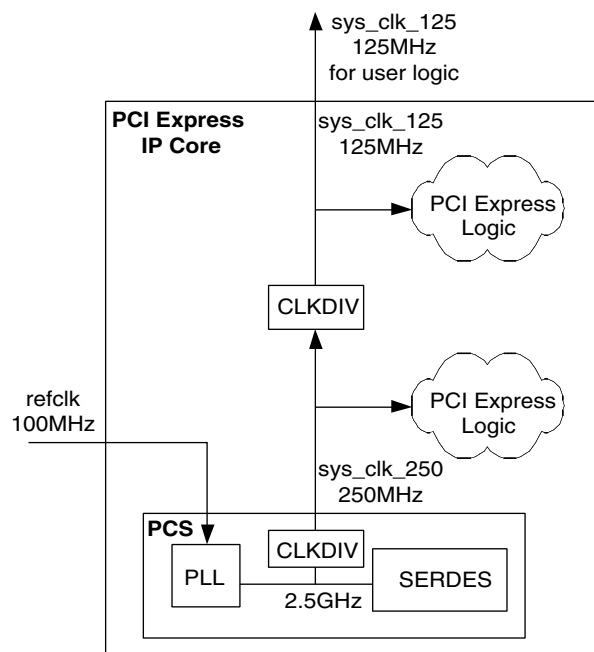
The PCS/SERDES memory map will be initialized during bitstream configuration using the `pcs_pipe.txt` file provided with the IP core. This file can be modified to set further options of the PCS/SERDES quad such as buffer options.

The system bus interface should only be used if the user requires run time access to the controls provided by the memory map.

## Clocking Scheme

The PCI Express IP core requires a 100MHz reference clock. This reference clock is used to clock the SERDES block of the physical layer as well as the remainder of the PCI Express protocol stack. Table 5 provides a block diagram of the clocking scheme.

**Figure 5. PCI Express IP Core Clocking Scheme**



Typically, PCI Express provides a 100MHz reference clock. This clock will need to be supplied as `refclk` to PCS.

Inside the IP core a clock (`sys_clk_250`) is driven from the PCS block at 250MHz. This clock is used to clock portions of the design that use a 8-bit data path. The 250MHz clock is then divided using a CLKDIV to create a 125MHz (`sys_clk_125`) clock for the remainder that use a 16-bit data path and the user interface.

## Using the PCI Express IP Core

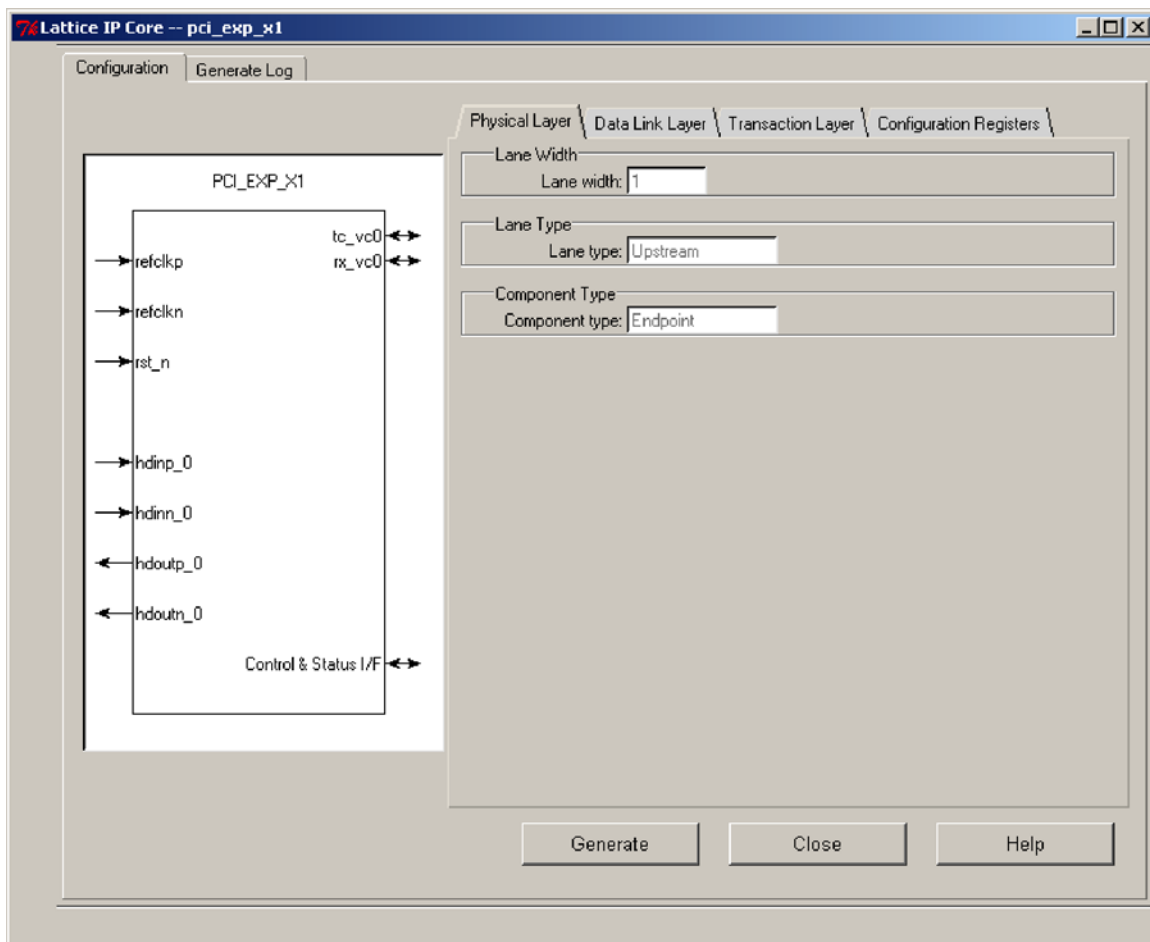
The PCI Express IP core is configured and created using the IPexpress GUI provided with the ispLEVER design tools. IPexpress creates all of the files necessary for the user to instantiate the IP core, simulate the core, synthe-

size the user's design using the core as a black box, and place and route the design in ispLEVER including the core.

## Creating the IP

IPexpress is used to create all IP and architectural modules in ispLEVER. For the PCI Express core the user is provided with several options.

## Configuring the Physical Layer

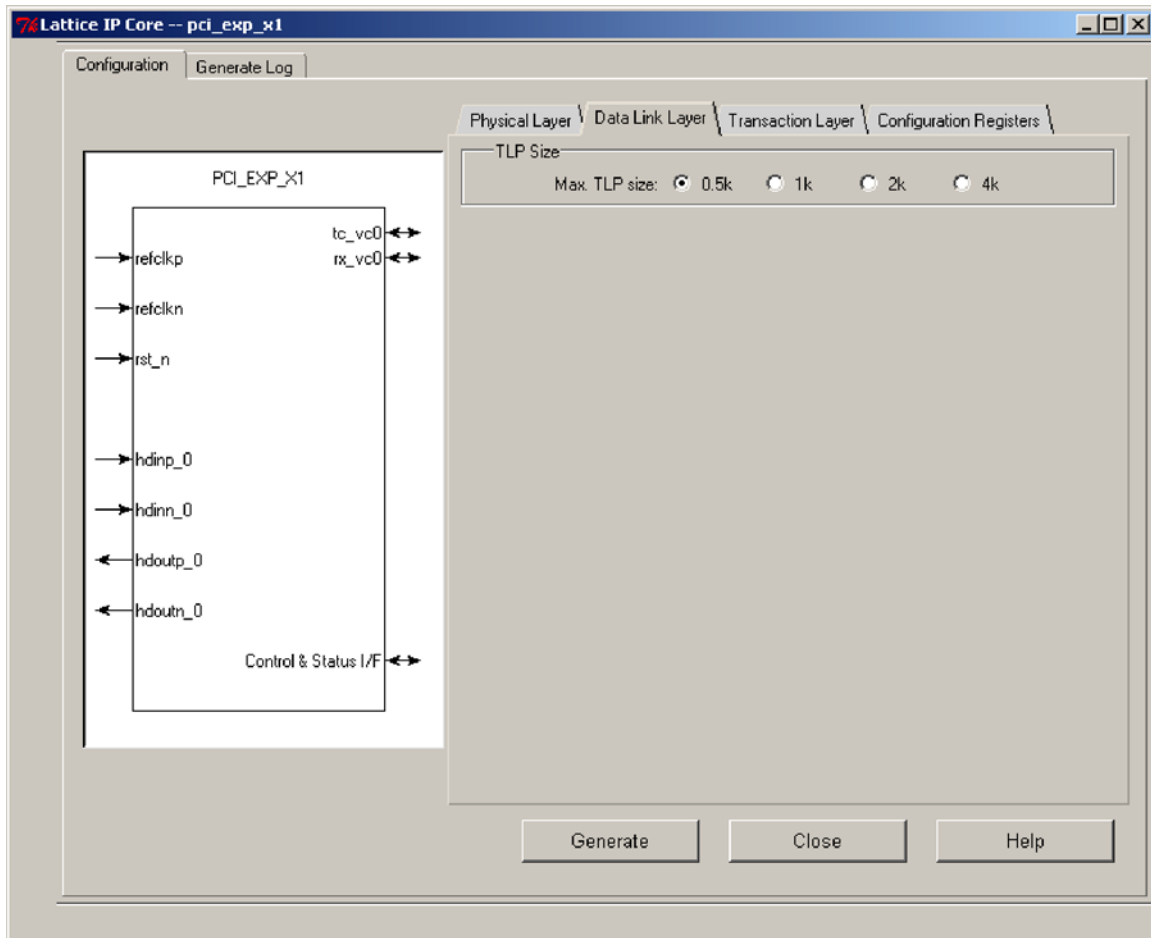


**LANE WIDTH:** Defines the lane width of the IP. The supported value is only 1.

**LANE WIDTH:** Defines the Upstream/Downstream lane. The supported value is only Upstream Lane.

**COMPONENT TYPE:** Defines the topology of the IP. The supported value is only as an endpoint.

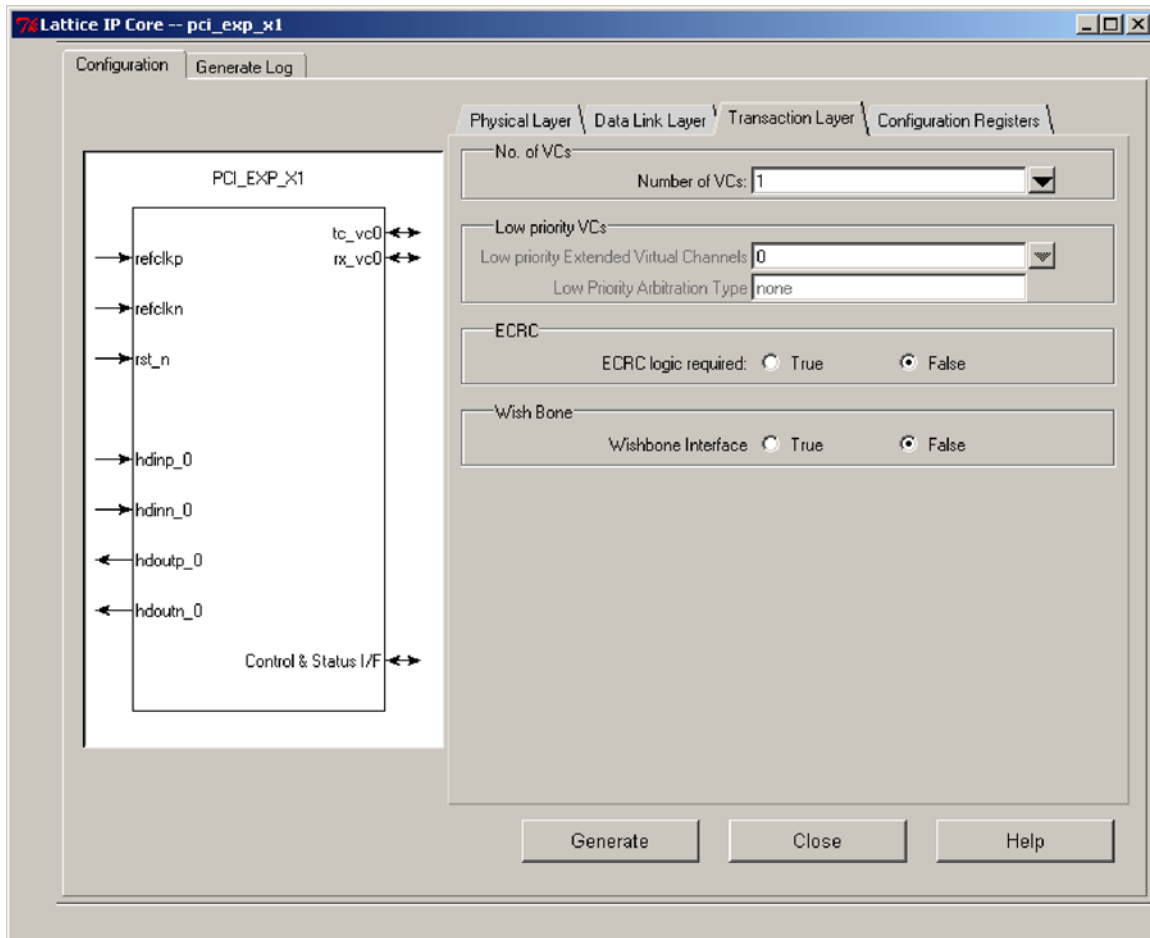
## Configuring the Data Link Layer



**Max TLP Size:** Defines the maximum TLP size required for the user. The supported values are 512, 1K, 2K and 4K bytes. This option impacts the size of the transmit retry buffer. The retry buffer is implemented using Embedded Block RAM (EBR).



## Configuring the Transaction Layer



**Number of VCs:** Defines the number of Virtual Channels required for the user. The supported values are 1, 2, 3, 4, 5, 6, 7 and 8.

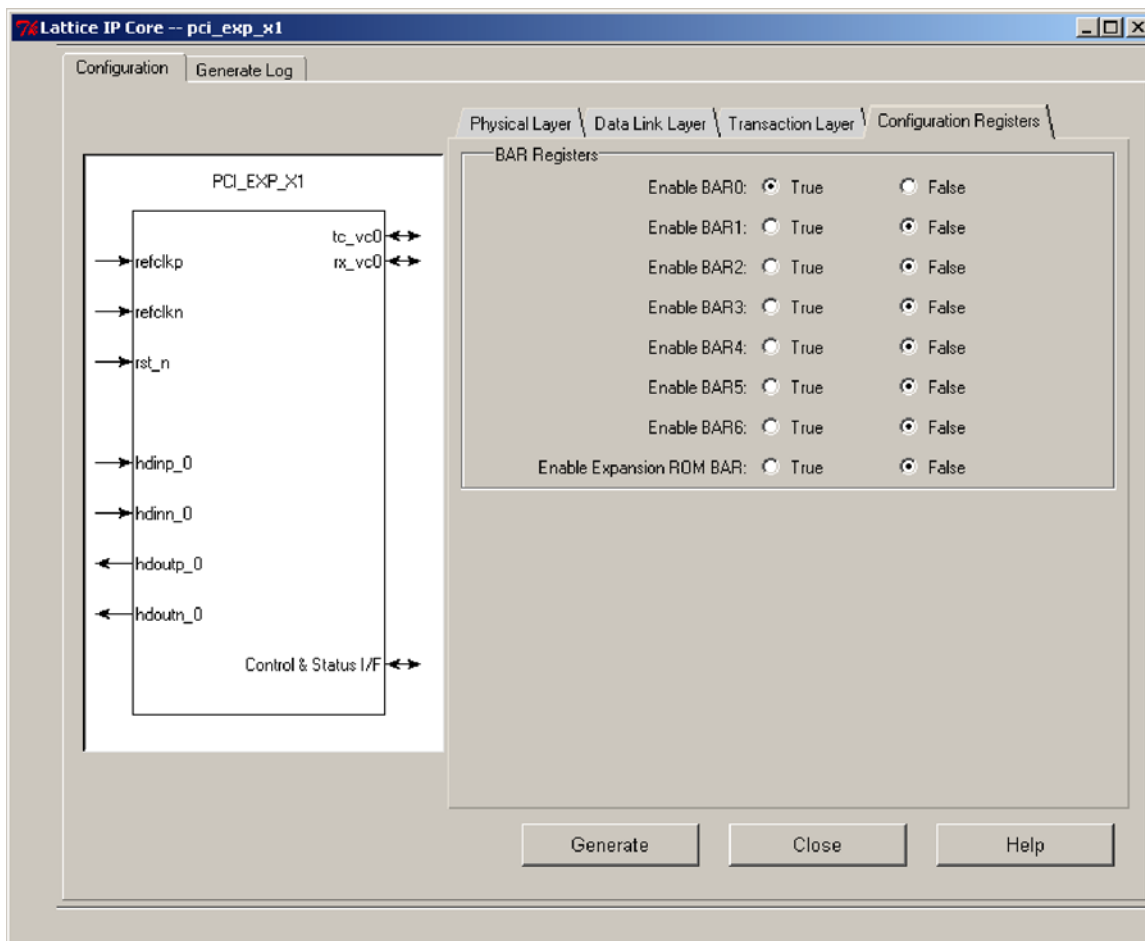
**Low Priority Extended Virtual Channel:** Selects the number of VCs that are low priority. The low priority VCs are at the top of the list of VCs. For example, if eight VCs are selected (VC0 to 7) and the number of low priority VCs is 4, then VC3, VC4, VC5, VC6, and VC7 will be low priority. Allowed values are = 3'b000 to `NUM\_VC-1.

**Low Priority Arbitration Type:** Specifies the arbitration type of multiple low priority VCs. Options include None and Round Robin.

**ECRC Logic Required:** Specifies whether the core needs to support ECRC generation/checking. ECRC is optional in the PCI Express specification. If ECRC is enabled in IPexpress, ports will be provided to control ECRC generation and checking. If ECRC is not enabled, the core will not include the ECRC logic and the ports will not be provided.

**Wishbone Interface:** Defines whether the Wishbone interface is present or not present at the user side.

## Configuring the BAR Registers



**Enable BARx:** Defines whether the BAR registers 0, 1, 2, 3, 4, 5 and 6 are present or not present.

**Enable Expansion ROM BAR:** Defines whether the expansion ROM is present or not present.

### Created Files

IPexpress creates several files that are used throughout the user's design cycle. Most of the files created are customized to the user's module name specified in IPexpress.

The design flow for IP created with IPexpress uses a pre-synthesized module (NGO) for synthesis and a protected model for simulation. The pre-synthesized module is customized and created during IPexpress generation. The protected simulation model is not customized during IPexpress and relies on parameters provided to customize behavior during simulation.

Table 5 provides a list of files created by IPexpress and how they are used.

**Table 5. File List**

File	Simulation	Synthesis/ ispLEVER	Description
<user name>_inst.v			This file provides an instance template for the IP core.
<user name>.v	Yes		This file provides the top level IP wrapper for simulation.
<user name>_beh.v	Yes		This file provides the IP simulation model.
pcs_exp_params.v	Yes		This file provides the user options of the IP for the simulation model.
<user name>_bb.v		Yes	This file provides the synthesis black box for the user's synthesis.
<user name>.ngo		Yes	This file provides the synthesized IP core used by ispLEVER.
pcs_pipe.txt	Yes	Yes	This file contains the PCS/SERDES memory map initialization. This file must be copied in to the simulation directory as well as the ispLEVER project directory.
<user name>.lpf		Yes	This file contains ispLEVER preferences to control place and route. These preferences should be included in the user's preference file.
<user name>.lpc			This file contains the IPexpress options used to recreate or modify the core in IPexpress.
pcie_x1_eval			This directory contains a sample design. This design is capable of performing a simulation and running through ispLEVER.

## Simulation

Simulation support for the IP core is provided for ModelSim® simulators. The simulation model for the IP core is generated with the core in IPexpress with the name <user name>.v. This file calls for the PCS/SERDES quad external library. The PCS/SERDES quad library is located in the ispTOOLS directory structure in the modelsim/lattice/verilog/ecp2/pcsc\_mti\_work directory.

A sample ModelSim .do file is available in the generated IP core.

## Implementation Details

The following section discusses several implementation details such as locating the IP and SERDES lanes, setting up the IP core for various modes, and applying design constraints in ispLEVER.

### Locating the IP

#### Board Layout Concerns for Add-in Cards

The PCI Express Add-in Card connector edge-finger is laid out for a certain orientation of lanes. The LatticeECP2M device package pinout also has a defined orientation of pins for the SERDES channels. The board layout will connect the PCI Express edge-fingers to the LatticeECP2M SERDES channels.

### Setting Design Timing Constraints

There are three design constraints that are required for the IP core. These constraints must be placed as preferences in the .lpf file. These preferences can be entered in the .lpf file through either the ispLEVER Design Planner or directly in the text based.lpf file.

Several interfaces inside the core run at 250MHz. This internal clock must be constrained for the place and route.

```
FREQUENCY NET "pclk" 250 MHz ;
```

The user interface clock clk\_125 must be constrained to run at 125MHz. Based on the connectivity of the design the name of this clock net may change.

```
FREQUENCY NET "sys_clk_125" 180 MHz ;
```

The 250 and 125 MHz clocks to the IP core need to be routed on a primary clock route to provide the best signal integrity.

```
USE PRIMARY NET "pclk" ;  
USE PRIMARY NET "sys_clk_125" ;
```

## References

- PCI Express Base Specification Revision 1.1 March 28, 2005
- PHY Interface for the PCI Express Architecture Version 1.00 6/19/2003
- LatticeECP2M Data Sheet

## Technical Support Assistance

Hotline: 1-800-LATTICE (North America)  
+1-503-268-8001 (Outside North America)  
e-mail: [techsupport@latticesemi.com](mailto:techsupport@latticesemi.com)  
Internet: [www.latticesemi.com](http://www.latticesemi.com)

## Revision History

Date	Version	Change Summary
September 2006	01.0	Initial release.
December 2006	01.1	Revised Appendix for LatticeECP2M Devices.
February 2007	01.2	Updated line for tx_data-vc0[vc1..vc7][15:0] port in Signal Descriptions table.
		Updated Transmit Interface Timing Diagram.
		Updated Receive Interface Timing Diagram.
February 2007	01.3	Added Test Ports section to PCI Express Port List table.
May 2007	01.4	Added GUI figures and CSR table.

## Appendix for LatticeECP2M Devices

**Table 6. Performance and Resource Utilization<sup>1</sup>**

IPexpress User-Configurable Mode	Slices	LUTs	Registers	sysMEM™ EBRs	f <sub>MAX</sub> (MHz)
Config 1	3997	5631	3814	4	162

1. Performance and utilization characteristics are generated using LFE2M-35E-6F672C, with Lattice's ispLEVER 6.1 software. When using this IP core in a different density, speed, or grade within the LatticeECP2M family, performance and utilization may vary.

### Ordering Part Number

The Ordering Part Number (OPN) for the PCI Express x1 Endpoint IP core targeting LatticeECP2M devices is PCI-EXP1-PM-U2. Table 7 lists the standard configurations that are available for the core, however any configuration can be created.

You can use the IPexpress software tool to help generate new configurations of this IP core. IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and ispLEVER help system. For more information on the ispLEVER design tools, visit the Lattice web site at: [www.latticesemi.com/software](http://www.latticesemi.com/software).

**Table 7. Parameter Settings for Standard Configurations**

Configuration	Config1	Config2	Config3
Maximum TLP Size	512K	2K	4K
Number of Virtual Channels	1	4	8
Low Priority Extended Virtual Channels	0	0	4
Enable ECRC	No	Yes	Yes
Wishbone Interface	No	Yes	Yes
Enable BAR 0	Yes	Yes	Yes
Enable BAR 1	No	Yes	Yes
Enable BAR 2	No	Yes	Yes
Enable BAR 3	No	No	Yes
Enable BAR 4	No	No	Yes
Enable BAR 5	No	No	Yes
Enable Expansion ROM BAR	No	No	Yes