



Gowin PCI Express Controller IP **User Guide**

IPUG1020-1.2E, 2024-07-05

Copyright © 2024 Guangdong Gowin Semiconductor Corporation. All Rights Reserved.

GOWIN is the trademark of Guangdong Gowin Semiconductor Corporation and is registered in China and/or other countries or regions. All other words and logos identified as trademarks or service marks are the property of their respective holders. No part of this document may be reproduced or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written consent of GOWINSEMI.

Disclaimer

GOWINSEMI assumes no liability and provides no warranty (either expressed or implied) and is not responsible for any damage incurred to your hardware, software, data, or property resulting from usage of the materials or intellectual property except as outlined in the GOWINSEMI Terms and Conditions of Sale. GOWINSEMI may make changes to this document at any time without prior notice. Anyone relying on this documentation should contact GOWINSEMI for the current documentation and errata.

Revision History

| Date | Version | Description |
|------------|---------|--|
| 05/25/2023 | 1.0E | Initial version published. |
| 03/29/2024 | 1.1E | PCIe Gen3 supported. |
| 07/05/2024 | 1.2E | Descriptions of Chapter 4 PCI Express Controller Configuration and Call updated. |

Contents

| | |
|--|------------|
| Contents | i |
| List of Figures | iii |
| List of Tables | iv |
| 1 About This Guide | 1 |
| 1.1 Purpose | 1 |
| 1.2 Related Documents | 1 |
| 1.3 Terminology and Abbreviations | 1 |
| 1.4 Support and Feedback | 2 |
| 2 Overview | 3 |
| 2.1 Features | 3 |
| 2.2 Functional Description | 4 |
| 3 PCI Express Controller | 5 |
| 3.1 Physical Layer Interface | 5 |
| 3.2 Clock and Reset | 5 |
| 3.3 Transaction Layer Interface | 6 |
| 3.3.1 TLP Receive Interface | 6 |
| 3.3.2 Receive Interface Timing | 7 |
| 3.3.3 TLP Transmit Interface | 8 |
| 3.3.4 Transmit Interface Timing | 8 |
| 3.4 Interrupt Interface | 10 |
| 3.4.1 Interrupt Interface Timing | 11 |
| 3.5 Controller Status Interface | 12 |
| 4 PCI Express Controller Configuration and Call | 14 |
| 4.1 Interface Configuration for SerDes IP | 14 |
| 4.2 PCI Express Controller Configuration Interface | 15 |

| | |
|--|-----------|
| 4.3 PCI Express Controller Configuration Options | 16 |
| 4.4 PCI Express Controller IP Generation and Instantiation | 19 |
| Appendix A PCIe Configuration Register Information..... | 21 |

List of Figures

| | |
|--|----|
| Figure 2-1 PCIe Controller Function Block Diagram..... | 4 |
| Figure 3-1 TLP Receiving Timing..... | 7 |
| Figure 3-2 Receive Timing of Non-posted TLP..... | 8 |
| Figure 3-3 pcie_tl_tx_wait_i and Transmitting Timing 1..... | 9 |
| Figure 3-4 pcie_tl_tx_wait_i and Transmitting Timing 2..... | 9 |
| Figure 3-5 pcie_tl_tx_wait_i and Transmitting Timing 3..... | 9 |
| Figure 3-6 TLP Transmitting Timing Examples..... | 10 |
| Figure 3-7 Single Interrupt Control Timing..... | 11 |
| Figure 3-8 Multiple Interrupt Control Timing..... | 11 |
| Figure 4-1 IP Core Generator..... | 14 |
| Figure 4-2 SerDes IP Configuration Interface 1..... | 15 |
| Figure 4-3 PCI Express Controller IP Configuration Interface..... | 16 |
| Figure 4-4 PCIe Controller Basic Configuration..... | 17 |
| Figure 4-5 PCI Express Controller BAR Configuration..... | 18 |
| Figure 4-6 PCI Express Controller Core Configuration..... | 19 |
| Figure 4-7 SerDes IP Configuration Interface 2..... | 19 |
| Figure 4-8 SerDes-PCI Express Controller IP Instantiation Snippet..... | 20 |

List of Tables

| | |
|--|----|
| Table 1-1 Terminology and Abbreviations | 1 |
| Table 3-1 Gowin PCI Express Controller IP Clock and Reset | 5 |
| Table 3-2 Gowin PCI Express Controller IP Receive Interface | 6 |
| Table 3-3 The Function Channel Corresponding to The Received Data | 7 |
| Table 3-4 Transmit Interface of Gowin PCI Express Controller IP Controller | 8 |
| Table 3-5 Gowin PCIe Controller IP Controller Interrupt Interface..... | 10 |
| Table 3-6 Gowin PCIe Controller IP Controller Status Interface | 12 |
| Table A-1 PCIe Capability | 21 |

1 About This Guide

1.1 Purpose

Gowin PCI Express Controller IP User Guide mainly include functional features, structure, port descriptions, configuration calls, etc. It is designed to help users quickly understand the features, characteristics and usage of Gowin PCI Express Controller IP.

1.2 Related Documents

The latest user guides are available on the GOWINSEMI website. You can find the related documents at www.gowinsemi.com:

- [DS1228, Arora V FPGA Products Overview](#)
- [DS981, Arora V 138K & 75K FPGA Products Data Sheet](#)

1.3 Terminology and Abbreviations

The terminology and abbreviations used in this manual are as shown in Table 1-1.

Table 1-1 Terminology and Abbreviations

| Terminology and Abbreviations | Full Name |
|-------------------------------|---|
| BAR | Base Address Register |
| LTSSM | Link Training and Status State Machine |
| MAC | Media Access Control |
| MSI | Message Signaled Interrupt |
| PCIe | Peripheral Component Interconnect Express |
| TLP | Transaction Layer Packet |

1.4 Support and Feedback

Gowin Semiconductor provides customers with comprehensive technical support. If you have any questions, comments, or suggestions, please feel free to contact us directly using the information provided below.

Website: www.gowinsemi.com

E-mail: support@gowinsemi.com

2 Overview

Arora V FPGA products include the PCIe integrated interface module, which is compliant with the PCIe Gen3 (8GT/s) specification as defined by the PCI Express Base Specification, V3.0, and can support x1, x2, x4, and x8 lanes.

PCI Express protocol enables custom FPGA-to-FPGA communication and attaches ASSP endpoint devices, such as Ethernet Controllers or Fiber Channel HBAs (Host Bus Adapter, HBA), to the FPGA. The PCIe controller module provides a high performance and low cost solution.

2.1 Features

The features of Arora V PCI Express Controller IP are as follows:

- Compliant to the PCI Express Base Specification 3.0
- Supports x1, x2, x4, x8 lanes
- Supports End Point
- Supports Gen1 (2.5GT/s), Gen2 (5GT/s), Gen3 (8GT/s)
- Up to six BARs (Base Address Register), resizable
- Supports up to 4KB data payload transfer
- Supports Autonomous link speed/width change
- Supports advanced configuration options, Advanced Error Reporting (AER), and End-to-End Cyclic Redundancy Check (ECRC)
- Configurable parameters: channel width, maximum payload size, FPGA logical interface speeds, reference clock frequency, base address register decoding and filtering, etc.

2.2 Functional Description

The basic functions of the PCIe controller integrated in the Gowin devices are shown in Figure 2-1.

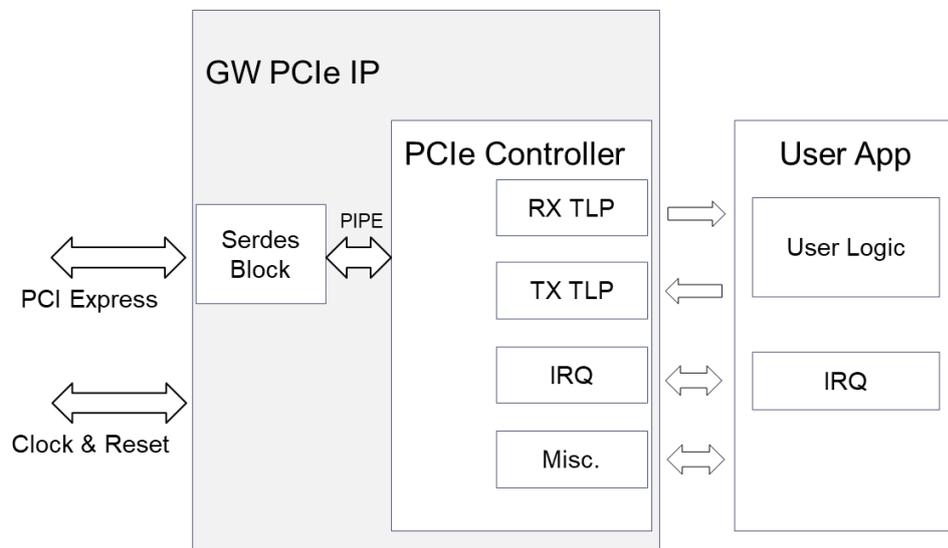
The controller uses GOWIN's 12G Serdes module as the physical layer to connect with external PCIe hosts or devices for data communication.

The PCIe controller provides the full functionality of the transaction Layer, data link layer and MAC.

For the user logic, the PCIe controller provides the TLP receive and transmit interfaces for data and common command transfers.

The controller supports interrupt modes such as MSI.

Figure 2-1 PCIe Controller Function Block Diagram



3 PCI Express Controller

This chapter mainly introduces the ports of the PCIe controller module. According to the practical applications, this chapter divides the controller ports into physical interfaces, transmission interfaces, interrupt interfaces, and configuration interfaces according to their functions.

3.1 Physical Layer Interface

Depending on the number of Lanes used by the PCIe controller, x1, x2, x4, you need to choose the specified IO pins and the fixed lane sequence design according to [Pinout](#). PCIe uses Q0 Lane0 or Q1 Lane0 for X1 pin. Q0 (or Q1) Ref0 is the reference clock input and the reference clock frequency is 100MHz.

3.2 Clock and Reset

Table 3-1 Gowin PCI Express Controller IP Clock and Reset

| Port | I/O | Description |
|---------------|-------|----------------------------|
| pcie_tl_clk_i | input | PCIe TLP user clocks |
| pcie_tl_rst_i | input | PCIe TLP user reset signal |

3.3 Transaction Layer Interface

3.3.1 TLP Receive Interface

Table 3-2 Gowin PCI Express Controller IP Receive Interface

| Port | I/O | Description |
|------------------------------|--------|---|
| pcie_tl_rx_sop_o | output | Receive the TLP Start packet, marking the first data packet of the TLP. |
| pcie_tl_rx_eop_o | output | Receive the TLP End packet, marking the last data packet of the TLP. |
| pcie_tl_rx_data_o [255:0] | output | Receive TLP data, 256 bit width by default |
| pcie_tl_rx_valid_o [7:0] | output | The validity flag of receiving the TLP data, each bit flag corresponds to whether the Dwords are valid or not. Bit 7 corresponds to pcie_tl_rx_data[255:224] and Bit 0 corresponds to pcie_tl_rx_data[31:0]. |
| pcie_tl_rx_bardec_o [5:0] | output | Receive the target BAR decoding signal of the TLP, indicating the functional channel corresponding to the received data, as shown in Table Table 3-3. |
| pcie_tl_rx_err_o[7:0] | output | Receive data error signal. Bit 0: ECRC Error. Bit 1: Invalid TLP, including: TLP packet with wrong identification bits in RX buffer TLP payload size does not match the actual data received Bit 2: RX buffer read error Bit 3: RX Config data packet Bit 4: Unsupported TLP formats Bit 5: ACS Violation Bits 7:6: reserved |
| pcie_tl_rx_wait_i | input | Receive wait signal, indicating that the RX buffer is not ready to receive data. |
| pcie_tl_rx_masknp_i | input | The mask signal of Non-posted TLP packet If the signal is set high, enables the controller to stop receiving the subsequent non-posted TLPs after the current TLP transmission is finished. |

Table 3-3 The Function Channel Corresponding to The Received Data

| Bit | BAR |
|-----|-------|
| 0 | BAR0 |
| 1 | BAR1* |
| 2 | BAR2 |
| 3 | BAR3* |
| 4 | BAR4 |
| 5 | BAR5* |

Note!

*, Bit1 is invalid when BAR0 is set to 64bit, Bit3 is invalid when BAR2 is set to 64bit, and Bit5 is invalid when BAR4 is set to 64bit.

3.3.2 Receive Interface Timing

Figure 3-1 shows the TLP receive timing. The figure shows the TLP data reception timing for a Header length of 4DWORD, a payload length of 15DWORD, and a target BAR0. The pcie_tl_rx_wait signal is set to 1 during reception.

Figure 3-1 TLP Receiving Timing

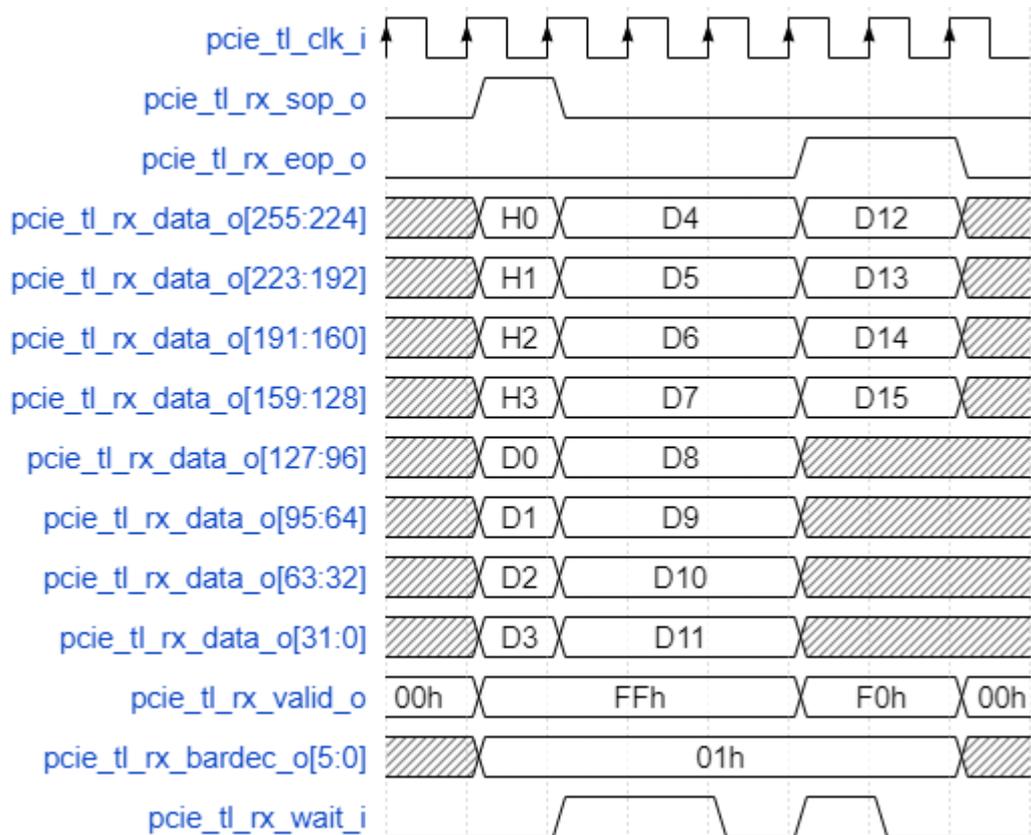
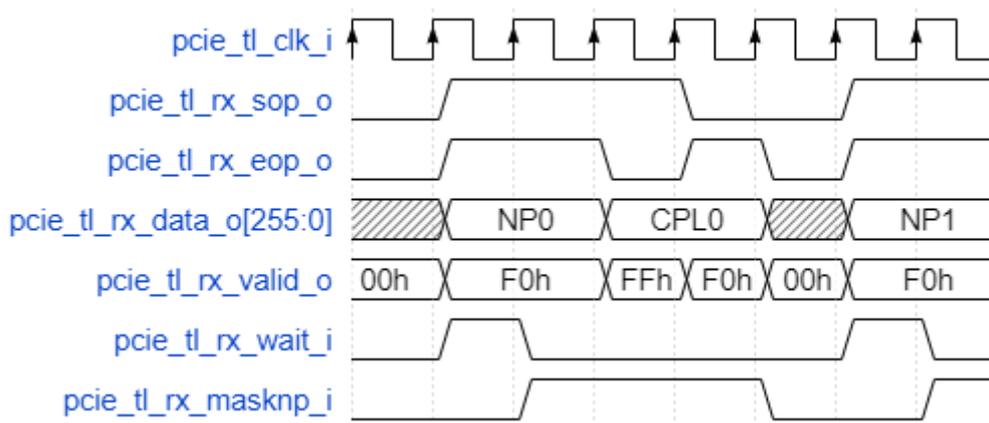


Figure 3-2 is the Receive Timing of the Non-posted TLP packet. During TLP reception, `pcie_tl_rx_wait_i` is set to 1 for one clock cycle. During the reception of the first TLP packet, `pcie_tl_rx_masknp_i` is set to 1, preventing the controller from receiving the subsequent non-posted TLP packets and only receiving the output completion packet (Completion TLP). The controller restarts receiving Non-posted TLP only after `pcie_tl_rx_masknp_i` is pulled down.

Figure 3-2 Receive Timing of Non-posted TLP



3.3.3 TLP Transmit Interface

Table 3-4 Transmit Interface of Gowin PCI Express Controller IP Controller

| Port | I/O | Description |
|---------------------------------------|--------|--|
| <code>pcie_tl_tx_sop_i</code> | input | Transmitting the TLP Start packet, marking the first data packet of the TLP packet. |
| <code>pcie_tl_tx_eop_i</code> | input | Transmitting the TLP End packet, marking the last data packet of the TLP packet. |
| <code>pcie_tl_tx_data_i[255:0]</code> | input | Transmit TLP Data. |
| <code>pcie_tl_tx_valid_i[7:0]</code> | input | A signal to indicate whether the TLP data transmission is valid or not. Each bit indicates whether the Dwords are valid or not. Bit 7 corresponds to <code>pcie_tl_tx_data[255:224]</code> and Bit 0 corresponds to <code>pcie_tl_tx_data[31:0]</code> . |
| <code>pcie_tl_tx_wait_o</code> | output | Transmitting wait signal, indicating that the TX buffer is not ready to transmit data. |

3.3.4 Transmit Interface Timing

Figure 3-3, Figure 3-4, and Figure 3-5 are the control timing when `pcie_tl_tx_wait_o` is set high during the TLP transmission. When the controller does not have enough TLP transmit space, `pcie_tl_tx_wait_o` is

pulled up. A valid TLP packet transmission needs to be recognized by the controller when the pcie_tl_tx_wait_o signal is 0.

Figure 3-3 pcie_tl_tx_wait_i and Transmitting Timing 1

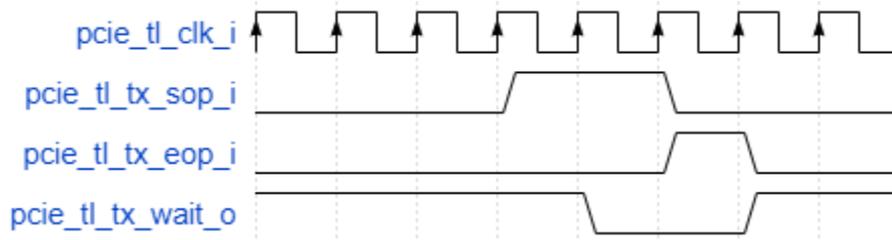


Figure 3-4 pcie_tl_tx_wait_i and Transmitting Timing 2

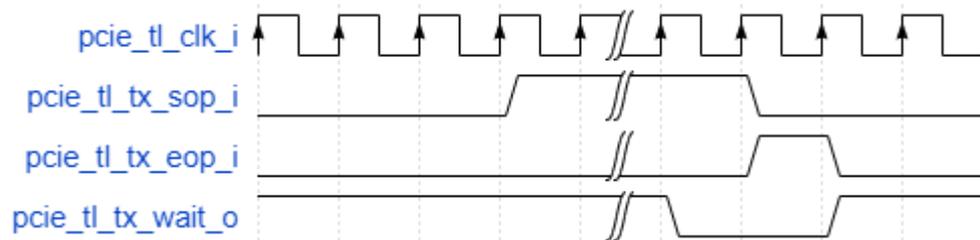


Figure 3-5 pcie_tl_tx_wait_i and Transmitting Timing 3

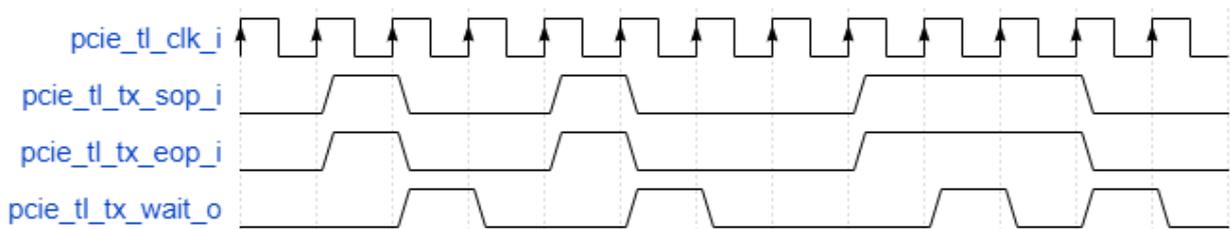
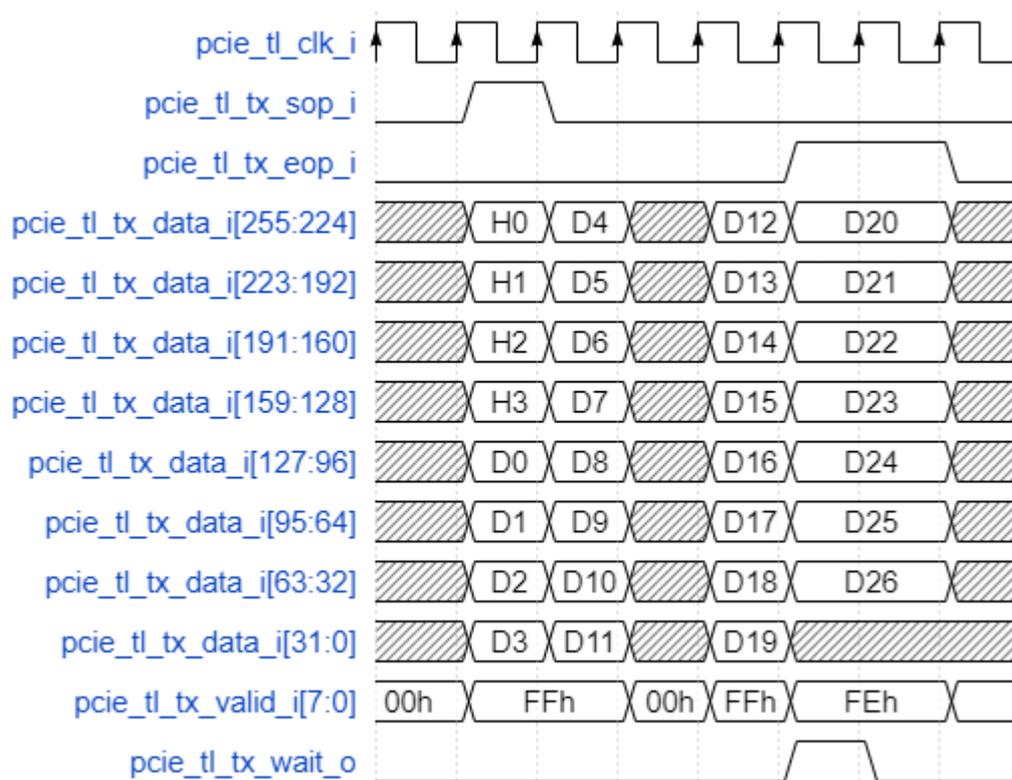


Figure 3-6 shows the complete timing reference for TLP Transmitting. The figure below shows the transmitting control timing for a Header length of 4DW and a payload length of 27DW. The controller identifies the valid transmit data by pcie_tl_tx_valid_i the feedback signal pcie_tl_tx_wait_o.

Figure 3-6 TLP Transmitting Timing Examples



3.4 Interrupt Interface

Table 3-5 Gowin PCIe Controller IP Controller Interrupt Interface

| Parameter | I/O | Description |
|---------------------------|--------|---|
| pcie_tl_int_status_i | input | Physical function interrupt status |
| pcie_tl_int_req_i | input | Interrupt request signal. A valid interrupt request is issued by setting one clock cycle high. The user side must wait for the pcie_tl_int_ack signal to be set high before issuing the next interrupt request. |
| pcie_tl_int_msinum_i[4:0] | input | The number of MSI interrupts. Indicates the number of MSI numbers corresponding to the interrupt request. If the MSI interrupt is not enabled, this signal needs to be set to 0. |
| pcie_tl_int_ack_o | output | The response of interrupt signal Indicates that a valid |

| Parameter | I/O | Description |
|-----------|-----|---|
| | | pcie_tl_int_req request was received by the controller and sent successfully. |

3.4.1 Interrupt Interface Timing

See Figure 3-7 for the interrupt interface timing. When enabling the interrupt of the physical function, the user pulls up the pcie_tl_int_status_i signal and issues the pcie_tl_int_req_i request while simultaneously issuing pcie_tl_int_msinum_i. After the first interrupt request is sent and the response pcie_tl_int_ack_o is received, the second interrupt request can continue to be sent. When all interrupt requests required by the application have been sent and responded to, the pcie_tl_int_status_i signal is pulled low and the interrupt is cleared.

Figure 3-7 Single Interrupt Control Timing

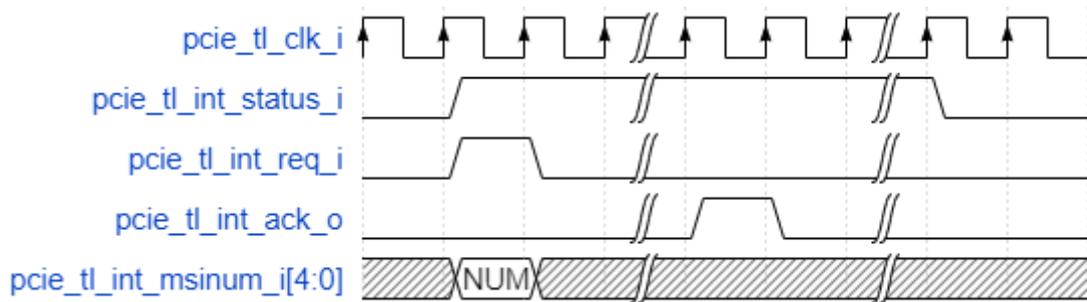
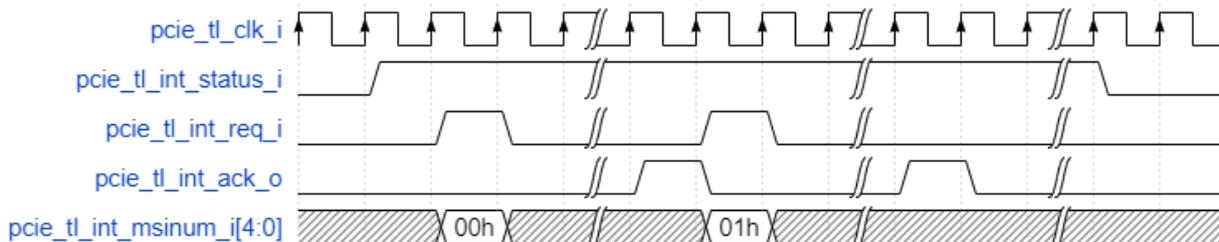


Figure 3-8 Multiple Interrupt Control Timing



3.5 Controller Status Interface

Table 3-6 Gowin PCIe Controller IP Controller Status Interface

| Parameter | I/O | Description |
|------------------------------|--------|---|
| ltssm_o[4:0] | output | LTSSM status <ul style="list-style-type: none"> ● 00h: detect.quiet ● 01h: detect.active ● 02h: polling.active ● 03h: polling.compliance ● 04h: polling.configuration ● 05h: config.linkwidthstart ● 06h: config.linkwidthaccept ● 07h: config.lanenumwait ● 08h: config.lanenumaccept ● 09h: config.complete ● 0Ah: config.idle ● 0Bh: recovery.receiverlock ● 0Ch: recovery.equalization ● 0Dh: recovery.speed ● 0Eh: recovery.receiverconfig ● 0Fh: recovery.idle ● 10h: L0 ● 11h: L0s ● 12h: L1.entry ● 13h: L1.idle ● 14h: L2.idle/L2.transmitwake ● 15h: reserved ● 16h: disable ● 17h: loopback.entry ● 18h: loopback.active ● 19h: loopback.exit ● 1Ah: hotreset |
| pcie_tl_tx_data_i[31:0] | output | Posted TLP controls the number of credits sent. [14:0]: the number of valid data credits [26:16]: the number of valid header credits [31]: sufficient number of credits to send TLP packets of max payload size |
| pcie_tl_tx_creditsnp_o[31:0] | output | Non-Posted TLP controls the number of |

| Parameter | I/O | Description |
|-------------------------------|--------|--|
| | | credits sent. [14:0]: the number of valid data credits [26:16]: the number of valid header credits [31]: sufficient number of credits to send TLP packets of max payload size |
| pcie_tl_tx_creditscpl_o[31:0] | output | Completion TLP controls the number of credits sent. [14:0]: the number of valid data credits [26:16]: the number of valid header credits [31]: sufficient number of credits to send TLP packets of max payload size |
| pcie_tl_cfg_busdev_o[12:0] | output | Bus Number and DeviceNumber information for PCIe devices. [12:5]: Bus Number [4:0]: Device Number |

4 PCI Express Controller Configuration and Call

Under “Tools” menu bar in Gowin Software interface, you can start the “IP Core Generator” tool to call and configure the PCI Express Controller.

4.1 Interface Configuration for SerDes IP

After creating a project, the user can click the "Tools" on the menu bar and select "IP Core Generator" from the dropdown list to open the IP Core Generator, as shown in Figure 4-1. Click "Soft IP Core > SerDes", then double-click the "SerDes" option to open the SerDes IP configuration interface, as shown in Figure 4-2.

Figure 4-1 IP Core Generator

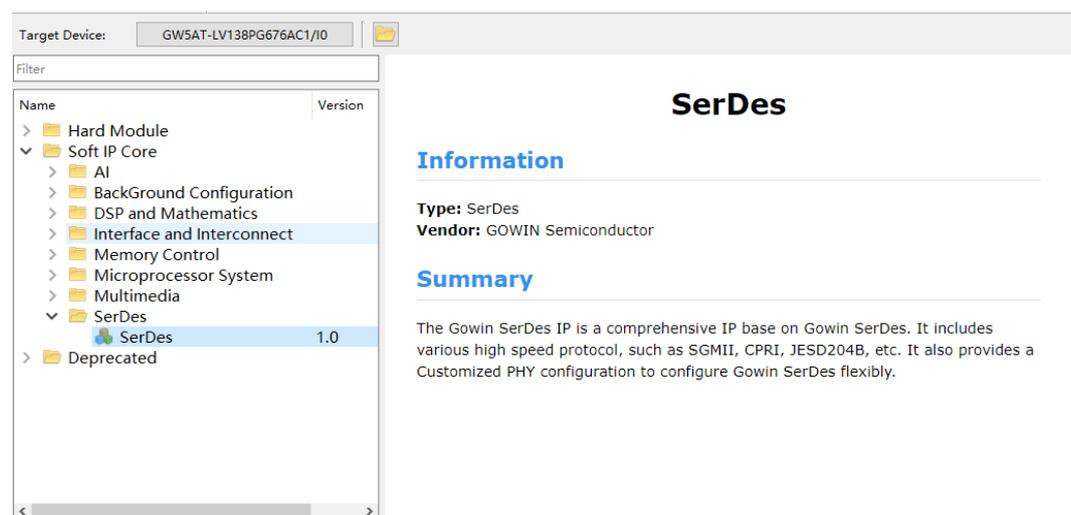
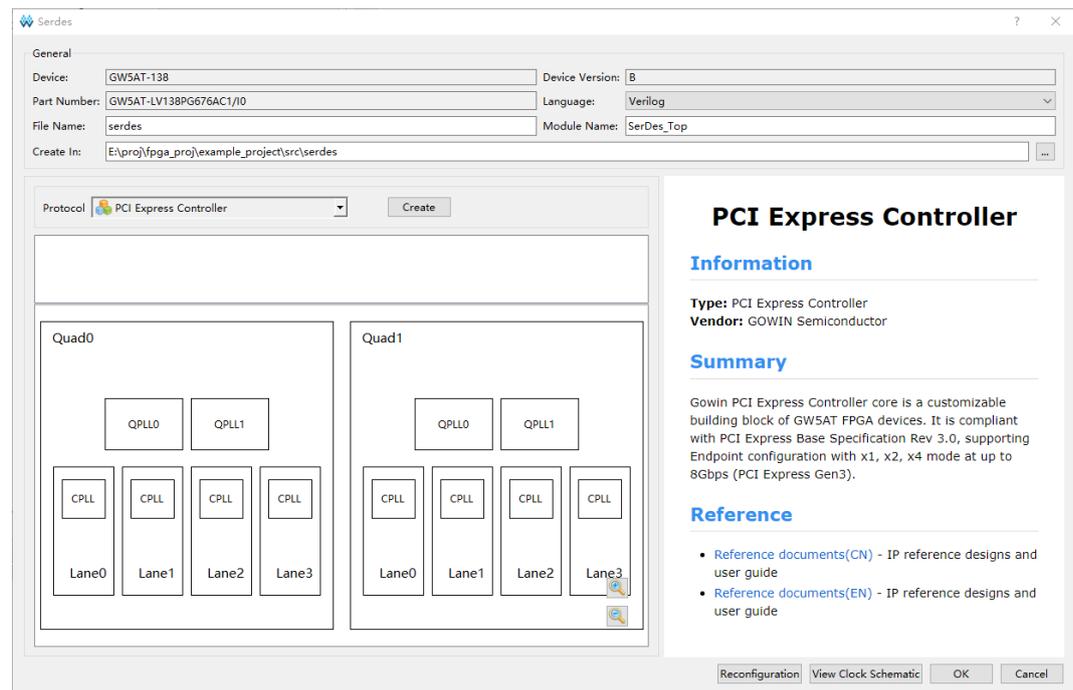


Figure 4-2 SerDes IP Configuration Interface 1



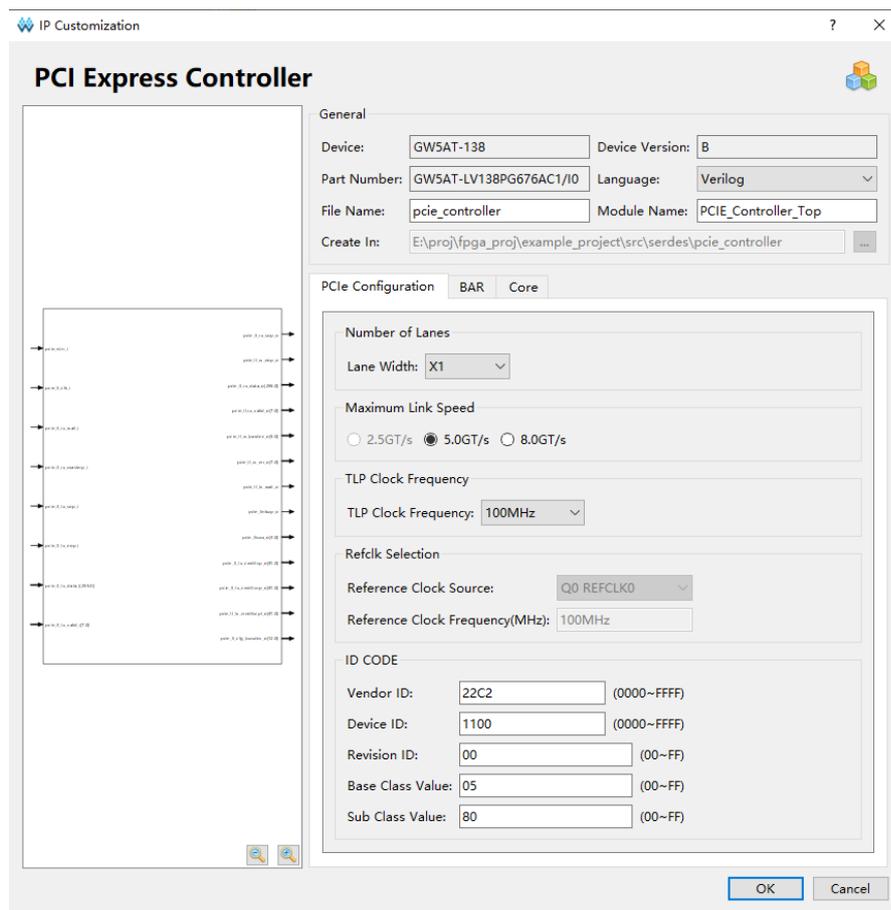
In the SerDes IP configuration interface, you can define the file name of the SerDes IP in the "File Name" box and the top-level module name of the SerDes IP in the "Module Name" box. In the "Protocol" drop-down list, you can select the desired protocol.

Choose "PCI Express Controller" and click the "Create" button to open the PCI Express Controller IP configuration interface, as shown in Figure 4-3.

4.2 PCI Express Controller Configuration Interface

The PCI Express Controller IP configuration interface is as shown in Figure 4-3. The interface diagram is on the left side, and the configuration parameters are on the right side. In the "File Name" box, you can define the IP file name, and you can define the top-level module name of the IP in in the "Module Name" box.

Figure 4-3 PCI Express Controller IP Configuration Interface



4.3 PCI Express Controller Configuration Options

The PCI Express Controller IP configuration interface includes the following three sections.

Figure 4-4 shows the basic configuration of the PCIe controller, including parameters such as operating speed, the number of lanes, and device ID.

Figure 4-5 shows the BAR configuration of the PCIe controller, and you can configure the size of BAR0-BAR5 and their corresponding IO and Memory attributes.

Figure 4-6 shows the interrupt configuration of the PCIe controller, including MSI mode and other configurations.

PCIe Controller Basic Configuration

Figure 4-4 PCIe Controller Basic Configuration

PCIe Configuration BAR Core

Number of Lanes

Lane Width:

Maximum Link Speed

2.5GT/s
 5.0GT/s
 8.0GT/s

TLP Clock Frequency

TLP Clock Frequency:

Refclk Selection

Reference Clock Source:

Reference Clock Frequency(MHz):

ID CODE

Vendor ID: (0000~FFFF)

Device ID: (0000~FFFF)

Revision ID: (00~FF)

Base Class Value: (00~FF)

Sub Class Value: (00~FF)

The basic configuration of PCIe is shown in Figure 4-4.

- **Lane Width:** Specifies the number of lanes for PCIe configuration, including x1, x2, and x4.
- **Maximum Link Speed:** Support for up to 8GT/s, corresponding to PCIe 3.0; the default configuration is 5GT/s, corresponding to PCIe 2.0.
- **TLP Clock Frequency:** The user clock used by the TLP layer. The clock source is from the user logic, and input the PCIe controller. You need to select the actual clock frequency, with options of 100MHz, 125MHz, and 150MHz.
- **Vendor ID:** Used to configure Vendor ID for the PCIe register space. The default value is 22C2h, which is the registered Vendor ID of Gowinsemi.
- **Device ID:** Used to configure Device ID.
- **Revision ID:** Used to configure Revision ID.
- **Base Class Value:** Used to configure Base Class Value.

- Sub Class Value: Used to configure Sub Class Value.

PCI Express Controller BAR Configuration

Figure 4-5 PCI Express Controller BAR Configuration

The screenshot displays the 'BAR' configuration tab of the PCI Express Controller configuration utility. It features six individual configuration blocks for BAR0 through BAR5. Each block contains the following fields:

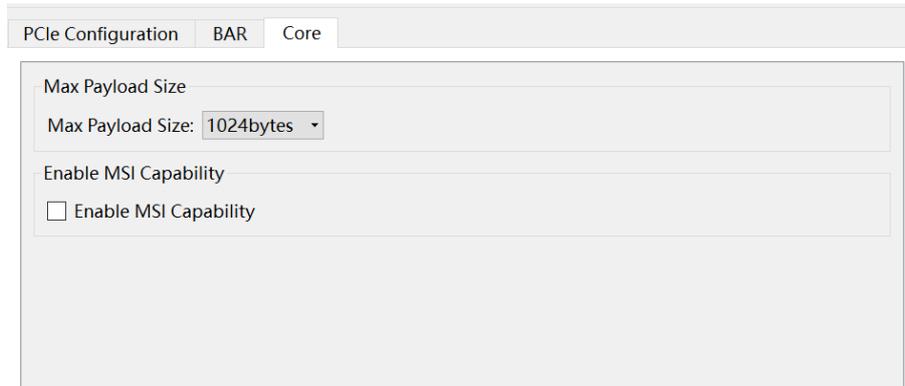
- Bar0 Enabled:** Type: Memory, 64 bit, Prefetchable, Size: 1KiloBytes, Value(Hex): FFFFFFFC00
- Bar1 Enabled:** Type: Memory, 64 bit, Prefetchable, Size: 2KiloBytes, Value(Hex): FFFFFFF800
- Bar2 Enabled:** Type: Memory, 64 bit, Prefetchable, Size: 2KiloBytes, Value(Hex): FFFFFFF800
- Bar3 Enabled:** Type: Memory, 64 bit, Prefetchable, Size: 2KiloBytes, Value(Hex): 00000000
- Bar4 Enabled:** Type: Memory, 64 bit, Prefetchable, Size: 2KiloBytes, Value(Hex): 00000000
- Bar5 Enabled:** Type: Memory, 64 bit, Prefetchable, Size: 2KiloBytes, Value(Hex): 00000000

Users can configure the application information for the BAR space under Bar tab. Gowin PCIe controller supports flexible configuration of BAR0-BAR5.

- BAR Enabled: if checked, BAR enabled.
- 64 bit: If checked, the nearest BAR space will be simultaneously occupied to share the address space.
- Prefetchable: Only valid when the 64 bit is enabled.
- Size: The BAR space size, supporting up to 1 Mbyte.

PCI Express Controller Core Configuration

Figure 4-6 PCI Express Controller Core Configuration

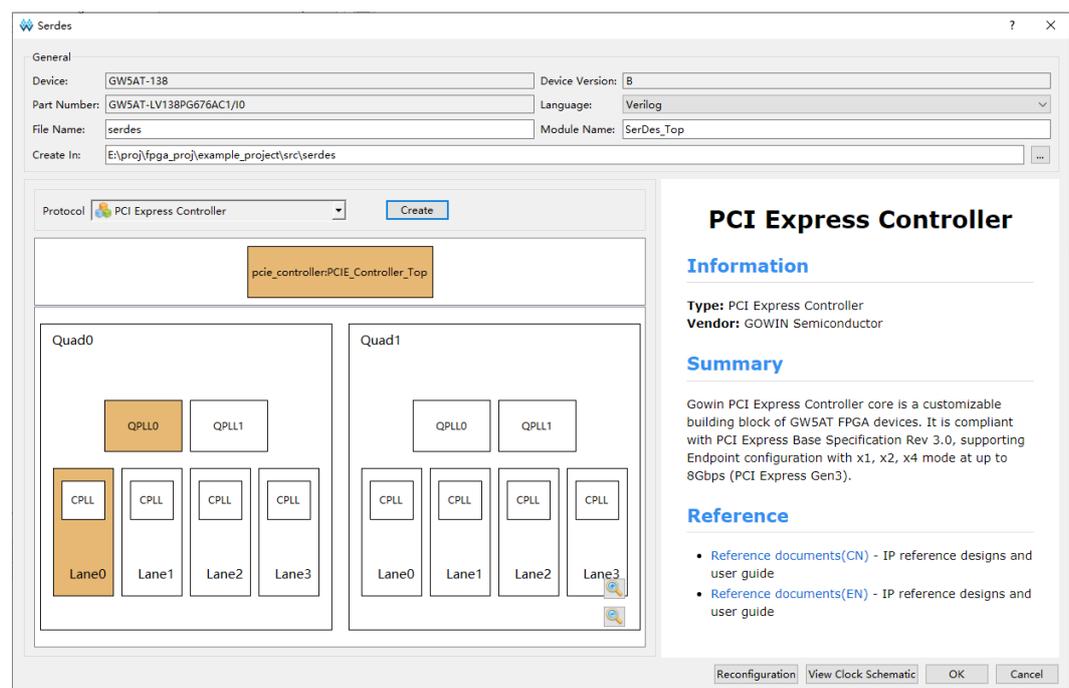


Users can configure the application information related to the PCIe Core functions under Core tab.

- **Max Payload Size:** The PCIe controller supports a maximum size of up to 4K bytes. Users can select the desired size from the drop-down list, with options including 128 bytes, 256 bytes, 512 bytes, 1024 bytes, 2048 bytes, and 4096 bytes.
- **MSI Capabilities:** Interrupt support, used to enable or disable the MSI interrupt interface in the application by checking the box.

4.4 PCI Express Controller IP Generation and Instantiation

Figure 4-7 SerDes IP Configuration Interface 2



After configuring PCI Express Controller, click the "OK" button to complete the IP configuration, as shown in Figure 4-7. Compared to Figure 4-2, this interface now shows the using of QPLL and lane.

Click the "OK" button in the SerDes IP configuration interface to generate the final PCI Express Controller and SerDes IP. Figure 4-8 shows a sample snippet instantiated in the user's Verilog code.

Figure 4-8 SerDes-PCI Express Controller IP Instantiation Snippet

```

10 //Change the instance name and port connections to the signal names
11 //-----Copy here to design-----
12
13 SerDes_Top your_instance_name (
14     .PCIE_Controller_Top_pcie_tl_rx_sop_o(PCIE_Controller_Top_pcie_tl_rx_sop_o), //output PCIE_Controller_Top_pcie_tl_rx_sop_o
15     .PCIE_Controller_Top_pcie_tl_rx_eop_o(PCIE_Controller_Top_pcie_tl_rx_eop_o), //output PCIE_Controller_Top_pcie_tl_rx_eop_o
16     .PCIE_Controller_Top_pcie_tl_rx_data_o(PCIE_Controller_Top_pcie_tl_rx_data_o), //output [255:0] PCIE_Controller_Top_pcie_tl_rx_data_o
17     .PCIE_Controller_Top_pcie_tl_rx_valid_o(PCIE_Controller_Top_pcie_tl_rx_valid_o), //output [7:0] PCIE_Controller_Top_pcie_tl_rx_valid_o
18     .PCIE_Controller_Top_pcie_tl_rx_bardec_o(PCIE_Controller_Top_pcie_tl_rx_bardec_o), //output [5:0] PCIE_Controller_Top_pcie_tl_rx_bardec_o
19     .PCIE_Controller_Top_pcie_tl_rx_err_o(PCIE_Controller_Top_pcie_tl_rx_err_o), //output [7:0] PCIE_Controller_Top_pcie_tl_rx_err_o
20     .PCIE_Controller_Top_pcie_tl_tx_wait_o(PCIE_Controller_Top_pcie_tl_tx_wait_o), //output PCIE_Controller_Top_pcie_tl_tx_wait_o
21     .PCIE_Controller_Top_pcie_tl_int_ack_o(PCIE_Controller_Top_pcie_tl_int_ack_o), //output PCIE_Controller_Top_pcie_tl_int_ack_o
22     .PCIE_Controller_Top_pcie_ltssm_o(PCIE_Controller_Top_pcie_ltssm_o), //output [4:0] PCIE_Controller_Top_pcie_ltssm_o
23     .PCIE_Controller_Top_pcie_tl_tx_creditsp_o(PCIE_Controller_Top_pcie_tl_tx_creditsp_o), //output [31:0] PCIE_Controller_Top_pcie_tl_tx_creditsp_o
24     .PCIE_Controller_Top_pcie_tl_tx_creditscp_o(PCIE_Controller_Top_pcie_tl_tx_creditscp_o), //output [31:0] PCIE_Controller_Top_pcie_tl_tx_creditscp_o
25     .PCIE_Controller_Top_pcie_tl_cfg_busdev_o(PCIE_Controller_Top_pcie_tl_cfg_busdev_o), //output [12:0] PCIE_Controller_Top_pcie_tl_cfg_busdev_o
26     .PCIE_Controller_Top_pcie_linkup_o(PCIE_Controller_Top_pcie_linkup_o), //output PCIE_Controller_Top_pcie_linkup_o
27     .PCIE_Controller_Top_pcie_rstn_i(PCIE_Controller_Top_pcie_rstn_i), //input PCIE_Controller_Top_pcie_rstn_i
28     .PCIE_Controller_Top_pcie_tl_clk_i(PCIE_Controller_Top_pcie_tl_clk_i), //input PCIE_Controller_Top_pcie_tl_clk_i
29     .PCIE_Controller_Top_pcie_tl_rx_wait_i(PCIE_Controller_Top_pcie_tl_rx_wait_i), //input PCIE_Controller_Top_pcie_tl_rx_wait_i
30     .PCIE_Controller_Top_pcie_tl_rx_masknp_i(PCIE_Controller_Top_pcie_tl_rx_masknp_i), //input PCIE_Controller_Top_pcie_tl_rx_masknp_i
31     .PCIE_Controller_Top_pcie_tl_tx_sop_i(PCIE_Controller_Top_pcie_tl_tx_sop_i), //input PCIE_Controller_Top_pcie_tl_tx_sop_i
32     .PCIE_Controller_Top_pcie_tl_tx_eop_i(PCIE_Controller_Top_pcie_tl_tx_eop_i), //input PCIE_Controller_Top_pcie_tl_tx_eop_i
33     .PCIE_Controller_Top_pcie_tl_tx_data_i(PCIE_Controller_Top_pcie_tl_tx_data_i), //input [255:0] PCIE_Controller_Top_pcie_tl_tx_data_i
34     .PCIE_Controller_Top_pcie_tl_tx_valid_i(PCIE_Controller_Top_pcie_tl_tx_valid_i), //input [7:0] PCIE_Controller_Top_pcie_tl_tx_valid_i
35     .PCIE_Controller_Top_pcie_tl_int_status_i(PCIE_Controller_Top_pcie_tl_int_status_i), //input PCIE_Controller_Top_pcie_tl_int_status_i
36     .PCIE_Controller_Top_pcie_tl_int_req_i(PCIE_Controller_Top_pcie_tl_int_req_i), //input PCIE_Controller_Top_pcie_tl_int_req_i
37     .PCIE_Controller_Top_pcie_tl_int_msinum_i(PCIE_Controller_Top_pcie_tl_int_msinum_i) //input [4:0] PCIE_Controller_Top_pcie_tl_int_msinum_i
38 );
39
40

```

Appendix **A** PCIe Configuration Register Information

Table A-1 PCIe Capability

| 31:24 | 23:16 | 15:8 | 7:0 | Byte Offset |
|-----------------------|-------|------------------|---------------|-------------|
| Capabilities Register | | Net Cap PTR | Capability ID | 080h |
| Device Capabilities | | | | 084h |
| Device Status | | Device Control | | 088h |
| Link Capabilities | | | | 08Ch |
| Link Status | | Link Control | | 090h |
| Slot Capabilities | | | | 094h |
| Slot Status | | Slot Control | | 098h |
| Root Capabilities | | Root Control | | 09Ch |
| Root Status | | | | 0A0h |
| Device Capabilities 2 | | | | 0A4h |
| Device Status 2 | | Device Control 2 | | 0A8h |
| Link Capabilities | | | | 0ACh |
| Link Status | | Link Control | | 0B0h |
| Slot Capabilities | | | | 0B4h |
| Slot Status | | Slot Control | | 0B8h |

