Libero® SoC v2021.2

SmartFusion®2 and IGLOO®2 High-Speed Serial Interface Configuration User Guide

Introduction

The High-Speed Serial Interface block in the SmartFusion2 and IGLOO2 families provides multiple high speed serial protocols, such as PCIe end-point and XAUI. In addition, it enables the FPGA fabric to connect with the External Physical Coding Sublayer (EPCS) Interface and implement any user-defined protocol in the fabric. The device might contain one or more High Speed Serial Interface blocks depending on the size. See the IGLOO2 FPGA and SmartFusion2 SoC FPGA Datasheet.

Note: The document is intended for Libero SoC SERDES_IF core. For more details about the High Speed Serial Interface, see the UG0447: SmartFusion2 and IGLOO2 FPGA High-Speed Serial Interfaces User Guide.

Table of Contents

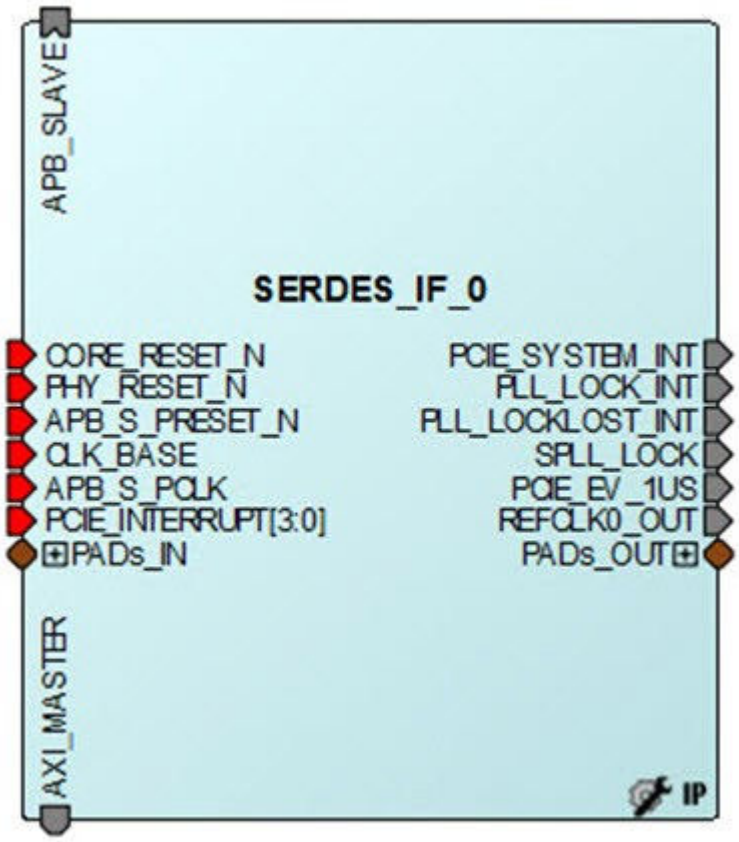
1. Overview

In this section, it is described how to configure a High Speed Serial Interface instance and define how the signals are connected.

To access the High Speed Serial Interface Configurator:

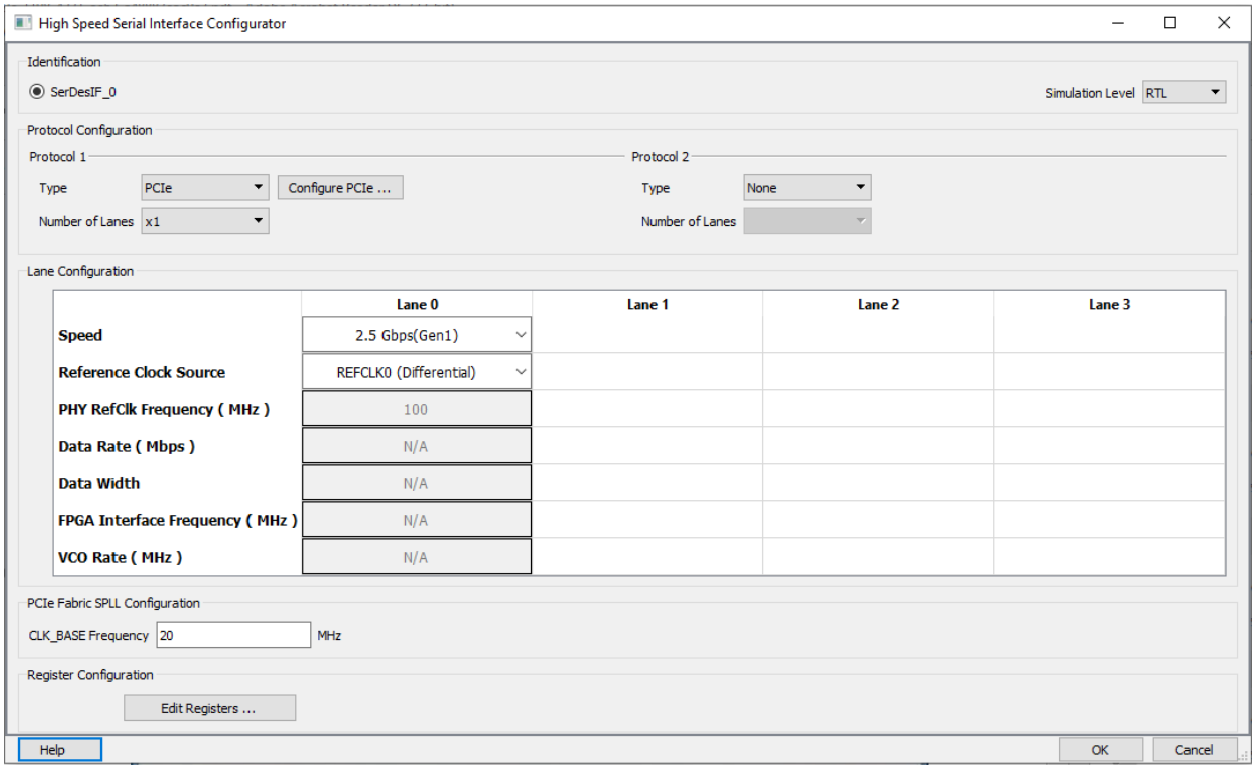
1. Instantiate the High Speed Serial Interface (SerDes) core from the Catalog in the SmartDesign Canvas, as shown in the following figure.

   Figure 1-1. SerDes Block Instantiation on the SmartDesign Canvas



2. Double-click each SerDes block on the Canvas to open the Configurator.
   By default, SERDESIF_0 is checked when you open the Configurator. If the instance name of the SerDes you have instantiated in the SmartDesign canvas is not SERDES_IF_0, uncheck SERDESIF_0 and check the correct SerDes identification box to match the correct SerDes instance name.

Figure 1-2. High Speed Serial Interface Configurator



As you make selections in the core configurator, it automatically narrows down the subsequent choices and defaults. The relevant ports appear in the generated macro.

2. Functionality

This section describes the functionality of the SERDES_IF Core.

2.1 Identification

The SmartFusion2 and IGLOO2 devices might contain one or more High Speed Serial Interface blocks. The first row of check-boxes allows you to identify the High Speed Serial Interface block (SERDESIF_0, SERDESIF_1, SERDESIF_2, and SERDESIF_3) that is configured.

Note: Some devices have only one High Speed Serial Interface block. If so, you must select SERDESIF_0. See the device datasheet (IGLOO2 FPGA and SmartFusion2 SoC FPGA Datasheet) for a list of resources available on a device.

2.2 Protocol Configuration

The IGLOO2 and SmartFusion2 devices have one SerDes block that supports two protocols: Protocol 1 and Protocol 2. For each Protocol, you must configure the Type and Number of Lanes.

2.2.1 Type

Select your Protocol type from the list:

- For Protocol 1, you can select any one of the following types:
  - PCIe
  - PCIe (Reverse)
  - XAUI
  - EPCS
- For Protocol 2, you can select the following types:
  - EPCS (available only when PCIe or PCIe Reverse is selected for Protocol 1)
    When you select the Protocol 1 as PCI or PCIe Reverse, you must click the **Configure PCIe** button to configure additional options for the PCI mode of SerDes. See 2.3 Configure PCIe for details.

Note: You must Configure Protocol 1 before configuring Protocol 2. Protocol 2 types are context sensitive. They depend on the options you have selected in Protocol 1. The Protocol 2 type selection is disabled when you select XAUI for Protocol 1.
Protocol 2 is activated only when PCIe, PCIe Reverse or EPCS is selected for Protocol 1 and you use less than four lanes. See Table 2-1 for the Protocol 1 and Protocol 2 configuration combinations.

2.2.2 Number of Lanes

Select the number of lanes to configure for Protocol 1 from the list:

- X1—Configure for 1 lane
- X2—Configure for 2 lanes
- X4—Configure for all 4 lanes

Note: The options in the list are context sensitive and depend on the Protocol Type.
If Protocol Type is selected as XAUI, all four lanes are selected by default.

2.2.3 Protocol 1 and 2 Combinations

The following table lists the protocol combinations that are feasible within a single High Speed Serial Interface block.

Table 2-1. Available Protocol Combinations

Protocol Type	Protocol #	Lane Width	Lane Assignment	Description	Speed Choices
PCIe	Protocol 1	x1	Lane 0	—	Gen1 (2.5 Gbps), Gen2 (5.0 Gbps)
		x2	Lane 0, Lane 1		
		x4	Lane 0, Lane 1, Lane 2, Lane 3		
PCIe Reverse	Protocol 1	x2	Lane 2, Lane 3	—	Gen1 (2.5 Gbps), Gen2 (5.0 Gbps)
		x4	Lane 0, Lane 1, Lane 2, Lane 3		
XAUI	Protocol 1	x4	Lane 0, Lane 1, Lane 2, Lane 3	—	3.125 Gpbs
EPCS	Protocol 1	x1	Lane 0, 1, 2 or 3	—	Custom Speed
		x2	Lane 0, Lane 1		
		x4	Lane 0, Lane 1, Lane 2, Lane 3		
	Protocol 2	x1	Lane 2 or 3	Available only when Protocol 1 is selected as PCIe, PCIe Reverse or EPCS	
		x2	Lane 2, Lane 3		

2.3 Configure PCIe

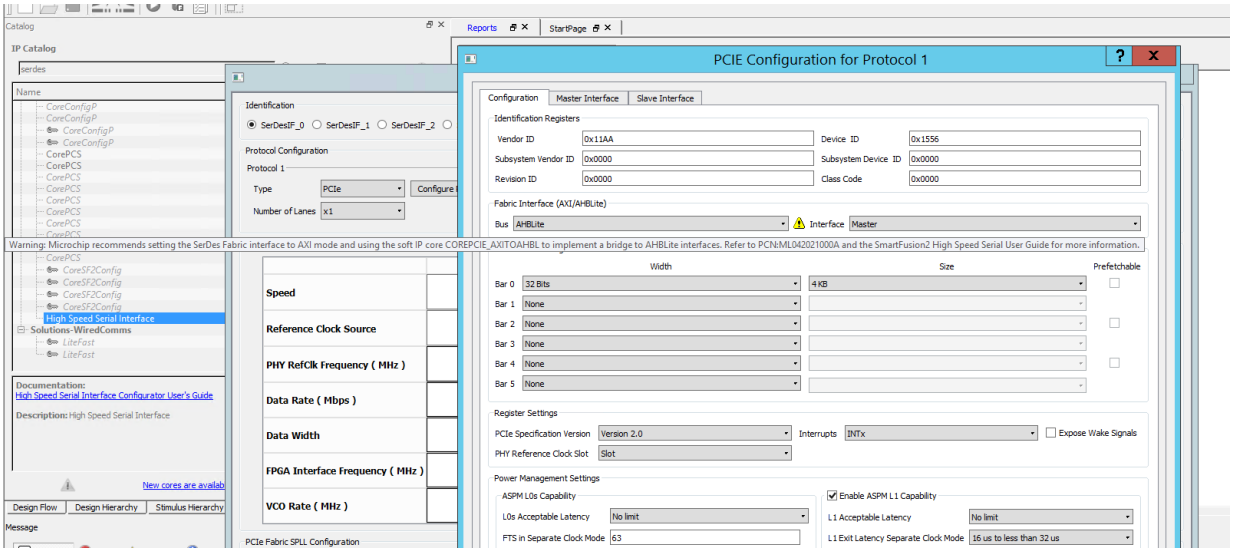
The **Configure PCIe** tab appears only when you select the PCIe or PCI Reverse Protocol.

2.3.1 Configuration

The Configuration tab allows you to configure Identification Registers, Fabric Interface, Base Address Registers, and Options.

In the RTG4 PCIe SerDes Configurator, a warning icon is added and the following warning message is displayed as a tooltip.

Figure 2-1. PCIe Configuration - Configuration Tab



Note: For SmartFusion2, the following tooltip is displayed: Microchip recommends setting the SerDes Fabric interface to AXI mode and using the soft IP core COREPCIE_AXITOAHBL to implement a bridge to AHBLite interfaces. For more information, see PCN_ML042021000A.

The following table lists the conditions when the warning message is displayed.

Table 2-2. Show\Hide Warning Symbol

Bus	Interfce	Description
AXI bus	Master	No Warning
AXI bus	Slave	No Warning
AXI bus	Both	No Warning
AHBLite	Master	Warning
AHBLite	Slave	No Warning
AHBLite	Both	Warning

2.3.1.1 Identification Registers

You can assign 16-bit hexadecimal signatures to the following six identification registers for PCIe:

- **Vendor ID**—0x11AA is the Vendor ID assigned to Microchip by PCI-SIG. Contact Microchip if you need to allocate subsystems under the Microchip vendor ID.
- **Vendor ID**—0x11AA is the Vendor ID assigned to Microchip by PCI-SIG. Contact Microchip if you need to allocate subsystems under the Microchip vendor ID.
- **Subsystem vendor ID—**Card manufacturer's ID.
- **Device ID**—Manufacturer's assigned part number is assigned by the vendor.
- **Revision ID**—Revision number, if available.
- **Subsystem Device ID**—Assigned by the subsystem vendor.
- **Class Code**—PCIe device's generic function.

2.3.1.2 Fabric Interface

Use this field to configure the bus standard (AXI or AHBLite) and interface (Master only, Slave only, or both) for the PCIe protocol.

In PCIe mode, the SerDes block can act as an AXI or AHBLite master.

You must instantiate a COREAXI or CoreAHBLite bus into the SmartDesign Canvas and then connect the Master and/or Slave Bus Interface (BIF) of the SerDes to the Master and/or Slave BIF of the COREAXI bus or CoreAHBLite bus.

2.3.1.3 Base Address Registers

The individual fields of the six Base address registers (Bar 0 to Bar 5) can be configured as follows:

- **Width**—The width on even registers can be 32 bit or 64 bit. If an even register is set to 64 bits wide, the subsequent (odd) register serves as the upper half of 64 bits. The width of odd registers is restricted to 32 bits.
- **Size**—The ranges from 4 KB to 2 GB when width is 32 bit. Some devices support only up to 1 GB. When the width is 64 bit, the size can range from 2 KB to 16 KB.
  Note: M2S150/M2GL150, M2S090/M2GL090, and M2S060/M2GL060 devices support 2 GB; all other devices support 1GB.

  See your device datasheet (IGLOO2 FPGA and SmartFusion2 SoC FPGA Datasheet) for more information.
- **Prefetchable**—Enabled only on even registers with 64 bit width.

2.3.1.4 Options

Options enables you to configure the following:

- **PHY Reference Clock Slot**—Sets your reference clock to slot or independent.
- **L2/P2**—Click the check box to add PCIE_WAKE_N, PCIE_WAKE_REQ, and PCIE_PERST_N ports.
  Note: SERDES_IF block does not support PCIE_PERST_N port.
- **PCIe Specification Version**—Sets the specification version to 1.0, 1.1 or 2.0.
- **Interrupt**—Sets the interrupt to:
  - MSI 1
  - MSI 2
  - MSI 4
  - MSI 8
  - MSI 16
  - MSI 32
  - INTx

Your Interrupt selection sets bit 16 of the PCIE_MSI_CTRL_STATUS register and bits [19:17] of the PCIE_MSI_CTRL_STATUS register as listed in the following table.

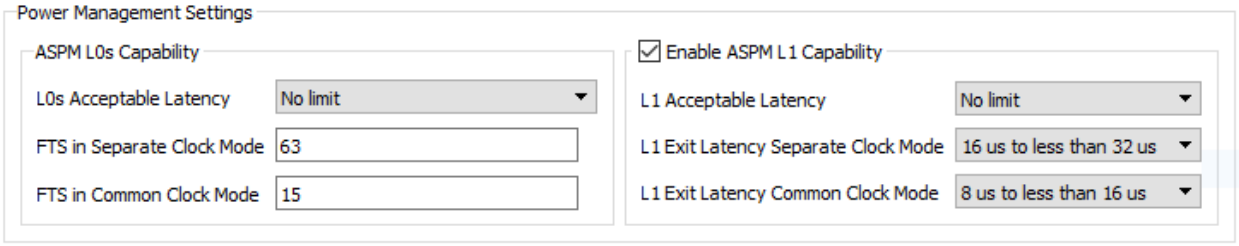
Table 2-3. MSI and Register Settings

Interrupt Selected	Setting for Bit 16 of PCIE_MSI_CTRL_STATUS Register	Setting for Bits [19:17] of PCIE_MSI_CTRL_STATUS Register
MSI 1	1	000
MSI 2	1	001
MSI 4	1	010
MSI 8	1	011
MSI 16	1	100
MSI 32	1	101
INTx	0 (Disable MSI)	000

2.3.2 Power Management Settings

Use the Power Management Settings to configure the Active State Power Management (ASPM) settings. The configurator sets the correct register values for your SerDes block based on the selections made.

Figure 2-2. Power Management Settings



2.3.2.1 ASPM L0s Capability

This is mandatory. Settings available are:

- **L0s Acceptable Latency**—Displays the list to choose one of the following latency values:
  - Maximum of 64 ns
  - Maximum of 128 ns
  - Maximum of 256 ns
  - Maximum of 512 ns
  - Maximum of 1 µs
  - Maximum of 2 µs
  - Maximum of 4 µs
  - No Limit
- **FTS in Separate Clock Mode**—Enter the number of FTS (Fast Training Sequences) required in separate clock mode. Valid values are from 0 through 255.
- **FTS in Common Clock Mode**—Enter the number of FTS (Fast Training Sequences) required in common clock mode. Valid values are from 0 through 255.

2.3.2.2 ASPM L1 Capability

By default, the ASPM L1 Capability is enabled. Configure the settings for ASPM L1 as follows:

- **L1 Acceptable Latency**—Displays the list to choose one of the of following for latency values:
  - Maximum of 1 µs
  - Maximum of 2 µs
  - Maximum of 4 µs
  - Maximum of 8 µs
  - Maximum of 16 µs
  - Maximum of 32 µs
  - Maximum of 64 µs
  - No Limit
- **L1 Exit Latency Separate Clock/Common Clock Mode**—Displays the list to choose one of the following for the exit latency value:
  - Less than 1 µs
  - 1 µs to less than 2 µs
  - 2 µs to less than 4 µs
  - 4 µs to less than 8 µs
  - 8 µs to less than 16 µs
  - 16 µs to less than 32 µs
  - 32 µs to 64 µs
  - More than 64 µs

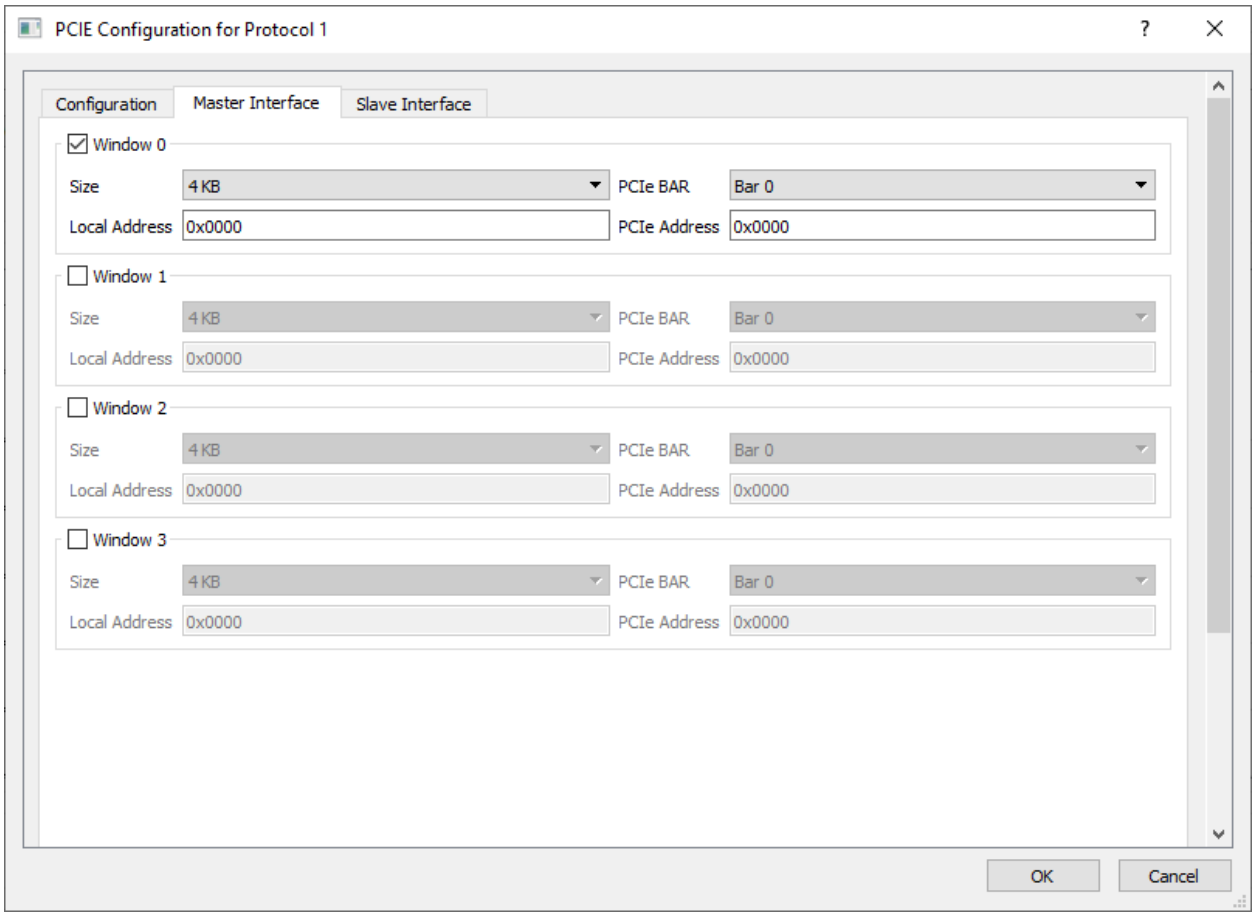
Note: The exit latency value you choose for the Common Clock mode must be smaller than the value of Separate Clock mode.

2.3.3 Master Interface

PCIe/PCIe Reverse Protocol 1 or Protocol 2 enables you to use the Master Interface tab to configure up to four PCI windows (Window 0 to Window 3) with the following parameters as shown in the following figure.

- Size
- PCIe BAR (Base Address Register)
- Local Address
- PCIe Address

Figure 2-3. PCIe Configuration - Master Interface Tab



2.3.3.1 Size

For each of the windows 0 through 3, select one of the available window sizes: 4 KB, 8 KB, 16 KB, 32 KB, 64 KB, 128 KB, 256 KB, 512 KB, 1 MB, 2 MB, 4 MB, 8 MB, 16 MB, 32 MB, 64 MB, 128 MB, 256 MB, 512 MB, 1 GB and 2 GB.

The default selection is 4 KB. The size selected is mapped to bits [31:12] of WindowsX_1 where X can be 0, 1, 2, or 3.

2.3.3.2 PCIe BAR

Select one of the following Base Address Registers (BAR):

- BAR0
- BAR1
- BAR2
- BAR3
- BAR4
- BAR5

- BAR0/1
- BAR2/3
- BAR4/5

Bar Size is mapped to bits [5:0] of WindowsX_2, where X can be 0, 1, 2 or 3, as listed in the following table.

Table 2-4. BAR Size and corresponding Bit Settings

BAR Size	Bit Settings
BAR0, BAR0/1	0x01
BAR 1	0x02
BAR 2, BAR2/3	0x04
BAR 3	0x08
BAR 4, BAR4/5	0x10
BAR5	0x20

2.3.3.3 Local Address

Local Address is 20 bits wide and is mapped to bits [31:12] of Base address AXI Master WindowsX_0. The LSB bits [11:0] of AXI Master WindowsX_0 are reserved and the configurator accounts for these bits. Do not include these reserved bits when you specify the local address.

2.3.3.4 PCIe Address

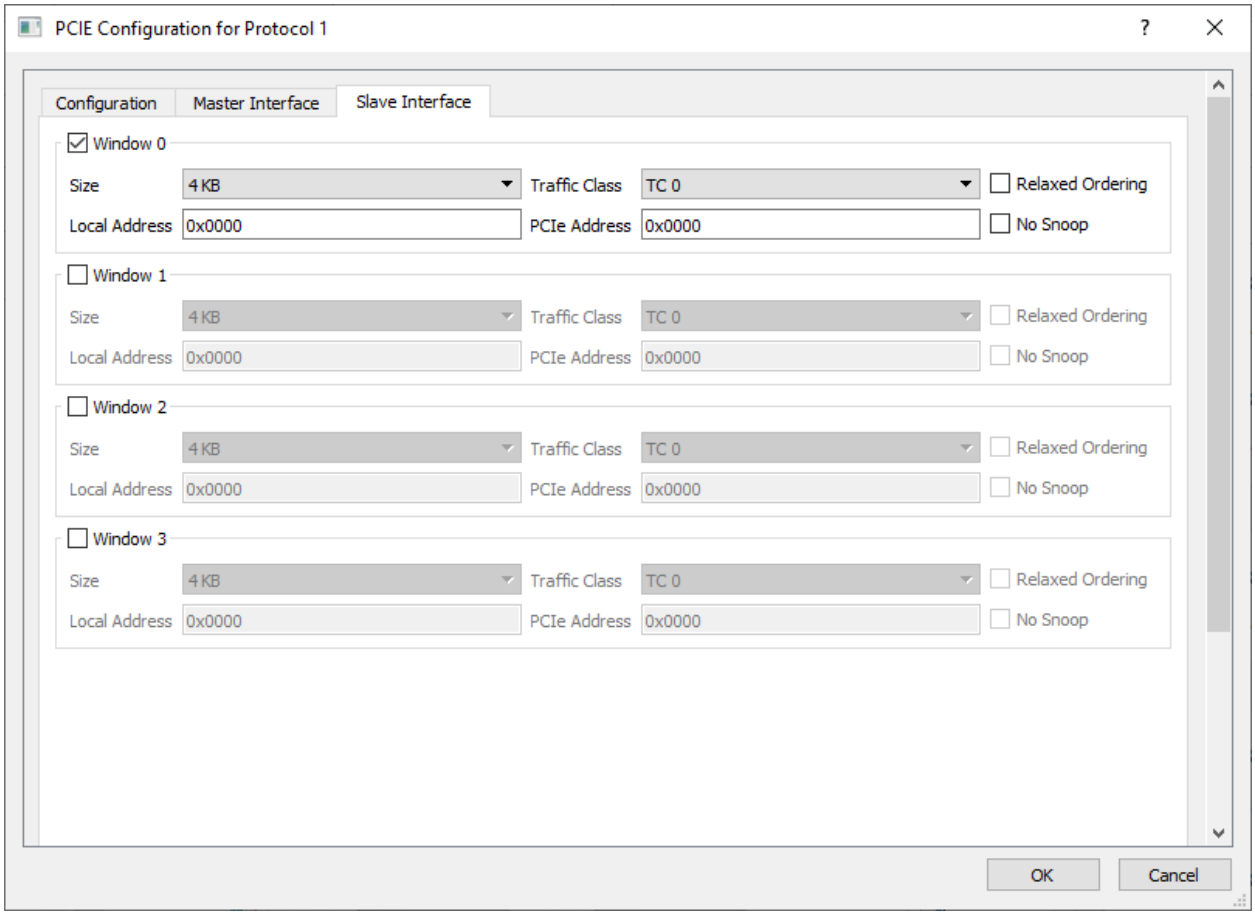
PCIe Address is mapped to bits [31:12] (LSB of Base address AXI Master WindowsX_2) and bits [31:0] (MSB of Base address AXI Master WindowsX_3).

2.3.4 Slave Interface

If you select PCIe or PCIe Reverse Protocol, the Slave Interface tab enables you to configure up to four PCI windows, Window 0 through Window 3, with the following parameters as shown in the following figure.

- Size
- Local Address
- PCIe Address
- Traffic Class: Selects the PCIe Traffic Class in the PCIe packet header.
- Relaxed Ordering: Enables you to generate the PCIe TLP using a selectable relaxed ordering bit.
- No Snoop: Enables you to generate the PCIe TLP using a selectable no snoop bit.

Figure 2-4. PCIe Configuration - Slave Interface Tab



The Size, Local Address, and PCIe Address options are the same as those for the Master Interface. See 2.3.3 Master Interface for more information.

2.3.4.1 Traffic Class

Traffic Class enables you to set your Traffic Class and corresponding register bits, as shown in Table 2-5. The traffic class is required by the PCI Express packet header. Its value determines the relative priority of a given transaction as it traverses the PCIe link. You can use the Traffic Class value to create a priority scheme for different packets.

- TC 0 (Default)
- TC 1
- TC 2
- TC 3
- TC 4
- TC 5
- TC 6
- TC 7

Table 2-5. Traffic Class and Corresponding Bit Setting

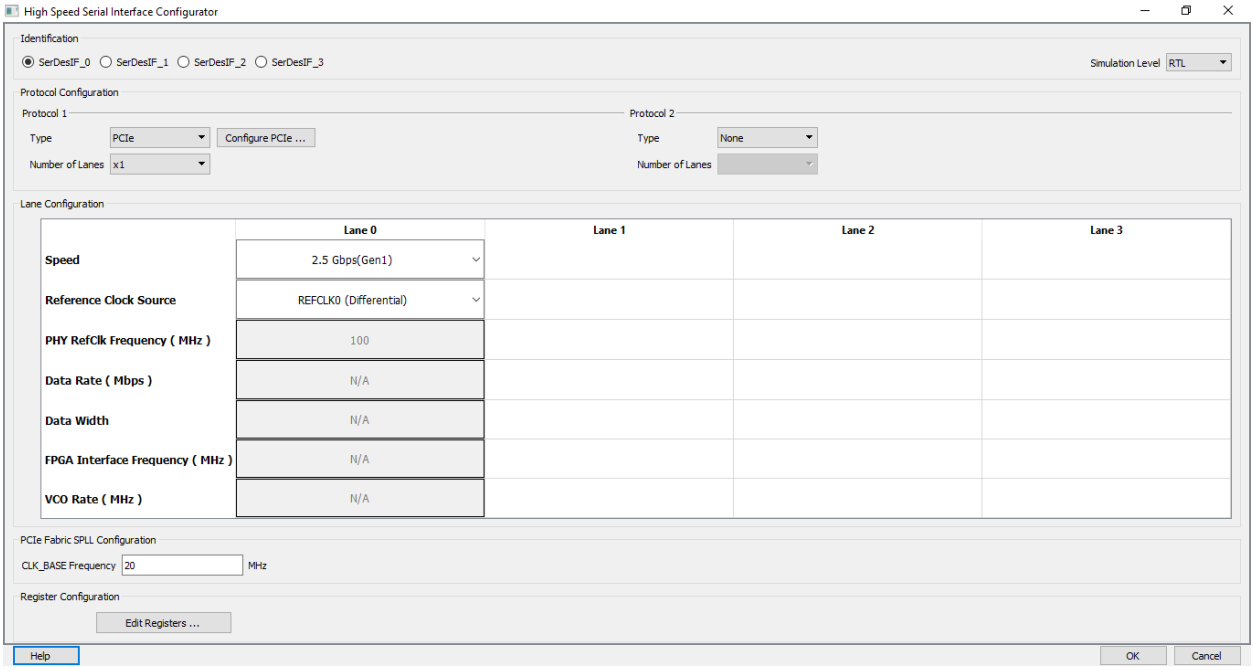
Traffic Class	Bit Setting for WindowX_2[4:2]
TC 0	000
TC 1	001
TC 2	010

.....continued	
Traffic Class	**Bit Setting for WindowX_2[4:2]**
TC 3	011
TC 4	100
TC 5	101
TC 6	110
TC 7	111

2.4 Lane Configuration

Use Lane Configuration to configure up to four lanes for your SerDes. The SerDes can be configured to run in dual-protocol mode. See Table 2-1 for lane configuration for dual mode operation.

Figure 2-5. High Speed Serial Interface Configurator



- **Speed**—Available selections depend on your selected Protocol. See Table 2-1 for the valid speeds.
- **Reference Clock Source**—Two clock sources are available: REFCLK0 and REFCLK1. Each can be differential or single-ended. You can select one of the following options for Protocol 1 and Protocol 2:
  - REFCLK0 (Differential),
  - REFCLK1 (Differential),
  - REFCLK0 (Single-Ended),
  - REFCLK1 (Single-Ended)
  - Fabric (Available only for EPCS Protocol)

  Note: Lane 0 and 1 share the same reference clock and Lane 2 and 3 share the same reference clock. The selected reference clock is always available as REFCLK0_OUT or REFCLK1_OUT and can be used as clock source for logic inside fabric.
- **PHY RefClk Frequency (MHz)**—This is a fixed value for all protocols except EPCS Custom Speed, in which case you can enter values between 100 and 160 MHz.
- **Data Rate (Mbps)**—Read-only fixed value for all protocols except EPCS custom speed, in which case you can select the data rate from the list. Data rates are computed based on the PHY RefClk frequency.

- **Data Width**—Read-only fixed value for all protocols except EPCS custom speed. For EPCS, the data width varies with data rate (Mbps) as follows:
  - 20 bits (for 5000 Mbps and 2500 Mbps)
  - 16 bits (for 4000 Mbps or 2000 Mbps)
  - 10 bits (2500 Mbps or 1250 Mbps)
  - 8 bits (for 2000 Mbps or 1000 Mbps)
  - 5 bits (for 1250 Mbps)
  - 4 bits (for 1000 Mbps)

  The displayed value is computed and updated based on your selected PHY RefClk frequency and data rate.
- **FPGA Interface Frequency (MHz)**—Read-only fixed value for all protocols except EPCS custom speed. The displayed value is computed and updated based on the PHY RefClk frequency and data rate you select.
- **VCO Rate (MHz)**—Read-only fixed value for all protocols except EPCS custom speed. The displayed value is computed and updated based on the PHY RefClk frequency and data rate you select.

2.5 PCIe/XAUI Fabric SPLL Configuration

The SPLL configuration fields are relevant only for PCIe and XAUI protocols (Figure 2-5). For the PCIe protocol, enter a valid value between 20 and 200 MHz for the CLK_BASE frequency.

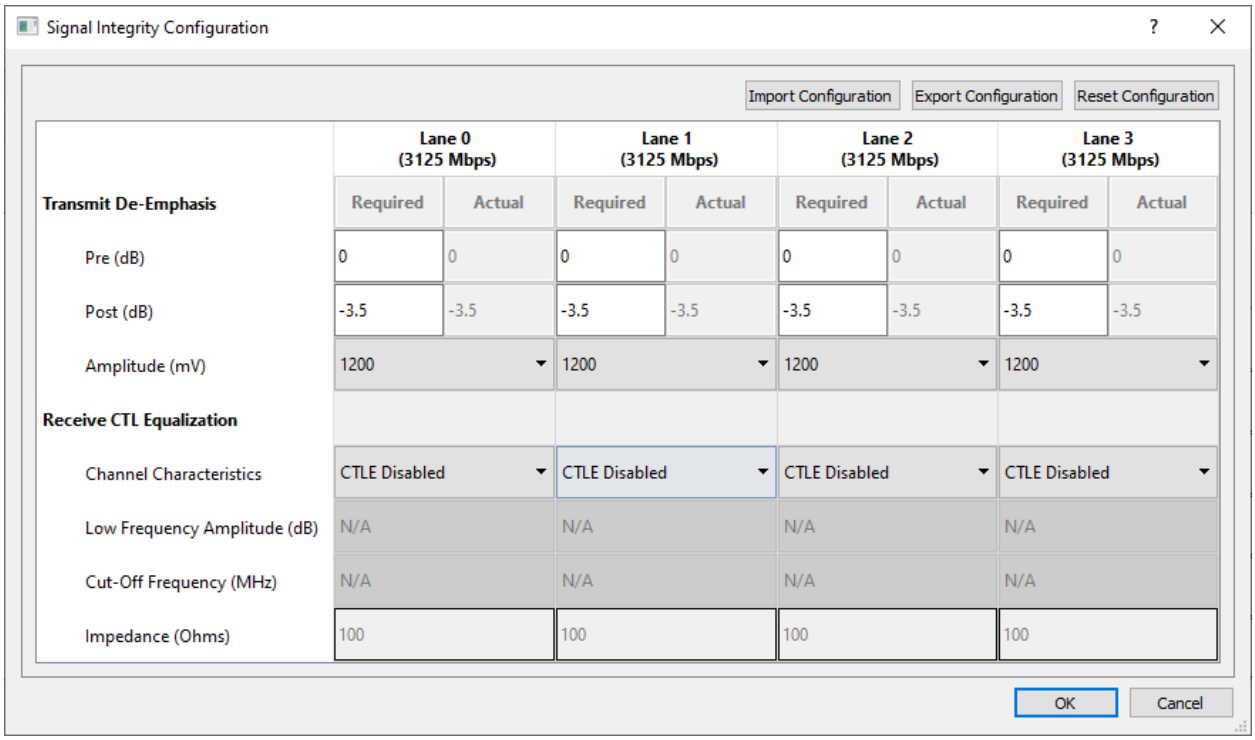
For the XAUI protocol, the CLK_BASE frequency is read-only and fixed at 156.25 MHz.

2.6 Signal Integrity Options

For the XAUI and EPCS protocols, click **Signal Integrity Options** to open the Signal Integrity Configuration dialog box.

The Signal Integrity Configuration dialog box gives you controls to maintain signal integrity and to mitigate signal integrity problems.

Figure 2-6. Signal Integrity Configuration Dialog Box



2.7 Transmit De-Emphasis

1. Enter any value between 0.0 and 36.1 (in dB) in the **Required** edit box for both Pre-Transmit and Post-Transmit stage. Not all values are supported. See the following table for all actual values supported.
2. The value you enter in the **Required** box is matched to the closest valid actual value and reported in the actual box.
3. The Configurator sets appropriate values for LANE<n>_TX_PRE_RATIO and LANE<n>_TX_PST_RATIO registers based on the Actual value, as shown in the following table.
4. The default value for **Required** is 0 for **Pre-Transmit** and 3.5 for **Post-Transmit.**
5. The LANE<n>_TX_AMP_RATIO lane register is always set to 0x80 for all lanes.

Table 2-6. EPCS Configuration for Different Data Width

Feature	Control Registers	Actual Value = Value Programmed in Register
De-Emphasis Pre	LANE<n>_TX_PRE_RATIO	0 dB = 0x0 0.1 dB = 0x1 0.3 dB = 0x2
De-Emphasis Post	LANE<n>_TX_PST_RATIO	0.4 dB = 0x3 0.5 dB = 0x4 0.7 dB = 0x5 0.9 dB = 0x6 1 dB = 0x7 1.2 dB = 0x8 1.3 dB = 0x9 1.5 dB = 0xa 1.6 dB = 0xb 1.8 dB = 0xc 2 dB = 0xd 2.1 dB = 0xe 2.3 dB = 0xf 2.5 dB = 0x10 2.7 dB = 0x11 2.9 dB = 0x12 3 dB = 0x13 3.3 dB = 0x14 3.5 dB = 0x15 3.7 dB = 0x16 3.9 dB = 0x17 4 dB = 0x18 4.3 dB = 0x19 4.5 dB = 0x1a 4.8 dB = 0x1b

.....continued		
Feature	**Control Registers**	**Actual Value = Value Programmed in Register**
De-Emphasis Post (contd.)	LANE<n>_TX_PST_RATIO (contd.)	5 dB = 0x1c 5.2 dB = 0x1d 5.5 dB = 0x1e 5.8 dB = 0x1f 6 dB = 0x20 6.3 dB = 0x21 6.5 dB = 0x22 7 dB = 0x23 7.2 dB = 0x24 7.5 dB = 0x25 7.8 dB = 0x26 8 dB = 0x27 8.5 dB = 0x28 9 dB = 0x29 9.3 dB = 0x2a 9.7 dB = 0x2b
De-Emphasis Post (contd.)	LANE<n>_TX_PST_RATIO (contd.)	10.1 dB = 0x2c 10.5 dB = 0x2d 11 dB = 0x2e 11.5 dB = 0x2f 12 dB = 0x30 12.6 dB = 0x31 13.2 dB = 0x32 13.8 dB = 0x33 14.5 dB = 0x34 15.2 dB = 0x35 16.1 dB = 0x36 17 dB = 0x37 18 dB = 0x38 19.2 dB = 0x39 20.5 dB = 0x3a 22.1 dB = 0x3b 24 dB = 0x3c 26.5 dB = 0x3d 30.1 dB = 0x3e 36.1 dB = 0x3f

Note: LANE<n> denotes the lane number where <n> can be 0, 1, 2 or 3.

For example, if you enter 2.4 dB in the required box, 2.5 dB (the closest match) is displayed in the actual box and the registers are set as follows:

1. LANE<n>_TX_PRE_RATIO registers are set to 0x10
2. LANE<n>_TX_PST_RATIO registers are set to 0x10
3. LANE<n>_TX_AMP_RATIO registers are set to 0x80

2.8 Receive CTL Equalization

You can set the values to control the Continuous Time Linear (CTL) equalization of the receiver. The list contains four selections (Channel Characteristics) to control equalization for each of the enabled lanes (Figure 2-6).

- User Defined
- Short-Reach (Default)
- Medium-Reach
- Long-Reach

For the User Defined selection, you can enter values for Low Frequency Amplitude (dB), Cut-Off Frequency (MHz), and Impedance value (Ohms). For all other selections (Short-Reach, Medium-Reach, and Long-Reach), the values for Low Frequency Amplitude (dB), Cut-Off Frequency (MHz), and Impedance (Ohm) are set and read-only.

The Configurator sets the register values based on your selections (Short/Medium/Long Reach) and the Data Rate of the lanes as per the following table.

For example, if **Short-Reach** is selected, LANE<n>_RE_AMP_RATIO and LANE<n>_RE_CUT_RATIO lane registers are automatically set with values based on the Data rate of that Lane for Short-Reach.

Table 2-7. Amplitude/Cut-Off Frequency Ratio Register Values

Feature	Control Registers	Condition	Value Programmed in Register
CTL Equalization	LANE<n>_RE_AMP_RATIO LANE<n>_RE_CUT_RATIO	Data rate of 1G - 3G	**Short-Reach** LANE<n>_RE_AMP_RATIO = 0x0 LANE<n>_RE_CUT_RATIO = 0x0 **Medium Reach** LANE<n>_RE_AMP_RATIO = 0x0 LANE<n>_RE_CUT_RATIO = 0x0 **Long Reach** LANE<n>_RE_AMP_RATIO = 0x0 LANE<n>_RE_CUT_RATIO = 0x0
		Data rate of 3G - 4G	**Short-Reach** LANE<n>_RE_AMP_RATIO = 0x0 LANE<n>_RE_CUT_RATIO = 0x0 **Medium Reach** LANE<n>_RE_AMP_RATIO = 0x0 LANE<n>_RE_CUT_RATIO = 0x0 **Long Reach** LANE<n>_RE_AMP_RATIO = 0x20 LANE<n>_RE_CUT_RATIO = 0x0
		Data rate of 4G - 5G	**Short-Reach** LANE<n>_RE_AMP_RATIO = 0x0 LANE<n>_RE_CUT_RATIO = 0x0 **Medium Reach** LANE<n>_RE_AMP_RATIO = 0x80 LANE<n>_RE_CUT_RATIO = 0x60 **Long Reach** LANE<n>_RE_AMP_RATIO = 0x80 LANE<n>_RE_CUT_RATIO = 0x80

Note: LANE<n> denotes the lane number where <n> can be 0, 1, 2, or 3.

2.9 Power Down Register Settings

This controls the physical reset behavior of the EPCS SerDes when there is a lack of RX activity.

- Enabled (Physical Reset behavior is enabled)
- Disabled (Physical Reset behavior is disabled)

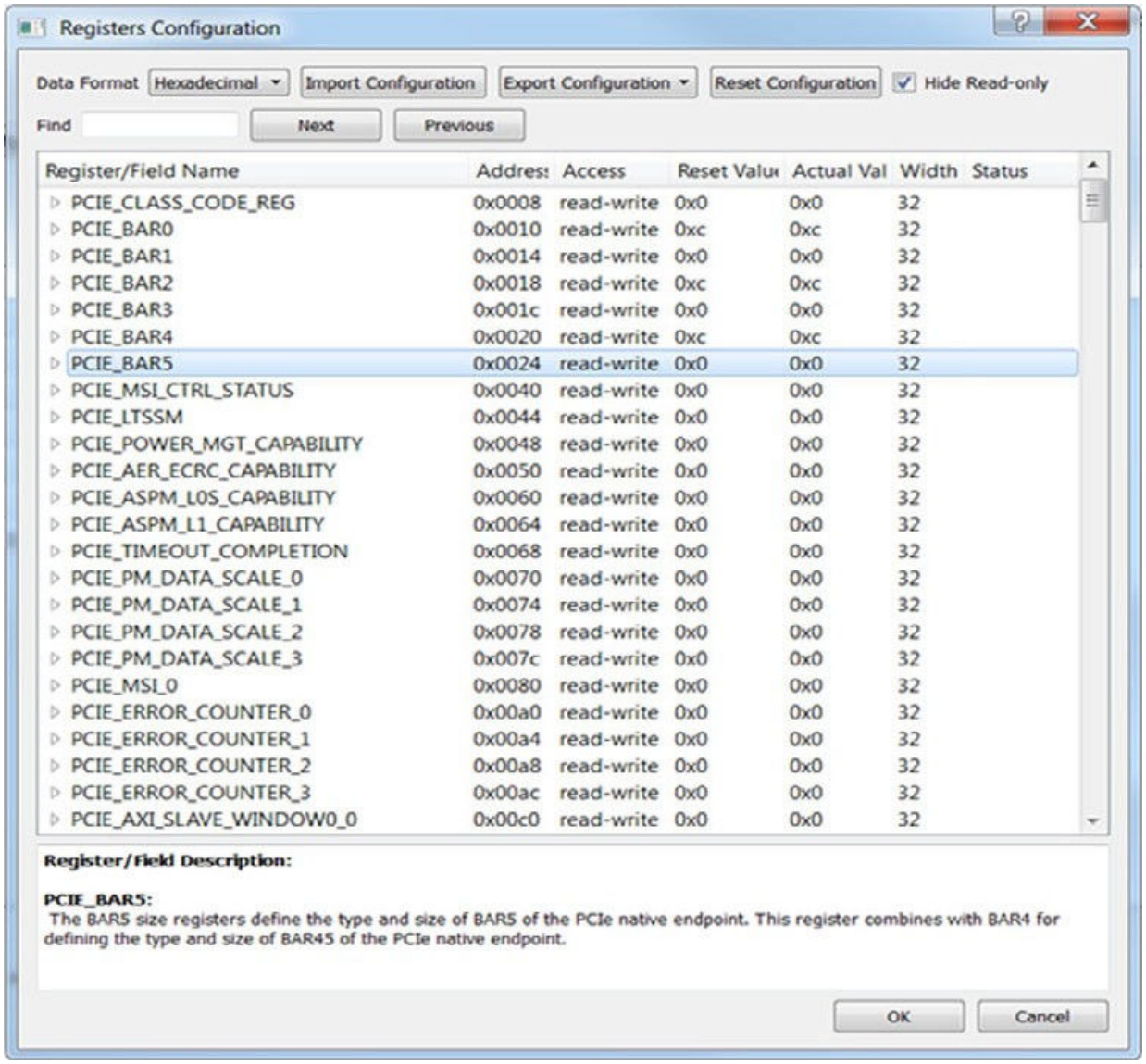
2.10 High Speed Serial Interface Control Registers

The High Speed Serial Interface has a set of registers that can be configured at run time. The configuration values for these registers represent different parameters, for example, AXI BAR Window. For details about these registers, see the UG0447: SmartFusion2 and IGLOO2 FPGA High-Speed Serial Interfaces User Guide.

2.10.1 High Speed Serial Interface Registers Configuration

To enter the High Speed Serial Interface configuration values, specify the register values when you configurie the High Speed Serial Interface. Click **Edit Registers** in the High Speed Serial Interface Configurator (Figure 1-2) to open the Registers Configuration dialog box (see the following figure). Data entered in this configurator is written at power-up in the High Speed Serial Interface registers, as described in the SmartFusion2 DDR Controller and Serial High Speed Controller Initialization Methodology document.

Figure 2-7. High Speed Serial Interface Registers Configuration Dialog Box



Alternatively, you can click **Import Configuration** and import an existing configuration text file to configure the registers.

The Registers Configuration dialog box enables you to enter High Speed Serial Interface register values using a graphical interface. The dialog box has the following features:

- **Registers Table**—Enter register values one-by-one using the Registers Table. To enter a register value, expand the register data tree (using the arrow or + sign), and click the **Actual Value** column to edit.
- **Import Configuration**—Import complete register configurations from text files. Register configuration syntax is shown in the following; Microchip recommends using this method.

- **Export Configuration**—You can export the current register configuration data into a text file. The syntax of the exported file is the same as that of importable register configuration text files. For example:

PCIE_AXI_MASTER_WINDOW0_0	0x00000000
PCIE_AXI_MASTER_WINDOW0_1	0xfffff001
PCIE_AXI_MASTER_WINDOW0_2	0x00000000
PCIE_AXI_MASTER_WINDOW0_3	0x00000000
PCIE_AXI_MASTER_WINDOW1_0	0x00001000
PCIE_AXI_MASTER_WINDOW1_1	0xfffff001
PCIE_AXI_MASTER_WINDOW1_2	0x4

- **Reset Configuration**—Click Reset Configuration to undo any changes you have made to the register configuration. This deletes all register configuration data and you must either re-import or reenter this data. The data are reset to the hardware reset values.
- **Hide Read-Only Registers**—Enables you to show or hide the read-only registers in the Register Table. These registers are mostly status registers and do not contribute to the configuration.

When you generate your FPGA, the configuration register data entered in this configurator is used to initialize the High Speed Serial Interface simulation model when performing a BFM simulation.

2.11 Firmware (SmartFusion2 Only)

When you generate the SmartDesign, the following files are generated in the <project dir>/firmware/ drivers_config/ sys_config directory. These files are required for the CMSIS firmware core to compile properly and contain information regarding your current design, including peripheral configuration data and clock configuration information for the MSS. Do not edit these files manually; they are recreated every time your root design is regenerated.

- sys_config.c
- sys_config.h
- sys_config_SERDESIF_<0-3>.h - High Speed Serial Interface configuration data
- sys_config_SERDESIF_<0-3>.c - High Speed Serial Interface configuration data

2.12 Simulation Level

There are three levels of ModelSim simulation supported for the High Speed Serial Interface block depending on the selected protocol. See the SmartFusion2 FPGA High Speed Serial Interface Simulation User Guide for more information.

- **BFM_CFG**—This level provides a bus functional model of only the APB configuration bus of the High Speed Serial Interface block. You can write and read the different configuration and status bits from the High Speed Serial Interface block through its APB slave interface. The status bits value does not change based on the APB state; they are kept at their reset values. This simulation level is available for all protocols.
- **BFM_PCIe**—This simulation level provides the BFM_CFG level plus the ability to communicate with the High Speed Serial Interface block through the master and slave AXI or AHB bus interfaces. Although, no serial communication actually goes through the High Speed Serial Interface block, this scenario enables you to validate the fabric interface connections. This simulation level is only available for the PCIe protocol.
- **RTL**—This simulation level enables you to fully simulate the High Speed Serial Interface block from the fabric interface to the serial data interface. This results in a longer simulation run time. This simulation level is available for all protocols.

2.12.1 Simulation Files - SmartFusion2

When you generate the SmartDesign associated with your MSS, the following simulation files are generated in the `<project dir>/simulation` directory:

- **test.bfm**—Top-level BFM file, first executed during any simulation that exercises the SmartFusion2 MSS' Cortex-M3 processor. It executes `peripheral_init.bfm` and `user.bfm`, in that order.
- **peripheral_init.bfm**—Contains the BFM procedure that emulates the CMSIS::SystemInit() function run on the Cortex-M3 before you enter the main() procedure. It copies the configuration data for any peripheral used in the design to the correct peripheral configuration registers and then waits for all the peripherals to be ready before asserting that you can use these peripherals.
- **SERDESIF_<0-3>_init.bfm**—Contains BFM write commands that simulate writes of the High Speed Serial Interface Configuration register data you entered (using the Edit Registers dialog box) into the High Speed Serial Interface registers.
- **SERDESIF_<0-3>_user.bfm**—Intended for user commands that simulate transactions being initiated off-chip (via the SerDes interface). You can simulate the datapath by adding your own commands in this file. Commands in this file are executed after `peripheral_init.bfm` has completed.
- **user.bfm**—Intended for user commands. You can simulate the datapath by adding your own commands in this file. Commands in this file are executed after `peripheral_init.bfm` has completed.

Using the preceding files, the configuration path is simulated automatically. You only need to edit the `user.bfm` file to simulate the datapath. Do not edit the `test.bfm`, `peripheral_init.bfm`, or `SERDESIF_<0- 3>_init.bfm` files as these files are recreated every time your root design is regenerated.

2.12.2 Simulation Files - IGLOO2

When you generate the SmartDesign associated with your HPMS, the following simulation files are generated in the `<project dir>/simulation` directory:

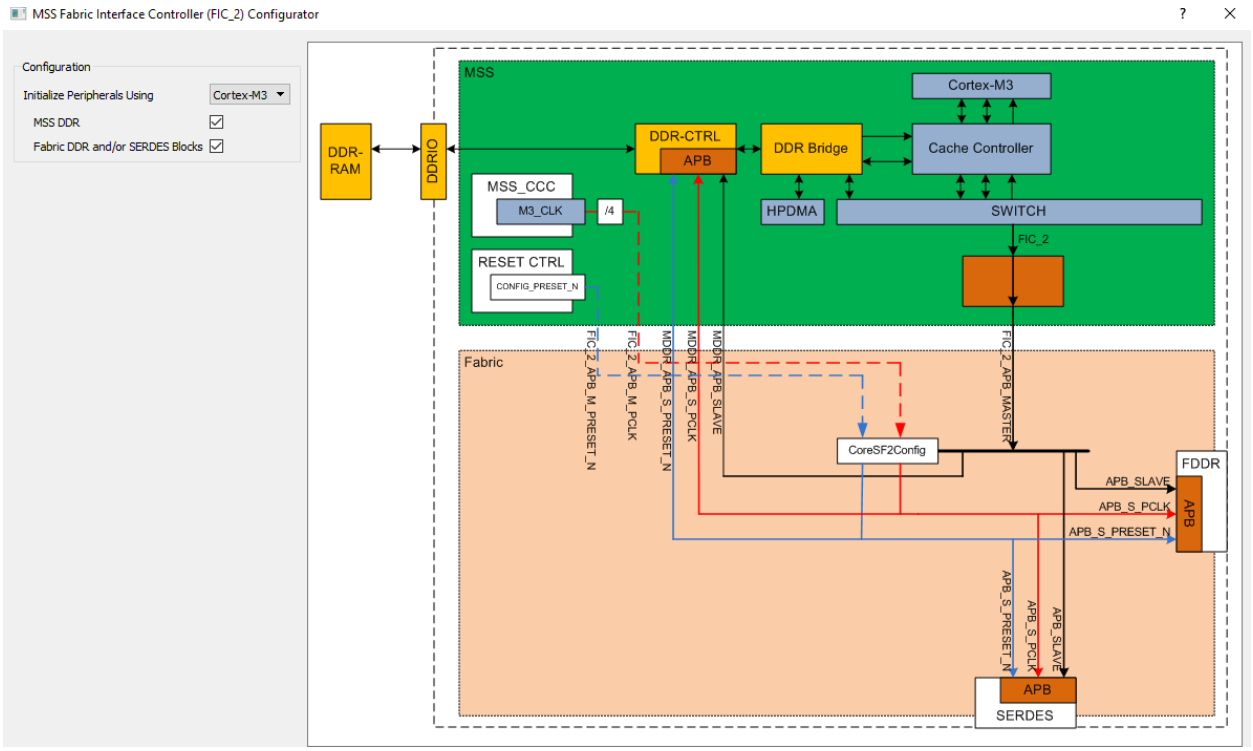
- **ENVM_init.mem**—In IGLOO2 designs, the configuration data for any peripheral used in the design are stored in the `ENVM_init.mem file`. The IGLOO2 simulation library uses this file. Libero SoC creates this file for the simulation just prior to the simulation run.
- **SERDESIF_<0-3>_init.bfm**—Contains BFM write commands that simulate writes of the High Speed Serial Interface Configuration register data you entered (using the Edit Registers dialog box) into the High Speed Serial Interface registers.
- **SERDESIF_<0-3>_user.bfm**—Intended for user commands that simulate transactions are initiated off-chip (via the SerDes interface). You can simulate the datapath by adding your own commands in this file. Commands in this file are executed after `peripheral_init.bfm` has completed.

2.13 High Speed Serial Interface Configuration Path - SmartFusion2

The configuration register data are used by the CMSIS::SystemInit() function compiled with your firmware application code. The SystemInit() function is run before the user main() function in your application. The Peripheral Initialization solution requires that, in addition to specifying High Speed Serial Interface Configuration register values, you configure the APB configuration data path in the MSS (FIC_2).

The SystemInit() function writes the data to the High Speed Serial Interface configuration registers via the FIC_2 APB interface.

Figure 2-8. FIC_2 Configurator Overview

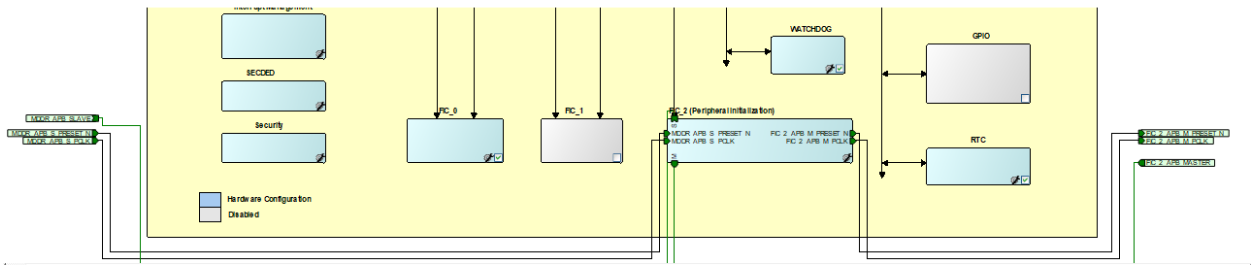


To configure the FIC_2 interface:

1. Open the FIC_2 configurator dialog box (Figure 2-8) from the MSS configurator.
2. Select **Initialize peripherals using Cortex-M3**. Ensure to click the check box to enable **Fabric DDR and/or SerDes blocks** and the MSS DDR option, if you use.
3. Click **OK** to save your settings. This exposes the FIC_2 configuration ports (Clock, Reset, and APB bus interfaces), as shown in Figure 2-9.
4. Generate MSS. The FIC_2 ports (FIC_2_APB_MASTER, FIC_2_APB_M_PCLK, and FIC_2_APB_M_RESET_N) are now exposed at the MSS interface and can be connected to the CoreSF2Config and CoreSF2Reset as per the **Peripheral Initialization** solution specification.

For details on configuring and connecting the CoreSF2Config and CoreSF2Reset cores, see SmartFusion2 DDR Controller and Serial High Speed Controller Initialization Methodology.

Figure 2-9. FIC_2 Ports



2.14 High Speed Serial Interface Configuration Path - IGLOO2 Initialization

You must use System Builder for IGLOO2 designs that use the SerDes block. System Builder generates an APB bus to handle the internals of the High Speed Serial Interface Configuration Path used for the initial SerDes block configuration and initialization.

The SerDes block is always configured as an APB Slave to the System Builder Block; it cannot be configured as a master. You must configure a fabric master to use the SerDes block as a slave.

After generating your System Builder Block, you must connect the SerDes configuration path signals to the System Builder Block:

- Connect the APB_SLAVE BIF of the SerDes block to the corresponding slave BIF of the System Builder block.
- Connect the APB_S_PCLK port of the SerDes block to the INIT_APB_S_PCLK port of the System Builder block.
- Connect the APB_S_PRESET_N port of the SerDes block to the INIT_APB_S_PRESET_N port of the System Builder block.

SerDes initialization data are automatically loaded into the SerDes initialization registers during device bootup.

3. Port Description

This chapter includes the following sections:

- APB Ports (3.1 APB Ports)
- PCIe Ports (3.2 PCIe Ports)
- PCIe AXI Master Ports (3.3 PCIe AXI Master Ports)
- PCIe AXI Slave Ports (3.4 PCIe AXI Slave Ports)
- PCIe AHBLite Master Ports (3.5 PCIe AHBLite Master Ports)
- PCIe AHBLite Slave Ports (3.6 PCIe AHBLite Slave Ports)
- XAUI Ports (3.7 XAUI Ports)
- EPCS Ports per Lane (3.8 EPCS Ports per Lane)
- PAD Ports (3.9 PAD Ports)

3.1 APB Ports

The following table lists the various APB ports.

Table 3-1. APB Ports

Ports	Direction	Port Group
APB_S_PRDATA[31:0]	OUT	APB_SLAVE
APB_S_PREADY	OUT	
APB_S_PSLVERR	OUT	
APB_S_PADDR[13:2]	IN	
APB_S_PENABLE	IN	
APB_S_PSEL	IN	
APB_S_PWDATA[31:0]	IN	
APB_S_PWRITE	IN	
APB_S_PCLK	IN	
APB_S_PRESET_N	IN	

3.2 PCIe Ports

The following table lists the various PCIe ports.

Table 3-2. PCIe Ports

Ports	Direction
CORE_RESET_N	IN
PHY_RESET_N	IN
CLK_BASE	IN
PCIE_INTERRUPT[3:0]	IN
PCIE_SYSTEM_INT	OUT
SPLL_LOCK	OUT
PLL_LOCK_INT	OUT

.....continued	
Ports	**Direction**
PLL_LOCKLOST_INT	OUT
PCIE_EV_1US	OUT
PCIE_WAKE_N	OUT
PCIE_WAKE_REQ	IN
PCIE_PERST_N	IN
REFCLK<x>_OUT where x can be 0 or 1 depending on whether REFCLK0 or REFCLK1 is selected as the Reference Clock Source.	OUT

3.3 PCIe AXI Master Ports

The following table lists the various PCIe AXI master ports.

Table 3-3. PCIe AXI Master Ports

Ports	Direction	Port Group
AXI_M_AWID[3:0]	OUT	AXI_MASTER
AXI_M_AWADDR[31:0]	OUT	
AXI_M_AWLEN[3:0]	OUT	
AXI_M_AWSIZE[1:0]	OUT	
AXI_M_AWSIZE[1:0]	OUT	
AXI_M_AWVALID	OUT	
AXI_M_AWREADY	IN	
AXI_M_WID[3:0]	OUT	
AXI_M_WSTRB[7:0]	OUT	
AXI_M_WLAST	OUT	
AXI_M_WVALID	OUT	
AXI_M_WDATA[63:0]	OUT	
AXI_M_WREADY	IN	
AXI_M_BID[3:0]	IN	
AXI_M_BRESP[1:0]	IN	
AXI_M_BVALID	IN	
AXI_M_BREADY	OUT	
AXI_M_ARID[3:0]	OUT	
AXI_M_ARADDR[31:0]	OUT	
AXI_M_ARLEN[3:0]	OUT	
AXI_M_ARSIZE[1:0]	OUT	
AXI_M_ARBURST[1:0]	OUT	
AXI_M_ARVALID	OUT	
AXI_M_ARREADY	IN	
AXI_M_RID[3:0]	IN	
AXI_M_RDATA[63:0]	IN	
AXI_M_RRESP[1:0]	IN	
AXI_M_RLAST	IN	
AXI_M_RVALID	IN	
AXI_M_RREADY	OUT	

3.4 PCIe AXI Slave Ports

The following table lists the various PCIe AXI slave ports.

Table 3-4. PCIe AXI Slave Ports

Ports	Direction	Port Group
AXI_S_AWID[3:0]	IN	AXI_SLAVE
AXI_S_AWADDR[31:0]	IN	
AXI_S_AWLEN[3:0]	IN	
AXI_S_AWSIZE[1:0]	IN	
AXI_S_AWBURST[1:0]	IN	
AXI_S_AWVALID	IN	
AXI_S_AWREADY	OUT	
AXI_S_AWLOCK[1:0]	IN	
AXI_S_WID[3:0]	IN	
AXI_S_WSTRB[7:0]	IN	
AXI_S_WLAST	IN	
AXI_S_WVALID	IN	
AXI_S_WDATA [63:0]	IN	
AXI_S_WREADY	OUT	
AXI_S_BID[3:0]	OUT	
AXI_S_BRESP[1:0]	OUT	
AXI_S_BVALID	OUT	
AXI_S_BREADY	IN	
AXI_S_ARID[3:0]	IN	
AXI_S_ARADDR[31:0]	IN	
AXI_S_ARLEN[3:0]	IN	
AXI_S_ARSIZE[1:0]	IN	
AXI_S_ARBURST[1:0]	IN	
AXI_S_ARVALID	IN	
AXI_S_ARLOCK[1:0]	IN	
AXI_S_ARREADY	OUT	
AXI_S_RID[3:0]	OUT	
AXI_S_RDATA[63:0]	OUT	
AXI_S_RRESP[1:0]	OUT	
AXI_S_RLAST	OUT	
AXI_S_RVALID	OUT	
AXI_S_RREADY	IN	

3.5 PCIe AHBLite Master Ports

The following table lists the various PCIe AHBLite master ports.

Table 3-5. PCIe AHBLite Master Ports

Ports	Direction	Port Group
AHB_M_HADDR[31:0]	OUT	AHB_MASTER
AHB_M_HBURST[1:0]	OUT	
AHB_M_HSIZE[1:0]	OUT	
AHB_M_HTRANS[1:0]	OUT	
AHB_M_HWRITE	OUT	
AHB_M_HWDATA[31:0]	OUT	
AHB_M_HREADY	IN	
AHB_M_HRESP	IN	
AHB_M_HRDATA[31:0]	IN	

3.6 PCIe AHBLite Slave Ports

The following table lists the various PCIe AHBLite slave ports.

Table 3-6. PCIe AHBLite Slave Ports

Ports	Direction	Port Group
AHB_S_HSEL	IN	AHB_SLAVE
AHB_S_HADDR[31:0]	IN	
AHB_S_HBURST[1:0]	IN	
AHB_S_HSIZE[1:0]	IN	
AHB_S_HTRANS[1:0]	IN	
AHB_S_HWRITE	IN	
AHB_S_HWDATA[31:0]	IN	
AHB_S_HREADYOUT	OUT	
AHB_S_HRESP	OUT	
AHB_S_HREADY	IN	
AHB_S_HRDATA[31:0]	OUT	

3.7 XAUI Ports

The following table lists the various XAUI ports.

Table 3-7. XAUI ports

Ports	Direction
XAUI_RXD[63:0]	OUT
XAUI_RXC[7:0]	OUT
XAUI_RX_CLK	OUT
XAUI_VNDRESLO[31:0]	OUT

.....continued	
Ports	**Direction**
XAUI_VNDRESHI[31:0]	OUT
XAUI_MMD_MDC	IN
XAUI_MMD_MDI	IN
XAUI_MMD_MDI_EXT	IN
XAUI_MMD_MDOE_IN	IN
XAUI_MMD_PRTAD[4:0]	IN
XAUI_MMD_DEVID[4:0]	IN
XAUI_LOOPBACK_IN	IN
XAUI_MDC_RESET	IN
XAUI_TX_RESET	IN
XAUI_RX_RESET	IN
XAUI_TXD[63:0]	IN
XAUI_TXC[7:0]	IN
XAUI_MMD_MDO	OUT
XAUI_MMD_MDOE	OUT
XAUI_LOWPOWER	OUT
XAUI_LOOPBACK_OUT	OUT
XAUI_MDC_RESET_OUT	OUT
XAUI_TX_RESET_OUT	OUT
XAUI_RX_RESET_OUT[3:0]	OUT
CORE_RESET_N	IN
PHY_RESET_N	IN
SPLL_LOCK	OUT
PLL_LOCK_INT	OUT
PLL_LOCKLOST_INT	OUT
XAUI_OUT_CLK	OUT
XAUI_PMA_READY_N	OUT
REFCLK<x>_OUT where x can be 0 or 1 depending on whether REFCLK0 or REFCLK1 is selected as the Reference Clock Source.	OUT

3.8 EPCS Ports per Lane

The following table lists the various EPCS ports per lane.

Table 3-8. EPCS Ports per Lane

Ports	Direction	Ports Group
EPCS_<n>_PWRDN	IN	EPCS_<n>_IN Where n can be 0, 1, 2 or 3 depending on the number of configured lanes.
EPCS_<n>_TX_VAL	IN	
EPCS_<n>_TX_OOB	IN	
EPCS_<n>_RX_ERR	IN	
EPCS_<n>_RESET_N	IN	
EPCS_<n>_TX_DATA[<wd>:0]	IN	
EPCS_FAB_REF_CLK When Fabric is selected as the Reference Clock Source in the Configurator	IN	
EPCS_<n>_READY	OUT	EPCS_<n>_OUT Where n can be 0, 1, 2 or 3 depending on the number of configured lanes.
EPCS_<n>_TX_CLK_STABLE	OUT	
EPCS_<n>_TX_CLK	OUT	
EPCS_<n>_RX_CLK	OUT	
EPCS_<n>_RX_VAL	OUT	
EPCS_<n>_RX_IDLE	OUT	
EPCS_<n>_TX_RESET_N	OUT	
EPCS_<n>_RX_RESET_N	OUT	
EPCS_<n>_RX_DATA[19:0]	OUT	
REFCLK<x>_OUT Where x can be 0 or 1 depending on whether REFCLK0 or REFCLK1 is selected as the Reference Clock	OUT	

Note: <n>: indicates the lane on which EPCS is configured.

Note: <wd>: valid values are 19,15, 9, 7, 4, and 3.

3.9 PAD Ports

The following table lists the various PAD ports.

Table 3-9. PAD Ports

Ports	Direction	Ports Group	Description
REFCLK<x>_SE	IN	PADs_IN	Where x can be 0 or 1, depending on whether REFCLK0 (Single-ended) or REFCLK1 (Single-ended) is selected as the Reference Clock Source.
RXD0_P, RXD0_N	IN		Differential input pair for lane 0 (Rx data).
RXD1_P, RXD1_N	IN		Differential input pair for lane 1 (Rx data).
RXD2_P, RXD2_N	IN		Differential input pair for lane 2 (Rx data).
RXD3_P, RXD3_N	IN		Differential input pair for lane 3 (Rx data).
REFCLK<x>_P, REFCLK<x>_N	IN		Differential input reference clock pair. These port names can be REFCLK0 or REFCLK1, depending on your selection (see Figure 1-1).
TXD0_P, TXD0_N	OUT	PADs_OUT	Differential output pair for lane 0 (Tx data)
TXD1_P, TXD1_N	OUT		Differential output pair for lane 1 (Tx data)
TXD2_P, TXD2_N	OUT		Differential output pair for lane 2 (Tx data)
TXD3_P, TXD3_N	OUT		Differential output pair for lane 3 (Tx data)

4. Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

Revision	Date	Description
B	10/2021	Updated 2.3.1.4 Options to add a note.
A	08/2021	Initial Revision

Microchip FPGA Support

Microchip FPGA products group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, and worldwide sales offices. Customers are suggested to visit Microchip online resources prior to contacting support as it is very likely that their queries have been already answered.

Contact Technical Support Center through the website at www.microchip.com/support. Mention the FPGA Device Part number, select appropriate case category, and upload design files while creating a technical support case.

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

- From North America, call **800.262.1060**
- From the rest of the world, call **650.318.4460**
- Fax, from anywhere in the world, **650.318.8044**

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.

- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, CryptoMemory, CryptoRF, dsPIC, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, Flashtec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet- Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, TrueTime, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, GridTime, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, NVM Express, NVMe, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SmartHLS, SMART-I.S., storClad, SQI, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, TSHARC, USBCheck, VariSense, VectorBlox, VeriPHY,

ISBN: 978-1-5224-9045-6

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: www.microchip.com/support Web Address: www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4485-5910 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-72400 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820