

---

# ***SmartFusion2 and IGL002***

## ***High Speed Serial Interface Configuration***



---

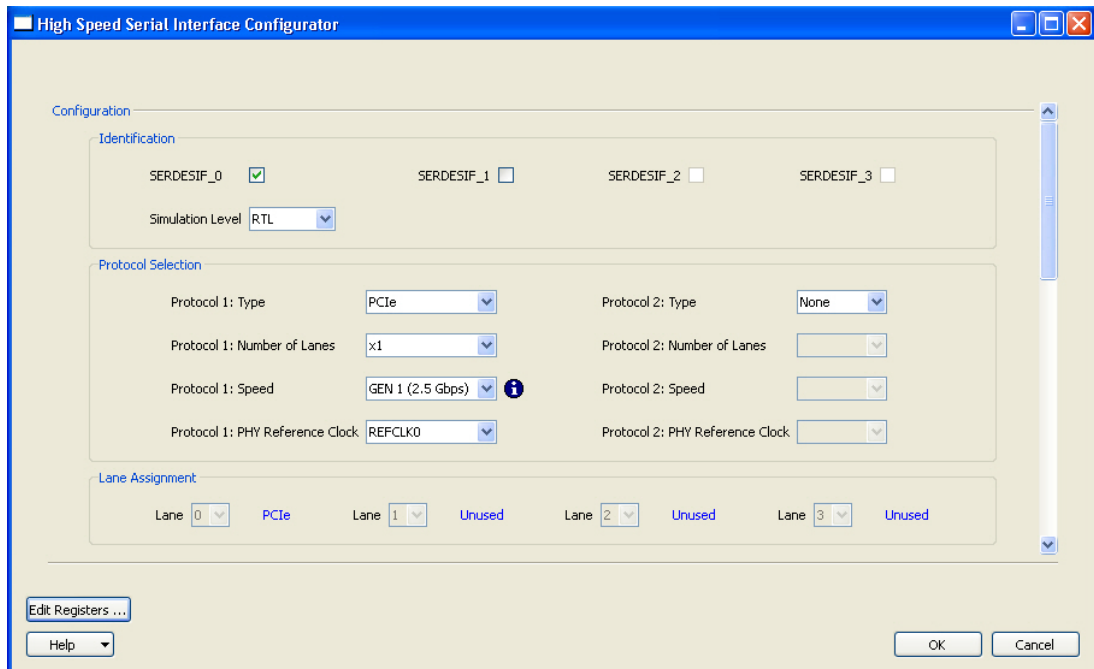
# Table of Contents

---

Introduction .....	3
<b>1 Functionality .....</b>	<b>5</b>
Identification .....	5
Protocol 1 and 2: Type, Number of Lanes, Speed .....	5
Protocol 1 and 2: PHY Reference Clock .....	6
Lane Assignment .....	6
PCIe/XAUI Fabric SPLL Configuration .....	7
PCIe Fabric Interface (AXI/AHBLite) .....	7
PCIe Base Address Registers .....	8
PCIe Identification Registers .....	8
PCIe Other Options .....	9
High Speed Serial Interface Control Registers .....	9
Firmware (SmartFusion2 Only) .....	11
Simulation Level .....	11
Simulation Files - SmartFusion2 .....	11
Simulation Files - IGLOO2 .....	12
High Speed Serial Interface Configuration Path - SmartFusion2 .....	12
High Speed Serial Interface Configuration Path - IGLOO2 Initialization .....	14
<b>2 Port Description .....</b>	<b>15</b>
<b>A Product Support .....</b>	<b>22</b>
Customer Service .....	22
Customer Technical Support Center .....	22
Technical Support .....	22
Website .....	22
Contacting the Customer Technical Support Center .....	22
ITAR Technical Support .....	23



SERDESIF\_0 and check the correct SERDES identification box to match the correct SERDES instance name.



The image shows a screenshot of the "High Speed Serial Interface Configurator" window. The window has a blue title bar and a light beige background. It is divided into three main sections: "Identification", "Protocol Selection", and "Lane Assignment".

- Identification:** Contains four checkboxes for SERDESIF\_0, SERDESIF\_1, SERDESIF\_2, and SERDESIF\_3. SERDESIF\_0 is checked. Below these is a "Simulation Level" dropdown menu set to "RTL".
- Protocol Selection:** Contains two columns of settings for Protocol 1 and Protocol 2.
  - Protocol 1: Type (PCIe), Number of Lanes (x1), Speed (GEN 1 (2.5 Gbps) with an information icon), and PHY Reference Clock (REFCLK0).
  - Protocol 2: Type (None), Number of Lanes (empty), Speed (empty), and PHY Reference Clock (empty).
- Lane Assignment:** Contains four lane settings: Lane 0 (PCIe), Lane 1 (Unused), Lane 2 (Unused), and Lane 3 (Unused).

At the bottom left, there are buttons for "Edit Registers ..." and a "Help" dropdown. At the bottom right, there are "OK" and "Cancel" buttons.

**Figure 2 • High Speed Serial Interface Configurator (Partial View)**

# 1 – Functionality

## Identification

SmartFusion2 and IGLOO2 devices may contain one or more High Speed Serial Interface blocks. The first row of checkboxes lets you identify which High Speed Serial Interface block (SERDESIF\_0, SERDESIF\_1, SERDESIF\_2, SERDESIF\_3) is being configured.

**Note:** Some devices have only one High Speed Serial Interface block, if so you must select SERDESIF\_0. Please refer to your device datasheet ([SmartFusion2 Datasheet](#); [IGLOO2 Datasheet](#)) for a list of resources available on a device.

## Protocol 1 and 2: Type, Number of Lanes, Speed

Table 1-1 shows the protocol combinations that are feasible within a single High Speed Serial Interface block.

**Table 1-1 • Available Protocols**

Protocol 1			Protocol 2		
Type	Number of Lanes	Speed	Type	Number of Lanes	Speed
PCIe, PCIe Lane Reversal Mode	x1, x2, x4	GEN 1 (2.5 Gbps), GEN 2 (5.0 Gbps)	None	-	-
XAUI	x4	3.125 Gbps	None	-	-
SGMII	x1	1.25 Gbps	None	-	-
EPCS	x1, x2, x4	1.25 Gbps 10-bit, 2.5 Gbps 20-bit, Custom	None	-	-
PCIe, PCIe Lane Reversal Mode	x1, x2	GEN 1 (2.5 Gbps), GEN 2 (5.0 Gbps)	SGMII	x1	1.25 Gbps
PCIe, PCIe Lane Reversal Mode	x1, x2	GEN 1 (2.5 Gbps), GEN 2 (5.0 Gbps)	EPCS	x1, x2	1.25 Gbps 10-bit, 2.5 Gbps 20-bit, Custom

The Protocol 2 selection can be made only if Protocol 1 is PCIe and the number of lanes for Protocol 1 is x2 or x1. Conversely, if you make any Protocol 2 selection, Protocol 1 is restricted to PCIe and the number of lanes for Protocol 1 is restricted to x2 or x1.

The PCIe Lane Reversal Mode reverses the order of lane assignments. Refer to the lane assignment section ([Table 1-2](#)) for more details.

The PCIe GEN 2 (5.0 Gbps) speed is not available in STD speed-grade devices. Please refer to your device datasheet ([SmartFusion2 Datasheet](#); [IGLOO2 Datasheet](#)) for more information.

The EPCS Custom mode can be further customized for other speeds and widths through the initialization of the related registers.

## Protocol 1 and 2: PHY Reference Clock

The PHY Reference clock supply to each protocol has two sources: REFCLK0 and REFCLK1, which are differential inputs. The exception is EPCS, which can receive its clock from a Fabric signal as well.

## Lane Assignment

The High Speed Serial Interface block has differential inputs (RXD) and outputs (TXD) for each lane responsible for carrying the high-speed serial data. The Lane assignment fields of the configurator shows the protocol selected for each of the four lanes in the High Speed Serial Interface block.

Table 1-2 shows the lane assignment for various possible configurations using the High Speed Serial Interface configurator.

**Table 1-2 • Lane Assignments**

Protocol 1		Protocol 2		PHYSICAL LANES to LOGICAL LANES Mapping			
Type	Number of Lanes	Type	Number of Lanes	LANE-0	LANE-1	LANE-2	LANE-3
PCIe	x1	None	-	PCIe-0	*	*	*
	x2			PCIe-0	PCIe-1	*	*
	x4			PCIe-0	PCIe-1	PCIe-2	PCIe-3
PCIe Lane Reversal Mode	x1			*	*	*	PCIe-0
	x2			*	*	PCIe-1	PCIe-0
	x4			PCIe-3	PCIe-2	PCIe-1	PCIe-0
XAUI	x4			XAUI-0	XAUI-1	XAUI-2	XAUI-3
SGMII	x1			*	*	*	SGMII-0
EPCS	x1			EPCS-0	*	*	*
	x2			EPCS-0	EPCS-1	*	*
	x4			EPCS-0	EPCS-1	EPCS-2	EPCS-3
PCIe	x1	SGMII	x1	PCIe-0	*	*	SGMII-0
	x2			PCIe-0	PCIe-1	*	SGMII-0
PCIe Lane Reversal Mode	x1			*	PCIe-0	*	SGMII-0
	x2			PCIe-1	PCIe-0	*	SGMII-0

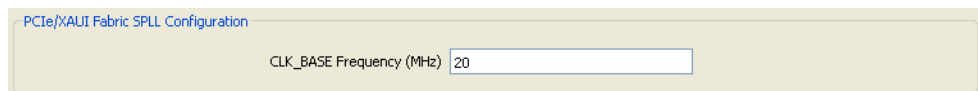
**Table 1-2 • Lane Assignments (continued)**

Protocol 1		Protocol 2		PHYSICAL LANES to LOGICAL LANES Mapping			
Type	Number of Lanes	Type	Number of Lanes	LANE-0	LANE-1	LANE-2	LANE-3
PCIe	x1	EPCS	x1	PCIe-0	*	*	EPCS-0
	x2			PCIe-0	PCIe-1	*	EPCS-0
	x1		x2	PCIe-0	*	EPCS-0	EPCS-1
	x2			PCIe-0	PCIe-1	EPCS-0	EPCS-1
PCIe Lane Reversal Mode	x1		x1	*	PCIe-0	*	EPCS-0
	x2			PCIe-1	PCIe-0	*	EPCS-0
	x1		x2	*	PCIe-0	EPCS-0	EPCS-1
	x2			PCIe-1	PCIe-0	EPCS-0	EPCS-1

NOTE: \* indicates that the lane is unused.

## PCIe/XAUI Fabric SPLL Configuration

The SPLL configuration fields are relevant only for PCIe and XAUI protocols ([Figure 1-1](#)).

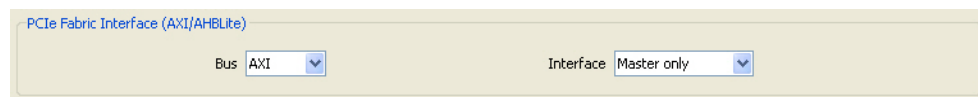


**Figure 1-1 • PCIe/XAUI Fabric SPLL Configuration**

**CLK\_BASE Frequency (MHz)** - The valid range for PCIe protocol is 20 to 200 MHz. Enter a value that matches the frequency of CLK\_BASE. For XAUI, it is fixed to 156.25 MHz.

## PCIe Fabric Interface (AXI/AHBLite)

PCIe fabric interface can be through an AXI or AHBLite bus as Master only, Slave only or both Master and Slave ([Figure 1-2](#)).



**Figure 1-2 • PCIe Fabric Interface (AXI/AHBLite)**

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.

## PCIe Base Address Registers

For PCIe, the individual fields of the 6 Base Address Registers can be configured as follows (Figure 1-3).



Register	Width	Size	Prefetchable
BAR 0	32 bit	4 KB	<input type="checkbox"/>
BAR 1	None		<input type="checkbox"/>
BAR 2	None		<input type="checkbox"/>
BAR 3	None		<input type="checkbox"/>
BAR 4	None		<input type="checkbox"/>
BAR 5	None		<input type="checkbox"/>

**Figure 1-3 • PCIe Base Address Registers**

**Width** - The width on even registers can be 32 bit or 64 bit. If an even register is set to 64 bits wide then the subsequent (odd) register serves as the upper half of 64 bits. Otherwise, the width of odd registers is restricted to 32 bits.

**Size** - Ranges from 4 KB to 2 GB. Some devices may go up to 1 GB. Please refer to your device datasheet ([SmartFusion2 Datasheet](#); [IGLOO2 Datasheet](#)) for more information.

**Prefetchable** - Enabled only on even registers with 64 bit width.

## PCIe Identification Registers

There are six identification registers for PCIe that can be assigned 16 bit hexadecimal signatures (Figure 1-4).



Field	Value
Vendor ID	0x11AA
Subsystem Vendor ID	0x0000
Revision ID	0x0000
Device ID	0x0000
Subsystem Device ID	0x0000
Class Code	0x0000

**Figure 1-4 • PCIe Identification Registers**

**Vendor ID** - 0x11AA is the Vendor ID assigned to Microsemi by PCI-SIG. Please contact Microsemi if you would like to allocate Subsystems under the Microsemi vendor ID.

**Subsystem vendor ID** - Card manufacturers ID.

**Device ID** - Manufacturers assigned part number assigned by the vendor.

**Revision ID** - Revision number, if available.

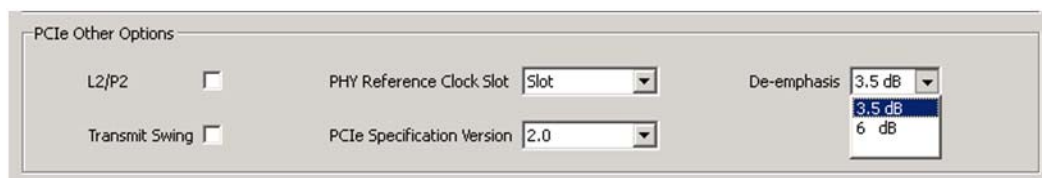
**Subsystem Device ID** - Assigned by the subsystem vendor.

**Class Code** - PCIe device's generic function.



## PCIe Other Options

PCIe Other options enables you to configure the following options (Figure 1-5).



**Figure 1-5 • PCIe Power Management and Other Options**

**L2/P2** - Selecting this option adds PCIE\_WAKE\_N, PCIE\_WAKE\_REQ and PCIE\_PERST\_N ports.

**PHY Reference Clock Slot** - Toggles between Slot and Independent.

**De-emphasis** - This option is enabled only for PCIe GEN 2 speed. You can then toggle between 3.5 dB and 6.0 dB.

**Transmit Swing** - Enabled only for PCIe GEN 2 speed.

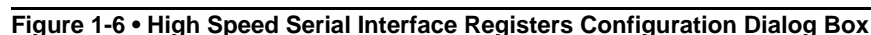
**PCIe Specification Version** - Sets the Specification version.

## High Speed Serial Interface Control Registers

The High Speed Serial Interface has a set of registers that can be configured at runtime. The configuration values for these registers represent different parameters, e.g. AXI BAR Window. For complete details about these registers, please refer to the Microsemi [SmartFusion2 or IGLOO2 High Speed Serial Interfaces User's Guide](#).

### High Speed Serial Interface Registers Configuration

To enter the High Speed Serial Interface configuration values, specify the register values when you are configuring the High Speed Serial Interface. Click **Edit Registers** in the High Speed Serial Interface Configurator (Figure 2 on page 4) to open the Registers Configuration dialog (Figure 1-6). Data entered



The Registers Configuration dialog enables you to enter High Speed Serial Interface register values using a graphical interface. The dialog 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 below; Microsemi 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:

**10**

- **Reset Configuration** - Click Reset Configuration to undo any changes you have made to the register configuration. Note that this deletes all register configuration data and you must either re-import or reenter this data. The data is 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.

## 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

## Simulation Level

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

**BFM\_CFG** - This level provides a Bus Functional Model of only the APB configuration bus of the High Speed Serial Interface block. You will be able to 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 will not change based on the APB state; they are kept at their reset values. This simulation level is available for all protocols.

**BFM\_PCl\_e** - 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 all the way from the fabric interface to the serial data interface at the expense of a longer simulation runtime. This simulation level is available for all protocols.

## 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 essentially 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 the user 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 above) 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 will be "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 will be "executed" after peripheral\_init.bfm has completed.

Using the files above, 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.

## Simulation Files - IGLOO2

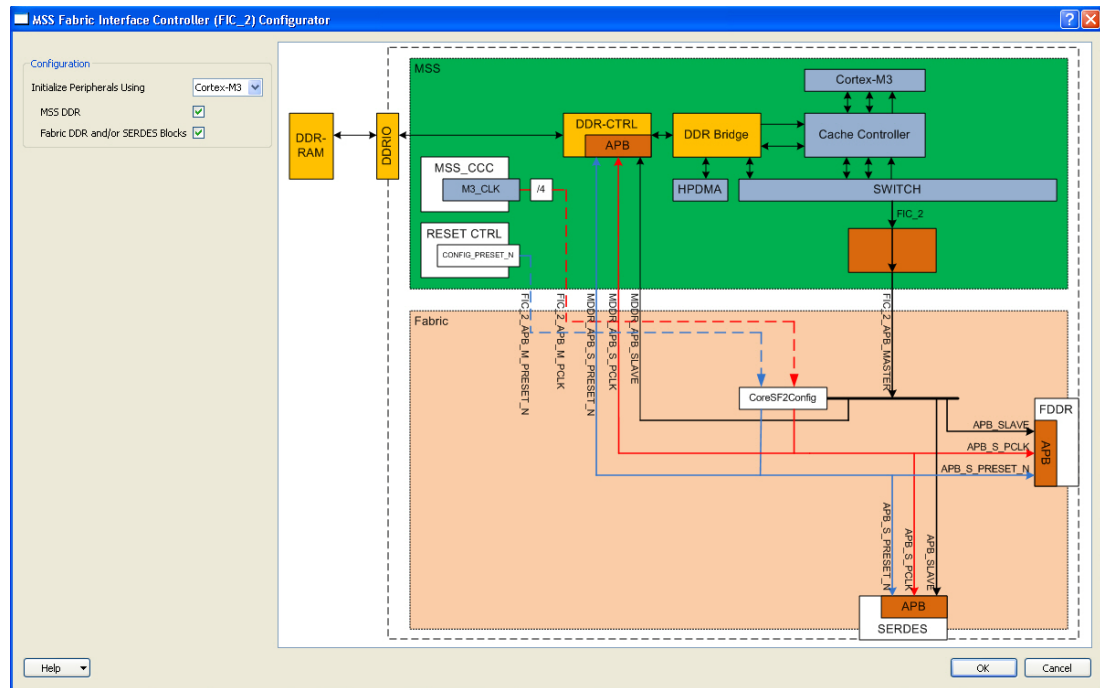
When you generate the SmartDesign associated with your HPMS, the following simulation files are generated in the <project dir>/simulation directory:

- **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 above) 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 will be "executed" after peripheral\_init.bfm has completed.

## High Speed Serial Interface Configuration Path - SmartFusion2

The configuration register data is 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.

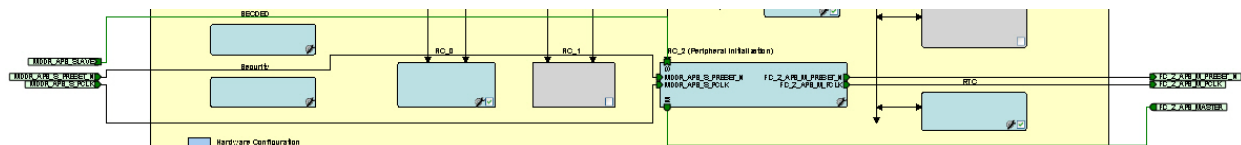


**Figure 1-7 • FIC\_2 Configurator Overview**

To configure the FIC\_2 interface:

1. Open the FIC\_2 configurator dialog (Figure 1-7) from the MSS configurator.
2. Select **Initialize peripherals using Cortex-M3**. Make sure that you have clicked the checkbox to enable **Fabric DDR and/or SERDES blocks** and the MSS DDR option (if you are using it).
3. Click **OK** to save your settings. This exposes the FIC\_2 configuration ports (Clock, Reset, and APB bus interfaces), as shown in Figure 1-8.
4. Generate the 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 complete details on configuring and connecting the CoreSF2Config and CoreSF2Reset cores, refer to the [SmartFusion2 DDR Controller and Serial High Speed Controller Initialization Methodology](#) document.



**Figure 1-8 • FIC\_2 Ports**

## High Speed Serial Interface Configuration Path - IGLOO2 Initialization

Once you make the proper connection for the Configuration path of the SERDES, the SERDES initialization is executed at runtime automatically.

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 APB\_S\_PCLK port of the System Builder block.
- Connect the APB\_S\_PRESET\_N port of the SERDES block to the APB\_S\_PRESET\_N port of the System Builder block.

---

## 2 – Port Description

---

**Table 2-1 • APB Ports**

Port	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	

**Table 2-2 • PCIe Ports**

Port	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
PLL_LOCKLOST_INT	OUT
PCIE_EV_1US	OUT
PCIE_WAKE_N	OUT
PCIE_WAKE_REQ	IN
PCIE_PERST_N	IN

**Table 2-3 • PCIe AXI Master Ports**

Port	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_AWBURST[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	



**Table 2-4 • PCIe AXI Slave Ports**

Port	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	

**Table 2-5 • PCIe AHBLite Master Ports**

Port	Direction	Ports 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	

**Table 2-6 • PCIe AHBLite Slave Ports**

Port	Direction	Ports 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	

**Table 2-7 • XAUI Ports**

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

**Table 2-8 • EPCS Ports per Lane**

Port	Direction	Ports Group
EPCS_<n>_PWRDN	IN	EPCS_<n>_IN
EPCS_<n>_TX_VAL	IN	
EPCS_<n>_TX_OOB	IN	
EPCS_<n>_RX_ERR	IN	
EPCS_<n>_RESET_N	IN	
EPCS_<n>_TX_DATA[19:0]	IN	
EPCS_<n>_READY	OUT	EPCS_<n>_OUT
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	

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

**Table 2-9 • SGMII Ports**

Port	Direction	Ports Group
SGMII_PWRDN	IN	SGMII_IN
SGMII_TX_VAL	IN	
SGMII_TX_OOB	IN	
SGMII_RX_ERR	IN	
SGMII_RESET_N	IN	
SGMII_TX_DATA[9:0]	IN	

**Table 2-9 • SGMII Ports (continued)**

Port	Direction	Ports Group
SGMII_READY	OUT	SGMII_OUT
SGMII_TX_CLK_STABLE	OUT	
SGMII_TX_CLK	OUT	
SGMII_RX_CLK	OUT	
SGMII_RX_VAL	OUT	
SGMII_RX_IDLE	OUT	
SGMII_TX_RESET_N	OUT	
SGMII_RX_RESET_N	OUT	
SGMII_RX_DATA[9:0]	OUT	

**Table 2-10 • PAD Ports**

Ports	Direction	Ports Group	Description
RXD0_P, RXD0_N	IN	PADs_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. This port names can be REFCLK0 or REFCLK1 depending on the user selection (refer to <a href="#">Figure 1 on page 3</a> )
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)

---

## A – Product Support

---

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

### Customer Service

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, 408.643.6913

### Customer Technical Support Center

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues, and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

### Technical Support

Visit the Customer Support website ([www.microsemi.com/soc/support/search/default.aspx](http://www.microsemi.com/soc/support/search/default.aspx)) for more information and support. Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on the website.

### Website

You can browse a variety of technical and non-technical information on the SoC home page, at [www.microsemi.com/soc](http://www.microsemi.com/soc).

### Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

#### Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is [soc\\_tech@microsemi.com](mailto:soc_tech@microsemi.com).

## My Cases

Microsemi SoC Products Group customers may submit and track technical cases online by going to [My Cases](#).

## Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support via email ([soc\\_tech@microsemi.com](mailto:soc_tech@microsemi.com)) or contact a local sales office. [Sales office listings](#) can be found at [www.microsemi.com/soc/company/contact/default.aspx](http://www.microsemi.com/soc/company/contact/default.aspx).

## ITAR Technical Support

For technical support on RH and RT FPGAs that are regulated by International Traffic in Arms Regulations (ITAR), contact us via [soc\\_tech\\_itar@microsemi.com](mailto:soc_tech_itar@microsemi.com). Alternatively, within [My Cases](#), select **Yes** in the ITAR drop-down list. For a complete list of ITAR-regulated Microsemi FPGAs, visit the [ITAR](#) web page.



**Microsemi**

**Microsemi Corporate  
Headquarters**

One Enterprise, Aliso Viejo CA 92656 USA  
Within the USA: +1 (949) 380-6100

Microsemi Corporation (NASDAQ: MSCC) offers a comprehensive portfolio of semiconductor solutions for: aerospace, defense and security; enterprise and communications; and industrial and alternative energy markets. Products include high-performance, high-reliability analog and RF devices, mixed signal and RF integrated circuits, customizable SoCs, FPGAs, and complete subsystems. Microsemi is headquartered in Aliso Viejo, Calif. Learn more at [www.microsemi.com](http://www.microsemi.com).

© 2012 Microsemi Corporation. All rights reserved. Microsemi and the Microsemi logo are trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.