# SmartFusion2 FPGA

## High Speed Serial Interface Simulation User's Guide

Microsemi

# Table of Contents

# Introduction

The High Speed Serial Interface (SERDES) in SmartFusion2 supports multiple high-speed serial protocols. The SERDESIF macro includes a PMA block, which is a serializer and de-serializer (SERDESIF) analog block that is capable of supporting multiple serial protocols on its physical lanes.

The data path from PMA varies with the protocol used. For the PCIe protocol, the data path from PMA includes the PCIE PCS, which is completely bypassed for all non-PCIe protocols. For the XAUI protocol, the data path includes an XAUI extender. For other serial protocols, the data path includes EPCS.

Protocols supported in SmartFusion2 are: PCIe 1.0, 1.1 and 2.0, XAUI and any user-defined high-serial protocol implemented in SmartFusion2 fabric accessing SERDESIF lanes through the EPCS interface.

This document describes the different simulation flows currently supported in the software for the SERDESIF block.

The SERDESIF block communicates with the fabric as well as the IO-PADs. The SERDESIF has the following interfaces towards the fabric:

- AXI/AHB Master Interface: a 64/32-bit AXI master interface for data communication with the fabric in PCIe mode.
- AXI/AHB Slave Interface: a 64/32-bit AXI slave interface for data communication with the fabric in PCIe mode.
- APB Slave Interface: 32-bit APB slave interface for configuration.
- External PCS Interface: 20-bit EPCS Interfaces available for 1, 2, 3 or 4 lane configurations at custom speed. Each 20-bit EPCS lane interface is independent.

# 1 – High Speed Serial Interface Simulation Support

The High Speed Serial Interface (SERDESIF) block is capable of operating in the following modes:

- PCIe
- XAUI
- EPCS

You can use the High Speed Serial Interface configurator to select the operating mode and the number of lanes to be used for each protocol. For details, refer to the High Speed Serial Interface Configurator User Guide.

In the example below (Figure 1-1), the MSS (APB Master) is configuring the SERDES block (APB Slave) using the APB bus (CoreAPB3_0) and the SERDES block (AHB Master) is writing into an AHBL SRAM (AHB Slave) using the AHBLite bus (CoreAHBLite_0).

Note: The CoreConfigP and CoreResetP blocks are for configuration and initialization of the SERDES blocks.

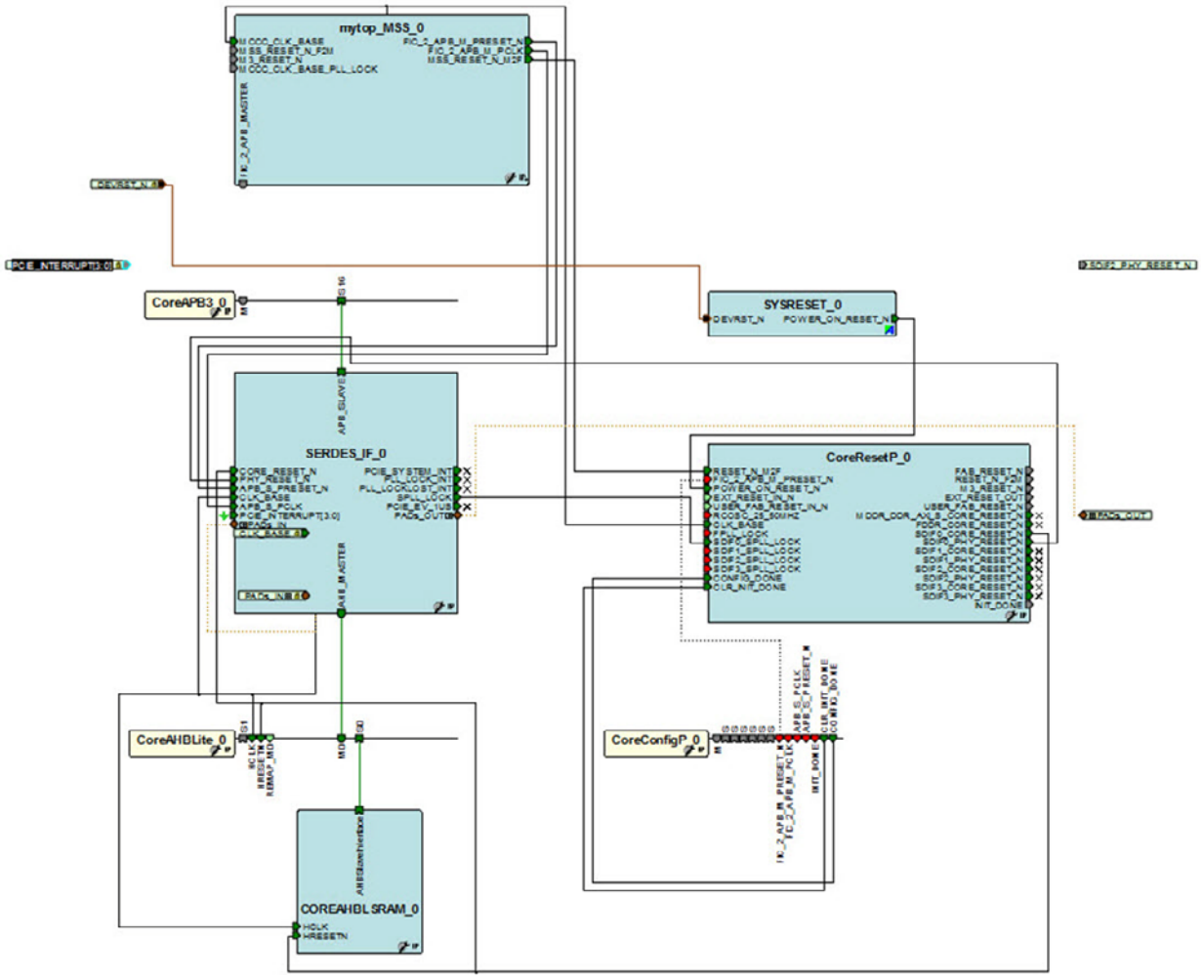


**Figure 1-1 • SERDES Block with APB Configuration Path Accessing an AHBLite Fabric Slave**

You can also select the simulation mode, depending on the SERDESIF mode of operation (as shown in Table 1-1). Refer to the SmartFusion2 and IGLOO2 High Speed Serial Interface Configuration User Guide for details on supported Protocols and their configurations.

**Table 1-1 • SERDESIF Modes and Corresponding Simulation Modes**

| SERDESIF Mode | Available Simulation Mode |
|---|---|
| PCIe | BFM_CFG, BFM_PCIe, RTL |
| XAUI, EPCS | BFM_CFG, RTL |

- **BFM_CFG** - In this mode, only the APB configuration bus of the SERDESIF is available for simulation. You can write/read the different configuration and status bits of the SERDESIF registers through its APB slave interface. Refer to "BFM_CFG Simulation Mode" on page 6.

  Note: In this simulation mode, the values of the status bits do not change in response to SERDESIF transactions - they are maintained at their reset values, while you are allowed to write

to the configuration bits. This simulation option is available for all the SERDESIF modes of operation.

Note: In this simulation mode, the physical interface of the SERDES IP block is inactive.

- **BFM_PCIe** - In this mode, you can access the APB bus similar to the BFM_CFG mode. In addition, you can also communicate with the SERDESIF block through the master and slave AXI or AHB bus interfaces. This mode is intended to allow you to validate your fabric interfaces to the SERDESIF block and is only available for the PCIe mode of operation.

  Note: In this simulation mode, the physical interface of the SERDES IP block is inactive.
- **RTL** - This mode enables you to fully simulate the SERDESIF block from the fabric interface to the serial I/O interface. This simulation mode is available for all the SERDESIF modes of operation. The simulation runtime for this mode is longer than the runtime for the other SERDESIF simulation modes.

## BFM_CFG Simulation Mode

This simulation mode enables you to interact with the APB slave interface of the SERDESIF block. The SERDESIF has three addressable spaces from the APB bus, namely: SERDESIF System Registers,

PCIe Bridge/Core Registers and SERDES Macro Registers. Configuration of the SERDESIF is done through these registers (see Figure 1-2 and Figure 1-3).

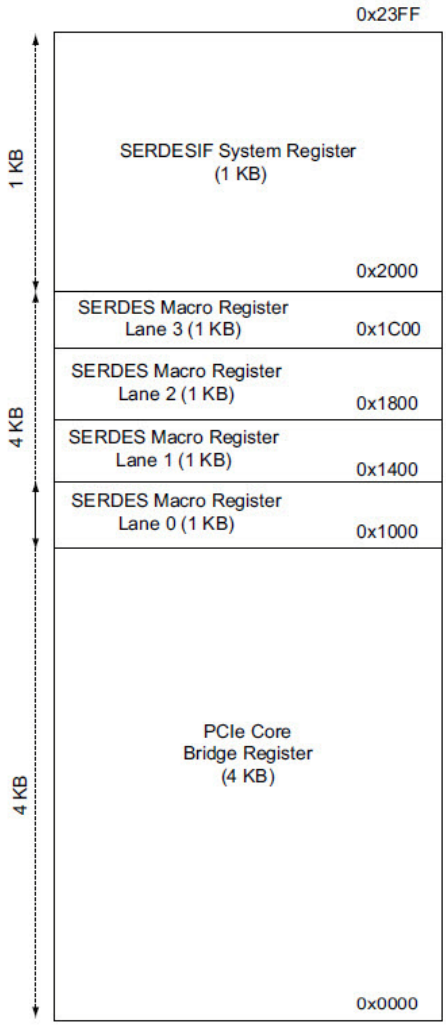


**Figure 1-2 • Memory Map of PCIe_0 (Single PCIe Mode)**

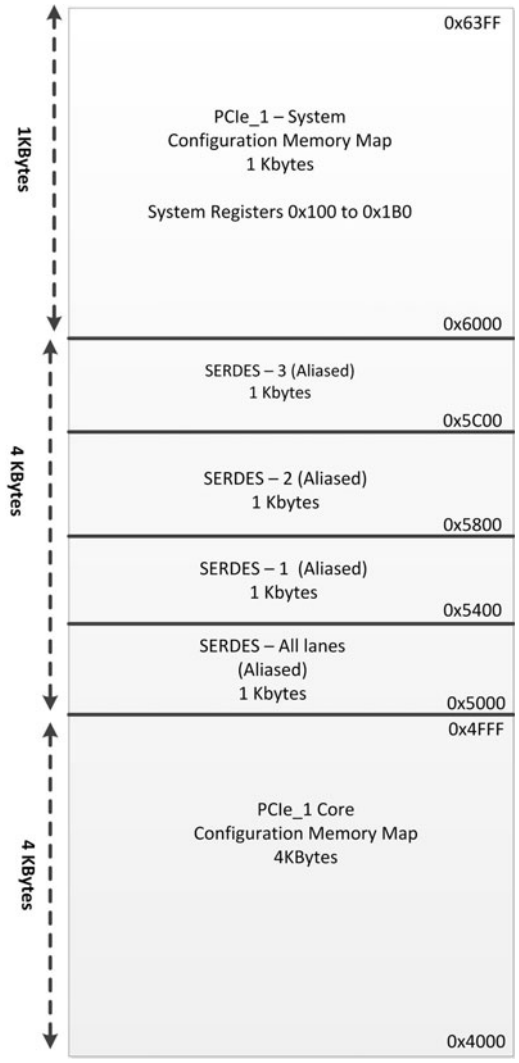


**Figure 1-3 • Memory Map of PCIe_1 (second PCIe controller in Dual PCIe mode) of SERDESIF2 and SERDESIF3 Core (Available on M2S090T/TS devices only)**

In BFM_CFG Simulation Mode, only one of these address spaces can be accessed. The accessible address space for PCIe_0 is from address offset 0x2000 to 0x23FF of the SERDESIF System Register memory map. However, address space 0x2100 through 0x3FFF is reserved for future use. Do not read and write into this reserved area as it may result in PSLVERR responses. For PCIe_1 (second SERDES controller in Dual PCIe Mode for M2S090T/TS devices), the accessible address space is 0x6000 through 0x63FF. However, address space 0x6100 through 0x7FFF is reserved for future use. Do not read and write into this reserved area as it may result in PSLVERR error.

You can write into all the write-allowed configuration registers of the SERDESIF blocks to configure their operation based on your needs using the APB interface. As a result, you must have an APB bus master to send the commands to the SERDESIF block, e.g., the MSS configuration APB FIC2 interface, or another APB master in the fabric.

From the simulation log, you can read back the configurations written to verify that the writes went through correctly. You can also read back the contents of the status registers that are accessible from the APB interface. However, the value of the status bits does not reflect the true state of the SERDESIF block. Only the reset values of these registers are read back.

In this mode, the application layer of the SERDESIF block is not used. The application layer is only used for configuring/setting up the SERDESIF, SERDES and PCIe APB register. As a result, the fabric datapath interface of the SERDESIF does not respond to external stimulations. You can use this simulation mode with any operational mode of the SERDESIF block.

## BFM_PCIe Simulation Mode

This simulation mode is only available for PCIe mode. In BFM_PCIe simulation mode (Figure 1-4), you can transmit/receive data from/to the fabric using the AXI/AHBLite interface of the SERDESIF. Using an APB Master, you can also read and write to the APB register interface of the SERDESIF to access the configuration registers. The PCIe BFM Model in the diagram is an AHBLite or AXI Master that accesses the AMBA High-Performance (AHB)-Lite Slaves over the AHBLite bus. The Memory Model is an AHBLite Slave that provides a simple read and write memory. The SERDESIF_n.init.bfm contains BFM write commands to initialize the SERDES prior to simulation.

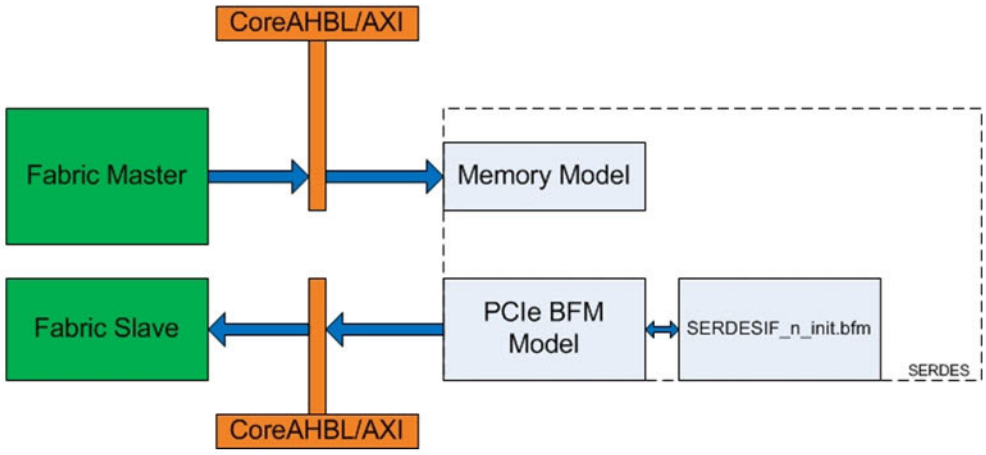


**Figure 1-4 • BFM_PCIe Simulation Mode Diagram**

Figure 1-5 shows the simulation mode diagram for SmartFusion2 M2S090T/TS devices that support PCIe for Protocol 1 and Protocol 2.

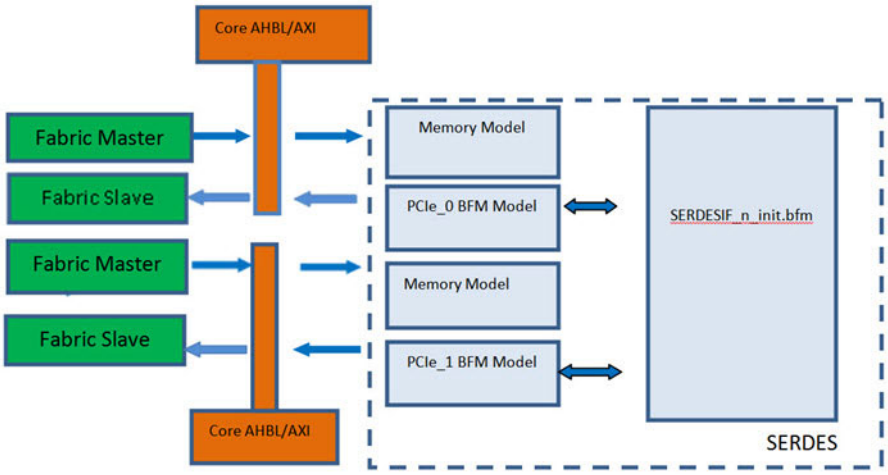


**Figure 1-5 • BFM_PCIe Simulation Mode Diagram for Dual PCIe Mode (M2S/M2GL090T/TS)**

The BFM_PCIe simulation mode uses BFM commands to emulate data being transferred through the SERDESIF block across the AXI/AHB bus interface to the fabric. The physical layer of the PCIe protocol is not implemented in this simulation mode and you cannot witness data transfer on the serial interface of the SERDESIF block.

The BFM_PCIe simulation mode provides you with an AXI/AHB bus master and an AXI/AHB bus slave, based on the SERDESIF PCIe bus configuration.

# SERDES AXI/AHB Bus Slave Interface

The AXI (Figure 1-6) and AHB bus slave (Figure 1-7) Interfaces available in the BFM_PCIe simulation mode provide you with a 64-bit or 32-bit slave interface for fabric communication. The slave acts as a memory that you can interact with by initiating write and read bus transactions using the appropriate bus master. Since the slave acts as a memory model, whatever is written into the slave can be read back from the same address.

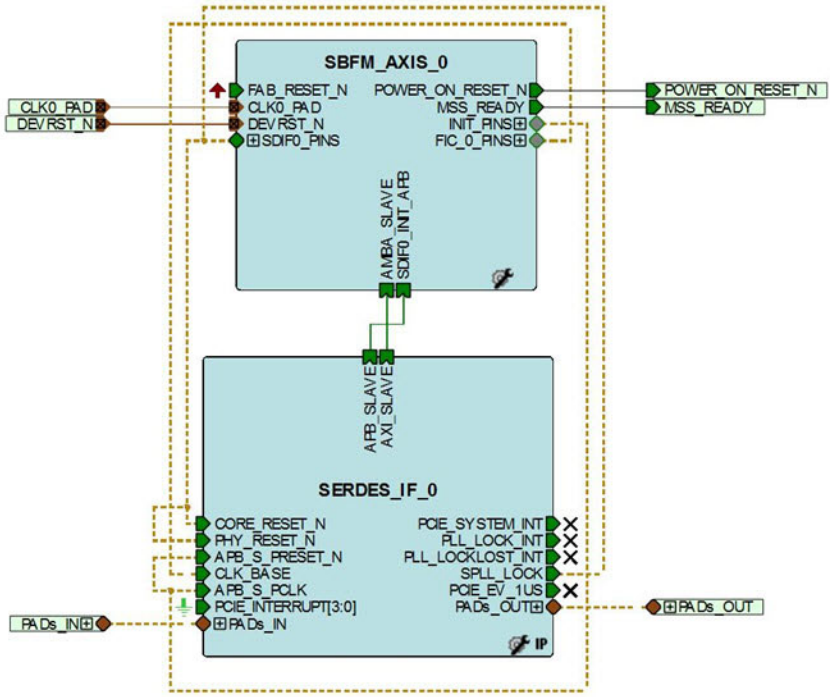


**Figure 1-6 • AXI Slave Simulation Mode**

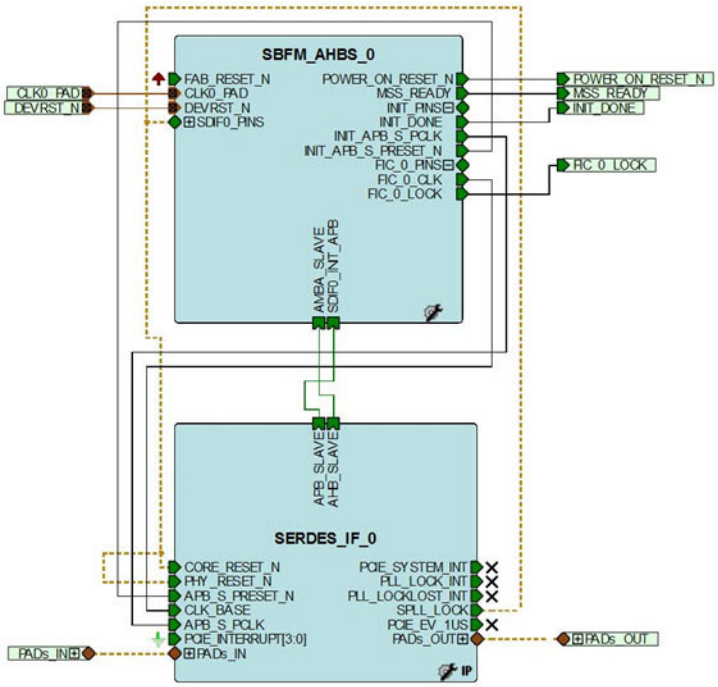


**Figure 1-7 • AHB Slave Simulation Mode**

# SERDES AXI/AHB Bus Master Interface

The AXI (Figure 1-8) and AHB bus master (Figure 1-9) Interfaces in the BFM_PCIe simulation mode enables you to emulate a 64-bit AXI or a 32-bit AHB master, depending on the bus configuration. In the bus master cases, you must write BFM bus commands to instruct the model to start bus transactions to the fabric, similar to the MSS. Refer to the SmartFusion2 FPGA Microcontroller Subsystem BFM Simulation Guide for more information.

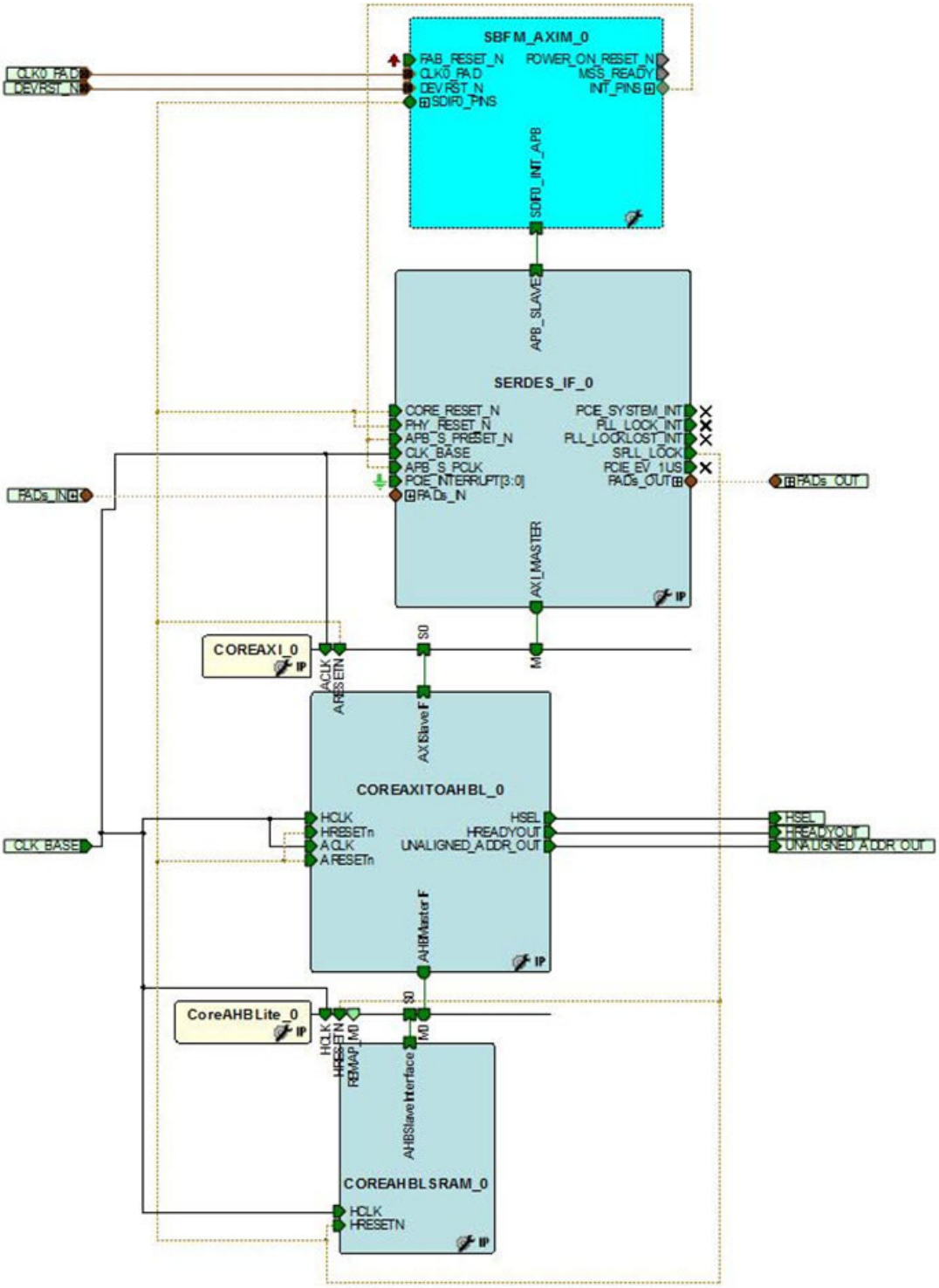


**Figure 1-8 • AXI Master Simulation Mode**

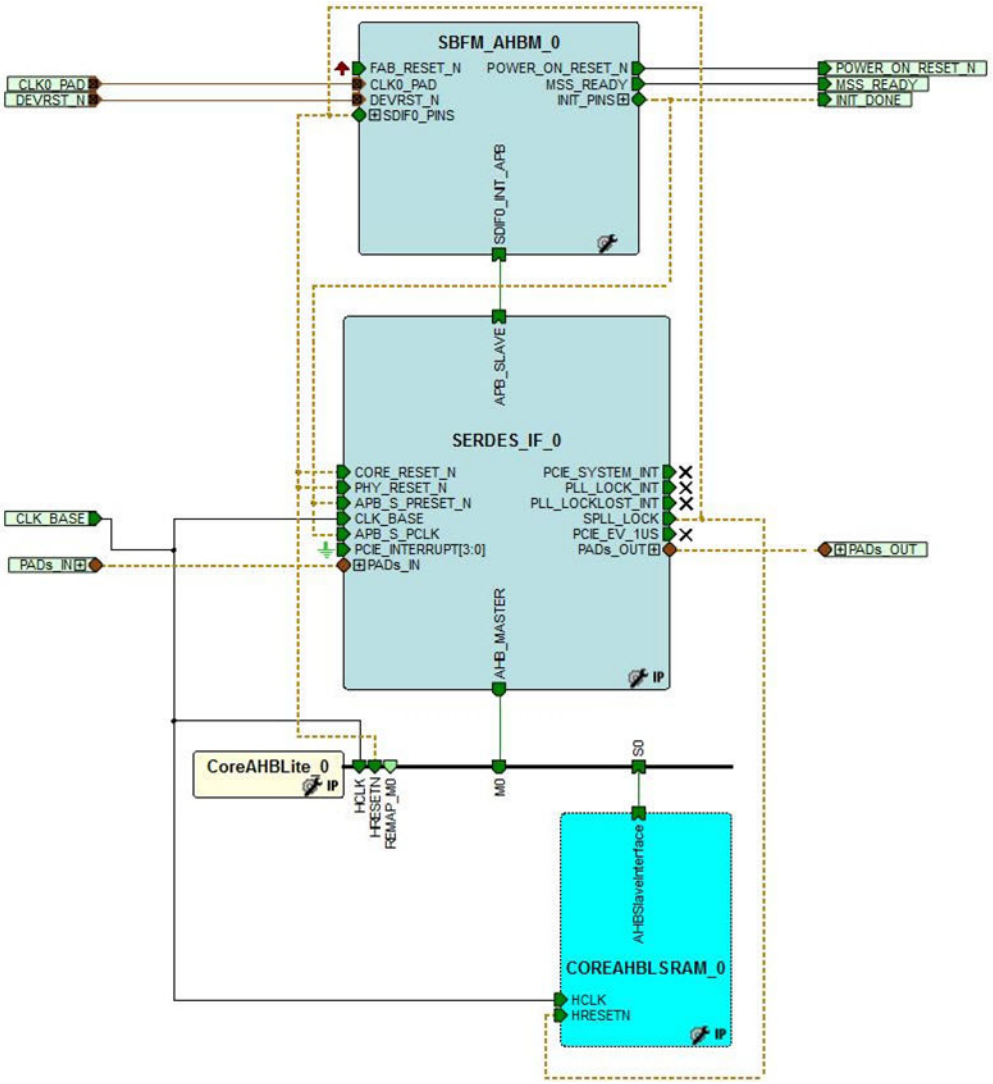


**Figure 1-9 • AHB Master Simulation Mode**

You must include your BFM instructions in the <project>/simulation/SERDESIF_<n>_user.bfm file, where n refers to the SERDESIF instance you are trying to simulate (e.g. SERDES_IF_0/1/2/3). The built-in SERDES BFM model interprets these instructions and initiates AHB/AXI transactions in sequence.

You can check your BFM commands before invoking the simulator. Libero performs a syntax check on your BFM commands and displays the results in the log tab. To check your BFM commands:

1. Open the BFM file SERDESIF_<n>_user.bfm in the Libero Text Editor.
2. Right-click and select Check BFM File (or click the Check File icon) (Figure 1-10).

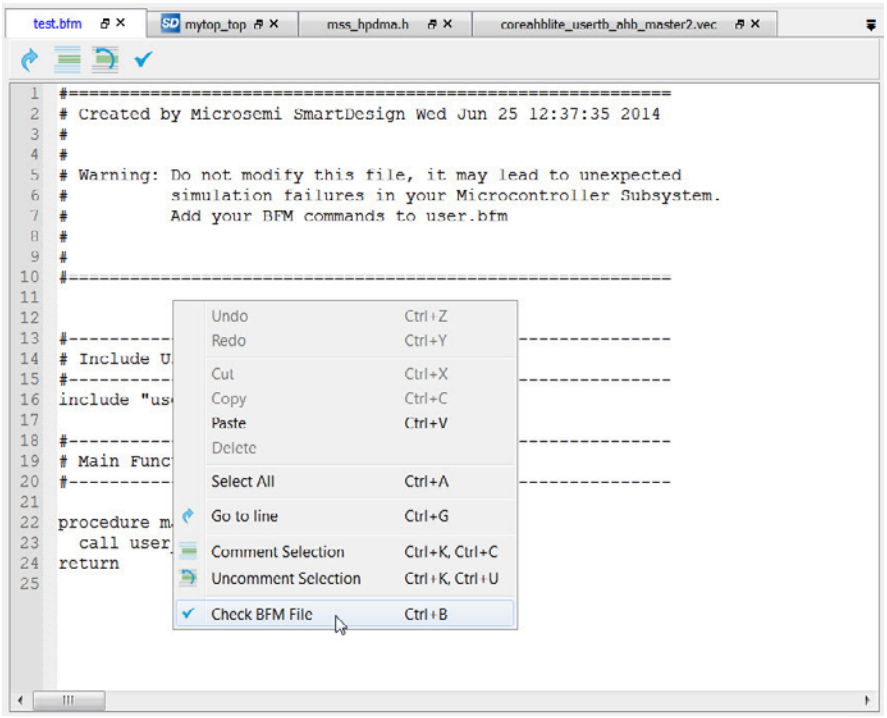


**Figure 1-10 • Check BFM File**

3. Select the Log tab to view the result.

The BFM commands used in the SERDES BFM files are similar to the BFM commands used by the MSS bus masters. Refer to Chapter 6 in the CoreAMBA BFM User's Guide for a complete list of available BFM commands. There are additional BFM commands that you can use only for PCIe AXI BFM simulation to emulate 64-bit AXI transactions:

- `write64 w <base_address> <base_address_offset> <32-bit MSB> <32-bit LSB>`

  This command makes the SERDES bus master start a 64-bit transaction on the external bus for a slave which address given by the <base_address> and <base_address_offset> using the data given by <32-bit MSB> and <32-bit LSB>. For example:

  ```
  write64 w 0x00000000 0x0 0xA0A1A2A3 0xB0B1B2B3;
  ```

- `readcheck64 w <base_address> <base_address_offset> <32-bit MSB> <32-bit LSB>`

  Makes the SERDES bus master start a 64-bit read transaction for the address given by <base_address> and <base_address_offset>. It will compare the 64-bit read data to the data given by <32-bit MSB> and <32-bit LSB>. If the data matches, the command will succeed, otherwise, it will error-out. For example:

  ```
  readcheck64 w 0x00000000 0x0 0xA0A1A2A3 0xB0B1B2B3;
  ```

- `read64 w <base_address> <base_address_offset>`

  This command will make the SERDES bus master start a 64-bit read transaction for the address given by <base_address> and <base_address_offset>. For example:

  ```
  read64 w 0x00000000 0x0 0xA0A1A2A3 0xB0B1B2B3;
  ```

- `writemult64 w <base_address> <base_address_offset> <32-bit MSB> <32-bit LSB>…`

  This command makes the SERDES bus master start a 64-bit transaction in burst mode on the external bus for a slave with address given by the <base_address> and <base_address_offset>

using the data given by <32-bit MSB> and <32-bit LSB>. This command performs as many writes as <32-bit MSB> and <32-bit LSB> pairs are specified by the user. Write operations are done to consecutive address locations starting from the base offset specified. For example:

```
writemult64 w 0x00000000 0x0 0xA1 0xB1 0xA2 0xB2 0xA3 0xB3 0xA4 0xB4 ;
```

- `readmult64  w <base_address> <base_address_offset> <n>`

  This command makes the SERDES bus master start a 64-bit burst read transaction for consecutive address locations starting from <base_address> and <base_address_offset> for <n> number of times. For example:

  ```
  readmult64  w 0x00000000 0x0 5;
  ```

- `readmultchk64 w <base_address> <base_address_offset> <32-bit MSB> <32-bit LSB>`

  This command makes the SERDES bus master start a 64-bit burst read transaction for the address given by <base_address> and <base_address_offset>. It will compare the 64-bit read data to the data given by <32-bit MSB> and <32-bit LSB>. Number of reads and corresponding compares depend on the number of <32-bit MSB> and <32-bit LSB> pairs specified in this command. If the data matches, the command will succeed;, otherwise, it errors out. For example:

  ```
  readmultchk64 w 0x00000000 0x0 0xA1 0xB1 0xA2 0xB2 0xA3 0xB3 0xA4 0xB4;
  ```

# RTL Simulation Mode

RTL Simulation mode is available for all the SERDESIF modes. This simulation mode enables you to simulate PCIe, XAUI, and EPCS. The RTL simulation mode supports all of the protocol communication layers, including the physical layer, and provides accurate cycle simulation for your design. However, using RTL simulation incurs some run-time penalties.

You are responsible for having your own model for an off-chip IP that can communicate with the SERDESIF block in the same protocol used by the SERDESIF block when using this mode. For example, if you are using a PCIe configured SERDESIF block, you must have your own PCIe IP off-chip block that can communicate with the SERDESIF block using the PCIe protocol (as shown in Figure 1-11). The same is true for all other SERDESIF block modes of operation. Since the IP user block is off-chip it must be connected to your design in the top level testbench.

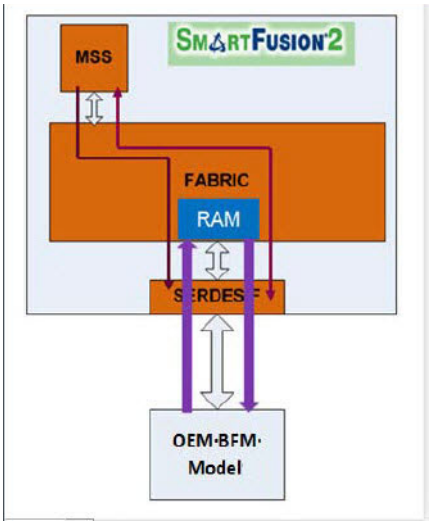


**Figure 1-11 • RTL Simulation Model**

# BFM Files for SERDES Simulation

Libero generates a list of BFM files in the simulation folder for SERDES simulation (Table 1-2). Some BFM files are not to be edited by the user. Editing them may lead to unexpected simulation failures.

**Table 1-2 • Generated BFM Files**

| BFM File Name | Function | Remarks |
|---|---|---|
| Test.bfm | Top level BFM<br>Contains the main function | Libero-generated.<br>Do not edit. |
| User.bfm | Contains user BFM commands<br>Calls subsystem_init function | Edit this file to add user BFM commands. |
| Subsystem.bfm | Contains subsystem memory map<br>Define name and base address of each subsystem resource<br>Call the init function to initialize subsystem | Do not edit. |
| Peripherals_init.bfm | Contains the Memory Map of all peripherals including SERDES<br>Calls the SERDES_<0/1/2/3>_init.bfm to initialize SERDES | Do not edit. |
| SERDES_<0/1/2/3>_init.bfm | Writes to SERDES registers to initialize SERDES | Do not edit. |
| SERDESIF_<0/1/2/3>_PCIE_<0/1>_user.bfm (Dual PCIe Mode Operation)<br>SERDESIF_<0/1/2/3>_user.bfm (Single PCIe Mode Operation) | Contains User BFM commands. | Edit this file to add your BFM commands for SERDES simulation. |

# 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) 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.

## 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.

## 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) or contact a local sales office. Sales office listings can be found at 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. 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 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**.



5-02-00463-3/08.14