OPB Device BFM

DS129 (v2.0) April 30, 2004 **Product Overview**

Introduction

The CoreConnect Toolkit OPB Device Bus Functional Model is a simulation hardware component that has an OPB bus interface and may act as a master, as a slave, or as both.

The master component contains logic to automatically initiate bus transactions on the bus. The slave component contains logic to respond to bus transactions based on an address decode operation.

The model maintains an internal memory which can be initialized through the bus functional language and may be dynamically checked during simulation, or when all bus transactions have completed.

Features

- Xilinx OPB bus interface
- May behave as a master, as a slave, or as both
  - The master initiates bus transactions
  - The slave responds to bus transactions
- In Command Mode, the behavior defined in a Bus Functional Language (BFL) file
- In Auto Mode, the master generates random bus transactions
- The model may be dynamically checked during simulation or when all bus transactions have completed

More Information

For detailed information on the IBM OPB Bus Functional Model Toolkit, you may register for the **CoreConnect Lounge** on the Xilinx web site to get access to the IBM CoreConnect documentation.

Core Facts

Core Specifics		
Supported Device Family	All	
Version of Core	opb_device _bfm	v1.00.a
Resources Used		
	Min	Max
I/O	N/A	N/A
LUTs	N/A	N/A
FFs	N/A	N/A
Block RAMs	N/A	N/A
Provided with Core		
Documentation	This document	
Design File Formats	VHDL	
Constraints File	N/A	
Verification	N/A	
Instantiation Template	N/A	
Reference Designs	None	
Design Tool Requirements		
Xilinx Implementation Tools	EDK 6.2 or later ISE 6.2 or later	
Verification	N/A	
Simulation	ModelSim SE/EE 5.6e or later	
Synthesis	N/A	
Support		
Support provided by Xilinx, Inc.		

Implementation

The OPB is a full-featured bus architecture with many features that increase bus performance. To obtain an efficient use of FPGA resource, Xilinx uses a subset of the OPB for Xilinx-developed OPB devices.

The CoreConnect OPB device model has an interface to a full-featured OPB bus. Xilinx has created an interface to the Core-Connect Toolkit components for them to be used within the Xilinx implementation of the OPB bus.

The CoreConnect OPB CoreConnect Toolkit contains standard Bus Functional Models and a Bus Functional Compiler for a rich Bus Functional Language specification. Xilinx utilizes these to provide the functionality and encapsulates the models around a customized interface that performs the translation between the two bus implementations domains. Figure 1 shows how this is done for the OPB device model.

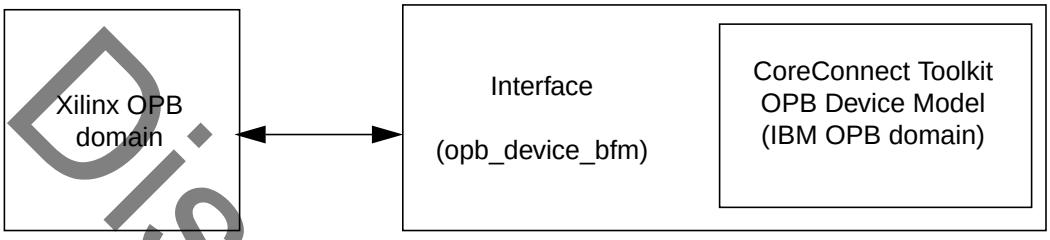

Figure 1: **Xilinx to IBM OPB domain interface**

MPD Parameters

The associated MPD (Microprocessor Peripheral Definition) file contains a list of the peripheral s parameters that are fixed at FPGA configuration time. The parameters are described in Table 1.

Table 1: **MPD Parameters**

Parameter	Description	Allowable Values	Type
NAME	Name	Any string	string
MASTER_ADDR_LO_0	Master base address 0(1)	Valid OPB Address	std_logic_vector
MASTER_ADDR_HI_0	Master high address 0(1)	Valid OPB Address	std_logic_vector
MASTER_ADDR_LO_1	Master base address 1(1)	Valid OPB Address	std_logic_vector
MASTER_ADDR_HI_1	Master high address 1(1)	Valid OPB Address	std_logic_vector
SLAVE_ADDR_LO_0	Slave base address 0	Valid OPB Address	std_logic_vector
SLAVE_ADDR_HI_0	Slave high address 0	Valid OPB Address	std_logic_vector
SLAVE_ADDR_LO_1	Slave base address 1(2)	Valid OPB Address	std_logic_vector
SLAVE_ADDR_HI_1	Slave high address 1(2)	Valid OPB Address	std_logic_vector

Notes:

1. Master address ranges are only used when the device is configured to automatically generate bus transactions by setting master_auto_mode to true in the BFL file. When the master model is configured for automatic mode, there is no need to initialize read/write commands since they are ignored.
2. A secondary slave address range may be used for the device to respond to bus transaction on more than one address range.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
4/27/04	1.0	Initial Xilinx release