XAPP1184 (v3.0) January 16, 2019

PIPE Mode Simulation Using Integrated Endpoint PCI Express Block in Gen3 x8 and Gen2 x8 Configurations

Vidya Gopalakrishnan and Murali Govinda Rao

Summary

The verification of designs involving high speed serial protocols such as PCI Express® can be complex and time consuming. Many verification projects use third-party bus functional models (BFMs) to reduce the complexity of the verification process and to speed up the time spent running the actual simulation.

Because of the complexity of serial transceivers, a significant number of processor cycles are consumed in simulation resulting in long simulation times. In addition, serial transceivers typically have little impact on the behavior of the upper PCI Express layers functionality. With this in mind, many verification projects bypass the serial transceivers for much of their verification and only simulate using transceivers to validate the design(s) at the end of a project.

A specification for interfacing between the PCI Express block and the serial transceivers is maintained by the PCISIG–called the PHY Interface for PCI Express (PIPE). Most PCI Express BFMs allow the device under test to be connected to a PIPE interface rather than a serial interface, effectively bypassing the transceivers, and greatly speeding up simulation times.

This application note describes integrating the PCI-Xactor Kit BFM from Avery Design Systems as a root complex with a Xilinx PCIe integrated block operating as:

- Gen3 x8 Endpoint (Xilinx® Virtex®-7 FPGA Gen3 Integrated Block, UltraScale™ Devices Gen3 Integrated Block, or UltraScale+™ Devices Integrated Block), or
- Gen2 x8 Endpoint (Xilinx 7 Series FPGAs Integrated Block, Virtex-7 FPGA Gen3 Integrated Block, UltraScale Devices Gen3 Integrated Block, or UltraScale+ Devices Integrated Block).

Currently the Avery Design Systems BFM Kit v1.0 supports PCIe link training, link up test, and configuration reads. For further support, contact Avery Design Systems [Ref 5]. Xilinx has tested PIPE mode simulation with Avery Design Systems BFM kit, but it should work with other third-party BFMs as well.

The Gen3 x8 configuration reference design files and the Gen2 x8 configuration reference design files are available for download. See Reference Design Files.

Reference Design and Requirements

Download the reference design files for this application note from the Xilinx website. See Libraries and Reference Design Files.

Tool Flow and Verification

The following checklist indicates the tool flow and verification procedures used for the provided reference design.

Table 1: **Reference Design Details**

Parameter	Description
General	
Developer Names	Vidya Gopalakrishnan, and Murali Govinda Rao
Target Devices	7 series, UltraScale, and UltraScale+
Source code provided	Yes
Source code format	Verilog
Design uses code and IP from existing Xilinx application note, reference designs, Vivado® tools or third party	Yes
Simulation	
Functional Simulation Performed	Yes
Timing simulation performed	No
Test bench used for functional and timing simulations	Yes
Test bench format	Verilog
Simulator software/version used	IES 15.20.042
SPICE/IBIS simulations	-NA-
Implementation	
Synthesis software tools/version used	-NA-
Implementation software tools/versions used	-NA-
Static timing analysis performed	-NA-
Hardware Verification	
Hardware verified	-NA-
Hardware platform used for verification	-NA-

Software Requirements

The following software is required to run the reference design in this application note:

- Vivado Design Suite 2018.2
- Avery Design Systems BFM Kit v1.0
- Cadence IES 15.20.042 simulator

Libraries and Reference Design Files

Avery Design Systems Libraries

To set up the Avery VIP package for use with this application note, follow these steps.

1. Download the Avery VIP for this Xilinx application note. To do so:
   a. Go to http://www.avery-design.com/avery-xilinx-app-note/
   b. Fill in name, company, and email.

   You will receive a direct download link in your email. The download is named `avery_xilinx_app.tar.gz`.

2. Install the Avery package. To do so:
   a. Save the downloaded file to the directory where you plan to install the Avery Verification IP package.
   b. Untar the file using the following command:

```
#> tar -xzvf avery_xilinx_app.tar.gz
```

3. Set the AVERY_VIP environment variable. To do so:
   a. Point the environment variable to the location where the `avery_xilinx_app.tar.gz` file was installed with the following command:

```
setenv AVERY_VIP <path to avery_xilinx_app/ directory>
```

Reference Design Files

The Gen3 x8 configuration reference design files and the Gen2 x8 configuration reference design files are available for download.

Note: In the `pcie4_uscale_plus_0_ex/pcie4_uscale_plus_0_ex.sim/sim_1/behav/ies` directory:

- The `compile.sh` file references the AVERY_VIP environment variable that is set in step 3 above.
- There are two additional files, `xil_sig2pipe.v` and `board_bfm.v`, that are needed as part of *Test Bench Integration*.

Introduction

The PHY Interface for the PCI Express Architecture (PIPE) is intended to enable the development of functionally equivalent PCI Express PHYs. The PCI Express PIPE 2.0 and PIPE 3.0 specification [Ref 1] defines the functionality that must be incorporated in a PIPE compliant PHY, and defines a standard interface between the PHY and a Media Access Layer (MAC) contained in a typical PCI Express block.

This application note provides a methodology to connect the PIPE interface of the Avery Design System PCI-Xactor BFM (in root complex mode) to the PIPE interface of a Xilinx 7 series FPGA Integrated Endpoint Block for PCI Express, an UltraScale Device Integrated Endpoint Block for PCI Express, or an UltraScale+ Device Integrated Endpoint Block for PCI Express. When configured with the proper options, the Xilinx PCI Express Endpoint has PIPE ports at the core top level. These ports can be connected to the Xactor RC BFM to bypass transceivers during simulation.

While this application note demonstrates specific connections to the Xactor BFM from Avery Design Systems, it can also serve as a model of how other third-party BFMs can be connected to the Integrated Endpoint Block for PCI Express through the PIPE interface.

PIPE Use Model

The PCI Express PIPE 2.0 and PIPE 3.0 specifications [Ref 1] provide information about combining multiple PIPE interfaces for multi-lane designs. The PIPE mode simulation uses a model from the Avery Design Systems BFM as a Root Complex (RC) and the Xilinx integrated PCI Express Endpoint block (EP) for an 8-lane design (Figure 1). For more details about shared signals and per-lane signals, see the PCI Express PIPE 2.0 and PIPE 3.0 specification [Ref 1].

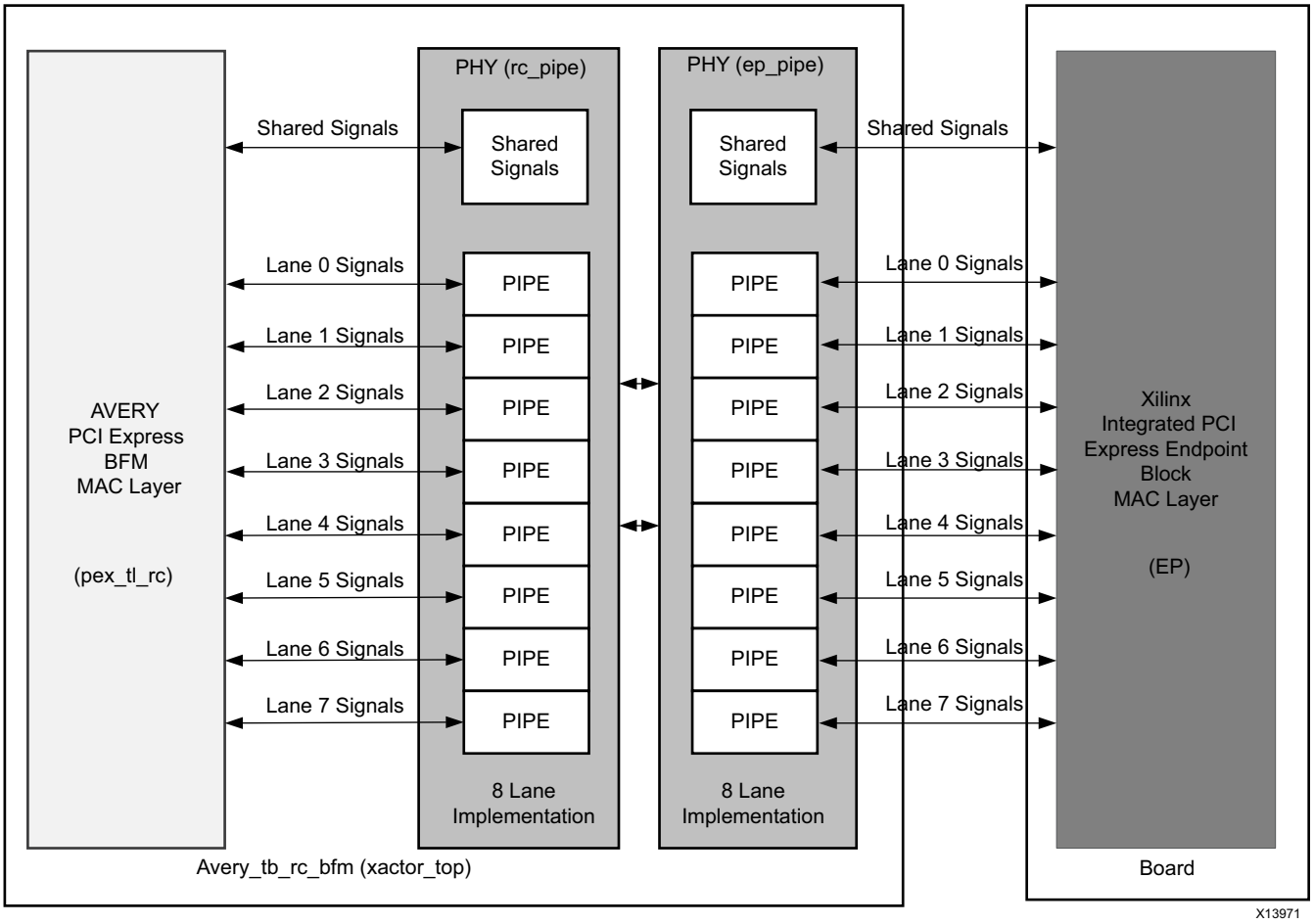

Figure 1: **Block Diagram**

Gen3 x8 Configuration

Xilinx Endpoint PIPE Port Descriptions

The PIPE signals on the Xilinx EP instantiation are encapsulated in buses that are available at the top level of the core. Each lane has one input bus (`pipe_rx_0_sigs[83:0]`, `pipe_rx_1_sigs[83:0]`,...) and one output bus (`pipe_tx_0_sigs[69:0]`, `pipe_tx_1_sigs[69:0]`,...). There are two common bus signals for providing commands, clocks, and status signaling (`common_commands_in[25:0]`, `common_commands_out[16:0]`). Table 2 and Table 3 describe the PIPE bus signals available at the top level of the core and their corresponding mapping inside the EP core (`pcie_top`) PIPE signals.

IMPORTANT: *A new file, `xil_sig2pipe.v`, is delivered in the simulation directory, and the file replaces `phy_sig_gen.v`. The Avery BFM interfaces with the `xil_sig2pipe` instance in the `board.v`.*

Table 2: **Common Input/Output Commands and Endpoint PIPE Signal Mappings**

In Commands	Endpoint PIPE Signal Mapping	Out Commands	Endpoint PIPE Signal Mapping
common_commands_in[25:0]	Not used(3)	common_commands_out[0]	pipe_clk(1)
		common_commands_out[2:1]	pipe_tx_rate_gt(2)
		common_commands_out[3]	pipe_tx_rcvr_det_gt
		common_commands_out[6:4]	pipe_tx_margin_gt
		common_commands_out[7]	pipe_tx_swing_gt
		common_commands_out[8]	pipe_tx_reset_gt
		common_commands_out[9]	pipe_tx_deemph_gt
		common_commands_out[16:10]	Not used(3)

Notes:

1. The pipe_clk signal is an output clock based on the core configuration. For Gen1, pipe_clk is 125 MHz. For Gen2 and Gen3, pipe_clk is 250 MHz.
2. The pipe_tx_rate_gt signal indicates the PIPE rate (Gen1: 2'b00, Gen: 2'b012 and Gen3: 2'b10).
3. This indicates that the port functionality has been deprecated, and the port can be left unconnected.

Table 3: **Input/Output Bus with Endpoint PIPE Signal Mappings**

Input Bus	Endpoint PIPE Signals Mapping	Output Bus	Endpoint PIPE Signals Mapping
pipe_rx0_sigs[31:0]	pipe_rx0_data_gt	pipe_tx0_sigs[31:0]	pipe_tx0_data_gt
pipe_rx0_sigs[33:32]	pipe_rx0_char_is_k_gt	pipe_tx0_sigs[33:32]	pipe_tx0_char_is_k_gt
pipe_rx0_sigs[34]	pipe_rx0_elec_idle_gt	pipe_tx0_sigs[34]	pipe_tx0_elec_idle_gt
pipe_rx0_sigs[35]	pipe_rx0_data_valid_gt	pipe_tx0_sigs[35]	pipe_tx0_data_valid_gt
pipe_rx0_sigs[36]	pipe_rx0_start_block_gt	pipe_tx0_sigs[36]	pipe_tx0_start_block_gt
pipe_rx0_sigs[38:37]	pipe_rx0_syncheader_gt	pipe_tx0_sigs[38:37]	pipe_tx0_syncheader_gt
pipe_rx0_sigs[83:39]	Not used	pipe_tx0_sigs[39]	pipe_tx0_polarity_gt
		pipe_tx0_sigs[41:40]	pipe_tx0_powerdown_gt
		pipe_tx0_sigs[69:42]	Not used(1)

Notes:

1. This port functionality has been deprecated, and the port can be left unconnected.
2. Lanes 1 to 15 (for UltraScale+) and lanes 1 to 7 (for 7 series Gen3 and UltraScale) use similar signal definitions.

Avery Design Systems BFM PIPE Port Descriptions

Xilinx PCI Express Endpoint PIPE ports must be connected with the Avery Design Systems BFM PIPE ports. Connections can be made hierarchically in the simulation top file `board.v` of the Xilinx EP, or they can be connected directly by instantiating the Xilinx EP in the `avery_tb_rc_bfm` top wrapper. Table 4 maps corresponding signals between Avery Design Systems BFM and the Xilinx Endpoint core.

Table 4: **Avery Design Systems BFM Port Mappings**

Avery Port Name	Endpoint Core PIPE Signal Name	Bus PIPE Port Name
aTxDetectRx	pipe_tx_rcvr_det_gt	common_commands_out[0]
aRate[1:0]	pipe_tx_rate_gt	common_commands_out[2:1]
aTxDeemph[143:0]	pipe_tx_deemph_gt	common_commands_out[3]
aTxMargin[2:0]	pipe_tx_margin_gt	common_commands_out[6:4]
aTxSwing	pipe_tx_swing_gt	common_commands_out[7]
apipe_txd[31:0]	pipe_tx0_data_gt	pipe_tx_0_sigs[31:0]
apipe_txk[1:0]	pipe_tx0_char_is_k_gt	pipe_tx_0_sigs[33:32]
aTxElecIdle[0]	pipe_tx0_elec_idle_gt	pipe_tx_0_sigs[34]
aTxDataValid[0]	pipe_tx0_data_valid_gt	pipe_tx_0_sigs[35]
aTxStartBlock[0]	pipe_tx0_start_block_gt	pipe_tx_0_sigs[36]
aTxSyncHeader[1:0]	pipe_tx0_syncheader_gt	pipe_tx_0_sigs[38:37]
aRxPolarity[0]	pipe_rx0_polarity_gt	pipe_tx_0_sigs[39]
aPowerDown[1:0]	pipe_tx0_powerdown_gt	pipe_tx_0_sigs[41:40]
aTxCompliance[0]	pipe_tx0_compliance_gt	pipe_tx_0_sigs[69]
apipe_rxd[31:0]	pipe_rx0_data_gt	pipe_rx_0_sigs[31:0]
apipe_rxk[1:0]	pipe_rx0_char_is_k_gt	pipe_rx_0_sigs[33:32]
aRxDataValid[0]	pipe_rx0_data_valid_gt	pipe_rx_0_sigs[34]
aRxElecIdle[0]	pipe_rx0_elec_idle_gt	pipe_rx_0_sigs[35]
aRxStartBlock[0]	pipe_rx0_start_block_gt	pipe_rx_0_sigs[36]
aRxSyncHeader[1:0]	pipe_rx0_syncheader_gt	pipe_rx_0_sigs[38:37]
aRxStatus[2:0]	pipe_rx0_status_gt	pipe_rx_0_sigs[41:39]
aRxValid[0]	pipe_rx0_valid_gt	pipe_rx_0_sigs[42]
aPhyStatus[0]	pipe_rx0_phy_status_gt	pipe_rx_0_sigs[43]
aRxPresetHint[23:0]1		
aFS[47:0]2		
aLF[47:0]2		
aRxEqEval[7:0]2		
aRxStandby[7:0]2		
aInvalidRequest[7:0]2		
aBlockAlignControl(1)		

Table 4: **Avery Design Systems BFM Port Mappings *(Cont'd)***

Avery Port Name	Endpoint Core PIPE Signal Name	Bus PIPE Port Name
aEncodeDecodeBypass[7:0](2)		
aTxUpconfig[7:0](2)		
aRxUpconfig[7:0](2)		
aLocalPresetIndex[31:0](2)		
aGetLocalPresetCoefficients[7:0](2)		
aWidth[1:0](3)		
aPCLK_Rate[2:0](4)		

Notes:

1. Assign 24'hFFFFFF.
2. Assign 0s.
3. If aRate is 2'b10 (Gen3 speed), assign 2'b10 (32-bit datapath/quad PIPE data). Otherwise, assign 2'b01 (16-bit datapath/dual PIPE data).
4. If aRate is 2'b00 (Gen1 speed), assign 3'b001 (pipe_clk is 125 MHz, and dual pipe data). Otherwise, assign 3'b010 (pipe_clk is 250 MHz, and quad pipe data). The required logic is added in the board.v file of the reference example design.

Test Bench Integration

These steps are required to integrate the Avery Design Systems BFM with the Xilinx Integrated PCI Express Endpoint block:

1. Download the Avery Design System BFM kit and the Xilinx reference example design for this application note. For details, see Avery Design Systems Libraries. Extract the files using `tar xfz <tarfile>`. For the BFM kit, follow the Avery Design Systems BFM integration procedure for setting environment variables, paths and simulation instructions located in the README file under `avery_bfm_kit`.
2. In the Vivado® integrated design environment, create a new design targeting a device that contains a Gen3 PCI Express block. Target a **VCU118** board, and ensure that the target language is set to **Verilog** in the Vivado Project Settings.
3. In the Flow Navigator, select **IP Catalog**.
4. Select **UltraScale+ PCI Express Integrated Block** to customize the PCI Express core.
5. Customize these settings:
   - Lane width = **X8**.
   - Link speed = **8.0 GT/s**.
   - Ensure that **Enable External PIPE Interface** is selected.

   Note: See the *UltraScale+ Devices Integrated Block for PCI Express LogiCORE IP Product Guide (PG213)* [Ref 2] for detailed information about customizing the core for Gen3 mode of operation, and see the *Xilinx Vivado Design Suite User Guide: Logic Simulation* (UG900) [Ref 4] for detailed information about using logic simulation tools in the Vivado Design Suite.

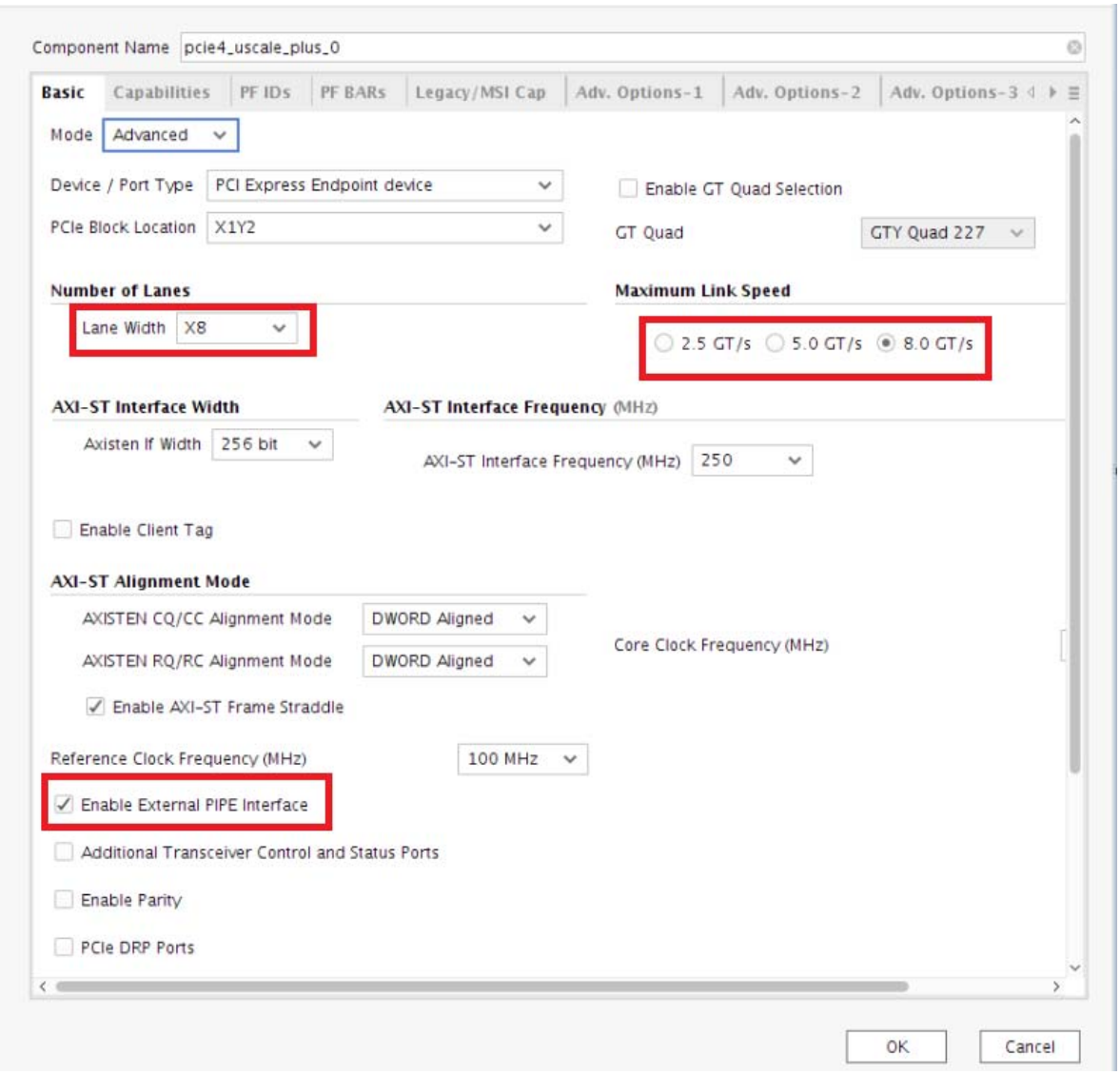

Figure 2: **Customization Options**

6. Generate the IP.
7. After the PCIe IP is generated, right-click the IP and select **Open IP Example Design**.
8. Select **IES** as Target simulator in Simulation Project Settings.
9. In the Flow Navigator, select **Run Simulation**.
10. In the Xilinx example reference design of this application note, locate the three script files in the simulation directory, which are `compile.sh`, `elaborate.sh`, and `simulate.sh`.
11. Open the `compile.sh` file. This file contains inclusion of the required files and directories from the Xilinx reference example design project, and includes the Avery specific files and directories. See the Avery README file for more information.
12. There is an updated `board_bfm.v` file in the Gen3 x8 configuration reference design ZIP file. Replace `board.v` with `board_bfm.v`. The `board_bfm.v` shows the connections between Xilinx and the Avery BFM.

13. The board_bfm.v calls the xil_sig2_pipe module, which instantiates the PIPE signals with the Avery BFM PIPE signals. The module already has the connectivity to tie the reset, clk, rate, powerdown and pipe_data ports. Please do not modify the instance.
14. Analyze the elaborate.sh and simulate.sh files. You can use the same commands in the new scripts for the new project to run simulation.
15. After you have made all connections and updated the three script files, run these commands one at a time.
    - ./compile.sh
    - ./elaborate.sh
    - ./simulate.sh
16. After the simulation is complete, analyze the results.

Note: The above procedure applies for other supported devices of the Xilinx PCIe Gen3 Integrated Block IPs.

Gen2 x8 Configuration

Xilinx Endpoint PIPE Port Descriptions

The PIPE signals on the Xilinx EP instantiation are encapsulated in buses that are available at the top level of the core. Each lane has one input bus (pipe_rx_0_sigs[24:0], pipe_rx_1_sigs[24:0],...) and one output bus (pipe_tx_0_sigs[22:0], pipe_tx_1_sigs[22:0],...). There are two common bus signals for providing commands, clocks, and status signaling (common_commands_in[3:0], common_commands_out[11:0]). Table 5 and Table 6 describe the PIPE bus signals available at the top level of the core and their corresponding mapping inside the EP core (pcie_top) PIPE signals.

IMPORTANT: *A new file, xil_sig2pipe.v, is delivered in the simulation directory, and the file replaces phy_sig_gen.v. The Avery BFM interfaces with the xil_sig2pipe instance in the board.v.*

Table 5: **Common Input/Output Commands and Endpoint PIPE Signal Mappings**

In Commands	Endpoint PIPE Signal Mapping	Out Commands	Endpoint PIPE Signal Mapping
common_commands_in[3:0](1)	Not used(1)	common_commands_out[0]	pipe_clk(2)
		common_commands_out[1]	pipe_tx_rate_gt(3)
		common_commands_out[2]	pipe_tx_rcvr_det_gt
		common_commands_out[3]	pipe_tx_deemph_gt
		common_commands_out[6:4]	pipe_tx_margin_gt

Table 5: **Common Input/Output Commands and Endpoint PIPE Signal Mappings *(Cont'd)***

In Commands	Endpoint PIPE Signal Mapping	Out Commands	Endpoint PIPE Signal Mapping
		common_commands_out[7]	Not used(1)

Notes:

1. This indicates that the port functionality has been deprecated, and the port can be left unconnected.
2. The pipe_clk port is an output clock based on the core configuration. For Gen1, pipe_clk is 125 MHz. For Gen2, pipe_clk is 250 MHz.
3. The pipe_tx_rate_gt port indicates the PIPE rate: 1'b0 for Gen1, and 1'b1 for Gen2.

Table 6: **Input/Output Bus with Endpoint PIPE Signal Mappings**

Input Bus	Endpoint PIPE Signals Mapping	Output Bus	Endpoint PIPE Signal Mapping
pipe_rx0_sigs[15:0]	pipe_rx0_data_gt	pipe_tx0_sigs[15:0]	pipe_tx0_data_gt
pipe_rx0_sigs[17:16]	pipe_rx0_char_is_k_gt	pipe_tx0_sigs[17:16]	pipe_tx0_char_is_k_gt
pipe_rx0_sigs[18]	pipe_rx0_elec_idle_gt	pipe_tx0_sigs[18]	pipe_tx0_elec_idle_gt
pipe_rx0_sigs[14:19]	Not used(1)	pipe_tx0_sigs[19]	pipe_tx0_compliance_gt
		pipe_tx0_sigs[20]	pipe_tx0_polarity_gt
		pipe_tx0_sigs[22:21]	pipe_tx0_powerdown_gt

Notes:

1. This ports functionality has been deprecated, and the port can be left unconnected.
2. Lanes 1 to 7 use similar signal definitions.

Avery Design Systems BFM PIPE Port Descriptions

Xilinx PCI Express Endpoint PIPE ports must be connected with the Avery Design Systems BFM PIPE ports. Connections can be made hierarchically in the simulation top file `board.v` of the Xilinx EP, or they can be connected directly by instantiating Xilinx EP in the `avery_tb_rc_bfm` top wrapper. Table 7 maps corresponding signals between Avery Design Systems BFM and the Xilinx Endpoint core.

Table 7: **Avery Design Systems BFM Port Mapping**

Avery Port Name	Endpoint Core PIPE Signal Name	Bus PIPE Port Name
aTxDetectRx	pipe_tx_rcvr_det_gt	common_commands_out[6]
aRate	pipe_tx_rate_gt	common_commands_out[7]
aTxDeemph	pipe_tx_deemph_gt	common_commands_out[8]
aTxMargin	pipe_tx_margin_gt	common_commands_out[11:9]
aclk250M(1)	pipe_clk	common_commands_in[0]
apipe_txd[15:0]	pipe_tx0_data_gt	pipe_tx_0_sigs[15:0]
apipe_txk[1:0]	pipe_tx0_char_is_k_gt	pipe_tx_0_sigs[17:16]

Table 7: **Avery Design Systems BFM Port Mapping *(Cont'd)***

Avery Port Name	Endpoint Core PIPE Signal Name	Bus PIPE Port Name
aRxPolarity[0]	pipe_rx0_polarity_gt	pipe_tx_0_sigs[18]
aTxCompliance[0]	pipe_tx0_compliance_gt	pipe_tx_0_sigs[19]
aTxElecIdle[0]	pipe_tx0_elec_idle_gt	pipe_tx_0_sigs[20]
aPowerDown	pipe_tx0_powerdown_gt	pipe_tx_0_sigs[22:21]
apipe_rxd[15:0]	pipe_rx0_data_gt	pipe_rx_0_sigs[15:0]
apipe_rxk[1:0]	pipe_rx0_char_is_k_gt	pipe_rx_0_sigs[17:16]
aRxValid[0]	pipe_rx0_valid_gt	pipe_rx_0_sigs[18]
aRxStatus[2:0]	pipe_rx0_status_gt	pipe_tx_0_sigs[22:20]
aPhyStatus	pipe_rx0_phy_status_gt	pipe_tx_0_sigs[23]
aRxElecIdle[0]	pipe_rx0_elec_idle_gt	pipe_tx_0_sigs[24]
	pipe_rx0_chanisaligned_gt(2) pl_ltssm_state(2)	pipe_rx_0_sigs[19] common_commands_out[5:0]

Notes:

1. `aclk250M` is expected to change from 125 MHz to 250 MHz when the link has trained to Gen2 speed.
2. These signals can be ignored.

Test Bench Integration

These steps are required to integrate the Avery Design Systems BFM with the Xilinx Integrated PCI Express Endpoint block:

1. Download the Avery Design System BFM kit and Xilinx reference example design of this application note. For details, see Libraries and Reference Design Files. Extract the files using `tar xfz <tarfile>`. For the BFM kit, follow the Avery Design Systems BFM integration procedure for setting environment variables, paths and simulation instructions located in the README file under `avery_bfm_kit`.
2. In the Vivado integrated design environment, create a new design targeting a device that contains a Gen2 PCI Express block. Target a **KC705** board, and ensure that the target language is set to **Verilog** in the Vivado project settings.
3. In the Flow Navigator, select **IP Catalog**.
4. Select the **7 Series Integrated Block for PCI Express** to customize the PCI Express core.
5. Customize these settings:
   - Lane width = **X8**.
   - Link speed = **5.0 GT/s**.
   - Ensure that **Enable External PIPE Interface** is selected.

Note: See the *Xilinx LogiCORE IP 7 Series FPGAs Integrated Block for PCI Express* (PG054) [Ref 3] for detailed information about customizing the core for Gen2 mode of operation, and see the *Xilinx Vivado Design Suite User Guide: Logic Simulation* (UG900) [Ref 4] for detailed information about using logic simulation tools in the Vivado Design Suite.

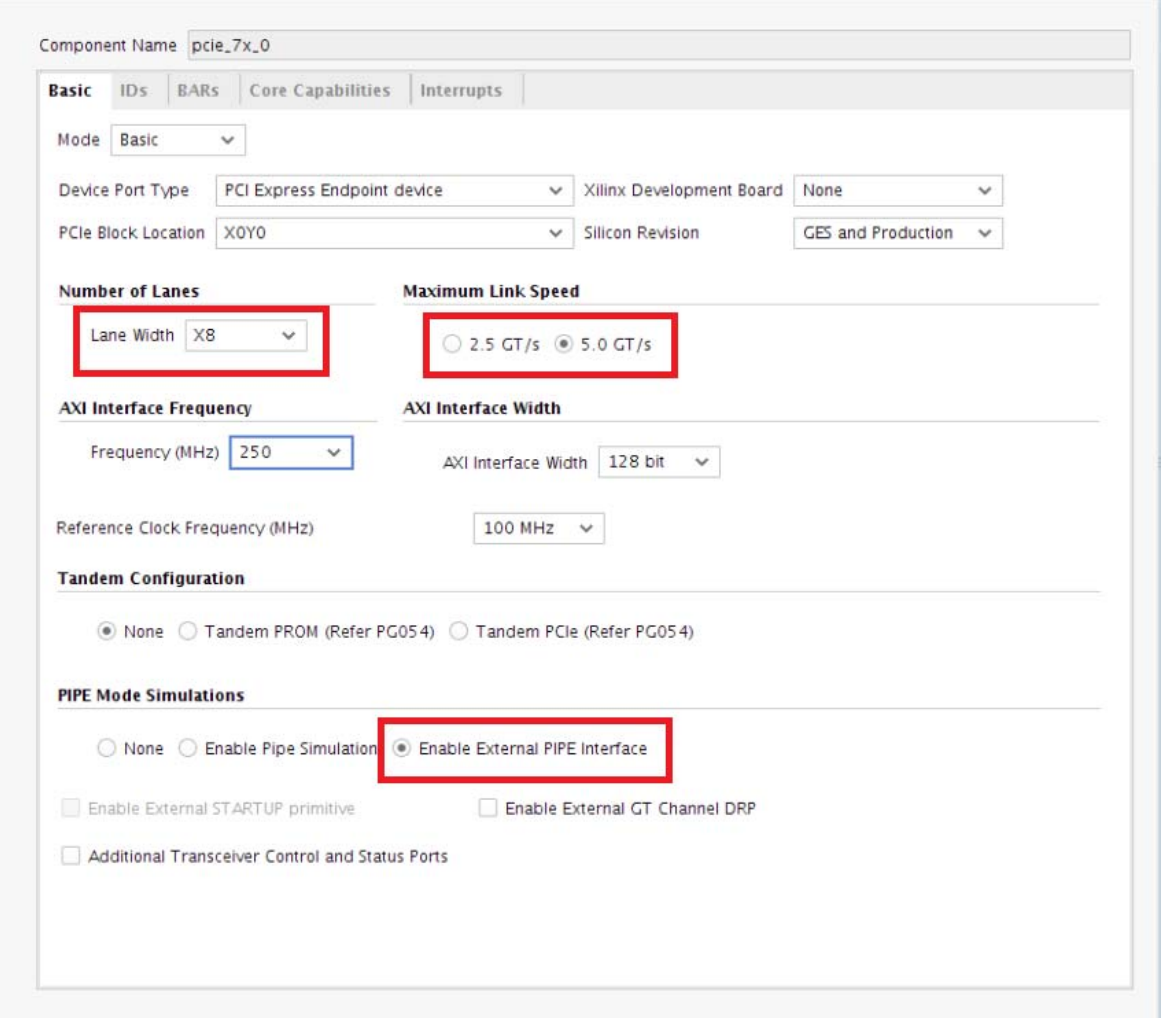

Figure 3: **Customization Options**

6. Generate the IP.
7. After the PCIe IP is generated, right-click the IP and select **Open IP Example Design**.
8. Select **IES** as Target simulator in Simulation Project Settings.
9. In the Flow Navigator, select **Run Simulation**.
10. In the Xilinx example reference design of this application note, locate the three script files in the simulation directory, which are `compile.sh`, `elaborate.sh`, and `simulate.sh`.
11. Open the `compile.sh` file. This file contains inclusion of the required files and directories from the Xilinx reference example design project, and includes the Avery specific files and directories.
12. There is an updated `board_bfm.v` file in the Gen2 X8 configuration reference design ZIP. Replace `board.v` with `board_bfm.v`. The `board_bfm.v` shows the PIPE connections between Xilinx and Avery BFM.

13. The `board_bfm.v` file calls the `xil_sig2_pipe` module, which instantiates the PIPE signals with the Avery BFM PIPE signals. The module already has the connectivity to tie the `reset`, `clk`, `rate`, `powerdown` and `pipe_data` ports. Please do not modify the instance.
14. Analyze the `elaborate.sh` and `simulate.sh` files. You can use the same commands in the new scripts for the new project to run simulation.
15. After you have made all connections and updated three script files, run these commands one at a time.
    - `./compile.sh`
    - `./elaborate.sh`
    - `./simulate.sh`
16. After the simulation is complete, analyze the results.

Simulation Logs

The following log files created during simulation in `sim_1/behav/`provide valuable information for analyzing and debugging issues.

- **simulation.log**: Provides details about simulation summary, such as Link up, BIOS Enumeration, Memory Write/Read Test Simulation PASSED or FAILED.
- **tracker_phy_rc.txt**, **tracker_dll_tc.txt** and **tracker_tl_rc.txt**: Provides details about exchange of PLP, DLLP, TLPs in the form of symbols across TX and RX lanes of Avery Design Systems RC BFM, such as TS1, TS2, SKP, SDP, STP, ACK, NAK, and the corresponding LTSSM changes.

Serial versus PIPE Modes

Table 8 provides guidelines about the simulation time differences between the Serial and PIPE modes of simulation in the Gen3 x8/Gen2 x8 configuration. The findings assume that the design contains Xilinx transceivers.

Note: Xilinx transceivers are found in serial mode simulation only. In PIPE mode simulation, they are bypassed.

Table 8: **Simulation Time Differences in Serial Mode and PIPE Simulation(1)**

Configuration	Simulation Mode	Simulation Time(2)	Wall Clock Time(3)
Gen3 x8	Serial mode	118 µs	8 minutes
	PIPE mode	32 µs	~45 seconds

Table 8: **Simulation Time Differences in Serial Mode and PIPE Simulation[1] *(Cont'd)***

Configuration	Simulation Mode	Simulation Time[2]	Wall Clock Time[3]
Gen2 x8	Serial mode	87 µs	6 minutes
	PIPE mode	23 µs	~20 seconds

Notes:

1. The numerical values in this table are captured on a typical Xilinx server with the Cadence IES simulator. Minor deviations can be expected based on the server configuration and simulator used.
2. Simulation time includes the simulation time as seen on the simulator only until the PCIe link is up.
3. Wall clock time shows compile, elaboration and simulation time until the PCIe link is up.

Conclusion

PIPE mode simulation is very useful for reducing the simulation time during verification of complex PCI Express applications. From this approach, this application note highlighted the reduction in simulation time and the differences between serial mode and PIPE mode. This document also provided simple integration steps between Avery Design Systems RC BFM and Xilinx EP core. Instructions are also provided for incorporating these changes in user designs for speeding up the PCIE application verification cycle.

Documentation Navigator and Design Hubs

Xilinx® Documentation Navigator provides access to Xilinx documents, videos, and support resources, which you can filter and search to find information. To open the Xilinx Documentation Navigator (DocNav):

- From the Vivado® IDE, select **Help > Documentation and Tutorials**.
- On Windows, select **Start > All Programs > Xilinx Design Tools > DocNav**.
- At the Linux command prompt, enter `docnav`.

Xilinx Design Hubs provide links to documentation organized by design tasks and other topics, which you can use to learn key concepts and address frequently asked questions. To access the Design Hubs:

- In the Xilinx Documentation Navigator, click the **Design Hubs View** tab.
- On the Xilinx website, see the Design Hubs page.

Note: For more information on Documentation Navigator, see the Documentation Navigator page on the Xilinx website.

References

The documents referenced in this application note are listed in this section.

1. PHY Interface for the PCI Express (PIPE) Specification at www.intel.com
2. *UltraScale+ Devices Integrated Block for PCI Express LogiCORE IP Product Guide* (PG213)
3. *7 Series FPGAs Integrated Block for PCI Express Product Guide* (PG054)
4. *Vivado Design Suite User Guide: Logic Simulation* (UG900)
5. Avery Design Systems
6. *Virtex-7 FPGA Gen3 Integrated Block for PCI Express v3.0 Product Guide* (PG023)
7. *UltraScale Devices Gen3 Integrated Block for PCI Express LogiCORE IP Product Guide* (PG156)
8. *UltraScale+ Devices Integrated Block for PCI Express Product Guide* (PG213)

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
01/16/2019	3.0	• Added support for UltraScale and UltraScale+ devices. • Updated Endpoint PIPE Signal Mappings for both Gen3 and Gen2. • Reference design for Gen3 x8 configuration now targets UltraScale+ device. • Updated to the IES simulator. • Moved the Gen3 configuration section before the Gen2 configuration section to reflect most common use case.
02/19/2014	2.0	Add Gen3 x8 configuration.
11/01/2013	1.0	Initial Xilinx release.

Please Read: Important Legal Notices

performance; you assume sole risk and liability for use of Xilinx products in such critical applications, please refer to Xilinx's Terms of Sale which can be viewed at https://www.xilinx.com/legal.htm#tos.

AUTOMOTIVE APPLICATIONS DISCLAIMER

AUTOMOTIVE PRODUCTS (IDENTIFIED AS "XA" IN THE PART NUMBER) ARE NOT WARRANTED FOR USE IN THE DEPLOYMENT OF AIRBAGS OR FOR USE IN APPLICATIONS THAT AFFECT CONTROL OF A VEHICLE ("SAFETY APPLICATION") UNLESS THERE IS A SAFETY CONCEPT OR REDUNDANCY FEATURE CONSISTENT WITH THE ISO 26262 AUTOMOTIVE SAFETY STANDARD ("SAFETY DESIGN"). CUSTOMER SHALL, PRIOR TO USING OR DISTRIBUTING ANY SYSTEMS THAT INCORPORATE PRODUCTS, THOROUGHLY TEST SUCH SYSTEMS FOR SAFETY PURPOSES. USE OF PRODUCTS IN A SAFETY APPLICATION WITHOUT A SAFETY DESIGN IS FULLY AT THE RISK OF CUSTOMER, SUBJECT ONLY TO APPLICABLE LAWS AND REGULATIONS GOVERNING LIMITATIONS ON PRODUCT LIABILITY.