LogiCORE™ IP Aurora 8B/10B v5.1

Getting Started Guide

UG352 (v5.1) December 2, 2009

XILINX®

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
03/24/08	2.9	Initial Xilinx release.
03/24/08	2.9.1	Post-release updates and corrections.
06/27/08	3.0	Virtex-5 FPGA Aurora v3.0 release. Added “Using ChipScope Pro Cores with the Aurora 8B/10B Core,” page 16section. Added “Using the Timer-Based Simplex Mode” section.
04/24/09	4.1	LogiCORE IP Aurora 8B/10B v1.4 core. Update tools to v11.1. Renamed document title to include the 8B/10B protocol. Added Virtex-6 FPGA RocketIO GTX transceiver support. Removed *Using the Timer-Based Simplex Mode*, *Using the ISE Software Flow to Generate the Aurora 8B/10B Core*, and *Simulating the Example Design using the ISIM Simulator (ISim)* sections. Updated the following sections: • “About the Core,” page 9 • “Generating the Core,” page 14 • “Implementing the Example Design,” page 16 • “Using ChipScope Pro Cores with the Aurora 8B/10B Core,” page 16 • “Simulating the Example Design,” page 15 • “Aurora 8B/10B Project Directory Structure,” page 19
06/24/09	4.2	LogiCORE IP Aurora 8B/10B v4.2 release. Update tools to v11.2. Miscellaneous changes to the content of “Aurora 8B/10B Project Directory Structure,” page 19 include: deleted <component name> directory; moved aurora_8b10b_readme.tx to the doc directory
12/02/09	5.1	LogiCORE IP Aurora 8B/10B v5.1 release. Update tools to v11.4. Added support for Spartan-6 FPGA GTP transceivers. Moved the project directory structure and tables to Chapter 4, “Detailed Example Design.”

Table of Contents

Revision History 2

Preface: About This Guide

Guide Contents 5
Additional Resources 5
Conventions 6
Typographical 6
Online Document 8

Chapter 1: Introduction

About the Core 9
Recommended Design Experience 9
Related Xilinx Documents 9
Additional Core Resources 10
Technical Support 10
Feedback 10
Core 10
Document 10

Chapter 2: Licensing the Core

Supported Tools and System Requirements 11
Operating Systems 11
Tools 11
Before You Begin 11
License Options 12
Obtaining Your Full License Key 12
Installing Your License File 12

Chapter 3: Quick Start Example Design

Overview 13
Generating the Core 14
Simulating the Example Design 15
Implementing the Example Design 16
Using ChipScope Pro Cores with the Aurora 8B/10B Core 16
Description 16

Chapter 4: Detailed Example Design

<project directory> 18
<project directory>/<component name> 18
<component name>/doc 18

<component name>/example_design 18
/example_design/cc_manager 19
/example_design/clock_module 19
/example_design/gt 19
/example_design/traffic_gen_check 19
<component name>/implement 20
<component name>/simulation 20
/simulation/functional 21
/simulation/timing 21
<component name>/src 22

About This Guide

The *LogiCORE IP Aurora 8B/10B v5.1 Getting Started Guide* provides information for generating a LogiCORE™ IP Aurora 8B/10B core using Virtex®-5 FPGA GTP/GTX transceivers, Virtex-6 FPGA GTX transceivers, and Spartan®-6 FPGA GTP transceivers. The core implements the Aurora 8B/10 protocol using the high-speed serial transceivers on the Virtex-5 LXT, SXT, FXT, and TXT family, the Virtex-6 LXT, SXT, CXT, and Lower Power family, and the Spartan-6 LXT family. The information includes customizing and simulating the core using the provided example design, and running the design files through implementation using the Xilinx tools.

Guide Contents

This guide contains the following chapters:

- Preface, "About this Guide" introduces the organization and purpose of this guide, a list of additional resources, and the conventions used in this document.
- Chapter 1, "Introduction" describes the core and related information, including recommended design experience, additional resources, technical support, and submitting feedback to Xilinx.
- Chapter 2, "Licensing the Core" provides information about installing and licensing the core.
- Chapter 3, "Quick Start Example Design" provides an overview of the Aurora 8B/10B protocol and core, and gives a step-by-step tutorial on how to generate Aurora 8B/10B designs with the CORE Generator™ software.
- Chapter 4, "Detailed Example Design" provides detailed information about the example design, including a description of files and the directory structure generated by the Xilinx CORE Generator tool, the purpose and contents of the provided scripts, the contents of the example HDL wrappers, and the operation of the demonstration test bench

Additional Resources

To find additional documentation, see the Xilinx website at:

http://www.xilinx.com/support/documentation/index.htm.

To search the Answer Database of silicon, software, and IP questions and answers, or to create a technical support WebCase, see the Xilinx website at:

http://www.xilinx.com/support/mysupport.htm.

Conventions

This document uses the following conventions. An example illustrates each convention.

Typographical

The following typographical conventions are used in this document:

Convention	Meaning or Use	Example
`Courier font`	Messages, prompts, and program files that the system displays	`speed grade: - 100`
`Courier bold`	Literal commands that you enter in a syntactical statement	**`ngdbuild`** *design_name*
Helvetica bold	Commands that you select from a menu	**File → Open**
	Keyboard shortcuts	**Ctrl+C**
Italic font	Variables in a syntax statement for which you must supply values	**`ngdbuild`** *design_name*
	References to other manuals	See the *User Guide* for more information.
	Emphasis in text	If a wire is drawn so that it overlaps the pin of a symbol, the two nets are *not* connected.
Dark Shading	Items that are not supported or reserved	This feature is not supported
Square brackets [ ]	An optional entry or parameter. However, in bus specifications, such as **`bus[7:0]`**, they are required.	**`ngdbuild`** [*option_name*] *design_name*
Braces { }	A list of items from which you must choose one or more	**`lowpwr =`**{**`on`**|**`off`**}
Vertical bar |	Separates items in a list of choices	**`lowpwr =`**{**`on`**|**`off`**}
Angle brackets < >	User-defined variable or in code samples	<directory name>

Vertical ellipsis
.
.
.
	Repetitive material that has been omitted	`IOB #1: Name = QOUT'`
		`IOB #2: Name = CLKIN'`
		.
		.
		.

Horizontal ellipsis ...	Repetitive material that has been omitted	**`allow block`** *block_name loc1 loc2 ... locn;*

Convention	Meaning or Use	Example
Notations	The prefix '0x' or the suffix 'h' indicate hexadecimal notation	A read of address 0x00112975 returned 45524943h.
	An '_n' means the signal is active low	**`usr_teof_n`** is active low.

Online Document

The following conventions are used in this document:

Convention	Meaning or Use	Example
Blue text	Cross-reference link to a location in the current document	See the section "Additional Resources" for details. Refer to "Title Formats" in Chapter 1 for details.
Red text	Cross-reference link to a location in another document	See Figure 2-5 in the *Virtex-II Platform FPGA User Guide.*
Blue, underlined text	Hyperlink to a website (URL)	Go to www.xilinx.com for the latest speed files.

XILINX®

Chapter 1

Introduction

This chapter introduces the LogiCORE™ IP Aurora 8B/10B v5.1 core and provides related information, including recommended design experience, additional resources, technical support, and how to submit feedback to Xilinx. The Aurora 8B/10B core is designed to support both Verilog and VHDL design environments. In addition, the example design delivered with the core is provided in both Verilog and VHDL.

About the Core

The Aurora 8B/10B core is a CORE Generator™ software IP core, included in the latest IP Update on the Xilinx IP Center. For detailed information about the core, see www.xilinx.com/aurora. For information about system requirements, installation, and licensing options, see Chapter 2, "Licensing the Core."

Recommended Design Experience

Although the Aurora 8B/10B core is a fully verified solution, the challenge associated with implementing a complete design varies depending on the configuration and functionality of the application. For best results, previous experience building high-performance, pipelined FPGA designs using Xilinx implementation software and user constraints files (UCF) is recommended. Contact your local Xilinx representative for a closer review and estimation for your specific requirements.

Related Xilinx Documents

Prior to generating an Aurora 8B/10B core, users should be familiar with the following:

- SP002: *Aurora 8B/10B Protocol Specification*
- SP006: *LocalLink Interface Specification*
- UG196: *Virtex-5 FPGA RocketIO GTP Transceiver User Guide*
- UG198: *Virtex-5 FPGA RocketIO GTX Transceiver User Guide*
- UG366: *Virtex-6 FPGA GTX Transceivers User Guide*
- UG386: *Spartan-6 FPGA GTP Transceivers User Guide*
- ISE software documentation: www.xilinx.com/ise

Additional Core Resources

For detailed information and updates about the Aurora 8B/10B core, see the following documents located on the Aurora 8B/10B core product page http://www.xilinx.com/products/ipcenter/aurora8b10b.htm:

- DS637: *LogiCORE IP Aurora 8B/10B v5.1 Data Sheet*
- Aurora 8B/10B Release Notes
- UG353: *LogiCORE IP Aurora 8B/10B v5.1 User Guide*
- UG058: *Aurora 8B/10B Bus Functional Model User Guide* (Contact: auroramkt@xilinx.com)

Technical Support

For technical support, go to www.xilinx.com/support. Questions are routed to a team of engineers with expertise using the Aurora 8B/10B core.

Xilinx will provide technical support for use of this product as described in the LogiCORE IP Aurora 8B/10B v5.1 license (see "Installing Your License File," page 12). Xilinx cannot guarantee timing, functionality, or support of this product for designs that do not follow the guidelines in *LogiCORE IP Aurora 8B/10B v5.1 User Guide* and the *LogiCORE IP Aurora 8B/10B v5.1 Getting Started Guide*.

Feedback

Xilinx welcomes comments and suggestions about the Aurora 8B/10B core and the accompanying documentation.

Core

For comments or suggestions about the Aurora 8B/10B core, please submit a WebCase from www.xilinx.com/support. Be sure to include the following information:

- Product name
- Core version number
- List of parameter settings
- Explanation of your comments

Document

For comments or suggestions about this document, please submit a WebCase from www.xilinx.com/support. Be sure to include the following information:

- Document title
- Document number
- Page number(s) to which your comments refer
- Explanation of your comments

XILINX®

Chapter 2

Licensing the Core

This chapter provides instructions for obtaining a license key for the Aurora 8B/10B core, which you must do before using it in your designs, and for installing the core.

Supported Tools and System Requirements

Operating Systems

Windows

- Windows XP Professional 32-bit/64-bit
- Windows Vista Business 32-bit/64-bit

Linux

- Red Hat Enterprise Linux WS v4.0 32-bit/64-bit
- Red Hat Enterprise Desktop v5.0 32-bit/64-bit (with Workstation Option)
- SUSE Linux Enterprise (SLE) desktop and server v10.1 32-bit/64-bit

Tools

- ISE® software v11.4
- Mentor Graphics ModelSim v6.4b and above

Check the release notes for the required Service Pack; ISE Service Packs can be downloaded from www.xilinx.com/xlnx/xil_sw_updates_home.jsp?update=sp.

Before You Begin

This chapter assumes you have installed the core using either the CORE Generator™ IP Software Update installer or by performing a manual installation after downloading the core from the web. For information about installing the core, see:

www.xilinx.com/aurora

License Options

The Aurora 8B/10B core is made available with a Full License key. After installing the required Xilinx ISE® software and IP updates, install a Full License key.

The Full license key provides full access to all core functionality both in simulation and in hardware, including:

- Functional simulation support
- Full implementation support including place and route and bitstream generation
- Full functionality in the programmed device with no time outs

Obtaining Your Full License Key

To obtain a Full license key:

1. Navigate to the product page for this core: www.xilinx.com/aurora
2. Click the **Aurora 8B/10B** link at the bottom of the page.
3. Follow the instructions to install the required Xilinx ISE software and IP updates and generate the required license key on the Xilinx Product Download and Licensing Site.

Installing Your License File

An email will be sent to you containing instructions for installing your license file. Additional details about IP license key installation can be found in the ISE Design Suite Installation, Licensing and Release Notes document available at www.xilinx.com/support/documentation/dt_ise.htm.

Chapter 3

Quick Start Example Design

This chapter introduces the example design that is included with the Aurora 8B/10B core. The quick start instructions are a step-by-step procedure for generating an Aurora 8B/10B core, implementing the core in hardware using the accompanying example design, and simulating the core with the provided demonstration testbench (demo_tb). For detailed information about the example design provided with the Aurora 8B/10B core, see the *LogiCORE™ IP Aurora 8B/10B User Guide*.

Overview

The quick start example consists of the following components:

- An instance of the Aurora 8B/10B core generated using the default parameters
  - Full-duplex with a single GTP/GTX transceiver
  - LocalLink interface
  - Virtex®-5 FPGA target device
- A top-level example design (aurora_8b10b_v5_1_example_design) with user constraints file (UCF) for an ML523 board
- A demonstration testbench to simulate two instances of the example design

The Aurora 8B/10B example design has been tested with XST for synthesis and Mentor Graphics ModelSim for simulation.

Figure 3-1 shows a block diagram of the default Aurora 8B/10B example design.

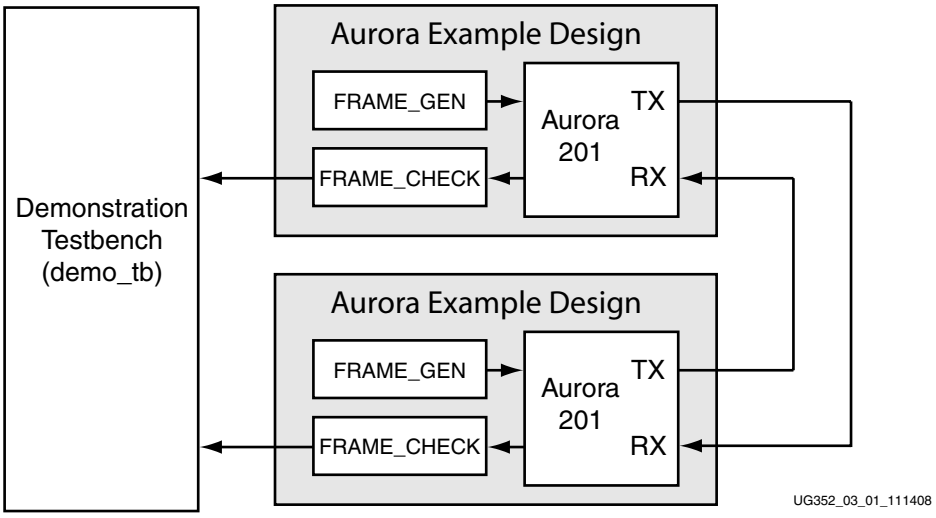

Figure 3-1: **Example Design**

Generating the Core

To generate an Aurora 8B/10B core with default values using the CORE Generator™ tool:

1. Start the CORE Generator software from a required directory.

   For help starting and using the CORE Generator software, see *CORE Generator Help* in the ISE® software documentation.

2. Choose **File** → **New Project**.
3. Type a project name.
4. To set project options:
   - On the Part tab, for Family select **Virtex5**. For Device, select an appropriate device that supports GTP/GTX transceivers, such as **xc5vfx70t**.

     Note: If an unsupported silicon family is selected, the Aurora 8B/10B appears light grey in the taxonomy tree and cannot be customized. Only devices containing GTP/GTX transceivers are supported by the core.

   - No further project options need to be set.
   - Optionally, on the Generation tab, set the Design Entry pull-down to **Verilog**.
5. After creating the project, locate the Aurora 8B/10B core v5.1 in the taxonomy tree under:

   `/Communication_&_Networking/Serial_Interfaces`

6. Double-click the core. If the license file is not properly configured, the CORE Generator software reports an error. See Chapter 2, "Licensing the Core."

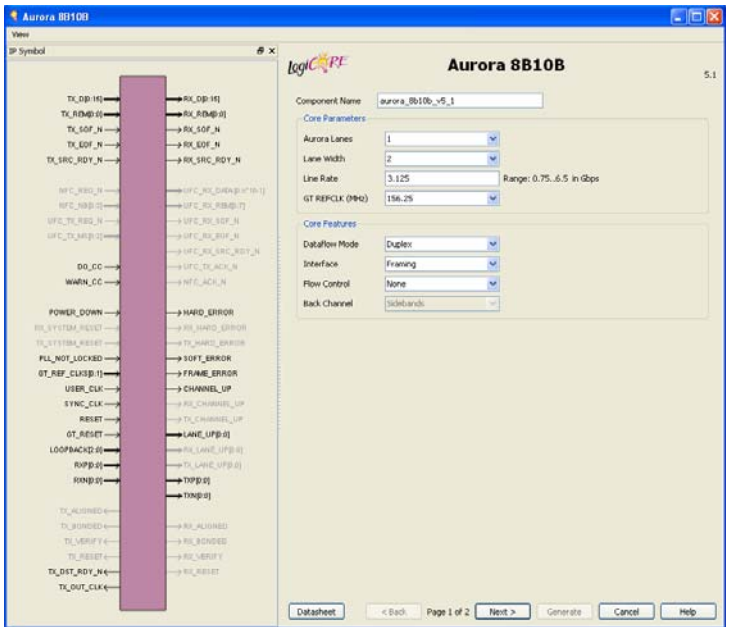

Figure 3-2: **CORE Generator Aurora 8B/10B Customization Screen**

7. In the Component Name field, enter a name for the core instance. This example uses the name **aurora_8b10b_v5_1**.
8. Click **Generate**.

The core and its supporting files, including the example design, are generated in the project directory. For detailed information about the example design files and directories, see Chapter 4, "Detailed Example Design."

Simulating the Example Design

The Aurora 8B/10B core provides a quick way to simulate and observe the behavior of the core using the provided example design. Prior to simulating the core, the functional (gate-level) simulation models must be generated. You must compile all source files in the following directories to a single library as shown in Table 3-1. Refer to the *Synthesis and Verification Design Guide* for ISE 11.4 software for instructions on how to compile ISE software simulation libraries.

Table 3-1: **Required Simulation Libraries**

HDL	Library	Source Directories
Verilog	UNISIMS_VER	<*Xilinx dir*>/verilog/src/unisims <*Xilinx dir*>/ secureip/<*SIMULATOR*>
VHDL	UNISIM	<*Xilinx dir*>/vhdl/src/unisims <*Xilinx dir*>/ secureip/<*SIMULATOR*>

Notes:

1. SIMULATOR can be Modelsim.

The Aurora 8B/10B core provides a command line script to simulate the example design. To run a VHDL or Verilog ModelSim simulation of the Aurora 8B/10B core, use the following instructions:

1. Launch the ModelSim simulator and set the current directory to:

   ***<project directory>*/aurora_8b10b_v5_1/simulation/functional**

2. Set the MTI_LIBS variable:

   `modelsim>` **setenv MTI_LIBS *<path to compiled libraries>***

3. Launch the simulation script:

   `modelsim>` **do simulate_mti.do**

The ModelSim script compiles the example design and testbench, and adds the relevant signals to the wave window. After the design is compiled and the wave window is displayed, run the simulation to see the Aurora 8B/10B core power up, followed by Aurora 8B/10B channel initialization and data transfer. Data transfer begins after the CHANNEL_UP signal goes High.

Implementing the Example Design

After the core is generated, the design can be processed by the Xilinx implementation tools. The generated output files include several scripts to assist the user in running the Xilinx software.

From the command prompt, navigate to the project directory and type the following:

For Windows

```
ms-dos> cd aurora_8b10b_v5_1\implement
ms-dos> .\implement.bat
```

For Linux

```
% cd aurora_8b10b_v5_1/implement
%./implement.sh
```

These commands execute a script that synthesizes, translates, maps, place-and-routes the example design and produces a bitstream file. The resulting files are placed in the results directory created within the implement directory.

Using ChipScope Pro Cores with the Aurora 8B/10B Core

Description

The ChipScope™ Pro ICON and VIO cores aid in debugging and validating the design in board. To assist with debugging, these cores are provided with the Aurora 8B/10B core, which is enabled by setting USE_CHIPSCOPE as 1 in the aurora_8b10b_v5_1_example_design file.

XILINX®

Chapter 4

Detailed Example Design

This chapter provides detailed information about the example design, including a description of files and the directory structure generated by the Xilinx CORE Generator™ tool, the purpose and contents of the provided scripts, the contents of the example HDL wrappers, and the operation of the demonstration test bench.

- **<project directory>**
  Top-level project directory; name is user-defined.
  - <project directory>/<component name>
    Top-level core directory; contains the core deliverables
    - <component name>/doc
      Product documentation
    - <component name>/example_design
      Example design and user constraints files
      - /example_design/cc_manager
        Verilog/VHDL design files for the clock compensation block
      - /example_design/clock_module
        Verilog/VHDL design files for the clocking blocks
      - /example_design/gt
        Verilog/VHDL wrapper files for the GTP/GTX transceivers
      - /example_design/traffic_gen_check
        Verilog/VHDL design files for the frame generator and checker
    - <component name>/implement
      Implementation scripts and support files
    - <component name>/simulation
      Simulation test bench and simulation script files
      - /simulation/functional
        Functional simulation files
      - /simulation/timing
        Timing simulation file
    - <component name>/src
      Verilog/VHDL files for the core

The Aurora 8B/10B core directories and their associated files are defined below.

<project directory>

The project directory contains the CORE Generator project files.

Table 4-1: **project Directory**

Name	Description
<project directory>	
<coregen project filename>.cgp	CORE Generator project file

Back to Top

<project directory>/<component name>

This top level core directory contains the core deliverables.

<component name>/doc

The doc directory contains the product documentation.

Table 4-2: **doc Directory**

Name	Description
<component name>/doc	
aurora_8b10b_ds637.pdf	*LogiCORE IP Aurora 8B/10B data sheet*
aurora_8b10b_gsg352.pdf	*LogiCORE IP Aurora 8B/10B Getting Started Guide*
aurora_8b10b_ug353.pdf	*LogiCORE IP Aurora 8B/10B User Guide*
aurora_8b10b_readme.txt	Release notes file

Back to Top

<component name>/example_design

The example_design directory contains the example design and user constraints files provided with the core.

Table 4-3: **example_design Directory**

Name	Description
<component name>/example_design	
<component name>_example_design.v[hd]	Example design source file
<component_name>_example_design.ucf--> Example design UCF file	Aurora 8B/10B example design constraints

Back to Top

/example_design/cc_manager

The cc_manager directory contains the clock compensation source file.

Table 4-4: **cc_manager Directory**

Name	Description
<component name>/example_design/cc_manager	
<component name>_standard_cc_module.v[hd]	Clock compensation module source file

Back to Top

/example_design/clock_module

The clock_module directory contains the clock module source file.

Table 4-5: **clock_module Directory**

Name	Description
<component name>/example_design/clock_module	
<component name>_clock_module.v[hd]	Clock module source file

Back to Top

/example_design/gt

The gt directory contains the Verilog/VHDL wrapper files for the GTP/GTX transceivers.

Table 4-6: **gt Directory**

Name	Description
<component name>/example_design/gt	
<component name>_transceiver_tile.v[hd] <component name>_transceiver_wrapper.v[hd]	Verilog/VHDL wrapper files for the transceiver

Back to Top

/example_design/traffic_gen_check

The traffic_gen_check directory contains frame generator and frame checker modules for Aurora 8B/10B core.

Table 4-7: **traffic_gen_check Directory**

Name	Description
<component name>/example_design/traffic_gen_check	
<component name>_frame_check.v[hd] <component name>_frame_gen.v[hd]	Example design traffic generation and checker files

Back to Top

<component name>/implement

The implement directory contains scripts and support files for both Linux and Windows operating systems. These scripts automate the process of synthesizing and implementing the files needed for the example design.

Table 4-8: **implement Directory**

Name	Description
<component name>/implement	
`implement.bat`	Windows batch file that processes the example design through the Xilinx tool flow
`implement.sh`	Linux shell script that processes the example design through the Xilinx tool flow
`xst.scr`	XST script file for the example design
`xst.prj`	XST project file for the example design
`icon.ngc` `vio.ngc`	Virtex-5 family NGC files for the debug cores compatible with the ChipScope Pro Analyzer tool
`v6_icon.ngc` `v6_vio.ngc`	Virtex-6 family NGC files for the debug cores compatible with the ChipScope Pro Analyzer tool
`s6_icon.ngc` `s6_vio.ngc`	Spartan-6 family NGC files for the debug cores compatible with the ChipScope Pro Analyzer tool

Back to Top

<component name>/simulation

The simulation directory contains the test bench files for the example design.

Table 4-9: **simulation Directory**

Name	Description
<component name>/simulation	
demo_tb.v[hd]	Test bench file for simulating the example design

Back to Top

/simulation/functional

The functional directory contains functional simulation scripts provided with the core.

Table 4-10: **functional Directory**

Name	Description
<component name>/simulation/functional	
simulate_mti.do	ModelSim macro file that compiles the example design sources, the structural simulation model, and the demonstration test bench then runs the functional simulation to completion
mti_wave.do	ModelSim macro file that opens a Wave window

Back to Top

/simulation/timing

The timing directory contains the timing simulation scripts provided with the core.

Table 4-11: **timing Directory**

Name	Description
<component name>/simulation/timing	
simulate_mti.do	Modelsim macro file that compiles the post place and route netlist of the example design along with standard delay format (SDF) back annotation then runs timing simulation to completion

Back to Top

<component name>/src

The src directory contains the source files related to the Aurora 8B/10B example design.

Table 4-12: **src Directory**

Name	Description
<component name>/src	
<component name>_aurora_lane.v[hd] <component name>_aurora_pkg.vhd (VHDL Only) <component name>_channel_error_detect.v[hd] <component name>_channel_init_sm.v[hd] <component name>_chbond_count_dec.v[hd] <component name>_error_detect.v[hd] <component name>_global_logic.v[hd] <component name>_idle_and_ver_gen.v[hd] <component name>_lane_init_sm.v[hd] <component name>_rx_ll.v[hd] <component name>_rx_ll_pdu_datapath.v[hd] <component name>_sym_dec.v[hd] <component name>_sym_gen.v[hd] <component name>_tx_ll.v[hd] <component name>_tx_ll_control.v[hd] <component name>_tx_ll_datapath.v[hd]	Aurora 8B/10B source files

Back to Top