LogiCORE IP Aurora 8B/10B v3.1 for Virtex-4 FX FPGA

Getting Started Guide

UG173 (v3.1) April 24, 2009

XILINX®

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
04/28/05	1.1	Initial Xilinx release.
01/10/06	2.0	LogiCORE Aurora v2.4 release.
09/12/06	2.5	LogiCORE Aurora v2.5 release.
11/30/06	2.5.1	LogiCORE Aurora v2.5.1 release. Updated Chapter 2, “Licensing the Core.”
03/01/07	2.6	LogiCORE Aurora v2.6 release.
05/17/07	2.7	LogiCORE Aurora v2.7 release. Updated “Generating the Core.” Added “Using ChipScope Pro Cores with the Aurora Core.”
08/22/07	2.7.1	LogiCORE Aurora v2.7.1 release.
10/10/07	2.8	LogiCORE Aurora v2.8 release. Added “Using ChipScope Pro Cores with the Aurora Core” and “Using the Timer-Based Simplex Mode.”
03/24/08	2.9	LogiCORE Aurora v2.9 release. Added ISIM support.
03/24/08	2.9.1	Post-release updates and corrections.

Date	Version	Revision
09/19/08	3.0	LogiCORE Aurora v3.0 release. Moved "System Requirements," page 13 to Chapter 2, "Licensing the Core." Updated "Using ChipScope Pro Cores with the Aurora Core," page 17. Added the *With timer* option to "Usage," page 17 in "Using the Timer-Based Simplex Mode." Updated the steps in "Using the ISE Software Flow to Generate the Aurora Core," page 18. Expanded the "Example Design Directory Structure," page 20.
04/24/09	3.1	LogiCORE Aurora v3.1 release. Tools updated to v1.1. Expanded title and core name to be more descriptive. Updated core to be specific for Virtex-4 FX FPGA. Removed *Simulating the Example Design using ISIM Simulator* and T*esting DUT-BFM Automatic Compliance of Example Design sections*. Updated "System Requirements," page 13, "Generating the Core," page 16, "Using the ISE Software Flow to Generate the Aurora Core," page 18, "Simulating the Example Design," page 18, and "Example Design Directory Structure," page 20.

Table of Contents

About This Guide

The *LogiCORE IP Aurora 8B/10B v3.1 for Virtex-4 FX FPGA Getting Started Guide* provides information about generating a LogiCORE™ IP Aurora8B/10B core using Virtex®-4 FPGA RocketIO™ multi-gigabit transceivers. The information includes customizing and simulating the core using the provided example design, and running the design files through implementation using the Xilinx tools.

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 protocol and core, and gives a step-by-step tutorial on how to generate Aurora designs with the CORE Generator™ software.

Additional Resources

To find additional documentation, see the Xilinx website at:

www.xilinx.com/literature.

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:

www.xilinx.com/support.

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 *Development System Reference 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.
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`**`}`

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;*

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.

Chapter 1

Introduction

This chapter introduces the Aurora core and provides related information, including recommended design experience, additional resources, technical support, and how to submit feedback to Xilinx.

The LogiCORE™ IP Aurora 8B/10B core is a high-speed serial solution based on the Aurora protocol and the Virtex®-4 FPGA RocketIO™ multi-gigabit transceivers. The core is delivered as open-source code and supports both Verilog and VHDL design environments. Each core comes with an example design and supporting modules.

About the Core

The Aurora 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 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 core, users should be familiar with the following:

- Documents located on the Aurora product page: www.xilinx.com/aurora
  - SP002, *Aurora Protocol Specification*
  - UG058, *Aurora Bus Functional Model User Guide* - Contact Auroramkt@xilinx.com
- Documents located on the LocalLink product page: www.xilinx.com/locallink
  - SP006, *LocalLink Interface Specification*
- Xilinx RocketIO Transceiver User Guides:
  - UG076, *Virtex-4 RocketIO Multi-Gigabit Transceiver User Guide*
- ISE® software documentation at www.xilinx.com/ise

Additional Core Resources

For detailed information and updates about the Aurora core, see the following documents, located on the Aurora product page at www.xilinx.com/aurora.

- DS128: *Aurora 8B/10B v3.1 for Virtex-4 FX FPGA Data Sheet*
- UG061: *LogiCORE IP Aurora 8B/10B v3.1 for Virtex-4 FX FPGA User Guide*
- Aurora 8B/10B v3.1 for Virtex-4 FX FPGA Release Notes

Technical Support

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

Xilinx will provide technical support for use of this product as described in the *LogiCORE IP Aurora 8B/10B v3.1 for Virtex-4 FX FPGA User Guide*. Xilinx cannot guarantee timing, functionality, or support of this product for designs that do not follow these guidelines.

Feedback

Xilinx welcomes comments and suggestions about the Aurora core and the accompanying documentation.

Core

For comments or suggestions about the Aurora 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 theAurora core, which you must do before using it in your designs.

System Requirements

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) v10.1 32-bit/64-bit

Software

- ISE® v11.1 software
- Mentor Graphics® ModelSim® v6.4b

Check the release notes for the required Service Pack; ISE Service Packs can be downloaded from www.xilinx.com/support/download.htm.

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

Obtaining Your License

To obtain your license for the Aurora core, do the following:

- Navigate to the Aurora product page: www.xilinx.com/aurora
- Click the Aurora LogiCORE IP link at the bottom of the page
- Click Order and Register

Follow the onscreen instructions to review and electronically sign the Aurora License Agreement and download your license file for the Aurora core.

Installing Your License File

After selecting a license option, an email will be sent to you that includes instructions for installing your license file. In addition, information about advanced licensing options and technical support is provided.

Chapter 3

Quick Start Example Design

The quick start instructions are a step-by-step procedure for generating the Aurora 8B/10B for Virtex®-4 FX FPGA v3.1 core, implementing the core in hardware using the accompanying example design (`aurora_example`), and simulating the core with the provided demonstration testbench (`example_tb`). To learn more about the example design provided with the Aurora core, see the *LogiCORE IP Aurora 8B/10B v3.1 for Virtex-4 FX FPGA User Guide*.

Overview

The quick start example consists of the following components:

- An instance of the Aurora core generated using the default parameters
  - Full-duplex with a single RocketIO™ MGT transceiver
  - Both flow control options
  - LocalLink interface
- A top-level example design with user constraints file (UCF) for an ML423 board
- A demonstration testbench (`example_tb`) to simulate two instances of the example design

The Aurora example design has been tested with XST for synthesis and ModelSim for simulation.

Figure 3-1 shows a block diagram of the default Aurora example design.

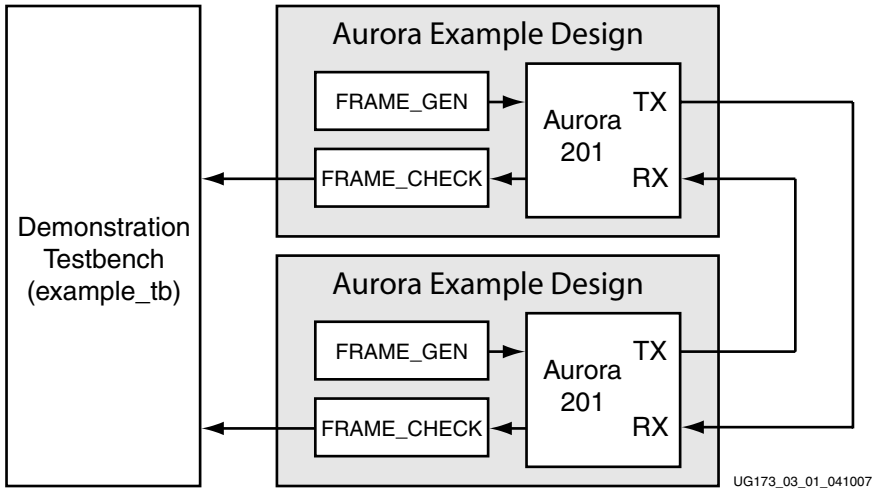


Figure 3-1: **Example Design**

Generating the Core

To generate an Aurora core with default values using the CORE Generator™ software:

1. Start the CORE Generator software.

   For help starting and using the CORE Generator software, see *CORE Generator Help* in the ISE® software documentation at www.xilinx.com/ise

2. Choose File → New Project.
3. Save CORE Generator project file in a specific project directory. This example uses the following location and directory name:

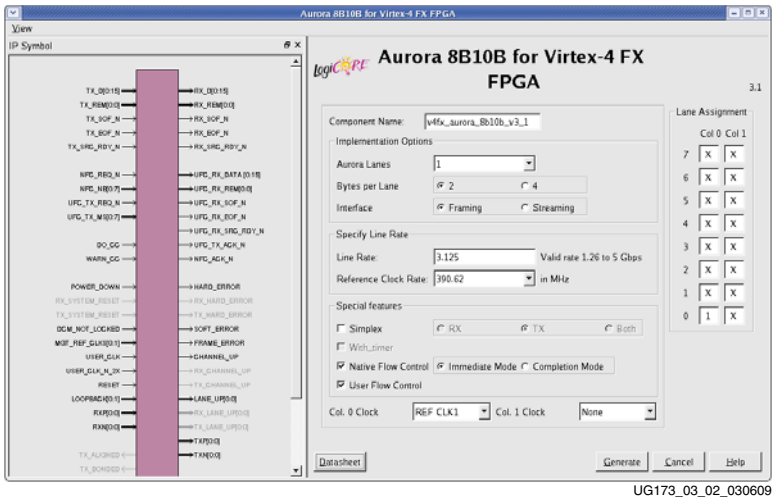
   `/Projects/aurora/`

4. Click OK to create the directory.
5. To set project options:
   - On the Part tab, select a Family and Device that support the Aurora core, such as Virtex4 and XC4VFX60.

     Note: If an unsupported silicon family is selected, the Aurora core appears light grey in the taxonomy tree and cannot be customized. For a list of supported architectures, see the *LogiCORE IP Aurora 8B/10B v3.1 for Virtex-4 FX FPGA User Guide*.
   - Select appropriate package and speed grade.
   - Optionally, on the Generation tab, set the Design Entry pulldown to Verilog.
6. After creating the project, locate the Aurora core in the taxonomy tree under:

   `/Communication_&_Networking/Serial_Interfaces`.

7. 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."



UG173_03_02_030609

Figure 3-2: **CORE Generator Software Aurora Customization Screen**

8. In the Component Name field, enter a name for the core instance. This example uses the name v4fx_aurora_8b10b_v3_1.
9. 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 "Example Design Directory Structure," page 20.

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 v4fx_aurora_8b10b_v3_1\scripts
ms-dos> xilperl make_aurora.pl -win
```

For Linux

```
% cd v4fx_aurora_8b10b_v3_1/scripts
% xilperl make_aurora.pl
```

These commands execute a script that synthesizes, builds, maps, place-and-routes the example design and produces a .bit file. The resulting files are placed in the scripts directory. See the *LogiCORE IP Aurora 8B/10B v3.1 for Virtex-4 FX FPGA User Guide* for information on how to use the `make_aurora.pl` build script to create an ISE software project for the Aurora core.

Using ChipScope Pro Cores with the Aurora Core

Description

The ChipScope™ Pro ICON and ILA cores aid in debugging and validating the design in board. To assist with debugging, these cores are provided along with the Aurora core from the CORE Generator software.

The Aurora example design includes the ICON and ILA cores and they are located in the example top-level module. The user must set the appropriate parameters to enable the core.

Using the Timer-Based Simplex Mode

Description

In simplex mode, the Aurora core RX channel partner communicates to the TX channel partner via sideband signals named RX_RESET, RX_ALIGNED, RX_VERIFY, and RX_BONDED. The Aurora core has been enhanced to include a timer-based simplex mode. This simplex mode does not rely on sideband signals, thereby saving trace routing at the board level, resulting in cost savings. In the timer-based simplex mode, the sideband signals are generated by a set of timers whose values were taken from sample Aurora design simulations. These timer values can be modified according to the Aurora core configuration and user requirements.

Usage

The *With timer* option must be checked for the Simplex design to generate with a built-in timer instead of sideband signals.

Using the ISE Software Flow to Generate the Aurora Core

1. Open ISE 11.1 software.
2. Select File → New Project… to open the New Project Wizard.
3. Enter the Project Name and Project Location.
4. Set the Top-Level Source Type to HDL, and click Next.
5. Select the Device and specify the Preferred Language, and click Next.
6. Click New Source to open the New Source Wizard.
7. Select IP (Coregen & Architecture Wizard). Enter a File name and click Next.
8. Select Communication & Networking → Serial Interfaces → Aurora 8B/10B for Virtex-4 FX FPGA v3.1 and click Next.
9. Click Finish.
10. If necessary, click Yes to create the new directory, and click Next twice to reach the Project Summary.
11. Click Finish to open the CORE Generator software and the LogiCORE IP Aurora 8B/10B for Virtex-4 FX FPGA v3.1 GUI.
12. Provide the necessary parameters for the Aurora configuration (see the *LogiCORE IP Aurora 8B/10B v3.1 for Virtex-4 FX FPGA User Guide*) and click Generate.

    Note: The CORE Generator software has added an XCO file to the source file tree for the project.
13. Add the constraints file from the `ucf` directory.
14. Perform synthesis and implementation steps to generate a `.bit` file.
15. Use IMPACT to transfer the `.bit` file to the target device.

When using the ISE software flow to generate and synthesize the Aurora CORE, it is important to set the Hierarchy Separator value to '/' and the Bus Delimiter value to '[ ]'.

To set these values:

1. Right-click Synthesize - XST in the Process tree and select Properties.
2. Select the Synthesis Options category.
3. Set the Property display level to Advanced.
4. Select '/' in the Hierarchy Separator list.
5. Select '[ ]' in the Bus Delimiter list.
6. Click OK.

Simulating the Example Design

The Aurora 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. See *Simulating Your Design* in the *Synthesis and Verification Design Guide* for ISE 11.1 software for instructions to compile simulation libraries for ISE software.

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

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

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

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

   ```
   <project directory>/v4fx_aurora_8b10b_v3_1/scripts
   ```

2. Set the MTI_LIBS variable:

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

3. Launch the simulation script:

   ```
   modelsim> v4fx_aurora_8b10b_v3_1_example_test.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 for about 50 µs to see the Aurora core power up, followed by Aurora channel initialization and data transfer. Data transfer begins after the CHANNEL_UP signal goes High.

Example Design Directory Structure

This section provides detailed information about the example design files and directory structure generated by the Xilinx CORE Generator software.

<project directory>
Top-level project directory containing documentation; name is user-defined

- <project directory>/cc_manager
  Optional clock compensation module source file in Verilog or VHDL
- <project directory>/clock_module
  Optional clock module source file in Verilog or VHDL
- <project directory>/examples
  Example design source files in Verilog or VHDL
- <project directory>/scripts
  Implementation scripts
- <project directory>/src
  Aurora source files in Verilog or VHDL
- <project directory>/testbench
  Test bench files
- <project directory>/ucf
  User constraints files

The Aurora 8B/10B for Virtex-4 FX FPGA v3.1 core directories and their associated files are defined in the following section.

<project directory>

Caution! The <project directory> contains all the CORE Generator software's project files and documentation.

Table 3-2: **Project Directory**

Name	Description
<project_dir>	
virtex-4fx_aurora_8b10b_readme.txt	Core release notes file
virtex-4fx_aurora_8b10b_ug061.pdf	*LogiCORE IP Aurora 8B/10B v3.1 for Virtex-4 FX FPGA User Guide*
virtex-4fx_aurora_8b10b_gsg173.pdf	*LogiCORE IP Aurora 8B/10B v3.1 for Virtex-4 FX FPGA Getting Started Guide*
virtex-4fx_aurora_8b10b_ds128.pdf	*Aurora 8B/10B v3.1 for Virtex-4 FX FPGA Data Sheet*

Back to Top

<project directory>/cc_manager

The optional cc_manager directory contains the clock compensation module file.

Table 3-3: **cc_manager Directory**

Name	Description
<project_dir>/cc_manager	
<project_dir>_standard_cc_module.v[hd]	Clock compensation module file

Back to Top

<project directory>/clock_module

The optional clock_module directory contains the clock module source file.

Table 3-4: **clock_module Directory**

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

Back to Top

<project directory>/examples

The examples directory contains the example design source files provided with the core.

Table 3-5: **examples Directory**

Name	Description
<project_dir>/examples	
<project_dir>_aurora_example.v[hd] <project_dir>_frame_gen.v[hd] <project_dir>_frame_check.v[hd]	Example design source files

Back to Top

<project directory>/scripts

The scripts directory contains the scripts provided with the core.

Table 3-6: **scripts Directory**

Name	Description
<project_dir>/scripts	
<project_dir>_example_test.do config.csh example_wave.do make_aurora.pl v4_icon.ngc v4_ila.ngc	Implementation script files

Back to Top

<project directory>/src

The src directory contains the Verilog or VHDL design files.

Table 3-7: **src Directory**

Name	Description
<project_dir>/src	
<project_dir>.v[hd] <project_dir>_aurora_lane.v[hd] <project_dir>_aurora_pkg.vhd (VHDL only) <project_dir>_channel_error_detect.v[hd] <project_dir>_channel_init_sm.v[hd] <project_dir>_chbond_count_dec.v[hd] <project_dir>_error_detect.v[hd] <project_dir>_global_logic.v[hd] <project_dir>_idle_and_ver_gen.v[hd] <project_dir>_lane_init_sm.v[hd] <project_dir>_mgt_wrapper.v[hd] <project_dir>_rx_ll.v[hd] <project_dir>_rx_ll_nfc.v[hd] <project_dir>_rx_ll_pdu_datapath.v[hd] <project_dir>_rx_ll_ufc_datapath.v[hd] <project_dir>_sym_dec.v[hd] <project_dir>_sym_gen.v[hd] <project_dir>_tx_ll.v[hd] <project_dir>_tx_ll_control.v[hd] <project_dir>_tx_ll_datapath.v[hd] <project_dir>_ufc_filter.v[hd] <project_dir>_unused_mgt.v[hd] <project_dir>_cal_block_v1_4_1.v[hd]	Aurora 8B/10B for Virtex-4 FX FPGA v3.1 source files

Back to Top

<project directory>/testbench

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

Table 3-8: **testbench Directory**

Name	Description
<project_dir>/testbench	
<project_dir>_example_tb.v[hd] <project_dir>_sim_reset_on_config.vhd (VHDL only)	Test bench file for simulating the example design

Back to Top

<project directory>/ucf

The ucf directory contains the constraints files provided with the core.

Table 3-9: **ucf Directory**

Name	Description
<project_dir>/ucf	
<project_dir>.ucf	Aurora 8B/10B for Virtex-4 FX FPGA v3.1 user constraints file
<project_dir>_aurora_example.ucf	Aurora example design constraints file

Back to Top