LogiCORE Aurora v2.4

Getting Started Guide

UG173 January 10, 2006

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.

Table of Contents

About This Guide

The *LogiCORE Aurora v2.4 Getting Started Guide* provides information about generating a Xilinx LogiCORE™ Aurora core, 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, "Installing and 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 tool.

Additional Resources

For additional information, go to http://www.xilinx.com/support. The following table lists some of the resources you can access from this website or by using the provided URLs.

Resource	Description/URL
Tutorials	Tutorials covering Xilinx design flows, from design entry to verification and debugging http://www.xilinx.com/support/techsup/tutorials/index.htm
Answer Browser	Database of Xilinx solution records http://www.xilinx.com/xlnx/xil_ans_browser.jsp
Application Notes	Descriptions of device-specific design techniques and approaches http://www.xilinx.com/xlnx/xweb/xil_publications_index.jsp?category=Application+Notes

Resource	Description/URL
Data Sheets	Device-specific information on Xilinx device characteristics, including readback, boundary scan, configuration, length count, and debugging http://www.xilinx.com/xlnx/xweb/xil_publications_index.jsp
Problem Solvers	Interactive tools that allow you to troubleshoot your design issues http://www.xilinx.com/support/troubleshoot/psolvers.htm
Tech Tips	Latest news, design tips, and patch information for the Xilinx design environment http://www.xilinx.com/xlnx/xil_tt_home.jsp

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 you enter in a syntactical statement	**`ngdbuild`** *`design_name`*
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 details.
	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}`**

Convention	Meaning or Use	Example

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

Horizontal ellipsis ...	Omitted repetitive material	`allow block block_name loc1 loc2 ... locn;`
Notations	The prefix '0x' or the suffix 'h' indicate hexadecimal notation	A read of address `0x00112975` returned `45524943`h.
	An '_n' means the signal is active low	`usr_teof_n` is active low.

Online Document

The following linking 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 Handbook.*
Blue, underlined text	Hyperlink to a website (URL)	Go to http://www.xilinx.com for the latest speed files.

XILINX®

Chapter 1

Introduction

The Xilinx LogiCORE Aurora core is a high-speed serial solution based on the Aurora protocol and the Xilinx RocketIO multi-gigabit transceiver (MGT). 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.

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.

About the Core

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

- SP002 *Aurora Protocol Specification*
- SP006 *LocalLink Interface Specification*
- Xilinx RocketIO Transceiver User Guides:
  - UG024 for Virtex-II Pro MGTs
  - UG035 for Virtex-II Pro X MGTs
  - UG076 for Virtex-4 MGTs
- ISE documentation
  http://www.xilinx.com/support/sw_manuals/xilinx8/index.htm

Additional Core Resources

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

- *LogiCORE Aurora v2.4 Data Sheet*
- Aurora Release Notes
- *LogiCORE Aurora v2.4 User Guide*

For updates to this document, see the *LogiCORE Aurora v2.4 Getting Started Guide*, also located on the Aurora product page.

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 Aurora v2.4 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

Installing and Licensing the Core

This chapter provides instructions for installing the Aurora core in the CORE Generator tool and how to obtain a free license to use the core.

System Requirements

Windows

- Windows® 2000 Professional with Service Pack 2-4
- Windows XP Professional with Service Pack 1

Solaris/Linux

- Sun Solaris™ 8/9
- Red Hat™ Enterprise Linux 3.0 (32-bit and 64-bit)

Software

- ISE 8.1i with Service Pack 1 (8.1.01i) or later

If necessary, ISE 8.1i Service Packs can be downloaded from http://www.xilinx.com/xlnx/xil_sw_updates_home.jsp?software=8.1i

Installing the Core

You can install the core in two ways: From the CORE Generator IP Updates Installer, which displays a list of compatible updates from which you select the desired core or core update, or by performing a manual installation after downloading the core from the web.

CORE Generator Installation

1. From the CORE Generator main GUI, choose **Tools** → **Updates Installer** to start the Updates Installer.
2. If prompted for a proxy host, contact your administrator to determine the proxy host address and port number you need to get through your firewall.
3. Select **81i_ip_update1** from the list of updates in the Available Packages panel.
4. Click **Add To Install Queue** to add the update ZIP file to the install queue.
5. Do one of the following:
   - If prompted to enter a log-in name and password, enter your Xilinx log-in and password.

- If you are new to Xilinx, click **Create an Account** and follow the instructions to create an account. (After creating an account, you will be redirected to the page to download the core.)

6. Click **Install All Packages from Queue** to download the update.

   After downloading the update, the Updates Installer terminates the CORE Generator session and installs the downloaded archive. After the download is complete, you can restart the CORE Generator.

7. To confirm the installation, check the following file:
   `C:\Xilinx\coregen\install\install_history`.

   Note: This step assumes your Xilinx software is installed in C:\Xilinx.

Manual Installation

1. Close the CORE Generator application if it is running.
2. Download the IP Update ZIP file (Windows) or tar.gz file (UNIX) from the following location and save it to a temporary directory:
   http://www.xilinx.com/xlnx/xil_sw_updates_home.jsp?update=ip&software=8.1i
   - If prompted to enter a log-in name and password, enter your Xilinx log-in and password.
   - If you are new to Xilinx, click Create an Account and follow the instructions to create an account.
3. Do one of the following:
   - For Windows, unpack the ZIP file using WinZip 7.0 SR-1 or later.
   - For UNIX, Xilinx recommends that you unpack the tar.gz file using the UNIX command line utilities gunzip and tar. WinZip and GNU tar are not recommended due to differences in the way they handle files with long path names. Please see Xilinx Answer 11162 for details.
4. Extract the ZIP file (81i_ip_update1.zip) or tar.gz (81i_ip_update1.tar.gz) archive to the root directory of your Xilinx software installation. Allow the extractor utility you use to overwrite all existing files and maintain the directory structure defined in the archive.
5. To verify the root directory of your Xilinx installation, do one of the following:
   - For Windows: Type `echo %XILINX%` from a DOS prompt.
   - For Unix: If you have already installed the Xilinx ISE software, the Xilinx variable defined by your set-up script identifies the location of the Xilinx installation directory. After sourcing the Xilinx set-up script, type `echo $XILINX` to determine the location of the Xilinx installation. Note that you may need system administrator privileges to install the update.
6. Confirm the directory structure in one of the following ways:
   - For Windows:
     <*Xilinx_root_directory*>\coregen\ip\xilinx\network_ip1_h\com\xilinx\ip
   - For UNIX:
     <*Xilinx_root_directory*>/coregen/ip/xilinx/network_ip1_h/com/xilinx/ip

   If you do not see this directory structure, recheck the directory to which you extracted the archive to and try again.

7. Restart the CORE Generator. During start-up, the CORE Generator automatically detects new or updated versions of IP available in your installation and lets you specify which IP customizers (cores) will be visible in your current CORE Generator project.
8. Choose one of the following options:
   - Display only the latest versions for all cores in the catalog
   - Update the catalog view to add only new cores to the display
   - Make a Custom selection of visible cores in your current project
9. Determine if the installation was successful by verifying that the new cores are visible in the CORE Generator GUI.
10. If the new cores are not visible, return to Step 6 to verify the directory structure. If the directory structure is incorrect, return to Step 4 to verify that the directory was extracted to the correct location.

    For additional assistance installing the IP Update, contact the Xilinx Technical Support.

Obtaining Your License

To obtain your license for the Aurora core, perform the following steps:

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

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 an Aurora core, implementing the core in hardware using the accompanying sample design, and simulating the core with the provided sample testbench. To learn more about the sample design provided with the Aurora core, see the *LogiCORE Aurora v2.4 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
  - Both flow control options
  - LocalLink interface
  - Virtex-II Pro target device
- A top-level sample design with user constraints file (UCF) for an ML321 board
- A demonstration testbench to simulate two instances of the sample design

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

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

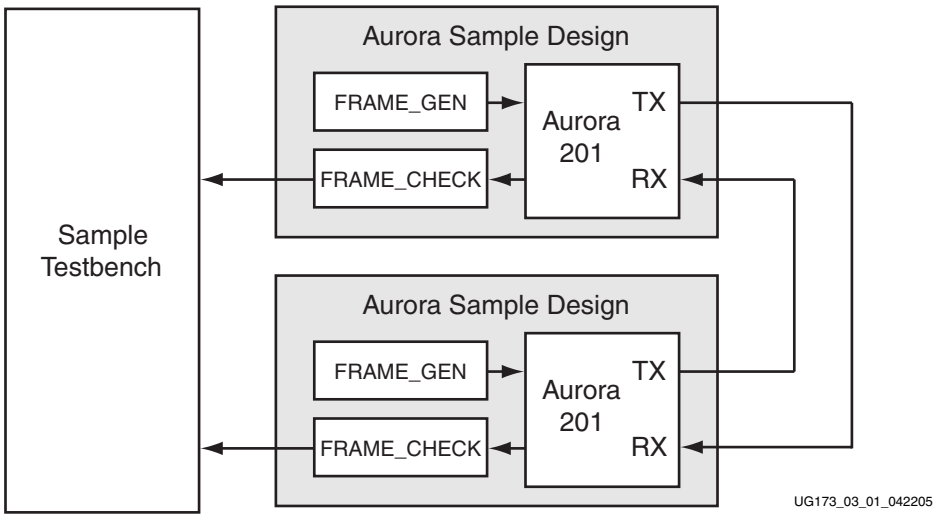


Figure 3-1: **Sample Design**

Generating the Core

To generate a sample Aurora core with default values using the Xilinx CORE Generator tool, do the following:

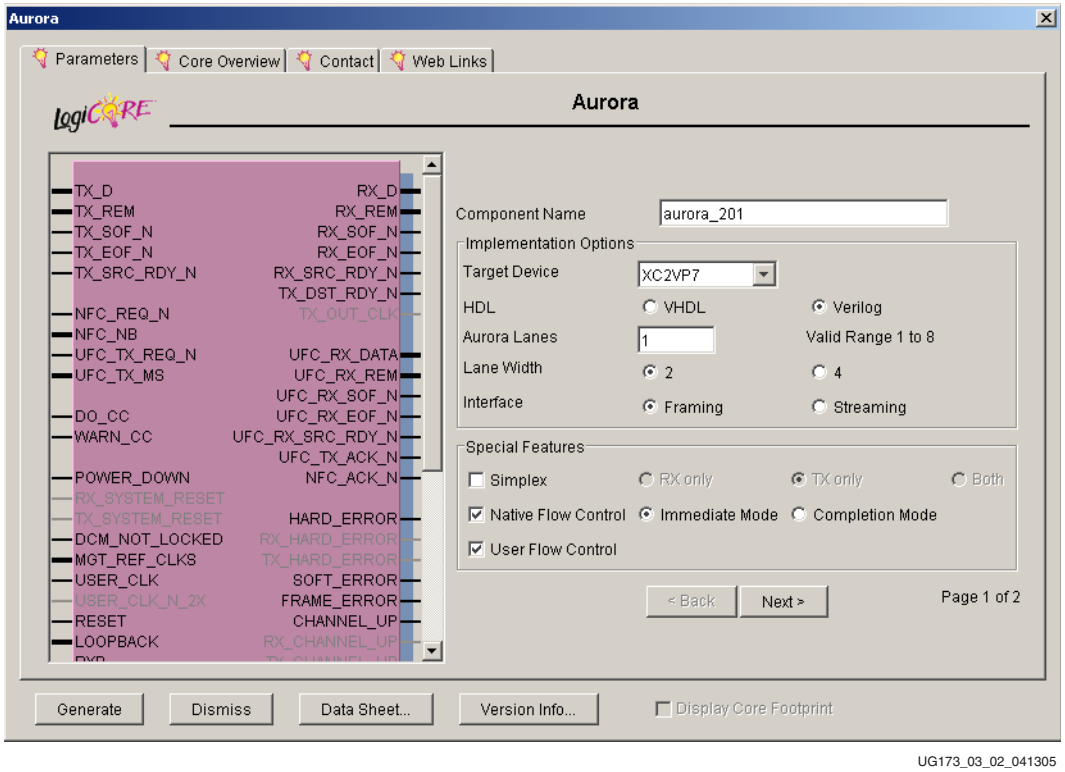
1. Start the CORE Generator tool.

   For help starting and using the CORE Generator tool, see the Xilinx CORE Generator Guide, available from the ISE documentation.
2. Choose **File** → **New Project**.
3. Type a location and a directory name. This example uses the following location and directory name:

   /Projects/aurora/201_v2_4
4. Do the following to set project options:
   - From **Target Architecture**, select an FPGA family that supports the Aurora core, for example, **Virtex2P**

     Note: If an unsupported silicon family is selected, the Aurora core does not appear in the taxonomy tree. For a list of supported architectures, see the *LogiCORE Aurora v2.4 User Guide*.
   - No further project options need to be set.
5. After creating the project, locate the Aurora core 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 reports an error. See Chapter 2, "Installing and Licensing the Core."

   Note: Ignore the License Status warning that pops up. Aurora *is* delivered as source code.



UG173_03_02_041305

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

7. In the Component Name field, enter a name for the core instance. This example uses the name **aurora_201**.
8. Optionally, set HDL to **VHDL**.
9. Click **Generate**.

   The core and its supporting files, including the sample design, are generated in the project directory. For detailed information about the sample design files and directories see *Chapter 3, Customizing the Aurora Core* in the *LogiCORE Aurora v2.4 User Guide*.

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_201\scripts
ms-dos> xilperl make_aurora.pl -sample -m -p -b
```

For UNIX

```
unix-shell% cd aurora_201/scripts
unix-shell% xilperl make_aurora.pl -sample -m -p -b
```

These commands execute a script that synthesizes, builds, maps, place-and-routes the sample design and produces a bitmap file. The resulting files are placed in the scripts directory. See the *LogiCORE Aurora v2.4 User Guide* for information on how to use the make_aurora.pl build script to create an ISE project for the Aurora core.

Simulating the Example Design

The Aurora core provides a quick way to simulate and observe the behavior of the core using the provided sample 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 *Chapter 6, Simulating Your Design* in the *Synthesis and Verification Design Guide* for ISE 8.1i for instructions on how to compile simulation libraries for ISE.

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

HDL	Library	Source Directories
Verilog	UNISIMS_VER	<*Xilinx dir*>/verilog/src/unisims <*Xilinx dir*>/smartmodel/nt/wrappers/mtiverilog
VHDL	UNISIM	<*Xilinx dir*>/vhdl/src/unisims <*Xilinx dir*>/smartmodel/nt/wrappers/mtivhdl Compile files in the following order: unisim_VPKG.vhd unisim_VCOMP.vhd unisim_VITAL.vhd unisim_SMODEL.vhd smartmodel_wrappers.vhd

The Aurora core provides a command line script to simulate the sample 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>*/aurora_201/scripts**

2. Set the MTI_LIBS variable:

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

3. Launch the simulation script:

   `modelsim>` **`do sample_test.do`**

The ModelSim script compiles the sample 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 20 microseconds to see the FPGA power up, followed by Aurora channel initialization and data transfer. Run the simulation for about 180 microseconds for Virtex-4 cores. Data transfer begins after the CHANNEL_UP signal goes high.

Example Design Hierarchy

The hierarchy for the design used in this quick start example is as follows:

```
sample_tb
|___aurora_sample
    |___aurora_201
    |   |___aurora_lane
    |   |   |___lane_init_sm
    |   |   |___chbond_count_dec
    |   |   |___sym_gen
    |   |   |___sym_dec
    |   |   |___error_detect
    |   |
    |   |___phase_align
    |   |___global_logic
    |   |   |___channel_init_sm
    |   |   |___idle_and_ver_gen
    |   |   |___channel_error_detect
    |   |
    |   |___tx_ll
    |   |   |___tx_ll_datapath
    |   |   |___tx_ll_control
    |   |
    |   |___rx_ll
    |       |___rx_ll_nfc
    |       |___ufc_filter
    |       |___rx_ll_pdu_datapath
    |       |___rx_ll_ufc_datapath
    |
    |___standard_cc_module
    |___frame_gen
    |___frame_check
```