Alliance Series 3.1i Quick Start Guide

Alliance Series 3.1i Quick Start Guide — 0401886 *Printed in U.S.A.*

About This Manual

This manual provides an overview of the Alliance Series 3.1i Software, including a basic tutorial. This manual is geared towards the user who has already installed their software and online documentation, and set up their user environment variables.

Other publications you can consult for related information include the *Design Manager/Flow Engine Guide* and the *Alliance Release Notes and Installation Guide*.

Note This Xilinx software release is certified as Year 2000 compliant.

Manual Contents

This manual covers the following topics.

- Chapter 1, "Introduction," provides general information on the Xilinx software.
- Chapter 2, "Implementation Tools Tutorial,"provides a basic tutorial to illustrate the Xilinx design flow.
- Appendix A, "Alliance FPGA Express Interface Notes,"describes installing and using FPGA Express for the Alliance 3.1i software.
- Appendix B, "Configuring Xprinter,"provides information on how to print from Xilinx graphical interface programs on the workstation.
- "Glossary of Terms,"provides definitions of basic terminology used in the Quick Start Guide.

Additional Resources

For additional information, go to `http://support.xilinx.com`. The following table lists some of the resources you can access from this Web site. You can also directly access these resources using the provided URLs.

Resource	Description/URL
Tutorials	Tutorials covering Xilinx design flows, from design entry to verification and debugging `http://support.xilinx.com/support/techsup/tutorials/index.htm`
Answers Database	Current listing of solution records for the Xilinx software tools Search this database using the search function at `http://support.xilinx.com/support/searchtd.htm`
Application Notes	Descriptions of device-specific design techniques and approaches `http://support.xilinx.com/apps/appsweb.htm`
Data Book	Pages from *The Programmable Logic Data Book*, which contain device-specific information on Xilinx device characteristics, including readback, boundary scan, configuration, length count, and debugging `http://support.xilinx.com/partinfo/databook.htm`
Xcell Journals	Quarterly journals for Xilinx programmable logic users `http://support.xilinx.com/xcell/xcell.htm`
Technical Tips	Latest news, design tips, and patch information for the Xilinx design environment `http://support.xilinx.com/support/techsup/journals/index.htm`

Conventions

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

Typographical

The following conventions are used for all documents.

- `Courier font` indicates messages, prompts, and program files that the system displays.

  ```
  speed grade: - 100
  ```

- **`Courier bold`** indicates literal commands that you enter in a syntactical statement. However, braces "{ }" in Courier bold are not literal and square brackets "[ ]" in Courier bold are literal only in the case of bus specifications, such as bus [7:0].

  `rpt_del_net=`

  `Courier bold` also indicates commands that you select from a menu.

  `File` → **`Open`**

- *Italic font* denotes the following items.

  - Variables in a syntax statement for which you must supply values

    `edif2ngd` *`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 "[ ]" indicate an optional entry or parameter. However, in bus specifications, such as bus [7:0], they are required.

  edif2ngd *[option_name] design_name*

- Braces "{ }" enclose a list of items from which you must choose one or more.

  lowpwr ={**on**|**off**}

- A vertical bar " | " separates items in a list of choices.

  lowpwr ={**on**|**off**}

- A vertical ellipsis indicates repetitive material that has been omitted.

  ```
  IOB #1: Name = QOUT'
  IOB #2: Name = CLKIN'
  .
  .
  .
  ```

- A horizontal ellipsis "...." indicates that an item can be repeated one or more times.

  `allow block` *block_name loc1 loc2locn;*

Online Document

The following conventions are used for online documents.

- Red-underlined text indicates an interbook link, which is a cross-reference to another book. Click the red-underlined text to open the specified cross-reference.

- Blue-underlined text indicates an intrabook link, which is a cross-reference within a book. Click the blue-underlined text to open the specified cross-reference.

Contents

Chapter 2 Implementation Tools Tutorial

Appendix A Alliance FPGA Express Interface Notes

Appendix B Configuring Xprinter

Chapter 1

Introduction

This chapter contains general information on the Alliance 3.1i Software, including supported devices, design flow, documentation, installation, and licensing.

Note Complete software installation information is located in the *Alliance 3.1i Release Notes and Installation Guide.*

For complete information on the new features in this software release, refer to the "What's New" file included on your Alliance Implementation Tools CD-ROM. You can access this file through **`Start`** → **`Programs`** → **`Xilinx Alliance`**.

For detailed information on configuring your third-party interface tools (such as Mentor Graphics, Synopsys, and Viewlogic) to work with the Alliance software flow, refer to the appropriate online interface guide at `http://support.xilinx.com/support/sw_manuals/3_1i/index.htm`. For technical tips on the third-party interface tools, see `http://support.xilinx.com/support/techsup/journals/index.htm`.

This chapter contains the following sections.

- "Support and Services"
- "Architecture Support"

Supported Netlists

You must use the Xilinx Unified Libraries to create your designs. Refer to the *Libraries Guide* for a list of components. The following table lists the netlist formats supported by the Xilinx software.

Netlist Format	Variations
EDIF	SEDIF, EDN, EDF, EDIF
XNF	SXNF, XFF, XTF, XNF

Operating System Compatibility

The software in this release supports several operating systems, as shown in the following table.

Table 1-1 Operating System Compatibility

Platform Type	Version Number
Solaris®	Solaris 2.6 / 2.7
HP Series 9000	HP-UX 10.20 / 11.0
Windows NT®	NT 4.0
Windows®	Windows 98 / 2000

Xilinx Design Flow

This section provides flow diagrams that illustrate the processing steps and flow of files in and out of the Design Manager for FPGAs and CPLDs.

Note For complete information on using the Design Manager, see the *Design Manager/Flow Engine Guide*.

FPGA Design Flow

The following figure shows the processing steps and flow of files in and out of the Design Manager for FPGAs.

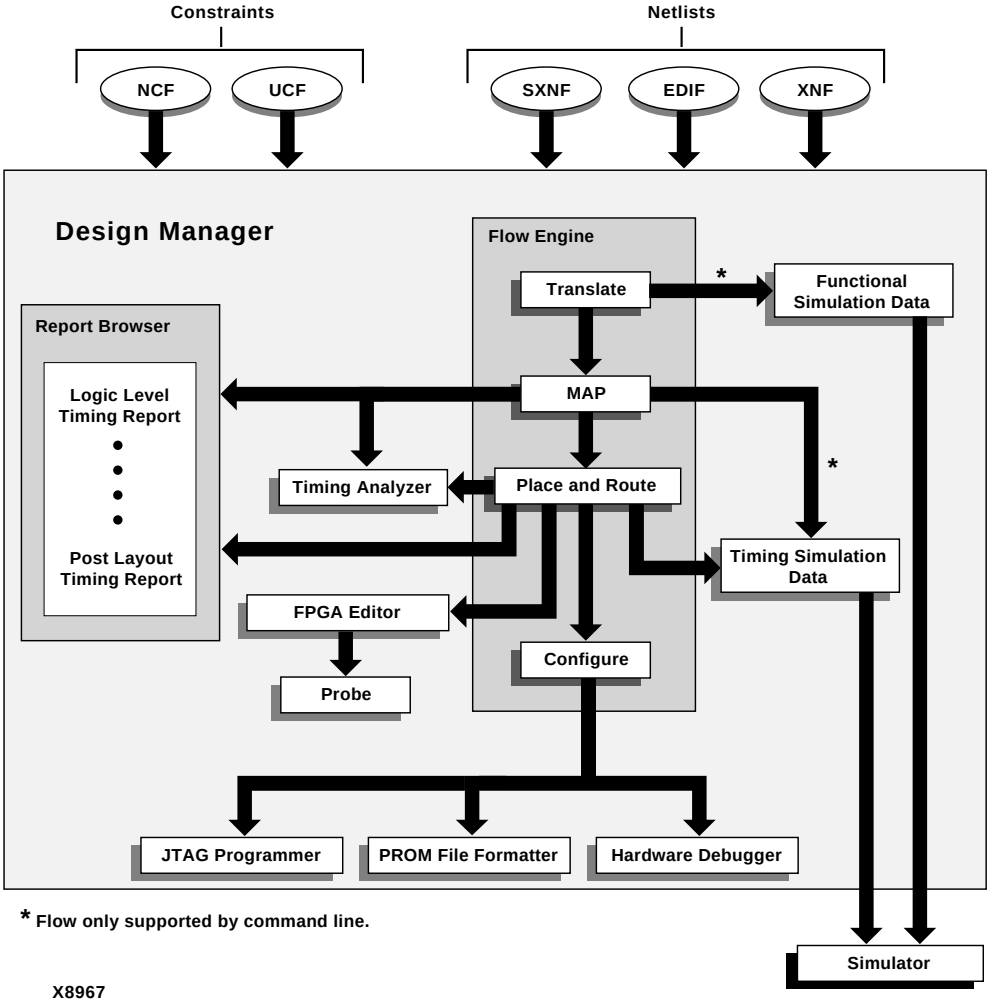

Figure 1-1 FPGA Design Flow

The following figure is a more detailed look at the various programs invoked during the FPGA design implementation process.

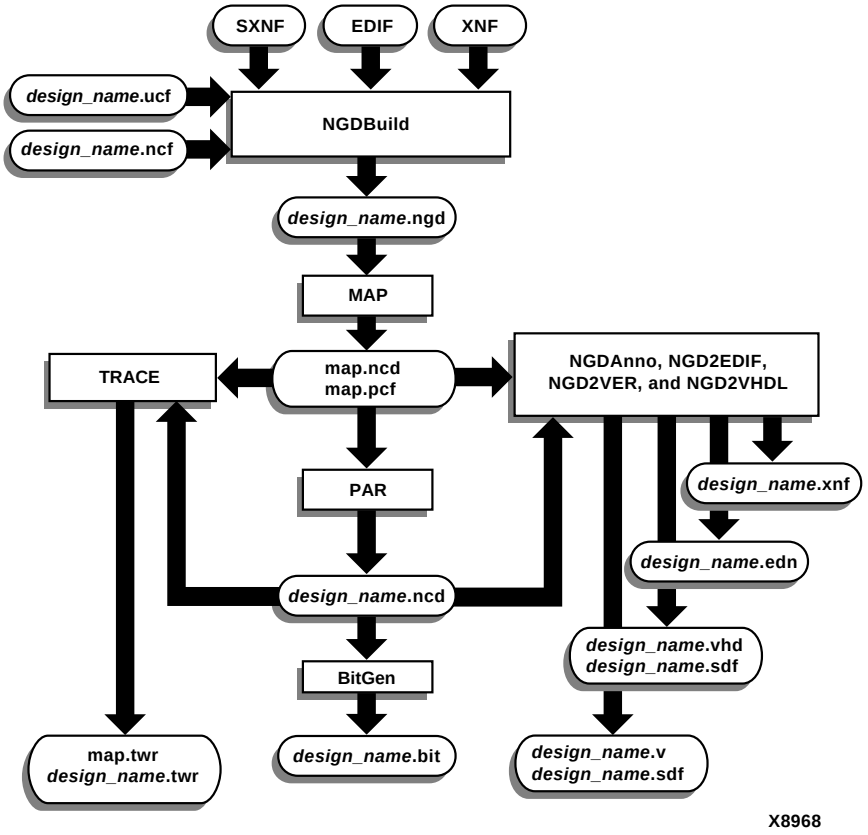

Figure 1-2 FPGA Detailed Design Flow

CPLD Design Flow

The following figure shows the processing steps and flow of files in and out of the Design Manager for CPLDs.

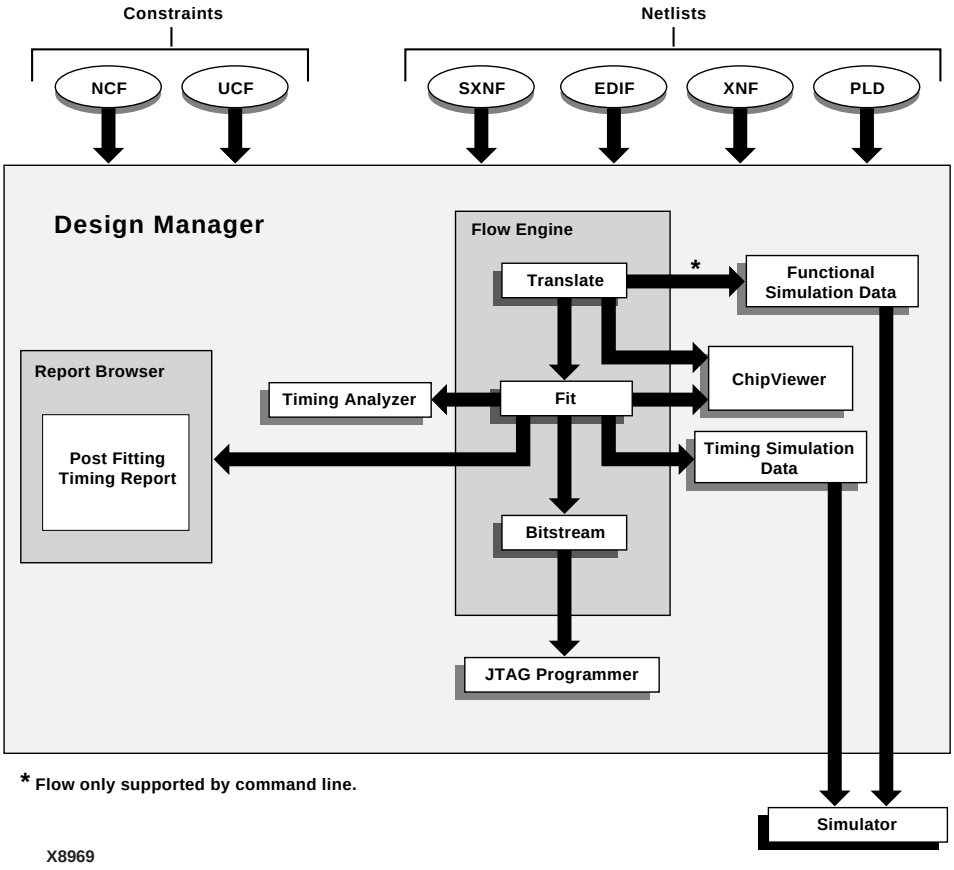

Figure 1-3 CPLD Design Flow

The following figure is a more detailed look at the various programs invoked during the CPLD design implementation process.

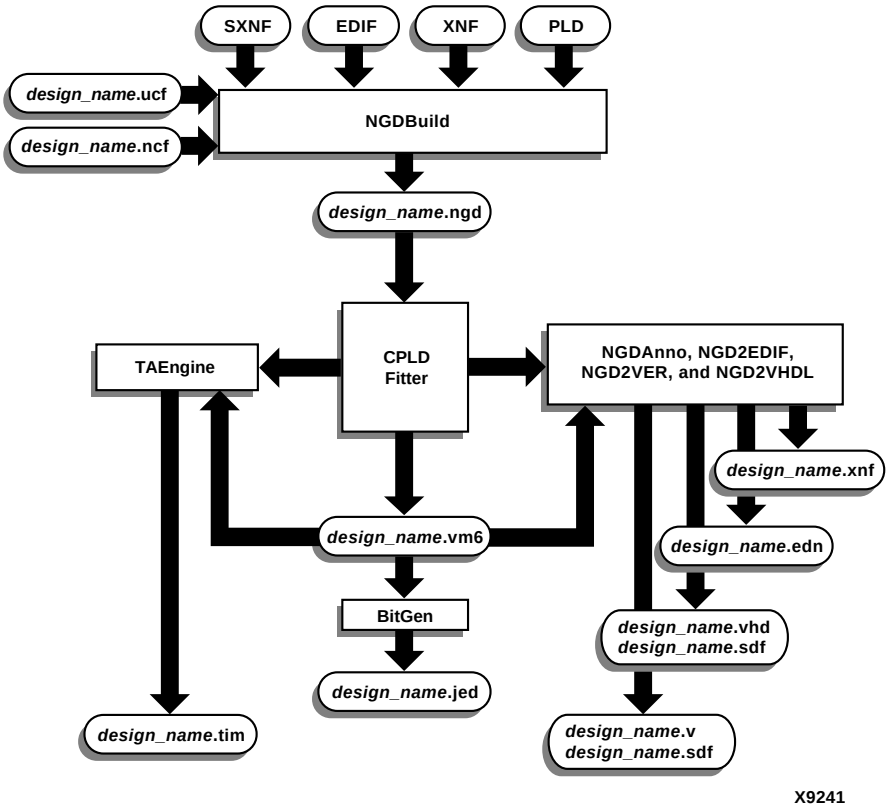

Figure 1-4 CPLD Detailed Design Flow

Xilinx Development System Tools and Features

This section lists the tools and the main features of the Xilinx software. The tutorial in this manual provides a brief overview of how to use these software tools.

For detailed information on using the following tools, refer to the appropriate online software manual.

Table 1-2 Xilinx Software Tools

Feature	Description
Design Manager	Top level software module in the Xilinx Development System. The Design Manager provides access to all the tools you need to read a file from a design entry tool and implement it in a device.
Flow Engine	Displays and executes all the steps needed to implement a Xilinx design, including translating design netlists; mapping logic to CLBs; placing and routing designs; creating a configuration file for downloading to a device; creating static timing reports; and creating timing simulation netlists in VHDL (Vital), Verilog, EDIF, or XNF.
LogiBLOX	Graphical tool used to create high-level modules, such as counters, shift registers, and multiplexers.
CORE Generator	The CORE Generator tool generates and delivers parameterizable cores optimized for Xilinx devices.
Floorplanner	Graphical tool used to control the placement of your design into a target FPGA using a “drag and drop” paradigm with the mouse pointer.
Constraints Editor	Graphical tool used after running NGDBuild to add timing constraints and I/O pin locations.
FPGA Editor	Graphical tool used to display and configure your designs before or after placing and routing.
Hardware Debugger	Used to download your design to a device, verify the downloaded configuration, and display the internal states of the programmed device.
PROM File Formatter	Creates files for serial or byte-wide configuration PROMs. Three formats are available: MCS, EXO, and TEK. The HEX format is also supported for microprocessor-based configuration.

Table 1-3 Xilinx Software Features

Feature	Description
Timing Specification Performance	Xilinx software supports timing-driven placement and routing.
Multi-Pass PAR	The place and route (PAR) software allows multiple place and route iterations on a single machine, a UNIX™ network, or on multiple machines running in parallel. This feature provides optimum performance and efficiency, utilizing CPU time to achieve faster design results.
Re-Entrant Routing	Re-entrant routing skips placement and routes your design. Routing begins with the existing placement and routing left in place.
Guide for Incremental Design Changes	You can select a previously mapped, routed, or fitted implementation revision to use as a guide for implementation.

Software Manuals and Online Help

Xilinx provides software manuals and online help for its graphical user interface (GUI) tools and associated EDA software interfaces. You can access the online help with the **`Help`** → **`Help Topics`** menu option in each software tool. The following sections provide information on accessing and using the online software manuals.

Online Software Manuals

Note Complete directions for installing the online manuals and the Alliance Implementation software are located in the *Alliance 3.1i Release Notes and Installation Guide.*

The software manuals are provided on the Web and on your Documentation CD-ROM. You can install the manuals locally, read them from the CD-ROM, or read them on the Xilinx Web site. To view the online documentation, you must use a Java™ compatible internet browser. The online manuals include a powerful search function, as well as easy print options.

Note For best performance, Xilinx recommends that you use Netscape Navigator™ version 4.5 (or higher) or Microsoft Internet Explorer™ version 5.0 (or higher) browser. You can install the Netscape browser from your software CD-ROM.

Software Manuals on the Web

The 3.1i software manuals are accessible from the Xilinx support Web site at the following location.

`http://support.xilinx.com/support/sw_manuals/3_1i/index.htm`

Online Help for Software Manuals

Online help instructions for reading, browsing, and searching the online manuals are available through the Web browser interface. Click the **Help** button in the upper left-hand corner of the browser window to access the help topics.

Printing Software Manuals

You can print the software manuals with the graphics and text inline. PDF format files of each of the software manuals are provided on the 3.1i Software Documentation CD-ROM and at `http://support.xilinx.com/support/sw_manuals/3_1i/index.htm`.

Printing PDF Files

To access and print using the PDF files, use the following steps.

1. Verify that you have Adobe Acrobat Reader (version 3.0 or above) installed on your network or local area.

   You can install Acrobat from the Alliance 3.1i Implementation tools software CD-ROM by selecting the Core Generator option, or from the Documentation CD.

2. Start Acrobat Reader

   - UNIX users

     Run the following command to start this tool.

     /usr/bin/*path_to_directory***/acrobat**

     Path_to_directory is the directory where your Acrobat program files are located.

   - PC users

     Select **Start** → **Programs** → **Acrobat Reader**

3. Insert the Xilinx 3.1i Software Documentation CD-ROM into your drive.
4. Access your CD-ROM directory.
   - UNIX users

     Enter the following command.

     /usr/bin/*path_to_CD-ROM_directory*

     Path_to_CD-ROM_directory is your mounted CD-ROM drive.

     Enter the **ls** command to view the contents of the CD-ROM directory.
   - PC users

     Select **Start** → **Programs** → **Windows Explorer** and select your CD-ROM drive to display the contents of the CD-ROM.
5. Open the Print directory or folder from the CD-ROM contents. This directory contains the PDF files.
6. Select a book and open it in Acrobat.

   The following table lists the book titles and corresponding PDF file names.

Table 1-4 List of Alliance 3.1i Software Manuals

Manual Title	PDF File Name
Xilinx / Concept-HDL Interface Guide	docchdl.pdf
Constraints Editor Guide	cst_edit.pdf
CORE Generator Guide	coregen.pdf
CPLD Schematic Design Guide	sdg_alli.pdf
CPLD Synthesis Design Guide	syn_cpld.pdf
Design Manager / Flow Engine Guide	dmfe.pdf
Development System Reference Guide	dev_ref.pdf
FPGA Editor Guide	fpedit.pdf
Floorplanner Guide	fplan.pdf
Hardware Debugger Guide	hdebug.pdf
Hardware User Guide	huguide.pdf

Table 1-4 List of Alliance 3.1i Software Manuals

Manual Title	PDF File Name
JTAG Programmer Guide	jtag.pdf
Libraries Guide	libguide.pdf
LogiBLOX Guide	lblox.pdf
Mentor Graphics Interface Guide	mentor.pdf
PROM File Formatter Guide	prom_fmt.pdf
Alliance Series 3.1i Quick Start Guide	docaqsg.pdf
Synthesis and Simulation Design Guide	gensim.pdf
Xilinx/Synopsys Interface Guide	xsi_int.pdf
Timing Analyzer Guide	timing.pdf
Viewlogic Interface Guide	vlifg.pdf

7. Print the book using the **File** → **Print** command from the Adobe Acrobat reader window.

Printing from the Online Document Viewer

You can print individual pages of the software manuals directly from your internet browser window. For example, if you are using a Netscape browser, use the **File** → **Print Frame** menu option in the browser window. (Make sure you clicked in the right-hand book view frame to select it for printing.)

Xilinx recommends that you use the online books for quick information access and searching, and the PDF files for best print quality. Many of the graphics in the Web-based manuals are not inline and will not print automatically. They are also sized for optimal online viewing and may not fit on a printed page.

If you do not have access to your Documentation CD-ROM, you can also access the software manuals in PDF format on the Web. You can FTP each manual to your local area.

Note This process is slower than accessing the PDF files directly from your CD-ROM.

EDA and Third Party Interface Support

The Alliance 3.1i software supports various third party interfaces. For the most current information on the latest vendor version support in the Alliance EDA partner program, refer to `http://www.xilinx.com/programs/alliance/alligen.htm`.

Software manuals for EDA interface users are provided on your software documentation CD-ROM. You can also access the manuals on the Web, from support.xilinx.com.

Software Installation and Licensing

Note Complete software installation instructions are located in the *Alliance 3.1i Release Notes and Installation Guide.*

You must register your software before you can complete the installation. Before you register, make sure you have the serial number from your software package available. You can register by telephone, fax, e-mail, or online.

Support and Services

This section provides information for contacting your technical support and customer service representatives.

Technical Support

If you experience problems with your software installation or operation, you can look for solutions and answers at `http://support.xilinx.com`. This support page contains thousands of online technical solutions and product information for Xilinx software and devices. The Xilinx Answers Database is updated daily with the latest patches, problem resolutions, application notes, and data sheets. You can receive immediate answers 24 hours a day.

If you cannot find your answers at `http://support.xilinx.com`, open a support case on the Web to access Xilinx application engineers worldwide. You can also use telephone hotlines, e-mail, or fax to open a support case.

If you need additional support, contact the Xilinx Technical Support hotline by phone or fax.

Note When e-mailing or faxing inquiries, provide your complete name, company name, and phone number. Also, provide a complete problem description including your design entry software and design stage. In North America, call 1-800-255-7778 if you cannot open a support case on the Web.

Location	Telephone	Electronic Mail	Facsimile (Fax)
United Kingdom	44-870-7350-610	ukhelp@xilinx.com	44-870-7350-620
France	33-1-3463-0100	frhelp@xilinx.com	33-1-3463-0959
Germany	49-89-93088-130	dlhelp@xilinx.com	49-89-93088-188
Japan	local distributor	jhotline@xilinx.com	local distributor
Korea	local distributor	korea@xilinx.com	local distributor
Hong Kong	local distributor	hongkong@xilinx.com	local distributor
Taiwan	local distributor	taiwan@xilinx.com	local distributor

Customer Service

Refer to the Xilinx Web site at `http://support.xilinx.com/support/custserv.htm` for customer service information. The customer service toll-free number for North America is 1-800-624-4782.

Architecture Support

The software supports the following architecture families in this release. Refer to the Programmable Logic Data Book for more information on these devices. The online version of the Data Book is at `http://www.xilinx.com/partinfo/databook.htm`. For updated information regarding speed grades and package support, search the Xilinx Answers Database and the latest Application Notes. You can search the technical documentation at `http://www.xilinx.com/support/searchtd.htm`.

- Spartan™ / XL / -II
- Virtex™ / -E / -II
- XC9500™ / XL / XV
- XC4000™E / L / EX / XL / XV / XLA
- XC3000™A / L

- XC3100™A / L
- XC5200™

Chapter 2

Implementation Tools Tutorial

This chapter provides a step-by-step tutorial that covers many functions of the Alliance 3.1i implementation tools. This tutorial is a good way to learn how the Alliance design flow works with basic designs. For detailed information on using the Xilinx Design Manager, see the *Design Manager/Flow Engine Guide*.

Note An updated version of this tutorial will be available after May 21, 2000 from the Xilinx Support web site as well as on the AppLINX CD. The web site location is `http://support.xilinx.com/support/techsup/tutorials/index.htm`. Contact your local sales representative to obtain a copy of the AppLINX CD.

For additional information on the Xilinx tools, see `http://www.support.xilinx.com/support/training/training.htm` and `http://www.support.xilinx.com/support/techsup/journals/index.htm`.

This chapter contains the following sections.

Installing the Tutorial Files

This tutorial demonstrates the Alliance design implementation tools flow. The front end design has already been compiled for you in an EDA interface tool and is described by an EDIF Netlist File (EDF). For a listing of EDA Interface tutorials, see the Xilinx Support area at `http://support.xilinx.com/support/techsup/tutorials/index.htm`.

This tutorial passes an input netlist from the front end tool to the back-end Alliance 3.1i design implementation tools, and incorporates placement constraints through a User Constraints File (UCF). Timing constraints are added with the Constraints Editor tool.

The tutorial design is named stopwatch and is designed to perform like a track coach's stopwatch. There are two inputs to the system, RESET and SRTSTP, and the configuration clock on the device is used as a ten Hz clock signal. Three seven bit outputs are generated by this system for output to three seven segment LED displays.

Before proceeding to Step 1 in the tutorial, create a working directory with the tutorial files as follows.

1. Create an empty working directory named Watch.
2. Copy the following files from the $XILINX/tutorial/qstart/ directory or from `http://support.xilinx.com/support/techsup/tutorials/` to your newly created working directory.

   Note In order for the $XILINX/tutorial/qstart directory to be present in your root Xilinx directory, you must first install the Userware Tutorial files from the Alliance Series 3.1i design implementation tools CD-ROM.

File Name	Description
stopwatch.edn	Input netlist file (EDIF)
tenths.edn	Coregen Counter netlist file (EDIF)
stopwatch.ucf	User constraints file

Step 1: Creating an Implementation Project

The design implementation tools are organized under a single program called the Design Manager. The Design Manager helps you manage the design flow process by keeping track of design versions and the implementation revisions within each version. The Design Manager also provides access to the entire suite of Xilinx implementation tools needed to complete a design.

While the Design Manager manages your design, the Flow Engine implements it. The Flow Engine is closely integrated with the Design Manager and shares many of the same menus and dialog boxes.

Use the following steps to create an implementation project.

1. On a workstation, enter the following to start the Design Manager.

   ```
   xilinx &
   ```

   On a PC, select **Start** → **Programs** → **Xilinx** → **Design Manager** to start the Design Manager.

   When you open the Design Manager for the first time, you must create a new project for your design. A project includes all design versions, implementation revisions, reports, and any data created while you work with a design.

   The Design Manager graphically displays information about these items in the project view. When you create a new project, you specify a design to open and a directory for the project. You can create as many projects as you want, but you can only work with one at a time.

2. To create a new implementation project for the tutorial design, select **`File`** → **`New Project`** from the Design Manager menu. The New Project dialog box appears.

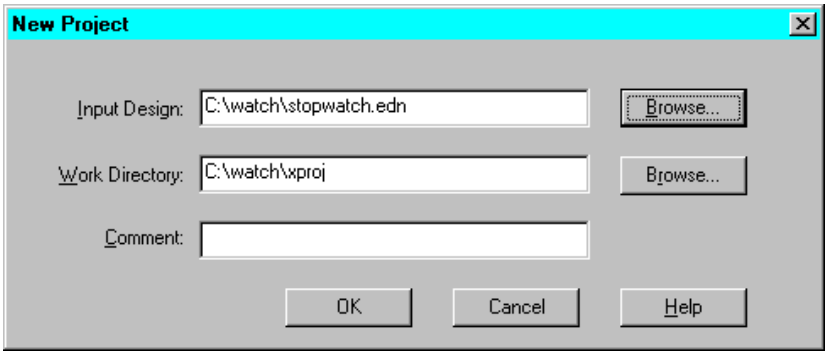

Figure 2-1 New Project Dialog Box

This dialog box contains the following fields.

Table 2-1 New Project Dialog Box Fields

Field	Description
Input Design	Top level netlist file containing the design definition
Work Directory	Directory used to store the implementation data created as the design is compiled
Comment	Use this field to enter any optional notation for the design

3. To specify your input design, click **Browse** next to the Input Design field to display the Browse dialog box.

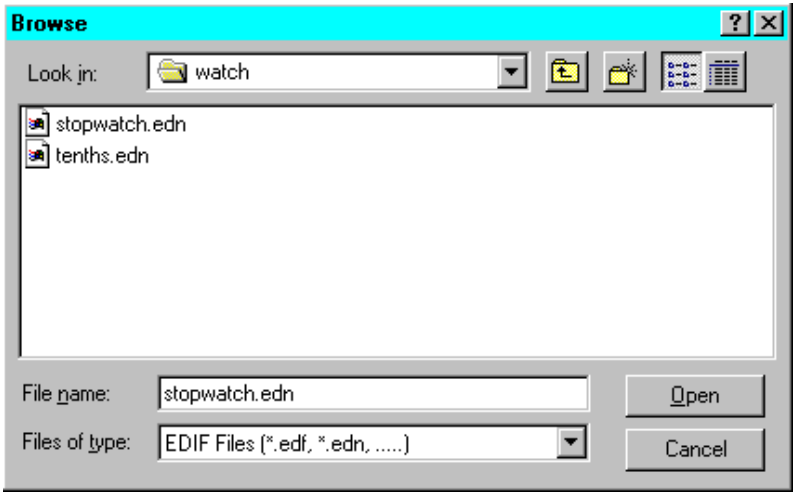

Figure 2-2 Browse Dialog Box

4. Select the appropriate file type from the pull-down list in the Files of Type field. For the tutorial design, EDIF is selected.

5. Select the stopwatch.edn file. The file name appears in the File Name field. Click **Open**.

   The Browse dialog box closes and the New Project dialog box is updated to include the specified input netlist. By default, the Work Directory field is set to the directory containing the input design. If preferred, you can set this to another directory. Because the files were previously copied to the Watch directory, this directory is used for the implementation project and resulting output files.

6. In the Comment field, enter the following.

   ```
   -tutorial
   ```

7. Click **OK** to close the New Project dialog box. The New Version dialog box appears.

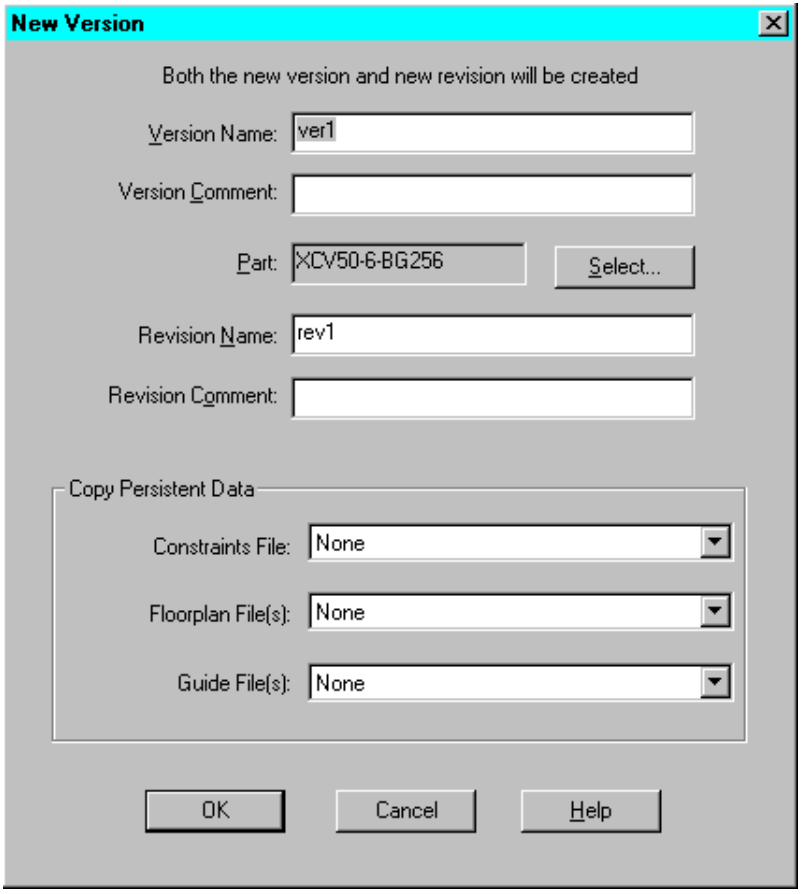

Figure 2-3 New Version Dialog Box

When initially creating a project, the New Version dialog box automatically appears to allow you to enter the information necessary to define the new design version. Also, if your input netlist changes as a result of changes made in the front-end tool, you are prompted by the Design Manager to generate a new design version.

Once a design version is created, you can try different implementation strategies on your design. The data associated with each of these implementation strategies is called an implementation revision. Because a new implementation revision is automatically created when you create a new version, the Version Name and Revision Name fields are already defined in the New Version dialog box.

The Version Name field shows ver1 as the default version, and the Revision Name field shows rev1 as the default revision. Comments on the options and strategies can be entered in the Version and Revision Comment fields.

8. The Part field automatically contains a part number if you specified the target device in your design entry tool. For the tutorial design, this field is empty and must be defined. Click **Select** to display the Part Selector dialog box.

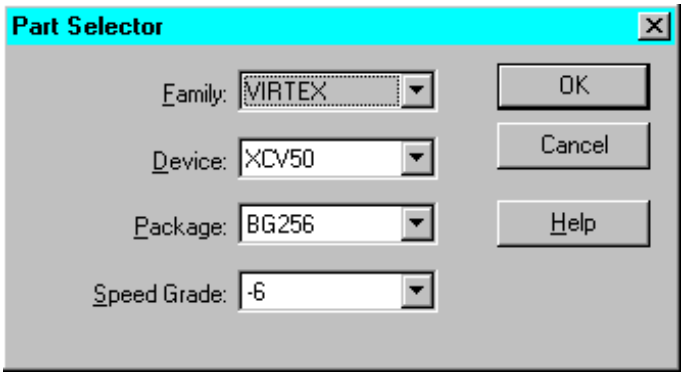

Figure 2-4 Part Selector Dialog Box

9. Use the pull-down lists for the fields in the Part Selector dialog box to enter the Family, Device, Package, and Speed Grade for the design. This design targets a Virtex XCV50-6-BG256. Click **OK**. The part number appears in the Part field in the New Version dialog box.

10. The Copy Persistent Data area of the dialog box allows you to copy constraint, guide, or floorplan data to the new revision that is about to be created. You can copy data from a custom file, or select None if you do not want to copy data. For this tutorial, None is selected for all three fields.

Note By default, the Design Manager copies floorplan and constraints file data from the "last" revision. The "last" revision is the final revision in the Design Manager project view. When initially creating a project, the Design Manager copies any existing constraints and floorplan file data from the project directory to the revision directory.

11. Click **OK** in the New Version dialog box.

    The Design Manager loads the design and displays a new design version and implementation revision in the project view as shown in the following figure.

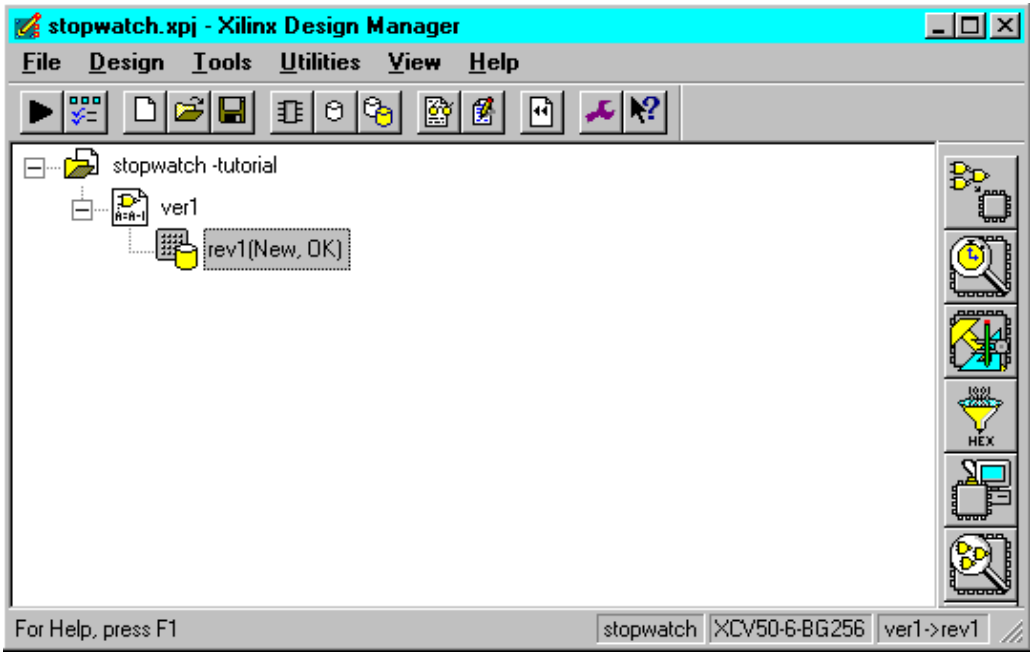

Figure 2-5 Stopwatch Project in Design Manager

Design Manager Status Bar

The status bar is located at the bottom of the Design Manager window. The status bar lists the current project, target device, and currently selected version / revision pair.

Figure 2-6 Design Manager Status Bar

Design Manager Toolbox

The toolbox, located on the right side of the Design Manager window, becomes active when a revision is selected. Icons in the toolbox (shown in the following figure) represent the Flow Engine, Timing Analyzer, Floorplanner, PROM File Formatter, Hardware Debugger, FPGA Editor, and JTAG Programmer tools.

Note The toolbox has drag and drop capability.

Figure 2-7 Design Manager Toolbox

Step 2: Specifying Options

An implementation revision contains data files and reports that are created based on a specific set of implementation strategies. Implementation strategies are defined by specifying a set of options. You can specify options that control how the Flow Engine implements a design, creates timing simulation data, creates netlist files, generates reports, and creates configuration data. The options available depend on the target device family.

You can create as many implementation revisions as you want for a design version. For example, if you want to try various implementation strategies on a netlist, several revisions can be created for a single design version. However, by default the Design Manager recompiles within the current revision.

In the Design Manager, notice how the project view displays rev1 under the initial version of the stopwatch project. The status of the revision is noted as (New, OK). *New* refers to the state of the design and is updated throughout the tutorial as the different compilation stages are completed. *OK* is the status of the current state and indicates there are no errors in the design processing.

Use the following steps to specify options for the tutorial design.

1. Select **`Design`** → **`Options`** to open the Options dialog box as shown in the following figure.

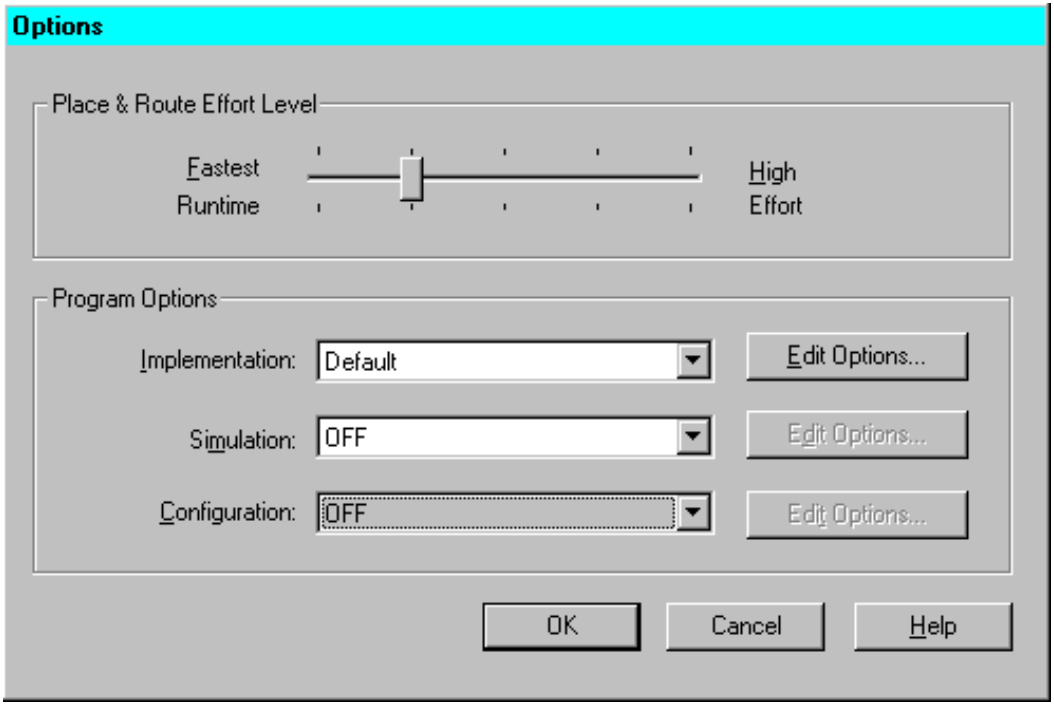

Figure 2-8 Options Dialog Box

This dialog box allows you to set options used in the implementation, simulation, and configuration flow. Changes made in this dialog box apply to the selected implementation revision. The dialog box above appears if you are targeting an FPGA. A slightly different Options dialog box is displayed if you are targeting a CPLD.

For more information on the fields in this dialog box, select **Help**.

2. Select **Edit Options** next to Implementation in the Program Options area of the dialog box.

   The following figure shows the top portion of the dialog box that appears. The implementation options control how the software maps, places, routes, and optimizes a design.

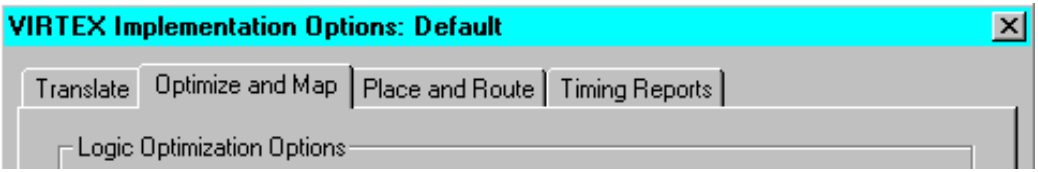

Figure 2-9 Top Portion of Virtex Implementation Options Dialog Box

3. Select the **Timing Reports** tab.
4. Select the **Produce Logic Level Timing Report** option. The Produce a Post Layout Timing Report option should already be selected by default. For both reports, select **Report Paths in Timing Constraints**.

   Timing reports are useful for evaluating design performance. These reports are analyzed later in this tutorial.

5. Click **OK** to save the Implementation options and return to the Options dialog box.
6. Select **OFF** from the Configuration Program Options pull-down list to disable bitstream generation for the design. This option is described later in this tutorial (after the evaluation of design performance is finished).
7. Click **OK** to exit the Options dialog box.

As previously mentioned, an implementation revision is created based on a specific set of implementation strategies. In addition to the program options that were just set, an implementation strategy is defined by the constraints applied to the design.

In this tutorial, the User Constraints File (UCF) is named stopwatch.ucf and is located in your design directory. By default, the Design Manager copies your constraints information into the new revision. You can use a text editor to open the stopwatch.ucf file in the

newly created rev1 directory and view the location constraints that are specified for the design.

The UCF provides a mechanism for constraining a logical design without returning to the design entry tools. However, it requires an understanding of the exact syntax needed to define constraints. Alternatively, the Xilinx Constraints Editor is a graphical tool that allows you to enter timing and pin location constraints. The Constraints Editor is used in this tutorial to view the current constraints specified in the stopwatch.ucf file, and to specify additional timing constraints.

To continue with the tutorial, select **Utilities** → **Constraints Editor** in the Design Manager.

You are prompted to run the Translate step before launching the Constraints Editor. The next step in the tutorial provides information on translating your design.

Step 3: Translating the Design

The Design Manager manages the files created during the implementation process while the Flow Engine controls the implementation process. The programs run by the Flow Engine use the settings specified by you in the Options dialog box. The Flow Engine gives you complete control over how a design is processed. Typically, you should set all of your options first, and then run through the entire flow by selecting the Implement command from the Design menu.

In this tutorial, you are attempting to further define the design by setting constraints after having defined options. To run the Constraints Editor, you must first run the Translate step.

Select **Yes** in the message dialog box to continue with the flow.

The Flow Engine is invoked for the first time. The steps in the design flow are graphically represented in the upper half of the Flow Engine window, and the status of each stage is also shown. Refer to the following figure.

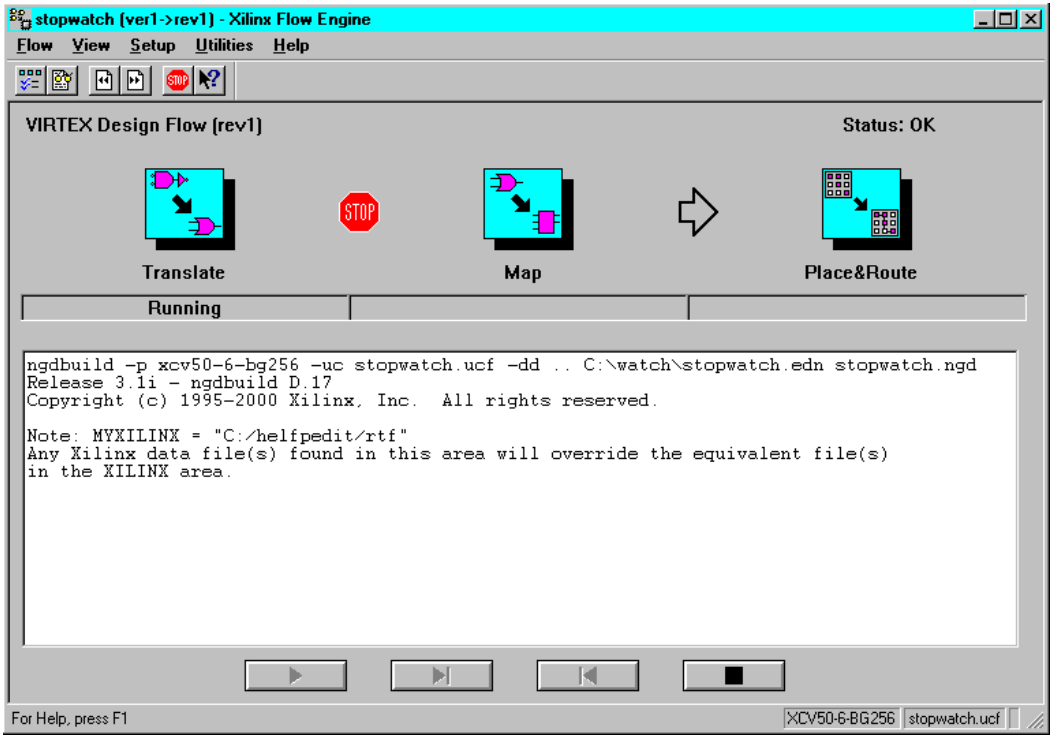

Figure 2-10 Translating Design

Notice the STOP sign placed between the Translate and MAP steps. This breakpoint has been automatically set to instruct the Flow Engine to stop after the Translate step is complete.

During translation, the NGDBuild program is launched, and performs the following functions.

- Converts input design netlists and writes results to a single merged NGD netlist. The merged netlist describes the logic in the design as well as any location and timing constraints.
- Performs timing specification and logical design rule checks
- Adds the UCF to the merged netlist

Once complete, the Flow Engine stops and the Constraints Editor is launched.

Step 4: Using the Constraints Editor

The Constraints Editor (see following figure) allows you to edit constraints previously defined (through a UCF), and to add new constraints to your design.

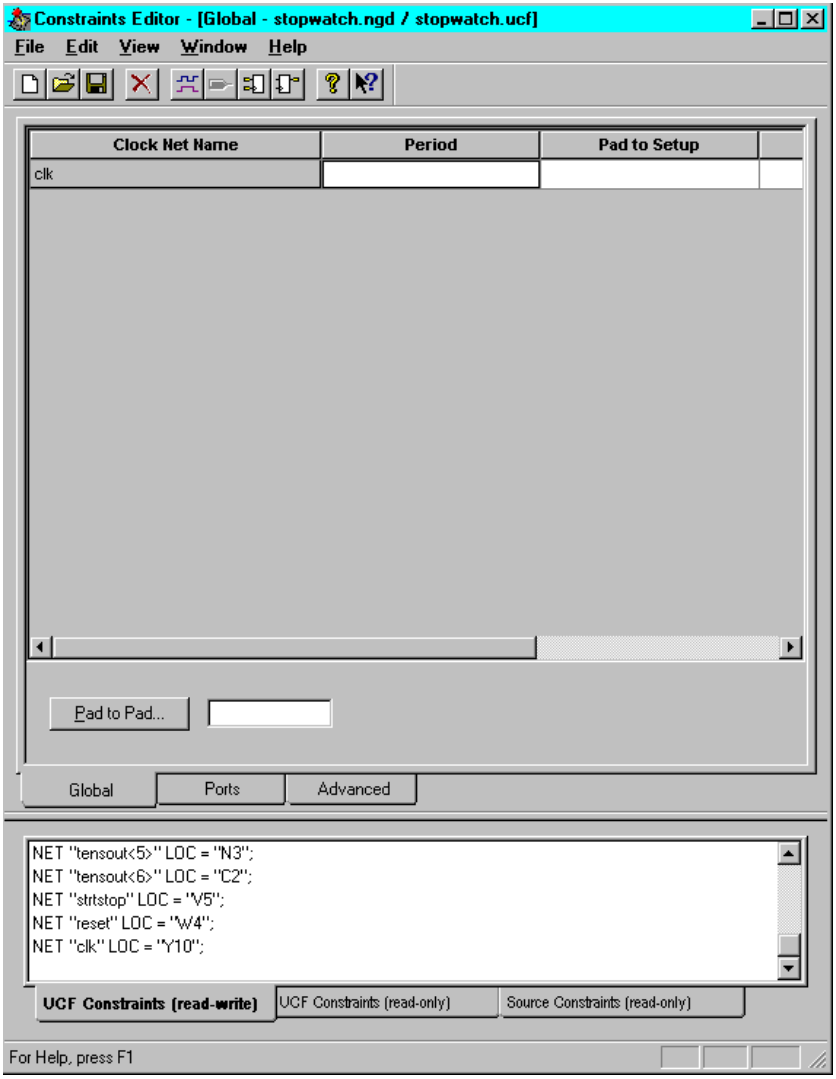

Figure 2-11 Constraints Editor Global Tab

Input files to the Constraints Editor include the following.

- NGD (Native Generic Database) file

  This file is input to the Map program, which then outputs the physical design database, an NCD (Native Circuit Description) file.

- Corresponding UCF

By default, when the NGD file is opened, an existing UCF file with the same base name as the NGD file is used. Alternatively, you can specify the name of the UCF file.

Upon successful completion, the Constraints Editor writes out a valid UCF file. NGDBuild uses the UCF file, along with design source netlists to produce a newer NGD file that incorporates the changes made. The NGD is then read by the MAP program (the next step in the design flow). In this tutorial, the stopwatch.ngd file and stopwatch.ucf file are automatically read into the Constraints Editor.

The Global tab appears in the foreground of the Constraints Editor window. This window automatically displays all of the clock nets in a design, and allows you to define the associated period, pad to setup, and/or clock to pad values.

Perform the following steps in the Constraints Editor.

1. Select the Period cell in the row associated with the clk clock net. Double-click the left mouse button to open the Clock Period dialog box.

   In the Clock Signal Definition area of the dialog box, select the **`Specific Time`** option to define an explicit period for the clock.

2. Enter 18.5 in the Time field. Verify that ns is selected from the Units pull-down list. Click **`OK`**.

   The period cell is updated with the newly defined global clock period constraint (with a default 50% duty cycle).

   Note In this tutorial, the Clock Period dialog box is used to specify constraints values. Alternatively, you can click once on a cell and directly enter constraints.

3. Select the Ports tab from the Constraints Editor main window.

Port Name	Port Direction	Location	Pad to Setup	Clock to Pad
clk	INPUT	Y10	N/A	N/A
onesout<0>	OUTPUT	M2	N/A	
onesout<1>	OUTPUT	P1	N/A	
onesout<2>	OUTPUT	M3	N/A	
onesout<3>	OUTPUT	N2	N/A	
onesout<4>	OUTPUT	M1	N/A	
onesout<5>	OUTPUT	R3	N/A	
onesout<6>	OUTPUT	N1	N/A	
reset	INPUT	W4		N/A

Figure 2-12 Constraints Editor Ports Tab

The left hand side displays a listing of the current user-defined ports. Notice that certain cells in the Location column contain device pins locking down ports to actual pins on the target device. This information is obtained by the Constraints Editor from the stopwatch.ucf file.

4. Enter the following pin locations by selecting the Location text box associated with each of the following signals.

Port Name	Location
onesout<0>	M2
onesout<1>	P1
onesout<2>	M3
onesout<3>	N2
onesout<4>	M1
onesout<5>	R3
onesout<6>	N1

5. Select **File** → **Save**.

   The changes made in the Constraints Editor are now saved to the stopwatch.ucf file in your current revision directory.

6. You are prompted to repeat the Translate step. Click **OK**.

7. Select **File** → **Exit**

Caution

Read the following procedure before you start to implement the design.

How to Stop the Design Processing Flow

Before continuing with the next step in the tutorial, review the procedure described in this section for stopping the processing of the design after the MAP step. Because the steps for the tutorial design often finish quickly, you should be familiar with this procedure before you start the Flow Engine.

Setting a break point to stop the design processing is useful for evaluating design performance before going on to the next step in the design flow. You can set a break point at any time during the processing of a design by the Flow Engine. For example, setting a break point after the Translate step is useful when you want to perform a functional simulation of a design and copy the resulting *design*.ngd file to your working directory. After copying the *design*.ngd file, you can run the appropriate NGD2XXX program on the file to create functional simulation data. For more information on the NGD2XXX programs, see the appropriate chapter in the *Development System Reference Guide*.

1. In this tutorial, you want to stop processing the design after the MAP step. To do this, you must set a break point to stop the Flow Engine. To stop after the MAP step from within the Flow Engine, click the stop sign toolbar icon while MAP is running.
2. The Stop After dialog box is displayed. The pull-down list box displays the break point appropriate for the current state of the design.

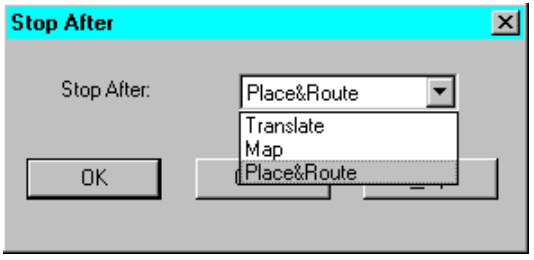

Figure 2-13 Stop After Dialog Box

3. Select **Map** in the list box and click **OK**.
4. The stop sign is added to the design flow between the Map and Place and Route steps as shown in the following figure.

Note The status bar at the bottom of the Flow Engine window is updated with the specified user constraints file (stopwatch.ucf).

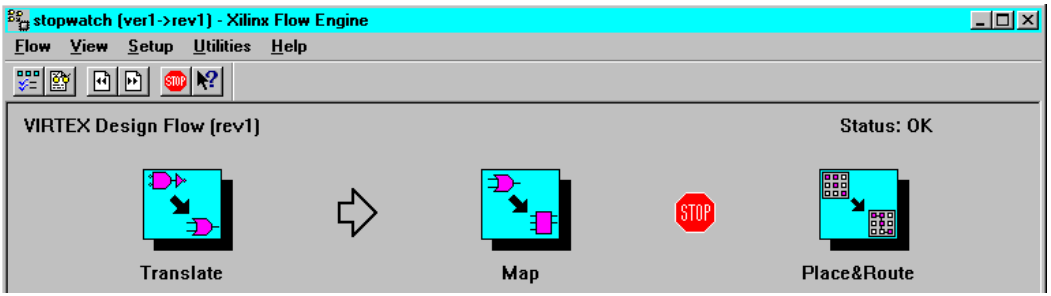

Figure 2-14 Mapping Design

Step 5: Mapping the Design

Now that all implementation strategies have been defined (options and constraints), the next step is implementing the design.

1. Select **Design** → **Implement** from the Design Manager.
2. The Flow Engine automatically detects that changes were made to your constraints file, and displays a message box indicating that the Translate step should be run again. Select **YES**.
3. The Flow Engine is invoked. Perform the procedure previously described in the "How to Stop the Design Processing Flow" section to stop the processing of the design after the Map step.

At this point, the input netlist is translated again, merged into a single design file, and mapped to CLBs and IOBs.

The Map program performs the following functions.

- Allocates CLB and IOB resources for all basic logic elements in the design.
- Processes all location and timing constraints, performs target device optimizations, and runs a design rule check on the resulting mapped netlist.

After the MAP step is done, the Flow Engine closes and the Implement Status dialog box appears.

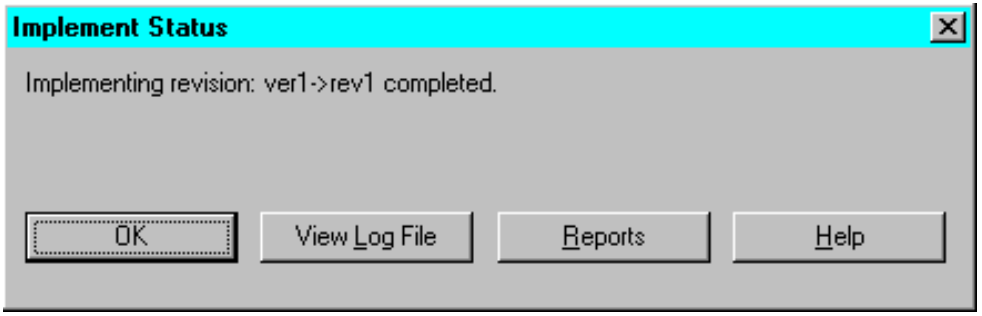

Figure 2-15 Implement Status Dialog Box

Follow these steps to view the generated reports.

1. Select **Reports** to invoke the Report Browser window. The Translation Report appears as the first report generated. The Map Report and Logic Level Timing Report files are created by the Map step. New reports that have not been read are denoted with a gold star in the upper left corner of the file icon.

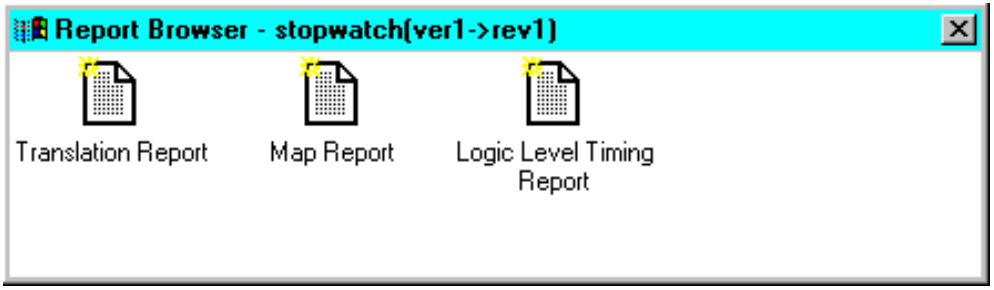

Figure 2-16 Report Browser after Running Map

2. Double-click on a report to review its output. The following table lists the types of reports available.

Table 2-2 Report Browser Reports

Report	Description
Translation Report	Includes warning and error messages from the translation process.
Map Report	Includes information on how the target device resources are allocated, references to trimmed logic, and device utilization. For detailed information on the Map report, refer to the Development System Reference Guide.
Logic Level Timing Report	Provides a summary analysis of your timing constraints based on block delays and estimates of route delays. This report is produced after Map and prior to PAR (Place And Route).

3. Select **OK** to close the Implement Status dialog. Keep the Report Browser open for now. Some of these reports are evaluated in more detail later in the tutorial.

Notice that the Design Manager project view displays the status of the revision as (Mapped, OK). “Mapped” refers to the state of the design and is updated throughout the tutorial as the different compilation stages are completed. “OK” refers to the status of the current state and indicates there are no errors in the design processing.

The design has now been mapped to the target architecture. The next step involves checking the design paths for block delays.

Step 6: Using Timing Analysis to Evaluate Block Delays After Mapping

After the design is mapped, you can use the Logic Level Timing Report to evaluate the logical paths in the design. Because the design is not placed and routed yet, actual routing delay information is not yet available. The timing report describes the logical block delays and estimated routing delays. The net delays that are provided are based

on an optimal distance between blocks (also referred to as *unplaced floors*).

Estimating Timing Goals With 50/50 Rule

You can get a preliminary idea of how realistic your timing goals are by evaluating a design after the Map step. A rough guideline (known as the 50/50 rule) specifies that the block delays in any single path make up approximately 50% of the total path delay after the design is routed. For example, a path with 10 ns of block delay should meet a 20 ns timing constraint after it is placed and routed. If your design is extremely dense, or if you are using an architecture with fewer routing resources (for example, a 4025E device versus a 4028XL), your net delays can be more than 50% of the total path delay.

Report Paths In Timing Constraints Option

Because timing constraints were defined for this tutorial design, the Report Paths in Timing Constraints option was selected. This option forces the Logic Level Timing Report to provide a period and path analysis on the constraints specified. Double-click on this report in the Report Browser. In the report, the period timing constraint is listed on top, as is the minimum period obtained by the tools after mapping. Because the report is limited to one path per timing constraint, there is a breakdown of a single path that contains 4 levels of logic. Notice the percentage of block (logic) delay versus routing delay for this calculation. The unplaced floors listed are estimates (indicated by the letter “e” next to the net delay) based on optimal placement of blocks.

If you do not generate a Logical Level Timing Report, the Place and Route (PAR) program still processes a design based on the relationship between the block delays, floors, and timing specifications for the design. For example, if a PERIOD constraint of 8 ns is specified for a path, and there are block delays of 7 ns and unplaced floor net delays of 3 ns, PAR stops and generates an error message. In this example, PAR fails because it determines that the total delay (10 ns) is greater than the constraint placed on the design (8 ns).

Review the Logic Level Timing Report to determine any timing violations that occurred prior to running PAR.

Step 7: Placing and Routing the Design

After the mapped design is evaluated to verify that block delays are reasonable given the design specifications, the design can be placed and routed. The Flow Engine can perform the following place and route algorithms.

- Timing Driven — run PAR with timing constraints specified from within the input netlist or from a constraints file
- Non-Timing Driven — run PAR and ignore all timing constraints

In this tutorial, timing driven placement and timing driven routing are automatically performed by PAR because timing constraints are specified for the design.

Close any open reports and the Report Browser.

To place and route the design, perform the following procedure.

1. In the Design Manager window, select **`Design`** → **`Implement`** to run the implementation flow.

   The Flow Engine is launched again. The Status:OK message in the upper right corner indicates that no errors are generated by PAR at this point.

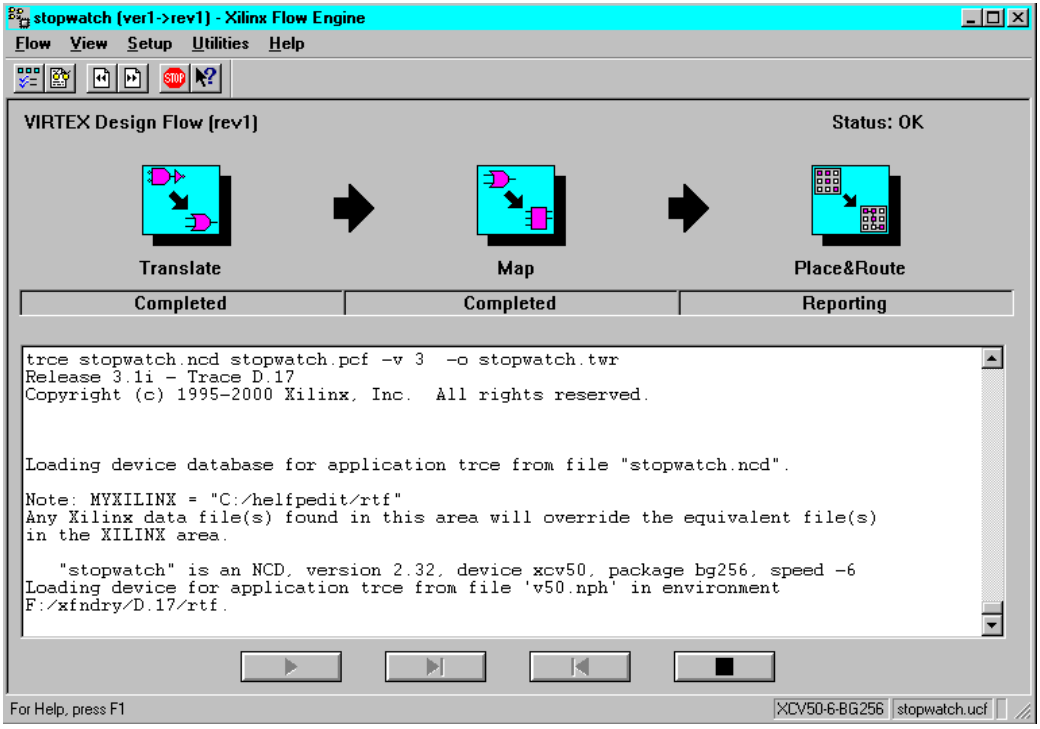

Figure 2-17 Placing and Routing Design

2. After the Flow Engine closes, select **Reports** in the Implement Status dialog box to display the Report Browser.

   Four new reports are created in the Report Browser: the Place and Route Report, the Pad Report, the Asynchronous Delay Report, and the Post Layout Timing Report.

   Review the reports generated to make sure the place and route process finished as expected.

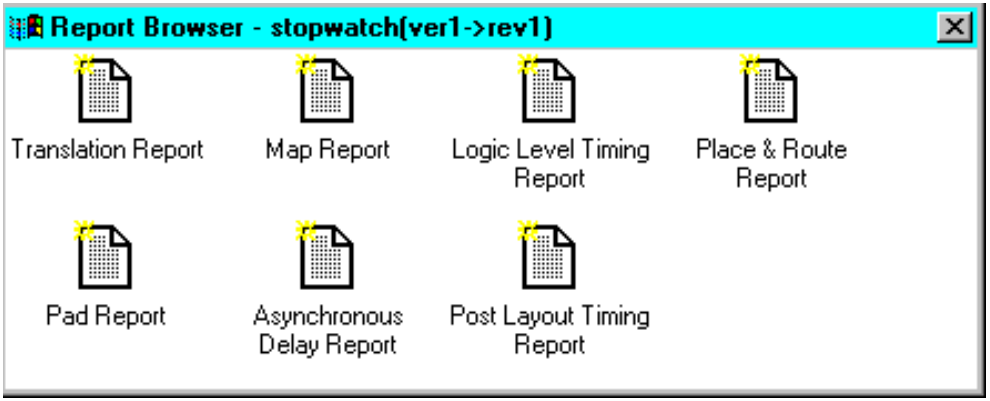

Figure 2-18 Reports Available After Place and Route

Table 2-3 Description of Reports Available After Place and Route

Report	Description
Place & Route Report	Provides a device utilization and delay summary. Use this report to verify that the design successfully routed and that all timing constraints were met.
Pad Report	Contains a report of the location of the device pins. Use this report to verify that pins locked down were placed in the correct location.
Asynchronous Delay Report	Lists all nets in the design and the delays of all loads on the net.
Post Layout Timing Report	Incorporates both the logic and routing delays to generate an evaluation of the design's timing constraints, clock frequencies, and path delays.

Note In the Design Manager window, the status of the current version / revision is now (Routed, OK).

Step 8: Evaluating Post Layout Timing

After the design is placed and routed, a Post Layout Timing Report is generated by default to verify that the design meets your specified

timing goals. This report evaluates the logical block delays and the routing delays. The net delays are now reported as actual routing delays after the place and route process (indicated by the letter “R” next to the net delay).

Double-click on the Post Layout Timing Report to open it. Following is a summary of this report.

- The minimum period value increased due to the actual routing delays.
- After the Map step, logic delay contributed to about 80% of the minimum period attained. The post layout report indicates that the logical delay value decreased somewhat. The total unplaced floors estimate changed as well. Routing delay after PAR now equals about 31% of the period; a true report of net delays after the place and route step.
- The post layout result does not necessarily follow the 50/50 rule previously described because the worst case path includes primarily component delays. After the design was mapped, block delays constituted about 80% of the period. After place and route, the majority of the worst case path is still made up of logic delay. Since total routing delay makes up only a small percentage of the total path delay, spread out across three nets, expecting this to be reduced any further is unrealistic. In general, you can reduce excessive block delays and improve design performance by decreasing the number of logic levels in the design.

Step 9: Creating Timing Simulation Data

After your design is placed and routed and the timing is statically verified, the next step is to create timing simulation data. To create timing simulation data, perform the following steps in the Design Manager.

1. Select **Design** → **Options** to open the Options dialog box.
2. Select the simulator that corresponds to your design entry tool from the Simulation pull-down list in the Program Options section of the dialog box.

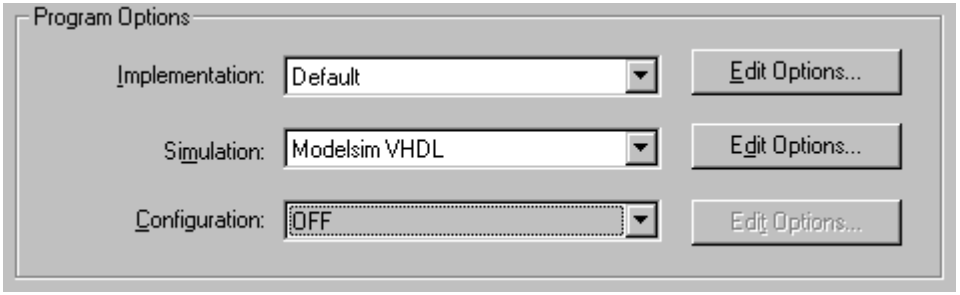

Figure 2-19 Options Dialog Box for Simulation

3. Click **OK** to close the Options dialog box.
4. Select **Design** → **Implement**.

   In the Flow Engine, a new stage appears directly after Place & Route called Timing(Sim). This new stage produces timing simulation data. This stage did not appear originally because a specific simulator was not selected in the initial pass. By default, this option is set to OFF. For your designs, you can select all options at the beginning of the design processing, or you can set them later.

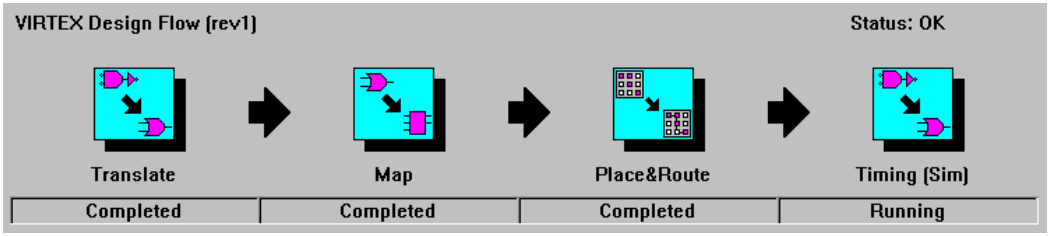

Figure 2-20 Flow Engine with Timing Simulation

During the Timing(Sim) step, the Flow Engine runs the NGDAnno program to create a back-annotated NGD file. The NGD file is then used as input to one of the NGD2XXX programs to produce the preferred simulation file format. By default, the files created are named *time_sim*. To make it easy to find the output files for your third-party simulation environment, the files are automatically copied to your working directory.

Step 10: Creating Configuration Data

To create a bitstream for the target device, follow these steps.

1. Select **Design** → **Options** to open the Options dialog box.
2. Select **Default** from the pull-down list for Configuration Program Options.
3. Click the **Edit Options** button corresponding to Configuration.

   The Virtex Configuration Options dialog box appears. The configuration templates set options that define the initial configuration parameters, start-up sequence, readback capabilities, and other advanced features.

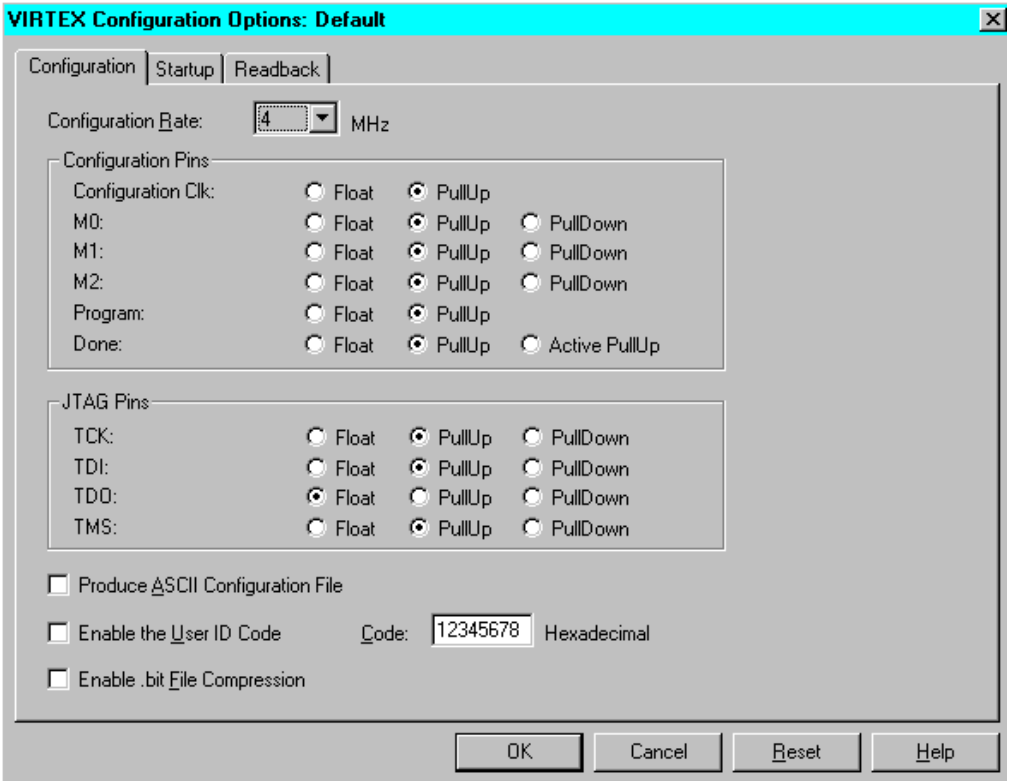

Figure 2-21 Virtex Configuration Options Tab

4. In the Configuration tab, verify that **PullUp** is selected for the Done pin.
5. Click **OK** to close the Virtex Configuration Options dialog box.
6. Click **OK** to close the Options dialog box.
7. Select **Design** → **Implement** from the Design Manager.

   The Flow Engine starts, running the BitGen program in the newly added Configure stage as shown in the following figure. BitGen creates the *design_name*.bit and *design_name*.ll files (in this tutorial, the stopwatch.bit and stopwatch.ll files). The *design_name*.bit file is the actual configuration data. The *design_name*.ll file is the logical allocation file that is used during hardware debugging to determine the location of the probable points in the design. These files are automatically copied to your working directory.

   The *design_name*.ll file is used to perform device readback with the Hardware Debugger tool. For more information on device readback, refer to the latest version of the Watch Design Hardware Verification Tutorial, located at `http://support.xilinx.com/support/techsup/tutorials/index.htm`.

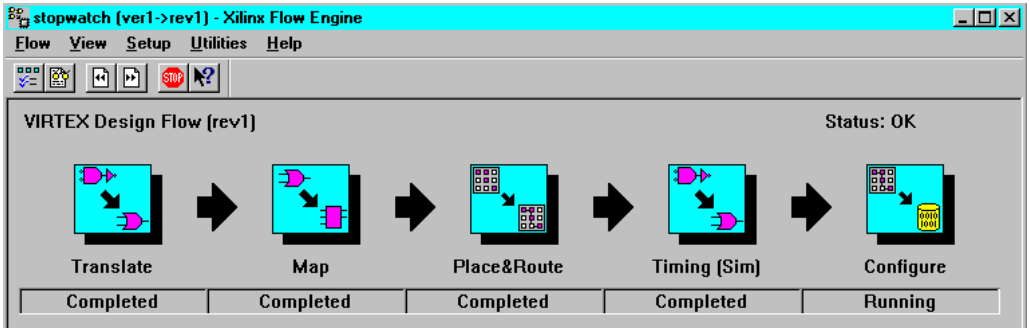

Figure 2-22 Configuring Design

8. The Flow Engine saves the configuration options in the BitGen Report. Review the report using the Report Browser. Verify that the specified options were used when creating the configuration data.

Step 11: Using the PROM File Formatter

If you are going to program a single device using the Hardware Debugger, all you need is a *design_name*.bit file. If you are going to program several devices in a daisy chain configuration, or program your devices using a PROM, you must use the PROM File Formatter (PFF) to create a PROM file. The PROM File Formatter accepts any number of bitstreams and creates one or more PROM files containing one or more daisy chain configurations.

1. To start the PROM File Formatter, click the PROM File Formatter icon in the toolbox in the Design Manager.

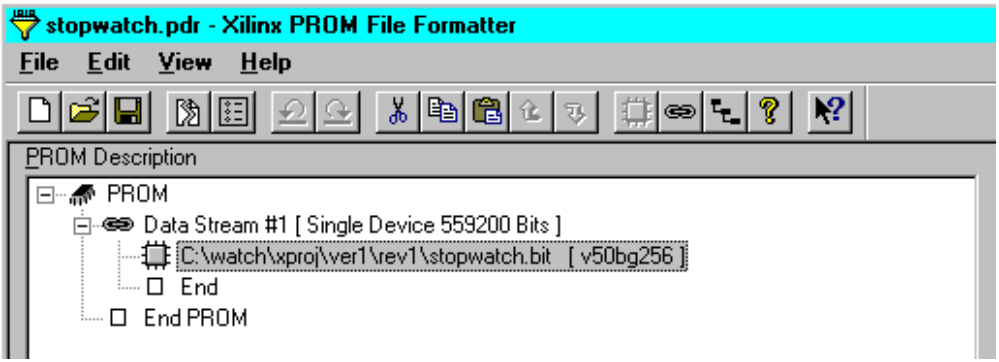

Figure 2-23 PROM File Formatter

The PFF starts with a default PROM that matches the currently selected (configured) revision. At this point, you can add additional bitstreams to the daisy chain, create additional daisy chains, remove the current bitstream and start over, or immediately save the current PROM file configuration.

The status bar at the bottom of the PFF window displays the PROM format, data format, current PROM size, and percentage of the selected PROM used by the current PROM configuration. The currently selected PROM is an XC1701. 559,232 bits of data are required to hold the configuration bitstream for the XCV50 target device for this tutorial. The PFF determined that an XC1701 is the correct PROM because it can hold up to 1,048,576 configuration bits (or 53% full).

The right half of the PFF window is a directory structure used for locating bitstreams. Only files with a .BIT extension are shown in

the list. For detailed information on using the PROM File Formatter to create daisy chains or complex PROM configurations, see the *PROM File Formatter Guide*. This tutorial describes how to save the default PROM file.

2. Select **File** → **PROM Properties** to open the PROM Properties dialog box.

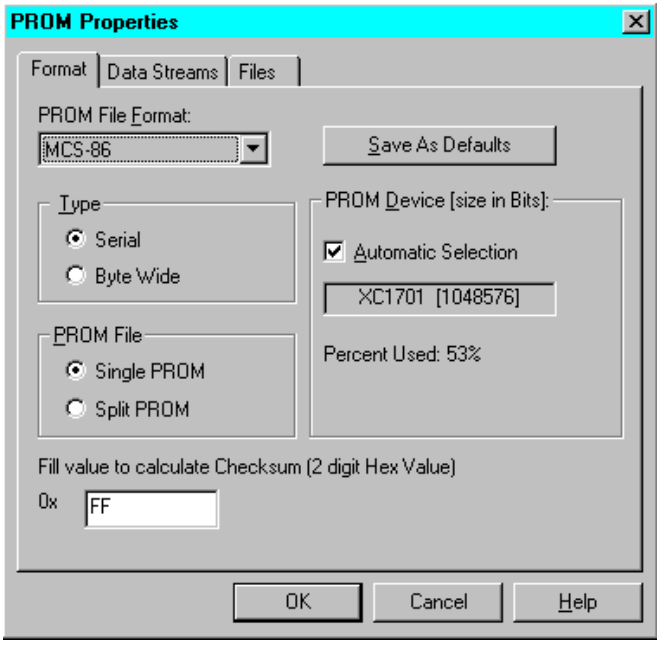

Figure 2-24 PROM Properties Dialog Box with Single PROM

3. Select appropriate options in this dialog box for PROM File Format, Type, and the number of PROMs used to hold the data.

   If you have more data than space available in the PROM, you must split the data into several individual PROMs with the Split PROM option. In this case, only a single PROM is needed. Click **OK** to accept the PROM Properties.

4. Select **File** → **Save Description** to save the PROM file.

5. Specify your working directory as the area where the PROM Description File will be saved.

The PROM File Formatter saves both the PROM file (stopwatch.mcs) and a PROM Description File (stopwatch.pdr). The PDR file can be re-opened if any changes are required.

6. Select **File** → **Exit** to close the PROM File Formatter.

   This completes the tutorial. For more information on the Alliance design flow and implementation methodologies (especially the tools and programs that were not covered as part of this tutorial), refer to the online version of the Alliance software manuals at `http://support.xilinx.com`.

Appendix A

Alliance FPGA Express Interface Notes

This appendix provides information on installing and using the Alliance FPGA Express with the Xilinx Alliance software release. Synopsys and the Xilinx CD-ROM documentation are referenced to help you find additional information. The Alliance FPGA Express is FPGA Express software purchased from Synopsys. Foundation Express is the FPGA Express software bundled with the current release of Foundation and is purchased from Xilinx. All references to FPGA Express in this appendix refer to Alliance FPGA Express. For more information on Foundation Express, refer to the *Foundation Series 3.1i Quick Start Guide*.

FPGA Express is a Verilog/VHDL compiler designed to work with Windows. FPGA Express can process either Verilog or VHDL files. This tool writes out XNF files (EDIF for Virtex designs) which are fully compatible with Alliance Series Design Implementation tools. Only the Xilinx implementation tools and a third party simulation tool are needed in addition to FPGA Express to fully create and simulate a design. For more information on FPGA Express, see `http://support.xilinx.com/support/techsup/journals/index.htm`. This appendix includes the following sections.

- "Using LogiBLOX with FPGA Express"

Additional Documentation

The following documentation is available for FPGA Express and the Alliance Series Design Implementation tools for the current release of software.

- For installation of the Alliance Series Design Implementation Tools, refer to the *Alliance Series 3.1i Release Notes and Installation Guide*.
- For installation of FPGA Express, HDL-entry flow, and mixed entry flows, refer to the *FPGA Express User's Guide*, a hard copy document included with your FPGA Express software from Synopsys.
- For additional information on FPGA Express and the Xilinx flow, refer to the *Synopsys FPGA Express Design Guide*, available via `ftp://ftp.xilinx.com/pub/swhelp/synopsys/xprsgde.zip`. This file is a Word for Windows (95) version 7.0 file.

Alliance FPGA Express/Xilinx Design Flow

FPGA Express is the top-level design tool in the design flow. FPGA Express writes out an XNF file (EDIF for Virtex) which is fully compatible with the Alliance Series Design Implementation tools. The XNF file written out by FPGA Express can be accepted by NGDBuild or the Design Manager for creation of a PROM file.

Refer to the following figure for a graphic representation of the design flow.

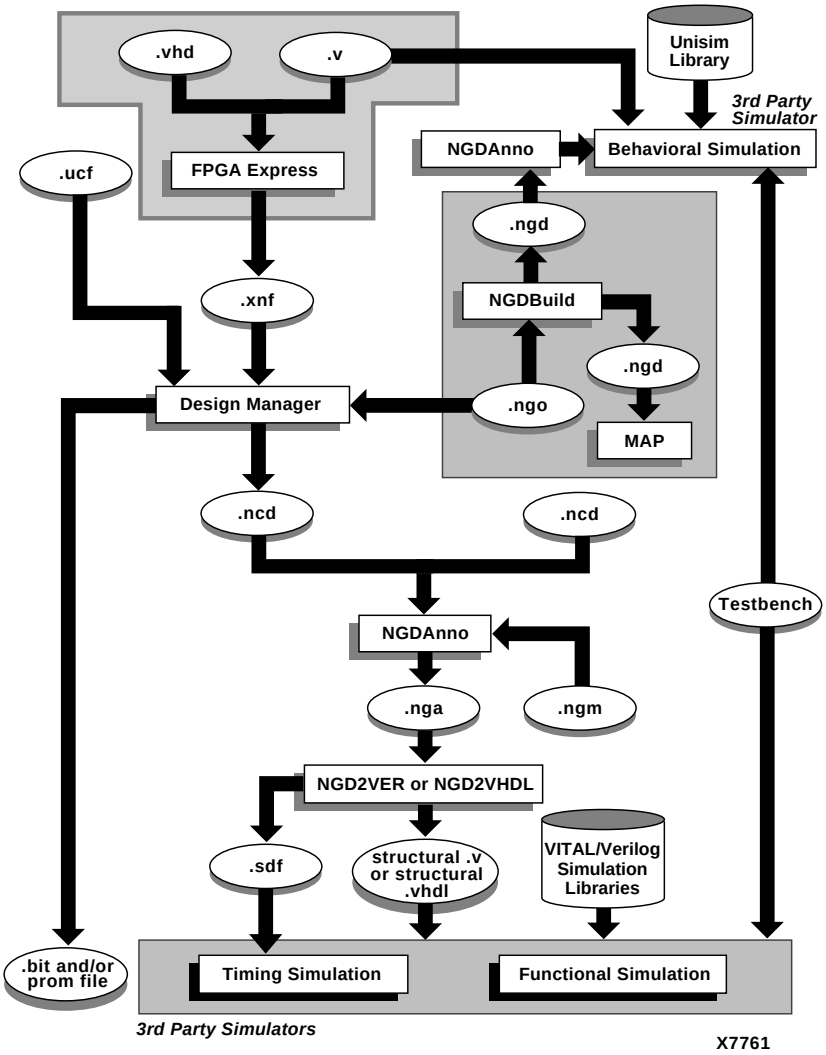

Figure A-1 Alliance FPGA Express/Xilinx Design Flow

Installing FPGA Express

Insert the FPGA Express CD into your CD-ROM drive. Start the Explorer and double-click on the CD-ROM icon. Double-click on setup.exe to start the install process.

For additional instructions on how to install FPGA Express, refer to the *FPGA Express User's Guide* included with the FPGA Express software from Synopsys.

Entering a Design

To enter a design, use the following steps.

1. Start FPGA Express by selecting the following.

   Program → Synopsys → FPGA Express

2. Use a text editor to enter your design in Verilog or VHDL.
3. Define your project in FPGA Express by selecting.

   File → New...

4. Identify the HDL files for synthesis by selecting the following.

   Synthesis → Identify Sources

5. Specify the top-level file in your project by selecting the top-level file in the top-level design drop-down list in the middle of the FPGA Express toolbar.
6. Create an implementation by selecting the following.

   Synthesis → Create Implementation

7. Optimize your design by selecting the following.

   Synthesis → Optimize Chip

8. Write an XNF file by selecting the following.

   Synthesis → Export Netlist

Verilog or VHDL designs are the input files for the FPGA Express design flow, and the output is an XNF file (EDIF for Virtex designs), which can be processed directly by the Xilinx implementation tools. For details on defining projects in FPGA Express, entering HDL code, defining constraints in FPGA Express, supported devices, and design

issues, refer to the *FPGA Express User's Guide* included with your FPGA Express software from Synopsys.

Simulating a Design

FPGA Express is a synthesis tool only. Simulation of designs with FPGA Express must be done with a third party simulation tool. For more information on simulation with FPGA Express, refer to the documentation for your third party simulation tool.

Timing Constraints

FPGA Express automatically inserts timespecs into the XNF file it writes out. For Virtex designs, a separate .ncf file containing timing constraints is created along with the .edf file. Optionally, the user can choose not to write out timespecs in the XNF file from FPGA Express. Instead, you can write the constraints in a .ucf file. The timespecs created by FPGA Express in the XNF file have the FROM: TO syntax.

Porting Code from FPGA Compiler to FPGA Express

Read this section if you are porting a design from FPGA / Design Compiler to FPGA Express. If you are compiling a design originally compiled with FPGA / Design Compiler and the code is one hundred percent behavioral, then no modification of the code is needed. But, if you have instantiated components from the XSI libraries, some of these components do not exist in the FPGA Express libraries.

Some of the components that can be instantiated in the Xilinx design flow cannot be instantiated in the FPGA Express tool, since there are slight differences in names. For example, the BUFGP_F in the XSI component library does not exist in the FPGA Express component library. In FPGA Express, the equivalent name of the BUFGP_F is BUFGP. For a complete listing of the library cells that can be instantiated in FPGA Express, refer to the contents of the following.

`fpgaexpress/lib/xc3000`

`fpgaexpress/lib/xc4000e`

`fpgaexpress/libxc5200`

The fpgaexpress directory is where FPGA Express is installed on your system. In these directories, there are files with a .dsn extension. The

string in front of .dsn is the name of the CELL that can be instantiated in FPGA Express.

In general, instantiation is not necessary. For the XC4000EX / XL / XLA / XV FPGA Express flow, you must instantiate the following components.

- I/O muxes
- Fast capture latches
- RAM
- BSCAN
- LogiBLOX

Using LogiBLOX with FPGA Express

For information on using LogiBLOX and FPGA Express, see `http://support.xilinx.com/support/techsup/journals/index.htm`.

Appendix B

Configuring Xprinter

This appendix provides detailed instructions on configuring a printer so you can print from the Xilinx application. The information in this appendix applies only to workstation applications. The following sections are in this Appendix.

Required Wind/U Files

You must have the following Wind / U files installed correctly to print from your application.

- PPD (PostScript Printer Description)

  PPD files provide Wind / U with model-specific information like paper tray configuration, supported paper sizes, available ROM fonts, and so forth.

- AFM (Adobe Font Metric) and TFM (Tagged Font Metric)

  AFM and TFM files provide font metric information for all fonts in PostScript and PCL5 printers, respectively.

- Additional files

  Various other files are copied to `$XILINX/data/xprinter` and are required by Wind / U when printing. These include xprinter.prolog (PostScript prolog), psstd.fonts and pclstd.fonts (provide PostScript / PCL5 to X Window System font name mappings), and rgb.txt.

Make a complete, as-is copy of the directory `$WUHOME/xprinter` and include it in the installation for your product. None of the files from this directory require modifications in most environments.

Configuring .WindU

Once you have installed the required printer configuration files on your system, you must configure the .WindU file in your home directory (or `SYS$LOGIN:WINDU.INI`) for printing.

You can modify the .WindU file either by using a text editor or the Xprinter Printer Setup dialog box. Using the dialog box is recommended, because it reduces the risk of error. The instructions in this appendix describe how to edit the .WindU file, and then provide step-by-step instructions for performing the same task with the Printer Setup dialog box. These instructions are provided to help you configure your printer.

Printer Information and PPD Files

When you configure Wind/U to print, you need to know the following information for each printer you want to access.

- Name of the printer description (PPD) file
- The command used to send output to a printer

The Wind/U installation media includes the PPD files for most commonly used printers. To verify that the PPD file associated with your printer is included, look at the Printer Devices dialog (from the Printer Setup dialog, click Install and Add Printer). If your printer model is listed, Wind/U has a PPD file for your printer. If your printer model is not listed, contact your printer vendor to obtain the PPD file for your printer.

Unix Print Command

The command used to send output to a specific printer depends on the platform, the printer, and how the printer is connected to your system. For example, if a printer is connected directly to your system, the following might be valid print commands.

Unix	**`lp -d ps -t$XPDOCNAME`**
OpenVMS	**`PRINT /QUEUE=OPTRA`**

If your printer is connected to a If your printer is connected to a different system on your network, your printer command will specify how to connect to that system. For example, if a printer is connected to the system bandit on your network, any of the following might be valid print commands.

```
rsh bandit lp -d ps -t$XPDOCNAME
```

Note In these examples, $XPDOCNAME represents the name of the output file sent to the printer with the specified command. If you use a multi-word file name, such as a print file, you must enclose the $XPDOCNAME in quotation marks as follows. You must escape the quotation marks in the remote command, because rsh strips them out if you do not.

The following table lists the commands needed to configure your remote or local printer.

Local Printer	`lp -d ps -t"$XPDOCNAME"`
Remote Printer	`rsh bandit lp -d ps -t\"$XPDOC-NAME\"`

Configuring Wind/U for Printing

Once you know the name of the PPD file and the print command for each printer you want to direct output to, you can configure Wind/U to recognize those printers. To configure Wind/U to recognize a printer, you must do the following.

1. Define a port, which is an alias for the print command.
2. Associate the port with the printer's PPD file.
3. Specify a default printer.
4. Set printer options.

Defining a Port

A printer port is an alias for the print command. It is defined in the Ports section of `$HOME/.WindU` and appears as part of the Printer Name in the Printer Setup dialog box. For example, the following is the first Printer Name in the Printer Setup dialog box before you make any changes to `$HOME/.WindU`.

```
AppleLaserWriter v23.0 PostScript on FILE:
```

In this Printer Name, FILE: is the port name. Port entries in the [ports] section have the following format.

port=print_command

This command sends output to the printer port. For example, if you have two printers: ORION and SIRIUS, your [ports] section may look like the following example.

```
[ports]

ORION=rsh bandit lp -d ps -t\"$XPDOCNAME\"

SIRIUS=rsh bandit lp -d ps -T pcl5 -t\"$XPDOCNAME\"
```

In this example, both printers are connected to the system bandit, so the print command is a remote shell command executed on bandit. ORION is a PostScript printer, so the command **`lp -d ps`** is executed on bandit to print to ORION. SIRIUS, however, is a PCL5 printer, so the print command executed on bandit to print to SIRIUS is **`lp -d ps -T pcl5`**.

If you have a printer connected to your local system, you need to add an entry for it. For the local printer, add an entry similar to the following.

```
[ports]

ORION=rsh bandit lp -d ps -t\"$XPDOCNAME\"

SIRIUS=rsh bandit lp -d ps -T pcl5 -t\"$XPDOCNAME\"

LOCAL=lp -d ps -t$XPDOCNAME
```

Your printer port can be any name, except FILE:, which is the only reserved port name. It causes HyperHelp to create a print file formatted specifically for the specified printer type.

You must create an entry in the [ports] section for every printer you want to print to.

To Define a New Port

To define a new port using the Printer Setup dialog box, perform the following steps.

1. To display the Ports dialog box, from the Printer Setup dialog box, click **Install**, **Add Printer**, and **Define New Port**.
2. Type the port definition in the Edit Port edit box.
3. Click **Add/Replace**.

   The new port is now included in the list of current port definitions.
4. Repeat steps 1–3 for each printer you want to print to.

Note To create a printer port for each available printer queue on HP700 systems, click the Spooler button in the Ports dialog box. This command creates a default printer port for each available printer queue returned by the lpstat -a command.

To Modify an Existing Port

To modify an existing port using the Printer Setup dialog box, perform the following steps.

1. To display the Ports dialog box, from the Printer Setup dialog box, click **Install**, **Add Printer**, and **Define New Port**.
2. Select the port you want to modify and edit the port information in the Edit Port edit box.
3. Click **Add/Replace**.

   The modified port is now included in the list of current port definitions.

Matching a Printer Type to a Defined Port

After you define a port for each printer, specify the type of printer associated with each port. Device types are listed in the [devices] section of the .WindU file. Each entry in the [devices] section has the following format.

```
alias=PPD_file driver,port
```

Note There must be a space between the PPD_file and driver and a comma between the driver and the port.

The following table describes each part of this entry.

Field	Description
alias	The alias is a descriptive name that identifies the printer. It can be anything. The alias is the name of the printer that appears in the Printer Setup dialog box, such as, HP LaserJet 4L PostScript).
PPD_file	The PPD_file is the name of the printer description (PPD) file used by the printer, without the .PPD extension.
driver	The driver is the type of driver the printer uses. Valid values are PostScript, PCL4, and PCL5.
port	The port is the printer port listed in the [ports] section of the .WindU file. (ORION, SIRIUS, and LOCAL appear in the example [ports] section.)

The following table lists the ports that correspond to each printer type.

Port	Printer Type	Output Type
ORION	HP LaserJet 4LPostScript	PostScript
SIRIUS	HP LaserJet 4M PCL Cartridge	PCL
LOCAL	QMS-PS 2200 v52.3	PostScript

Following is an example procedure for configuring three printers.

1. Choose an alias for each printer.

   To easily identify the printer you want to use from the Printer Setup dialog box, use these aliases.

   - HP LaserJet PS
   - HP LaserJet PCL
   - QMS PS

   Note If you use the Printer Setup dialog box to associate ports and PPD files, you cannot specify a printer alias. You must choose an alias from the predefined list that appears in the Printer Devices list box in the Add Printer dialog box. The corresponding PPD file is already associated with the printer aliases in this list box.

2. Identify the PPD file associated with each of these printers. In this example, the [devices] section of the .WindU file appears as follows.

```
[devices]

HP LaserJet PS=HP3SI523 PostScript,ORION

HP LaserJet PCL=HP4M PCL,SIRIUS

QMS PS=Q2200523 PostScript,LOCAL
```

After you add these entries to your .WindU file, the following printer choices are available from the Printer Setup dialog box.

```
HP LaserJet PS on ORION

HP LaserJet PCL on SIRIUS

QMS PS on LOCAL
```

To Match a Printer Device to a Port

To match a printer device to a port using the Printer Setup dialog, perform the following steps.

1. To display the Add Printer dialog box, from the Printer Setup dialog box, click **`Install`** and **`Add Printer`**.
2. In the Printer Devices field, select the description that matches the printer you are installing.

   If no description matches your printer, contact your printer vendor for a printer description (PPD) file and install it in the `$WUHOME/xprinter/ppds` directory.
3. Select the desired port in the Current Port Definitions list box and click **`Add Selected`**.

   The new printer is now included in the list of currently installed printers.

To Remove an Installed Printer

To remove a printer device/port combination using the Printer Setup dialog box, perform the following steps.

1. To display the Printer Installation dialog box, from the Printer Setup dialog box, click **Install**.
2. In the Currently Installed Printers list box, select the printer you want to remove and click **Remove Selected.**

Specifying a Default Printer

After all available printers are configured, you can make one of them the default printer. To specify a default printer in the Printer Setup dialog box, add an entry in the following format to the [windows] section of the .WindU file.

```
[windows]

device=PPD_file,driver,port
```

Provide the same information that you used in the [devices] section. Only the format of the entry is different; there is a comma between the *PPD_file* and the *driver* instead of a space.

For example, if you want the default printer to be the printer at port ORION, your [windows] section appears as follows.

```
[windows]

device=HP4L,PostScript,ORION
```

The printing-related sections of your .WindU file look like the following.

```
[windows]

device=HP4L,PostScript,ORION

[ports]

ORION=rsh bandit "lp -d ps -t"

SIRUS=rsh bandit "lp -d ps -T pcl5"
```

```
LOCAL=lp -d ps

[devices]

HP LaserJet PS=HP4L PostScript,ORION

HP LaserJet PCL=HP4M PCL,SIRIUS

QMS PS=Q2200523 PostScript,LOCAL
```

Whenever you make and save a change with the Printer Setup dialog box, the changes are written to the .WindU file in your home directory.

In your default .WindU file, the [windows] entry appears as follows.

```
[windows]

device=NULL,PostScript,FILE:
```

Because no PPD file is listed (NULL), the default in the Printer Setup dialog box is to print generic PostScript to a file. You can specify the file name and change the type of output to PCL in the Printer Setup dialog box.

To Specify a Default Printer

To specify a default printer using the Printer Setup dialog box, do the following.

1. To display the Options dialog box, from the Printer Setup dialog, click **Options**.
2. From the Printer Name drop-down list, select the desired printer and click **OK**.
3. Click **Save** in the Printer Setup dialog box.

Setting Printer Options

Because printer options vary between printers, use the Printer Setup dialog box to set them. Xprinter reads the PPD file to identify the specific options available for each printer.

1. Display the Printer Setup Dialog box.
2. Set all fields to the desired values.

The following table describes all printer setup fields.

Option	Description
Output Format	Specify whether to send output to a file or to a printer. If you choose Printer Specific, you can send output to any printer type/port combination configured in your `$HOME/.WindU` file. If the port is FILE: (as in this example), Xprinter creates an output file specifically for the specified printer type. If you choose Generic (File Only), print output is sent to an Encapsulated PostScript or generic PCL file.
Printer	Appears only if you select Output Format: Printer Specific. It specifies the name of the default printer to send print output to. Click the Options button to specify a different printer.
File Name	Appears only if you select Output Format: Generic (File Only). Type the name of the print file to create. To pipe print output to a command, type a ! character as the first character, then specify the command to pipe output to. For example, to pipe output to the lp command, enter the following: !lp -d ps.
EPSF PCL4 PCL5	Appears only if you select Output Format: Generic (File Only). Click this button to display a list of output file types and select the desired type. Available types are EPSF (Encapsulated PostScript), PCL4, and PCL5.
Orientation	Specify portrait or landscape.
Scale	To increase the size of the output, specify a value greater than 1.00. To reduce the size, specify a value less than 1.00. For example, a value of 2.00 doubles the size of the output; a value of 0.50 reduces it by half.
Copies	Specify the number of copies to print.

3. To set additional options, such as selecting a new printer or changing the page size, click **`Options.`**
4. Set all options to the desired values.

   The following table describes all printer options.

Option	Description
Printer Name	Changes the Printer in the Setup dialog box. Click the down arrow to display a list of configured printers.
Resolution	Specify printer resolution. Values vary.

Page Size	Specify paper size. Values vary.
Paper tray	Specify tray where paper is located. Values vary.

5. Click **Save** to apply your changes and make them the new default values.

Sending Output to a File

The default `$HOME/.WindU` file contains many printer devices, including the following.

```
HP LaserJet 4L PostScript=HP4L PostScript,FILE:

HP LaserJet 4M PCL Cartridge PCL5=HP4M PCL,FILE:
```

In all of the default entries, the port is FILE:, which is the only reserved port name. If you specify FILE: as the port, Wind/U creates a print file instead of sending output to a printer. When you use a PPD file, you generate PostScript or PCL output that is specific to the printer. If you use Output Format: Generic (File Only), you generate generic Encapsulated PostScript or PCL output.

For example, the HP LaserJet 4L PostScript entry creates a PostScript file that includes the characteristics of the HP 4L PostScript printer. The HP LaserJet 4M PCL entry creates a PCL file that includes the characteristics of the HP LaserJet 4M PCL printer.

You can also print to a file instead of a printer by selecting the Output Format: Generic (file only) option in the Printer Setup dialog box, but doing so creates a generic EPS or PCL print file that does not take advantage of any special characteristics of your particular printer.

Solving Printing Problems

If you have problems printing, use the following hints.

- Start with just printing to a PostScript file. You can use PostScript previewers (on Sun's pageview), to see the file. Adding spooling and PCL support later is easy.
- Ensure that the `$WUHOME/xprinter` files are installed correctly.
- Ensure that you have .WindU in your home directory.
- Check that the printing sample application works with the configuration you are using with your application.
- Ensure that there is a PPD file for the printer you are using. Xprinter requires a PPD file that describes the attributes (paper size, resolution, color capabilities, paper trays, and so forth). for each printer device you want to use. Wind/U includes a number of PPD files for common printers; however, these do not represent all supported printers. If you have a printer that is not included in the PPD files supplied with Wind/U, try the following.
  - Contact the printer manufacturer for the PPD file.
  - Download the PPD file from the Adobe FTP site (`ftp.adobe.com:/pub/adobe/PPDfiles`).
  - Use the PPD file for a similar output device.

If you continue to have problems, submit an SPR to Bristol Technical Support. Be sure to include a copy of your printer output and your .WindU file.

Glossary of Terms

aliases

Aliases, or signal groups, are useful for probing specific groups of nodes.

attribute

Attributes are instructions placed on symbols or nets in an FPGA schematic to indicate their placement, implementation, naming, direction, or other properties.

AutoRoute

AutoRoute automatically routes the objects you specify.

block

A group consisting of one or more logic functions.

component

A component is an instantiation or symbol reference from a library of logic elements that can be placed on a schematic.

constraint

Constraints are specifications for the implementation process. There are several categories of constraints: routing, timing, area, mapping, and placement constraints. Using attributes, you can force the placement of logic (macros) in CLBs, the location of CLBs on the chip, and the maximum delay between flip-flops. PAR does not attempt to change the location of constrained logic. CLBs are arranged in columns and rows on the FPGA device. The goal is to place logic in columns on the device to attain the best possible placement from the point of view of performance and space.

Constraints Editor

The Constraints Editor is a Graphical User Interface (GUI) that can be used to modify or delete existing constraints or to add new constraints to a design.

DC2NCF

DC2NCF (design constraints to netlist constraints file) translates a Synopsys DC file to a Netlist Constraints File (NCF). The DC file is a Synopsys setup file containing constraints for the design.

guided mapping

An existing NCD file is used to "guide" the current MAP run. The guide file may be used at any stage of implementation: unplaced or placed, unrouted or routed.

HDL

HDL (Hardware Description Language).

Implementation Tools

A set of tools that comprise the mainstream programs offered in the Xilinx design implementation tools. The tools are: NGDBuild, MAP, PAR, NGDAnno, TRCE, all the NGD2 translator tools, BitGen, PROMGen, and the FPGA Editor.

LCA file

An LCA file is a mapped file of a Xilinx design produced by an earlier release.

LCA2NCD

LCA2NCD converts an LCA file to an NCD file. The NCD file produced by LCA2NCD can be placed and routed, viewed in the FPGA Editor, analyzed for timing, and back-annotated.

LogiBLOX

Xilinx design tool for creating high-level modules such as counters, shift registers, and multiplexers.

locking

Lock placement applies a constraint to all placed components in your design. This option specifies that placed components cannot be unplaced, moved, or deleted.

macro

A macro is a component made of nets and primitives, flip-flops or latches, that implements high-level functions, such as adders, subtractors, and dividers. Soft macros and RPMs are types of macros. A macro can be unplaced, partially placed or fully placed, and it can also be unrouted, partially routed, or fully routed. See also “physical macro.”

MCS file

An MCS file is an output from the PROMGen program in Intel’s MCS-86 format.

MDF file

An MDF (MAP directive file) file is a file describing how logic was decomposed when the design was originally mapped. The MDF file is used for guided mapping using Xilinx Development System software.

MFP File

An MFP file is generated by the Floorplanner and controls the mapping and placement of logic in the design according to the floorplan created by the user.

MRP file

An MRP (mapping report) file is an output of the MAP run. It is an ASCII file containing information about the MAP run.

NCD file

An NCD (netlist circuit description) file is the output design file from the MAP program, LCA2NCD, PAR, or the FPGA Editor. It is a flat physical design database which may or may not be placed and routed.

NCF file

An NCF (netlist constraints file) file is produced by a synthesis vendor toolset, or by the DC2NCF program. This file contains constraints specified within the toolset. EDIF2NGD and XNF2NGD reads the constraints in this file and adds the constraints to the output NGO file.

NGC File

Binary file containing the implementation of a module in the design. If an NGC file exists for a module, NGDBuild reads this file directly, without looking for a source EDIF or XNF netlist. In HDL design flows, LogiBLOX creates an NGC file to define each module.

NGDAnno

The NGDAnno program distributes delays, setup and hold time, and pulse widths found in the physical NCD design file back to the logical NGD file. NGDAnno merges mapping information from the NGM file, and timing information from the NCD file and puts all this data in the NGA file.

NGA file

An NGA (native generic annotated) file is an output from the NGDAnno run. An NGA file is subsequently input to the appropriate NGD2 translation program.

NGD2EDIF

NGD2EDIF is a program that produces an EDIF 2.1.0 netlist in terms of the Xilinx primitive set, allowing you to simulate pre- and post-route designs.

NGD2VER

NGD2VER is a program that translates your design into a Verilog HDL file containing a netlist description of the design in terms of Xilinx simulation primitives for simulation only.

NGD2VHDL

NGD2VHDL is a program that translates your design into a Vital 3 compliant VHDL file containing a netlist description of your design in terms of Xilinx simulation primitives for simulation only.

NGDBuild

The NGDBuild program performs all the steps necessary to read a netlist file in XNF or EDIF format and create and NGD file describing the logical design.

NGD file

An NGD (native generic database) file is an output from the NGDBuild run. An NGD file contains a logical description of the design expressed both in terms of the hierarchy used when the design was first created and in terms of lower-level Xilinx primitives to which the hierarchy resolves. The NGD file is the input to MAP.

NGM file

An NGM (native generic mapping) file is an output from the MAP run and contains mapping information for the design. The NGM file is an input file to the NGDAnno program.

PAR (Place and Route)

PAR is a program that takes an NCD file, places and routes the design, and outputs an NCD file. The NCD file produced by PAR can be used as a guide file for reiterative placement and routing. The NCD file can also be used by the bitstream generator, BitGen.

path delay

A path delay is the time it takes for a signal to propagate through a path.

PCF file

The PCF file is an output file of the MAP program. It is an ASCII file containing physical constraints created by the MAP program as well as physical constraints entered by you. You can edit the PCF file from within the FPGA Editor.

physical Design Rule Check (DRC)

Physical Design Rule Check (DRC) is a series of tests to discover logical and physical errors in the design. Physical DRC is applied from the FPGA Editor, BitGen, PAR, and Hardware Debugger. By default, results of the DRC are written into the current working directory.

physical macro

A physical macro is a logical function that has been created from components of a specific device family. Physical macros are stored in files with the extension .nmc.

pin

A pin can be a symbol pin or a package pin. A package pin is a physical connector on an integrated circuit package that carries signals into and out of an integrated circuit. A symbol pin, also referred to as an instance pin, is the connection point of an instance to a net.

pinwires

Pinwires are wires which are directly tied to the pin of a site (i.e. CLB, IOB, etc.)

route

The process of assigning logical nets to physical wire segments in the FPGA that interconnect logic cells.

route-through

A route that can pass through an occupied or an unoccupied CLB site is called a route-through. You can manually do a route-through in the FPGA Editor. Route-throughs provide you with routing resources that would otherwise be unavailable.

states

The values stored in the memory elements of a device (flip-flops, RAMs, CLB outputs, and IOBs) that represent the state of that device for a particular readback (time). To each state there corresponds a specific set of logical values.

TRCE

TRCE (Timing Reporter and Circuit Evaluator) “trace” is a program that will automatically perform a timing analysis on a design using available timing constraints. The input to TRCE is a mapped NCD file and, optionally, a PCF file. The output from TRCE is an ASCII timing report which indicates how well the timing constraints for your design have been met.
(Historical note: TRCE should not be confused with the UNIX trace command. The UNIX trace command is used to trace system calls and signals).

TWR file

A TWR (Timing Wizard Report) file is an output from the TRCE program. A TWR file contains a logical description of the design expressed both in terms of the hierarchy used when the design was first created and in terms of lower-level Xilinx primitives to which the hierarchy resolves.

wire

A wire is either: 1) a net or 2) a signal.

UCF file

A UCF (user constraints file) contains user-specified logical constraints.

Index

P

R

S

T

U

V

W

X