Single Chip Crypto Lab Using PR/ISO Flow with the Virtex-5 Family

XAPP1135 (v1.1.3) June 19, 2013





Notice of Disclaimer

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of the Limited Warranties which can be viewed at http://www.xilinx.com/warranty.htm. IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe performance; you assume sole risk and liability for use of Xilinx products in Critical Applications: http://www.xilinx.com/warranty.htm#critapps.

© 2009–2013 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.

Revision History

| Date | Version | Revision |
|----------|---------|--|
| 06/15/09 | 1.0 | Initial Xilinx release. |
| 04/15/10 | 1.1 | Updated for ISE 11.4 and Trusted Routing throughout. |
| | | Removed "Synthesize Modules for Use in the Partial Reconfiguration Flow" section from Chapter 1 and put into new Chapter 2. Removed "Using the PlanAhead Tool to Place the Trusted Bus Macros" section in Chapter 3. |
| | | Removed Appendix A: Trusted Bus Macro Rules. |
| 09/21/10 | 1.1.1 | Added EAR banner to first page. |
| 08/29/11 | 1.1.2 | Removed EAR banner. Updated disclaimer. |
| 06/19/13 | 1.1.3 | In Reference Design Files, added URL to download reference design. |

The following table shows the revision history for this document.

Table of Contents

| Revision History | . 2 |
|--|----------------|
| Preface: About This Guide | |
| Guide Contents | . 5 |
| Additional Resources | . 5 |
| Conventions Typographical Online Document | .6 .6 .6 |

Chapter 1: Design Challenge

| Reference Design Files | 10 |
|--|----|
| Reference Design Checklist | 10 |
| Install Reference Design Files into Target Directories | 11 |

Chapter 2: Synthesizing Modules for Partial Reconfiguration Flow

| Synthesize the top Module | 13 |
|----------------------------------|----|
| Synthesize the aes Module | 20 |
| Synthesize the aes_r Module | 27 |
| Synthesize the compare Module | 34 |

Chapter 3: Implementing the System

| PlanAhead Project Entry 43 Launch the PlanAhead Tool 43 |
|--|
| PlanAhead Project Creation |
| Place I/Os with the PlanAhead Tool 49 |
| Place DCM with the PlanAhead Tool |
| Setting Up Timing Constraints with the PlanAhead Tool |
| Defining RP or ISO Partitions with the PlanAhead Tool |
| Defining Attributes for each RP or ISO Partition with the PlanAhead Tool 62 |
| Set Up the Area Groups for the RP Regions with the PlanAhead Tool 64 |
| Running Design Rule Checks88 |
| Running TimeAhead with the PlanAhead Tool 90 |
| Design Flow Progress |
| Exporting the Design |

Chapter 4: Running the Isolation Verification Tool Against the UCF

| Creating the File Used to Run the IVT UCF Test | 93 |
|--|----|
| Creating the Pin Isolation Group File | 94 |
| Running the IVT UCF Test | 95 |

| Examining the Output from the IVT UCF Test | |
|--|---|
| Chapter 5: Implementing the Design with the PlanAhead Tool Generating and Running an Implementation | |
| Chapter 6: Verifying the Design with the NCD Isolation Verification Tool | ו |
| Creating the File Used to Run the IVT NCD Test | |
| Running the IVT NCD Test 100 | |
| Examining the Output from the IVT NCD Test | |
| Design Flow Progress 101 | |



Preface

About This Guide

This lab covers the creation and implementation of a single-chip crypto (SCC) system, utilizing an isolated, redundant AES module. Complete step-by-step instructions are given for the entire process.

Guide Contents

This manual contains the following chapters:

- Chapter 1, "Design Challenge," describes the SCC design and the goals of the lab.
- Chapter 2, "Synthesizing Modules for Partial Reconfiguration Flow," describes the steps in the bottom-up synthesis flow.
- Chapter 3, "Implementing the System," gives step-by-step instructions for implementing the SCC design.
- Chapter 4, "Running the Isolation Verification Tool Against the UCF," covers running IVT against the pre-placed-and-routed design.
- Chapter 5, "Implementing the Design with the PlanAhead Tool," details placing and routing the SCC design.
- Chapter 6, "Verifying the Design with the NCD Isolation Verification Tool," describes running the verification tool on the placed-and-routed design.

Additional Resources

To find additional documentation, see the Xilinx website at:

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

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

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

Conventions

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

Typographical

The following typographical conventions are used in this document:

| Convention | Meaning or Use | Example |
|----------------|--|--|
| Courier font | Messages, prompts, and program files that the system displays | speed grade: - 100 |
| Courier bold | Literal commands that you enter in a syntactical statement | ngdbuild design_name |
| Helvetica bold | Commands that you select from a menu | $File \to Open$ |
| | Keyboard shortcuts | Ctrl+C |
| | Variables in a syntax statement for which you must supply values | ngdbuild design_name |
| Italic font | References to other manuals | See the <i>Command Line Tools User Guide</i> for more information. |
| | Emphasis in text | If a wire is drawn so that it overlaps the pin of a symbol, the two nets are <i>not</i> connected. |

Online Document

The following conventions are used in this document:

| Convention | Meaning or Use | Example |
|-----------------------|--|---|
| Blue text | Cross-reference link to a location in the current document | See the section "Additional Resources" for details. Refer to "Title Formats" in Chapter 1 for details. |
| Blue, underlined text | Hyperlink to a website (URL) | Go to http://www.xilinx.com for the latest speed files. |



Chapter 1

Design Challenge

From the instructions in this chapter, a Single Chip Crypto system can be created and implemented utilizing an isolated, redundant AES module targeting a Virtex®-5 or Virtex-5Q FPGA utilizing the ISE® 11.4 and PlanAheadTM 11.4 development tools. It is necessary to use the ISE 11.4 software for this lab. Figure 1-1 is a hierarchical diagram of the various VHDL sub-blocks used in the implementation of the design. This Single Chip Crypto lab shows how to develop a dual-AES design. The user can expand upon this to create their own custom design based on the design methodology.

Type 1 Virtex-5 FPGA Crypto applications require defense-grade (XQ) devices for mask control. The design example used in this application note was created using a non defense-grade Virtex-5 XC5VLX85-FF676-2 device.



Figure 1-1: VHDL Design Hierarchical Block Diagram

Figure 1-2 shows the design flow used for the Single Chip Crypto lab.



Figure 1-2: SCC System Design Flow

Figure 1-3 shows the partitioning and I/O mapping for an XC5VLX85-2FF676 device.



X1135_c1_02_040710



Figure 1-4 is a view of the completed XSCC Reference design being displayed in the Xilinx FPGA Editor tool. The design incorporates two AES algorithm blocks: AES and AES_r. Both of the AES blocks are driven by the same input data and key. In Figure 1-4, the two AES blocks are the sideways C-shaped regions. Both of the AES blocks are tied to the

compare block, which compares the outputs of the AES blocks. If the outputs of the AES blocks do not match, the compare block sends an alert to the user indicated by an LED. Both AES blocks have an input that allows the user to inject an error. However, only the AES block has the error injection input tied to an external pushbutton.



Figure 1-4: FPGA Editor View: Implementation of the Single Chip Crypto Flow

Reference Design Files

Reference Design Checklist

The design files for this application note can be downloaded from:

https://secure.xilinx.com/webreg/clickthrough.do?cid=343393

The design checklist in Table 1-1 includes simulation, implementation, and hardware details for the reference design.

| Parameter | Description |
|---|--------------------|
| General | |
| Developer Name | Xilinx |
| Target devices | Virtex-5 LX85 FPGA |
| Source code provided | Y |
| Source code format | VHDL |
| Design uses code/IP from existing Xilinx application note/ reference designs, CORE Generator [™] software, or 3rd party | N |
| Simulation | |
| Functional simulation performed | Y |
| Timing simulation performed | Y |
| Testbench used for functional and timing simulations | Y |
| Testbench format | VHDL |
| Simulator software/version used | ISE 11.4 software |
| SPICE/IBIS simulations | N |
| Implementation | |
| Synthesis software tools/version used | XST |
| Implementation software tools/versions used | ISE 11.4 software |
| Static timing analysis performed | Y |
| Hardware Verification | |
| Hardware verified | N |
| Hardware platform used for verification | N/A |

Install Reference Design Files into Target Directories

These steps describe the process for installing the reference design files:

Note: It is best if the design files are unzipped onto a directory without any spaces in the path.

- 1. Copy XAPP1135.zip to the Windows desktop.
- 2. Double-click on XAPP1135. zip and unzip the contents to the desired location.
- 3. The project files are placed in the following directories:

\Xilinx_Design\source\ \Xilinx_Design\ivt \Xilinx_Design\synthesis\

- \Xilinx_Design\PlanAhead
- \Xilinx_Design\BuildScripts
- \Xilinx_Design\results



Chapter 2

Synthesizing Modules for Partial Reconfiguration Flow

This chapter describes the steps that take the user through a bottom-up synthesis flow, which is the flow used for SCC designs to maintain isolation. The top module is synthesized first followed by each of the lower-level modules (AES, AES_R, COMPARE).

Synthesize the top Module

These steps describe how to synthesize the top module:

1. Start the ISE 11.4 software:

Start $\rightarrow~$ All Programs $\rightarrow~$ Xilinx ISE Design Suite 11 $\rightarrow~$ ISE $~\rightarrow~$ Project Navigator

2. Create a new ISE 11.4 software project:

 $\textbf{File} \rightarrow \textbf{ New Project}$

3. As shown in Figure 2-1, set the Project Location to **\Xilinx_Design\synthesis**, set the Project Name to **top**, and click **Next**.

| | 3 | XILINX | 8 |
|--|---|--------|---|
|--|---|--------|---|

| 🜁 New Project Wiza | rd | × |
|---|--|--------|
| Create New Project Specify project loc | cation and type. | |
| Enter a name, locati | ons, and comment for the project | 1 |
| Name: | SCC_LAB_TOP | |
| Location: | C:\Xilinx_Design\synthesis\SCC_LAB_TOP | |
| Working Directory: | C:\Xilinx_Design\synthesis\SCC_LAB_TOP | |
| Description: | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| Select the type of to | op-level source for the project | |
| Top-level source typ | e: | |
| HDL | | |
| | | |
| | | |
| | | |
| More Info | Next > Cancel | |
| | X1135 c1 07 | 040710 |

Figure 2-1: New Project Wizard (Create New Project)

4. Choose an XC5VLX85-FF676-2 FPGA as the target device (see Figure 2-2).

| electione device and design now for the | is project | |
|---|---------------------|---|
| Property Name | Value | |
| Product Category | All | ~ |
| Family | Virtex5 | ~ |
| Device | XC5VLX85 | ~ |
| Package | FF676 | * |
| Speed | -2 | * |
| Top-Level Source Type | HDL | ~ |
| Synthesis Tool | XST (VHDL/Verilog) | * |
| Simulator | ISim (VHDL/Verilog) | * |
| Preferred Language | VHDL | ~ |
| Property Specification in Project File | Store all values | * |
| Manual Compile Order | | |
| Enable Enhanced Design Summary | | |
| Enable Message Filtering | | |
| Display Incremental Messages | | |

X1135_c1_08_040710

Figure 2-2: New Project Wizard (Device Properties)

- 5. Click **Next** twice, click the **Add Source** button, navigate to the source\design directory, and add the top_package.vhd and SCC_LAB_TOP.vhd files to the project.
- 6. Uncheck **Copy to Project** for all sources and then click **Next** (see Figure 2-3).

| 📧 Nev | v Project Wizard | | |
|-------|--|---|------------------|
| Ado | Existing Sources Adding existing sources is optional. Additional sources can >Add Source" or "Project->Add Copy of Source" command | be added after the project is created using s. | the "Project- |
| Add | existing sources | | |
| | Source File | Copy to Project | Add Source |
| 1 | top_package.vhd | | Pemoue |
| 2 | SCC_LAB_TOP.vhd | | Kelliove |
| 3 | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| Mo | e Info | < Back Next > | Cancel |
| | | | X1135 c1 09 0407 |

Figure 2-3: New Project Wizard (Add Existing Sources)

7. Click **Finish** (see Figure 2-4).

| 🐱 New Project Wizard | × |
|--|---|
| Project Summary Project Navigator will create a new project with the following specifications. | |
| Project: Project Name: SCC_LAB_TOP Project Path: C:\Xilinx_Design\synthesis\SCC_LAB_TOP Working Directory: C:\Xilinx_Design\synthesis\SCC_LAB_TOP Description: Top Level Source Type: HDL | |
| Device: Device Family: Virtex5 Device: xc5v1x85 Package: ff676 Speed: -2 | |
| Synthesis Tool: XST (VHDL/Verilog) Simulator: ISim (VHDL/Verilog) Preferred Language: VHDL Property Specification in Project File: Store all values Manual Compile Order: false | |
| Enhanced Design Summary: enabled Message Filtering: disabled Display Incremental Messages: disabled | |
| Existing Sources: SCC_LAB_TOP.vhd - use from current location top_package.vhd - use from current location | |
| More Info < Back Finish Cancel | כ |

Figure 2-4: New Project Wizard (Project Summary)

8. Click **OK** (see Figure 2-5).

| 1 | File Name | Association | | Libra | ry |
|-----|-------------------|-------------|---|-------|----|
| 1 (| SCC_LAB_TOP.vhd | All | * | work | ~ |
| 2 | 🕗 top_package.vhd | All | * | work | ~ |
| | | | | | |
| | | | | | |
| | | | | | |

Figure 2-5: Adding Source Files

The ISE Project Navigator Window should look like the window shown in Figure 2-6.

Note: The question marks (?) by all blocks that are not coded at the top level are expected because only the top level is being synthesized at this time. All other blocks are out of context and are treated as black boxes.



X1135_c1_12_031810

Figure 2-6: Project Navigator Window

9. Open the SCC_LAB_TOP.vhd file and locate the buffer_type attributes:

```
attribute buffer_type: string;
attribute buffer_type of push_button : signal is "none";
attribute buffer_type of reset : signal is "none";
attribute buffer_type of led : signal is "none";
```

The buffer_type attribute directs XST to disable I/O insertion on these ports. The buffer_type attribute is necessary to guarantee I/Os are included in the lower-level modules, and therefore are part of the isolated regions. However, in SCC_LAB_TOP.vhd, clk is driven to all modules. Clock networks are not required to be isolated. As such, do not apply this attribute to the CLK pin of the top-level code.

10. To run XST synthesis, either right-click on the Synthesize-XST icon in the Processes window and click **Run** or simply double-click **Synthesize-XST**. After synthesis is complete, close the top project.

Synthesize the aes Module

These steps describe how to synthesize the aes module:

- 1. Start the ISE 11.4 software. Start \rightarrow All Programs \rightarrow Xilinx ISE Design Suite 11 \rightarrow ISE \rightarrow Project Navigator
- 2. Create a new ISE 11.4 software project:

$\textbf{File} \rightarrow \textbf{New Project}$

3. Set the Project Location to:

\Xilinx_Design\synthesis

- 4. Set the Project Name to **aes**.
- 5. Click **Next** (see Figure 2-7).

| - Enter a name, ioco | itions, and comment for the project | |
|----------------------|-------------------------------------|---|
| Name: | aes | |
| Location: | C:\Xilinx_Design\synthesis\aes | |
| Working Directory | : C:\Xilinx_Design\synthesis\aes | |
| Description: | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| Select the type of | top-level source for the project | |
| Top-level source t | ype: | |
| HDL | | ~ |
| | | |

Figure 2-7: New Project Wizard (Create New Project)

6. Click **Next** (see Figure 2-8).

| Property Name | value | |
|--|---------------------|---|
| Duaduat Cata anns | | |
| Product Category | All | ~ |
| Pamily | Virtex5 | ~ |
| Device | AC34LA03 | ~ |
| Раскаде Стора | 0 | Y |
| Speed | -2 | ~ |
| Top-Level Source Type | HDL | v |
| Synthesis Tool | XST (VHDL/Verilog) | ~ |
| Simulator | ISim (VHDL/Verilog) | ~ |
| Preferred Language | VHDL | ~ |
| Property Specification in Project File | Store all values | ~ |
| Manual Compile Order | | |
| Enable Enhanced Design Summary | | |
| Enable Message Filtering | | |
| Display Incremental Messages | | |
| | | |
| | | |
| | | |

Figure 2-8: New Project Wizard (Device Properties)

- 7. Click **Next** twice.
- 8. To add the key_expander.vhd, aes.vhd, and aes_package.vhd files to the project, click the **Add Source** button and navigate to the source\design directory.
- 9. Uncheck Copy to Project for all sources and then click Next (see Figure 2-9).

| Nev | w Project Wizard | E E |
|-----|--|--|
| Ado | d Existing Sources Adding existing sources is optional. Additional sources ca >Add Source" or "Project->Add Copy of Source" comma | an be added after the project is created using the "Project- nds. |
| Add | d existing sources | |
| | Source File | Copy to Project Add Source |
| 1 | aes.vhd | |
| 2 | aes_package.vhd | |
| 3 | key_expander.vhd | |
| 4 | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| Mor | re Info | <pre>< Back Next > Cancel</pre> |

X1135_c1_15_040710

Figure 2-9: New Project Wizard (Add Existing Sources)

10. Click **Finish** (see Figure 2-10).

| New Project Wizard |
|---|
| Project Summary Project Navigator will create a new project with the following specifications. |
| Project: |
| Project Name: aes |
| Project Path: C:\Xilinx_Design\synthesis\aes |
| Working Directory: C:\Xilinx_Design\synthesis\aes |
| Description: |
| Top Level Source Type: HDL |
| Device: |
| Device Family: Virtex5 |
| Device: xc5v1x85 |
| Package: ff676 |
| Speed: -2 |
| Synthesis Tool: XST (VHDL/Verilog) |
| Simulator: ISim (VHDL/Verilog) |
| Preferred Language: VHDL |
| Property Specification in Project File: Store all values |
| Manual Compile Order: false |
| Enhanced Design Summary: enabled |
| Message Filtering: disabled |
| Display Incremental Messages: disabled |
| Existing Sources: |
| aes.vhd - use from current location |
| aes_package.vhd - use from current location |
| key_expander.vhd - use from current location |
| |
| More Info < Back Finish Cancel |
| X1135 ct 16 0 |

Figure 2-10: New Project Wizard (Project Summary)

11. Click **OK** (see Figure 2-11).

| 15 | 🔤 Adding Source Files 🔀 | | | | | | |
|----|-------------------------|--|--|--------------|---|--------|--|
| | The allo :he | following allows you to se ws you to specify the Desi project. | ee the status of the ign View association | e so n fo | ource files being added to the project, and or sources which are successfully added to | | |
| | | File Name | Association | | Library | | |
| | 1 | 📀 aes.vhd | All 🔤 | ~ | work | - | |
| | 2 | 🧭 aes_package.vhd | All | ~ | work | - | |
| | 3 | 🧭 key_expander.vhd | All | ~ | work | - | |
| | | | | | | | |
| | Add | ling files to project: | <u></u> | | 3 of 3 files (0 errors | s) | |
| | | | | _ | OK Cancel Help | | |
| | | | | | X1135_c1_17 | _04071 | |

Figure 2-11: Adding Source Files

The ISE Project Navigator Window should look like the window shown in Figure 2-12.

Note: All blocks are defined. There should be no question marks (?) by any module. All modules at this level and below are in context.



Figure 2-12: Project Navigator Window

12. Open the aes.vhd file and locate the section containing the buffer_type attributes:

| attribute | buffer_type: | : st | cring; | | | | |
|-----------|------------------------|------|-----------|---|--------|----|---------|
| attribute | <pre>buffer_type</pre> | of | clk | : | signal | is | "none"; |
| attribute | <pre>buffer_type</pre> | of | start | : | signal | is | "none"; |
| attribute | <pre>buffer_type</pre> | of | mode | : | signal | is | "none"; |
| attribute | <pre>buffer_type</pre> | of | load | : | signal | is | "none"; |
| attribute | <pre>buffer_type</pre> | of | key | : | signal | is | "none"; |
| attribute | <pre>buffer_type</pre> | of | data_in | : | signal | is | "none"; |
| attribute | <pre>buffer_type</pre> | of | reset_out | : | signal | is | "none"; |
| attribute | <pre>buffer_type</pre> | of | data_out | : | signal | is | "none"; |
| attribute | <pre>buffer_type</pre> | of | done | : | signal | is | "none"; |

The buffer_type attribute directs the XST tool to disable I/O insertion on the specified ports. By default, XST inserts an I/O on all ports. The buffer_type attribute is necessary to guarantee I/Os are not placed at this level in the hierarchy either because the I/Os are placed at the top level (such as CLK) or because the port is a direct connection to a port of another instance.

Note: The reset and push_button signals do not have an attribute associated with them because they are "owned" by AES and therefore require an I/O to be inferred within the aes module rather than the top. The clk I/O was inferred when the top was synthesized. The other signals are internal to the FPGA; therefore they have the attribute **signal is "none"** applied.

13. In the aes.vhd source code file, the LUT instantiation between reset and reset_out is necessary for Trusted Routing rules. Refer to XAPP1134, Developing Secure Designs

Using the Virtex-5 Family, for more details on Trusted Routing rules. A section of the VHDL code for the LUT instantiation is shown here:

-- Instantiate LUT buffers on nets that either drive two different regions -- or are feedthrough's from one region to the next. This prevents a single

-- net being placed "shorting" three regions together. Trusted Bus Macros

-- fulfilled this requirement automatically.

lut_reset_out : LUT1
GENERIC MAP (INIT => X"2")
PORT MAP (I0 => reset, 0 => reset_out);

14. Right-click on the Synthesize-XST icon in the Processes window and select **Process Properties...** Set Optimization Goal to **Area**, set Optimization Effort to **High**, and click **OK** (see Figure 2-13).

| Synthesis Options optmode Optimization Goal Area optlevel Optimization Effort Ipp power Power Reduction Ipp -uc Synthesis Constraints File Ipp -uc Global Optimization Goal AltClotAltets -reduct_hierarchy Neel Herarchy As Optimized -reducers Read Cores Ipp -reducers Read Cores Ipp -reducers Cross Search Directories Ipp -reducers Cross Search Directories Ipp -sdc_uctilization_ratio DSP Utilization Ratio 100 -dram_utilization_ratio DSP Utilization Ratio 100 -case Case Case Case | Synthesis Options HDL Options Xilinx Specific Options | -opt_mode -opt_level -power -iuc | Optimization Goal Optimization Effort Power Reduction | Area High |
|--|---|---|---|--|
| HDL Options villex Specific Options inp. Jevel Optimization Effort power Power Reduction iuc Use Synthesis Constraints File iso Library Search Order iso Cores Search Directories iso Cores Search Directories iso Cores Search Directories iso_deliniter Bus Deliniter usite_deliniter Bus Deliniter usite_deliniton_ratio DO iste_deliniter BRAM Utilization Ratio 100 iste_deliniter BRAM Utilization Ratio 100 iste_deliniter Generics, Parameters | HDL Options Xilinx Specific Options | -opt_level -power -iuc | Optimization Effort Power Reduction | High |
| power Power Reduction -uc Use Synthesis Constraints File -uc Synthesis Constraints File -uc Synthesis Constraints File -uc Use Synthesis Constraints File -uc Global Optimization Goal -restlactore Ves -restlactore Ves -restlactore Ves -restlactore Read Cores -sd Cores Search Directories | | -power -iuc | Power Reduction | |
| +uc Use Synthesis Constraints File -uc Synthesis Constraints File -lso Lthrary Search Order -keep_hierarchy Keep Hierarchy -netlist_hierarchy No -netlist_hierarchy No -glob_opt Global Optimization Goal AllClockNets -glob_opt Global Optimization Goal AllClockNets -read_cores V - -sd Cores Search Directories -vwrke_timing_constraints -ross_clock_analysis Cross Clock Analysis -ross_clock_analysis Cross Clock Analysis -ross_clock_analysis -ross_clock_analysis Cross Clock Analysis -ross_clock_analysis -ross_clock_analysis -reacy_ulitaction_ratio BUF Perior Nation 100 -des_ulitaction_ratio DSP Utilization Ratio 100 -case Case Maintain -verilog2001 Verilog 2001 Verilog 2001 -verilog2001 Verilog | | -iuc | | |
| -uc Synthesis Constraints File -lso Library Search Order -leep_hierarchy Keep Hierarchy No -hetist_hierarchy Netist Hierarchy As Optimized -jolo_opt Global Optimization Goal AllClockNets -rtWiew Generate RTL Schematic Yes -read_cores V - -ed Cores Search Directories | | | Use Synthesis Constraints File | |
| -Iso Library Search Order -keep_hierarchy Keep Hierarchy No -netlist_hierarchy Netlist Hierarchy As Optimized -glob_opt Global Optimization Goal AllClockNets -triview Generate RTL Schematic Yes -read_cores ✓ - -sd Cores Search Directories ✓ -vrite_timing_constraints Write Timing Constraints | | -uc | Synthesis Constraints File | |
| -keep_hierarchy Keep Hierarchy No -netist_hierarchy Netist Hierarchy As Optimized -glob_opt Global Optimization Goal AllClockNets -rtiview Generate RTL Schematic Yes -read_ores Read Cores V | | -lso | Library Search Order | |
| -netlist_hierarchy Netlist Hierarchy As Optimized -glob_opt Global Optimization Goal AllClockNets -rthview Generate RTL Schematic Yes -read_cores Read Cores Y -sd Cores Search Directories ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,, | | -keep_hierarchy | Keep Hierarchy | No |
| -glob_opt Global Optimization Goal AllClockNets -rtiview Generate RTL Schematic Yes -read_cores Read Cores ✓ -sd Cores Search Directories ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,, | | -netlist_hierarchy | Netlist Hierarchy | As Optimized |
| rtView Generate RTL Schematic Yes read_cores Read Cores V sd Cores Search Directories ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,, | | -glob_opt | Global Optimization Goal | AllClockNets |
| read_cores Read Cores sd Cores Search Directories write_timing_constraints Write Timing Constraints cross_clock_analysis Cross Clock Analysis cross_clock_analysis Cross Clock Analysis -hierarchy_separator Hierarchy Separator -hierarchy_separator Hierarchy Separator -hierarchy_separator Bus Delimiter Subcedilization_ratio LUT-FF Pairs Utilization Ratio ofsde_utilization_ratio DSP Utilization Ratio ofsde_utilization_ratio DSP Utilization Ratio odsp_utilization_ratio DSP Utilization Ratio verilog2001 Verilog 2001 verilog2001 Verilog 2001 -verilog2001 Verilog Include Directories -verilog2001 Verilog Include Directories -verilog2001 Verilog Macros -define Verilog Macros -define Verilog Macros -define Verilog Macros | | -rtlview | Generate RTL Schematic | Yes |
| sd Cores Search Directories •write_timing_constraints Write Timing Constraints •cross_clock_analysis Cross Clock Analysis •cross_clock_analysis Cross Clock Analysis •hierarchy_separator Hierarchy Separator / •hierarchy_separator Hierarchy Separator / •hierarchy_separator Bus Delimiter <> •slic_dilization_ratio D10 •bram_utilization_ratio DSP Utilization Ratio 100 •define Case Maintain •case Case Maintain •verilog2001 Work Directory :C-LAB-UsersGuide_V5-Design/Xilimx_Design/synthesis/ses/x •verilog2001 Werilog Include Directories •verilog2001 Werilog Include Directories •define Werilog Macros | | -read_cores | Read Cores | |
| write_timing_constraints Write Timing Constraints -cross_clock_analysis Cross Clock Analysis -cross_clock_analysis Cross Clock Analysis -dross_clock_analysis Cross Clock Analysis -hierarchy_separator Hierarchy Separator -bus_delimiter Bus Delimiter -slice_utilization_ratio LUT-FF Pairs Utilization Ratio -drosm_utilization_ratio DSP Utilization Ratio -verilog2001 Work Directory -verilog2001 Werilog Include Directories -verilog20201 Werilog Macros -define Werilog Macros -define Verilog Macros -define Verilog Macros | | -sd | Cores Search Directories | |
| -cross_clock_analysis Cross Clock Analysis □ -hierarchy_separator Hierarchy Separator / -bus_delimiter Bus Delimiter <> -slice_utilization_ratio LUT-FF Pairs Utilization Ratio 100 -dram_utilization_ratio DSP Utilization Ratio 100 -dsp_utilization_ratio DSP Utilization Ratio 100 -case Case Maintain -verilog2001 Work Directory :C-LAB-UsersGuide_V5-Design/Xilinx_Design/synthesis/aes/x -verilog2001 Werlog 2001 ✓ -verilog2001 Werlog Macros | | -write_timing_constraints | Write Timing Constraints | |
| -hierarchy_separator Hierarchy Separator / -bus_delimiter Bus Delimiter <> -bus_delimiter Bus Delimiter <> -slice_utilization_ratio LUT-FF Pairs Utilization Ratio 100 -bram_utilization_ratio DSP Utilization Ratio 100 -dsp_utilization_ratio DSP Utilization Ratio 100 -case Case Maintain -case Case Maintain -verilog2001 Werlog 2001 ✓ -verilog2001 Werlog Include Directories … -define Verlog Macros … -define Verlog Macros … -define Verlog Verlog Verlog ✓ -Verperty display level: Advanced Y ✓ Display switch names Def | | -cross_clock_analysis | Cross Clock Analysis | |
| -bus_delimiter Bus Delimiter <> -bus_delimiter Bus Delimiter <>> -slice_utilization_ratio LUT-FF Pairs Utilization Ratio 100 -bram_utilization_ratio BRAM Utilization Ratio 100 -dsp_utilization_ratio DSP Utilization Ratio 100 -csse Case Maintain -csse Case Maintain -verilog2001 Work Directory C-LAB-UsersGuide_VS-Design/Xilinx_Design/synthesis/aes/xx -verilog2001 Werlog Include Directories … -verliog2001 Werlog Include Directories … -define Werlog Macros … -define Verliog Macros … -define Verlog Store | | -hierarchy_separator | Hierarchy Separator | 1 |
| +slice_utilization_ratio LUT-FF Pairs Utilization Ratio 100 -bram_utilization_ratio BRAM Utilization Ratio 100 -dsp_utilization_ratio DSP Utilization Ratio 100 -case Case Maintain -case Case Maintain -verilog2001 Work Directory C-LAB-UsersGuide_V5-Design/Xilinx_Design/synthesis/aes/x -verilog2001 Verilog 2001 ✓ -verilog2001 Verilog Include Directories | | -bus_delimiter | Bus Delimiter | \diamond |
| -bram_utilization_ratio BRAM Utilization Ratio 100 -dsp_utilization_ratio DSP Utilization Ratio 100 -case Case Maintain -case Work Directory :C-LAB-UsersGuide_V5-Design/Xilinx_Design/synthesis/aes/x -verilog2001 Verilog 2001 V -verilog2001 Verilog Include Directories | | -slice_utilization_ratio | LUT-FF Pairs Utilization Ratio | 100 |
| -dsp_utilization_ratio DSP Utilization Ratio 100 -case Case Maintain Work Directory :C-LAB-UsersGuide_V5-Design/Xilinx_Design/synthesis/aes/x set -xsthdpini HDL INI File -verligo2001 Verlog 2001 -verligo2001 Verlog Include Directories -qenerics Generics, Parameters -define Verlog Macros Other XST Command Line Options | | -bram_utilization_ratio | BRAM Utilization Ratio | 100 |
| -case Case Maintain Work Directory IC-LAB-UsersGuide_V5-Design/Xilinx_Design/synthesis/aes/x set -xsthdpini HDL INI File -verliog2001 Verliog 2001 -verliog2001 Verliog Include Directories -verliog2001 Verliog Include Directories -verliog2001 Verliog Macros -define Verliog Macros Other XST Command Line Options Verliog Synthesize | | -dsp_utilization_ratio | DSP Utilization Ratio | 100 |
| Work Directory IC-LAB-UsersGuide_V5-Design/Xilinx_Design/synthesis/aes/X set -xsthdpini HDL INI File -verlig2001 Werliog 2001 -vligincdir Verliog Include Directories -qenerics Generics, Parameters -define Werliog Macros Other XST Command Line Options | | -case | Case | Maintain |
| set -xsthdpini HDL INI File -verilog2001 Verilog 2001 V -vlgincdir Verilog Include Directories -generics Generics, Parameters -define Verilog Macros Other XST Command Line Options Property display level: Advanced V Display switch names Def | | | Work Directory | :C-LAB-UsersGuide_V5-Design/Xilinx_Design/synthesis/aes/xs |
| -verilog2001 Verilog 2001 V -vlgincdir Verilog Include Directories -generics Generics, Parameters -define Verilog Macros Other XST Command Line Options Property display level: Advanced V Display switch names Define | | set -xsthdpini | HDL INI File | |
| -vlgincdir Verllog Include Directories -generics Generics, Parameters -define Verllog Macros Other XST Command Line Options | | -verilog2001 | Verilog 2001 | |
| -generics Generics, Parameters -define Verilog Macros Other XST Command Line Options Property display level: Advanced ♥ ♥ Display switch names Defined | | -vlgincdir | Verilog Include Directories | |
| -define Verilog Macros Other XST Command Line Options Property display level: Advanced ♥ ♥ Display switch names Definition | | -generics | Generics, Parameters | |
| Other XST Command Line Options Property display level: Advanced V Display switch names Def | | -define | Verilog Macros | |
| Property display level: Advanced 💙 🔽 Display switch names Def | | | Other XST Command Line Options | ; |
| Property display level: Advanced 💙 🗸 Display switch names 🛛 Def | | | | |
| | | | Property disp | olay level: Advanced V Display switch names Defa |

Figure 2-13: Process Properties

- 15. To run XST synthesis, either right-click on the Synthesize-XST icon in the Processes window and select **Run** or simply double-click **Synthesize-XST**.
- 16. After synthesis is complete, close the aes project.

Synthesize the aes_r Module

These steps describe how to synthesize the aes_r module:

- 1. Start the ISE 11.4 software: Start \rightarrow All Programs \rightarrow Xilinx ISE Design Suite 11 \rightarrow ISE \rightarrow Project Navigator
- 2. Create a new ISE 11.4 project:

$\textbf{File} \rightarrow \textbf{New Project}$

- 3. Set the Project Location to **\Xilinx_Design\synthesis**.
- 4. Set the Project Name to **aes_r**.
- 5. Click **Next** (see Figure 2-14).

| Enter a name, locat | ons, and comment for the project | |
|----------------------|----------------------------------|--|
| Name: | aes_r | |
| Location: | C:\Xilinx_Design\synthesis\aes_r | |
| Working Directory: | C:\Xilinx_Design\synthesis\aes_r | |
| Description: | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| Select the type of t | p-level source for the project | |
| Top-level source typ | be: | |
| HDL | | |
| | | |
| | | |
| | | |
| | | |

Figure 2-14: New Project Wizard (Create New Project)

6. Choose an XC5VLX85-FF676-2 as the target device (see Figure 2-15).

| | 1 | |
|--|---------------------|---|
| Property Name | Value | |
| Product Category | All | ~ |
| Family | Virtex5 | ~ |
| Device | XC5VLX85 | ~ |
| Package | FF676 | ~ |
| Speed | -2 | ~ |
| Top-Level Source Type | HDL | ~ |
| Synthesis Tool | XST (VHDL/Verilog) | ~ |
| Simulator | ISim (VHDL/Verilog) | ~ |
| Preferred Language | VHDL | |
| Property Specification in Project File | Store all values | ~ |
| Manual Compile Order | | |
| Enable Enhanced Design Summary | | |
| Enable Message Filtering | | |
| Display Incremental Messages | | |
| | | |
| | | |
| | | |
| | | |

Figure 2-15: New Project Wizard (Device Properties)

- Click Next twice. Add the key_expander.vhd, aes_r.vhd, and aes_package.vhd files to the project by clicking the Add Source button and navigating to the source\design directory.
- 8. Uncheck **Copy to Project** for all sources and then click **Next** (see Figure 2-16).

| Nev | w Project Wizard | | |
|-----|---|---|-------------|
| Add | I Existing Sources Adding existing sources is optional. Additional sources can be a >Add Source" or "Project->Add Copy of Source" commands. | added after the project is created using th | e "Project- |
| Add | existing sources | | |
| | Source File | Copy to Project Ac | ld Source |
| 1 | aes_r.vhd | | Pemove |
| 2 | aes_package.vhd | | Cennove |
| 3 | key_expander.vhd | | |
| 4 | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| Mor | re Info | < Back Next > | Cancel |

X1135_c1_22_040710

Figure 2-16: New Project Wizard (Add Existing Sources)

9. Click **Finish** (see Figure 2-17).

| 🔤 New Project Wizard | × |
|---|--------|
| Project Summary Project Navigator will create a new project with the following specifications. | |
| Project: | ٦ |
| Project Name: aes_r | |
| Project Path: C:\Xilinx_Design\synthesis\aes_r | |
| Working Directory: C:\Xilinx_Design\synthesis\aes_r | |
| Description: | |
| Top Level Source Type: HDL | |
| Device: | |
| Device Family: Virtex5 | |
| Device: xc5v1x85 | |
| Package: ff676 | |
| Speed: -2 | |
| Synthesis Tool: XST (VHDL/Verilog) | |
| Simulator: ISim (VHDL/Verilog) | |
| Preferred Language: VHDL | |
| Property Specification in Project File: Store all values | |
| Manual Compile Order: false | |
| Enhanced Design Summary: enabled | |
| Message Filtering: disabled | |
| Display Incremental Messages: disabled | |
| Existing Sources: | |
| aes package.vhd - use from current location | |
| aes r.vhd - use from current location | |
| key_expander.vhd - use from current location | |
| More Info Cancel | ר ר |

X1135_c1_23_040710

Figure 2-17: New Project Wizard (Project Summary)

10. Click **OK** (see Figure 2-18).

| 🖾 Adding Source Files 🛛 🛛 🔀 | | | | | × |
|-----------------------------|---|-------------|---|----------------|------|
| The allo the | The following allows you to see the status of the source files being added to the project, and allows you to specify the Design View association for sources which are successfully added to the project. | | | | |
| | File Name | Association | | Library | |
| 1 | of aes_package.vhd | All | ~ | work | • |
| 2 | 🧭 aes_r.vhd | All | ~ | work | • |
| 3 | 📀 key_expander.vhd | All | ~ | work | • |
| | | | | | |
| Ad | Adding files to project: | | | | ;) |
| | | | | OK Cancel Help | |
| | | | | X1135 c1 24 | 0407 |

Figure 2-18: Adding Source Files

The ISE Project Navigator Window should look like the window shown in Figure 2-19.

Note: All modules are defined. There should be no question marks (?) by any module. All modules at this level and below are in context.



X1135_c1_25_031810

Figure 2-19: Project Navigator Window

11. Open the aes_r.vhd file and locate the section containing the buffer_type attributes:

```
attribute buffer_type: string;
attribute buffer_type of clk
                                   : signal is "none";
attribute buffer_type of reset
                                  : signal is "none";
attribute buffer_type of push_button : signal is "none";
attribute buffer_type of start : signal is "none";
attribute buffer_type of mode
                                  : signal is "none";
                                  : signal is "none";
attribute buffer_type of load
                                  : signal is "none";
attribute buffer_type of key
attribute buffer_type of data_in
                                   : signal is "none";
attribute buffer_type of data_out
                                   : signal is "none";
attribute buffer_type of done
                                    : signal is "none";
```

The buffer_type attribute directs the XST tool to disable I/O insertion on the specified ports. By default, XST inserts an I/O on all ports. The buffer_type attribute is necessary to prevent this insertion at this level in the hierarchy either because the I/Os are placed at the top level (such as CLK) or the port is a direct connection to a port of another instance.

Note: The clk I/O is inferred when top is synthesized. All other ports are internal to the FPGA; therefore they have the attribute **signal is "none"** applied.

- 12. Right-click the Synthesize-XST icon in the Processes window, select **Process Properties...**
- 13. Set Optimization Goal to **Area**, set Optimization Effort to **High**, and click **OK** (see Figure 2-20).

| Process Properties - Synth | esis Options | | E Constantino de Const |
|-------------------------------|---------------------------|----------------------------------|--|
| Category | Switch Name | Droperty Name | Value |
| Synthesis Options | -opt mode | Optimization Coal | Value |
| HDL Options | -opt_level | Optimization Effort | |
| ····· Xilinx Specific Options | -opc_iever | Power Reduction | |
| | iuc | Lice Supplierie Constrainte File | |
| | | Support Constraints File | |
| | leo | Library Search Order | |
| | -keep bierarchy | Keen Hierarchy | |
| | -seep_nerarchy | Notlict Hierarchy | |
| | | Clabel Optimization Cool | As Optimized |
| | -glob_opc | Global Optimization Goal | AllClockNets |
| | -rtiview | Generate RTL Schematic | Yes |
| | -read_cores | Read Cores | |
| | -50 | Cores Search Directories | |
| | -write_timing_constraints | Write Timing Constraints | |
| | -cross_clock_analysis | Cross Clock Analysis | |
| | -hierarchy_separator | Hierarchy Separator | / |
| | -bus_delimiter | Bus Delimiter | <u> </u> |
| | -slice_utilization_ratio | LUT-FF Pairs Utilization Ratio | 100 |
| | -bram_utilization_ratio | BRAM Utilization Ratio | 100 |
| | -dsp_utilization_ratio | DSP Utilization Ratio | 100 |
| | -case | Case | Maintain |
| | | Work Directory | -LAB-UsersGuide_V5-Design/Xilinx_Design/synthesis/aes_r/xst |
| | set -xsthdpini | HDL INI File | |
| | -verilog2001 | Verilog 2001 | |
| | -vlgincdir | Verilog Include Directories | (+ |
| | -generics | Generics, Parameters | |
| | -define | Verilog Macros | |
| | | Other XST Command Line Options | |
| | | • | |
| | | Property displa | ay level: Advanced 🔽 🗹 Display switch names 🛛 Default |
| | | | OK Cancel Apply Help |

X1135_c1_26_031810

Figure 2-20: Process Properties

- 14. To run XST synthesis, either right-click on the **Synthesize-XST** icon in the Processes window and select **Run** or simply double-click **Synthesize-XST**.
- 15. After synthesis is complete, close the aes_r project.

Synthesize the compare Module

These steps describe how to synthesize the compare module:

1. Start the ISE 11.4 software:

Start $\rightarrow~$ All Programs $\rightarrow~$ Xilinx ISE Design Suite 11 $\rightarrow~$ ISE $\rightarrow~$ Project Navigator

2. Create a new ISE 11.4 project:

File \rightarrow **New Project**

3. Set the Project Location to **\Xilinx_Design\synthesis**, set the Project Name to **compare**, and click **Next** (see Figure 2-21).

| 🚾 New Project Wiza | rd | X |
|---|------------------------------------|---------|
| Create New Project Specify project loo | cation and type. | |
| Enter a name, locati | ons, and comment for the project | |
| Name: | compare | |
| Location: | C:\Xilinx_Design\synthesis\compare | |
| Working Directory: | C:\Xilinx_Design\synthesis\compare | |
| Description: | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| Select the type of to | op-level source for the project | |
| Top-level source typ | e: | |
| HDL | ✓ | |
| | | |
| | | |
| | | |
| More Info | Next > Cancel | |
| | V112E at 2 | 7.04071 |

Figure 2-21: New Project Wizard (Create New Project)

4. Choose an XC5VLX85-FF676-2 as the target device (see Figure 2-22).

| S S S S S S S S S S S S S S S S S S S |
|---|
| 5 |
| 5 |
| ✓ ✓ |
| × |
| × |
| |
| DL/Verilog) 🛛 🗠 |
| DL/Verilog) 🗸 🗸 |
| ~ |
| values 👻 |
| |
| |
| |
| |
| |
| |
| |

Figure 2-22: New Project Wizard (Device Properties)

- 5. Click **Next** twice. Add the compare.vhd file to the project by clicking the **Add Source** button and navigating to the source\design directory.
- 6. Uncheck **Copy to Project** and click **Next** (see Figure 2-23).

| 🚾 Ne | w Project Wizard | Σ |
|------|--|--|
| Ade | I Existing Sources Adding existing sources is optional. Additional sources can b >Add Source" or "Project->Add Copy of Source" commands | e added after the project is created using the "Project- |
| Add | existing sources | |
| | Source File | Copy to Project Add Source |
| 1 | compare.vhd | Remove |
| | | |
| Мо | re Info | <pre></pre> |
| | | X1135_c1_29_040 |

Figure 2-23: New Project Wizard (Add Existing Sources)
7. Click **Finish** (see Figure 2-24).

| Project Summary Project Navigator will create a new project with the following specifications. Project: Project Path: C:\XIlinx_Design\synthesis\compare Working Directory: C:\XIlinx_Design\synthesis\compare Description: Top Level Source Type: HDL Device: Device Family: VirtexS Device: xcSv1x85 Package: ff676 Speed: -2 Synthesis Tool: XST (VHDL/Verilog) Simulator: ISim (VHDL/Verilog) Preferred Language: VHDL Property Specification in Project File: Store all values Manual Compile Order: false Enhanced Design Summary: enabled Message Filtering: disabled Display Incremental Messages: disabled Existing Sources: compare.vhd - use from current location | 🔤 New Project Wizard | × |
|--|---|---|
| Project: Project Name: compare Project Path: C:\Xilinx_Design\synthesis\compare Working Directory: C:\Xilinx_Design\synthesis\compare Description: Top Level Source Type: HDL Device: Device: Device: xcSv1x85 Package: ff676 Speed: -2 Synthesis Tool: XST (VHDL/Verilog) Simulator: ISim (VHDL/Verilog) Preferred Language: VHDL Property Specification in Project File: Store all values Manual Compile Order: false Enhanced Design Summary: enabled Message Filtering: disabled Display Incremental Messages: disabled Existing Sources: compare.vhd - use from current location | Project Summary Project Navigator will create a new project with the following specifications. | |
| Project Name: compare Project Path: C:\Xilinx_Design\synthesis\compare Working Directory: C:\Xilinx_Design\synthesis\compare Description: Top Level Source Type: HDL Device: Device: xcSv1x85 Package: ff676 Speed: -2 Synthesis Tool: XST (VHDL/Verilog) Simulator: ISim (VHDL/Verilog) Preferred Language: VHDL Property Specification in Project File: Store all values Manual Compile Order: false Enhanced Design Summary: enabled Message Filtering: disabled Display Incremental Messages: disabled Existing Sources: compare.vhd - use from current location | Project: | ٦ |
| Project Path: C:\Xilinx_Design\synthesis\compare Working Directory: C:\Xilinx_Design\synthesis\compare Description: Top Level Source Type: HDL Device: Device: xc5v1x85 Package: ff676 Speed: -2 Synthesis Tool: XST (VHDL/Verilog) Simulator: ISim (VHDL/Verilog) Preferred Language: VHDL Property Specification in Project File: Store all values Manual Compile Order: false Enhanced Design Summary: enabled Message Filtering: disabled Display Incremental Messages: disabled Existing Sources: compare.vhd - use from current location | Project Name: compare | |
| <pre>Working Directory: C:\Xilinx_Design\synthesis\compare Description: Top Level Source Type: HDL Device: Device: xc5v1x85 Package: ff676 Speed: -2 Synthesis Tool: XST (VHDL/Verilog) Simulator: ISim (VHDL/Verilog) Preferred Language: VHDL Property Specification in Project File: Store all values Manual Compile Order: false Enhanced Design Summary: enabled Message Filtering: disabled Display Incremental Messages: disabled Existing Sources: compare.vhd - use from current location</pre> | Project Path: C:\Xilinx Design\synthesis\compare | |
| Description: Top Level Source Type: HDL Device: Device: xc5v1x85 Peckage: ff676 Speed: -2 Synthesis Tool: XST (VHDL/Verilog) Simulator: ISim (VHDL/Verilog) Preferred Language: VHDL Property Specification in Project File: Store all values Manual Compile Order: false Enhanced Design Summary: enabled Message Filtering: disabled Display Incremental Messages: disabled Existing Sources: compare.vhd - use from current location | Working Directory: C:\Xilinx Design\synthesis\compare | |
| Top Level Source Type: HDL Device: Device Family: Virtex5 Device: xc5v1x85 Package: ff676 Speed: -2 Synthesis Tool: XST (VHDL/Verilog) Simulator: ISim (VHDL/Verilog) Preferred Language: VHDL Property Specification in Project File: Store all values Manual Compile Order: false Enhanced Design Summary: enabled Message Filtering: disabled Display Incremental Messages: disabled Existing Sources: compare.vhd - use from current location | Description: | |
| Device: Device Family: Virtex5 Device: xc5v1x85 Package: ff676 Speed: -2 Synthesis Tool: XST (VHDL/Verilog) Simulator: ISim (VHDL/Verilog) Preferred Language: VHDL Property Specification in Project File: Store all values Manual Compile Order: false Enhanced Design Summary: enabled Message Filtering: disabled Display Incremental Messages: disabled Existing Sources: compare.vhd - use from current location More Info | Top Level Source Type: HDL | |
| Device Family: Virtex5 Device: xc5v1x85 Package: ff676 Speed: -2 Synthesis Tool: XST (VHDL/Verilog) Simulator: ISim (VHDL/Verilog) Preferred Language: VHDL Property Specification in Project File: Store all values Manual Compile Order: false Enhanced Design Summary: enabled Message Filtering: disabled Display Incremental Messages: disabled Existing Sources: compare.vhd - use from current location | Device: | |
| Device: xc5v1x85 Package: ff676 Speed: -2 Synthesis Tool: XST (VHDL/Verilog) Simulator: ISim (VHDL/Verilog) Preferred Language: VHDL Property Specification in Project File: Store all values Manual Compile Order: false Enhanced Design Summary: enabled Message Filtering: disabled Display Incremental Messages: disabled Existing Sources: compare.vhd - use from current location More Info < Back | Device Family: Virtex5 | |
| Package: ff676 Speed: -2 Synthesis Tool: XST (VHDL/Verilog) Simulator: ISim (VHDL/Verilog) Preferred Language: VHDL Property Specification in Project File: Store all values Manual Compile Order: false Enhanced Design Summary: enabled Message Filtering: disabled Display Incremental Messages: disabled Existing Sources: compare.vhd - use from current location More Info < Back | Device: xc5v1x85 | |
| Speed: -2 Synthesis Tool: XST (VHDL/Verilog) Simulator: ISim (VHDL/Verilog) Preferred Language: VHDL Property Specification in Project File: Store all values Manual Compile Order: false Enhanced Design Summary: enabled Message Filtering: disabled Display Incremental Messages: disabled Existing Sources: compare.vhd - use from current location | Package: ff676 | |
| Synthesis Tool: XST (VHDL/Verilog) Simulator: ISim (VHDL/Verilog) Preferred Language: VHDL Property Specification in Project File: Store all values Manual Compile Order: false Enhanced Design Summary: enabled Message Filtering: disabled Display Incremental Messages: disabled Existing Sources: compare.vhd - use from current location | Speed: -2 | |
| Simulator: ISim (VHDL/Verilog) Preferred Language: VHDL Property Specification in Project File: Store all values Manual Compile Order: false Enhanced Design Summary: enabled Message Filtering: disabled Display Incremental Messages: disabled Existing Sources: compare.vhd - use from current location More Info Reack Finish Cancel | Synthesis Tool: XST (VHDL/Verilog) | |
| Preferred Language: VHDL Property Specification in Project File: Store all values Manual Compile Order: false Enhanced Design Summary: enabled Message Filtering: disabled Display Incremental Messages: disabled Existing Sources: compare.vhd - use from current location More Info < Back | Simulator: ISim (VHDL/Verilog) | |
| Property Specification in Project File: Store all values Manual Compile Order: false Enhanced Design Summary: enabled Message Filtering: disabled Display Incremental Messages: disabled Existing Sources: compare.vhd - use from current location More Info More Info | Preferred Language: VHDL | |
| Manual Compile Order: false Enhanced Design Summary: enabled Message Filtering: disabled Display Incremental Messages: disabled Existing Sources: compare.vhd - use from current location | Property Specification in Project File: Store all values | |
| Enhanced Design Summary: enabled Message Filtering: disabled Display Incremental Messages: disabled Existing Sources: compare.vhd - use from current location | Manual Compile Order: false | |
| Message Filtering: disabled Display Incremental Messages: disabled Existing Sources: compare.vhd - use from current location | Enhanced Design Summary: enabled | |
| Display Incremental Messages: disabled Existing Sources: compare.vhd - use from current location | Message Filtering: disabled | |
| Existing Sources: compare.vhd - use from current location More Info | Display Incremental Messages: disabled | |
| Compare.vhd - use from current location | Existing Sources: | |
| More Info Cancel | compare.vhd - use from current location | |
| More Info Cancel | | _ |
| | More Info < Back Finish Cancel |) |
| | | |

Figure 2-24: New Project Wizard (Project Summary)

8. Click **OK** (see Figure 2-25).

| The following allows you to see the status of the source files being added to the project, and allows you to specify the Design View association for sources which are successfully added to the project. File Name Association Library 1 Compare.vhd All Adding files to project: 0K Cancel Help | 🖾 Adding Source Files 🔀 | | | | |
|---|---|---|--|--|--|
| File Name Association 1 Compare.vhd All Work Image: state of the st | The following allows yo allows you to specify th the project. | u to see the status of ne Design View associ | the source files being added to the project, and ation for sources which are successfully added to | | |
| 1 Compare.vhd All work Adding files to project: OK Cancel Help | File Name | Association | Library | | |
| Adding files to project: | 1 🔇 compare.vhd | All 🔽 | work 💌 | | |
| | | | 1 of 1 files (0 errors) | | |
| | Adding files to project: | | OK Cancel Help | | |

Figure 2-25: Adding Source Files

The ISE Project Navigator Window should look like the window shown in Figure 2-26.

Note: All modules are defined. There should be no question marks (?) by any module. All modules at this level and below are in context.



X1135_c1_32_031810

Figure 2-26: Project Navigator Window

9. Open the compare.vhd file and locate the buffer attributes:

| attribute | buffer_type: | st | tring; | | | | |
|-----------|------------------------|----|-----------|---|--------|----|---------|
| attribute | buffer_type | of | clk | : | signal | is | "none"; |
| attribute | buffer_type | of | reset | : | signal | is | "none"; |
| attribute | <pre>buffer_type</pre> | of | reset_out | : | signal | is | "none"; |
| attribute | buffer_type | of | done1 | : | signal | is | "none"; |
| attribute | <pre>buffer_type</pre> | of | done2 | : | signal | is | "none"; |
| attribute | <pre>buffer_type</pre> | of | start | : | signal | is | "none"; |
| attribute | buffer_type | of | load | : | signal | is | "none"; |
| attribute | buffer_type | of | data_in1 | : | signal | is | "none"; |
| attribute | buffer_type | of | data_in2 | : | signal | is | "none"; |

The buffer_type attribute directs the XST tool to disable I/O insertion on the specified ports. By default, XST inserts an I/O on all ports. The buffer_type attribute is necessary to prevent this insertion at this level in the hierarchy either because the I/Os are placed at the top level (such as CLK) or because the port is a direct connection to a port of another instance.

Note: The led signal does not have an attribute associated with it because the led signal is driven directly by an output buffer from within the compare code. All other ports are either ports within the FPGA or had their I/Os inferred at a different level; therefore they have the attribute **signal is "none"** applied.

10. In the compare.vhd source code file, the LUT instantiation between certain inputs and outputs is necessary for Trusted Routing rules. Refer to <u>XAPP1134</u>, *Developing Secure Designs Using the Virtex-5 Family*, for more details on Trusted Routing rules. A section of the VHDL code for the LUT instantiation is shown here:

-- Instantiate LUT buffers on nets that either drive two different regions -- or are feedthrough's from one region to the next. This prevents a single -- net being placed "shorting" three regions together. Trusted Bus Macros -- fulfilled this requirement automatically.

```
lut_reset_out : LUT1
GENERIC MAP (INIT => X"2")
PORT MAP (IO => reset_i, O => reset_out );
lut_start_aes1_out : LUT1
GENERIC MAP (INIT => X"2")
PORT MAP (IO => start_i, O => start_aes1 );
lut_start_aes2_out : LUT1
GENERIC MAP (INIT => X"2")
PORT MAP (IO => start_i, O => start_aes2 );
lut_load_aes1_out : LUT1
GENERIC MAP (INIT => X"2")
PORT MAP (IO => load_aes1 );
```

```
lut_load_aes2_out : LUT1
GENERIC MAP (INIT => X"2")
```

```
PORT MAP (IO => load_i, O => load_aes2 );
```

11. Right-click on the Synthesize-XST icon in the Processes window and select **Process Properties** (see Figure 2-27).

| gory | Switch Name | Property Name | Value |
|----------------------------------|---------------------------|--------------------------------|---|
| Synthesis Options HDL Options | -opt_mode | Optimization Goal | Area |
| Xilinx Specific Options | -opt_level | Optimization Effort | High |
| | -power | Power Reduction | |
| | -iuc | Use Synthesis Constraints File | |
| | -uc | Synthesis Constraints File | |
| | -lso | Library Search Order | |
| | -keep_hierarchy | Keep Hierarchy | Yes |
| | -netlist_hierarchy | Netlist Hierarchy | As Optimized |
| | -glob_opt | Global Optimization Goal | AllClockNets |
| | -rtlview | Generate RTL Schematic | Yes |
| | -read_cores | Read Cores | |
| | -sd | Cores Search Directories | . (|
| | -write_timing_constraints | Write Timing Constraints | |
| | -cross_clock_analysis | Cross Clock Analysis | |
| | -hierarchy_separator | Hierarchy Separator | 1 |
| | -bus_delimiter | Bus Delimiter | \diamond |
| | -slice_utilization_ratio | LUT-FF Pairs Utilization Ratio | 100 |
| | -bram_utilization_ratio | BRAM Utilization Ratio | 100 |
| | -dsp_utilization_ratio | DSP Utilization Ratio | 100 |
| | -case | Case | Maintain |
| | | Work Directory | B-UsersGuide_V5-Design/Xilinx_Design/synthesis/compare/xs |
| | set -xsthdpini | HDL INI File | |
| | -verilog2001 | Verilog 2001 | |
| | -vlgincdir | Verilog Include Directories | |
| | -generics | Generics, Parameters | |
| | -define | Verilog Macros | |
| | | Other XST Command Line Options | |
| | | | |
| | | Property displa | ay level: Advanced 🔽 🗹 Display switch names 🛛 Defa |
| | | | OK Cancel Apply Hel |

Figure 2-27: Process Properties

- 12. Set the Optimization Goal to **Area** and the Optimization Effort to **High**.
- 13. Set the Keep Hierarchy option to **Yes**.

Note: This extra step is different than the SCC_LAB_TOP, aes, and aes_r blocks. Compare, in this lab, is an isolated module, not a PR module. Keeping its hierarchy prevents it from being flattened with the top level or other isolated blocks. This only keeps its top-level instance, not any sub-hierarchy. As such, it has no additional impact to optimization.

- 14. Click OK.
- 15. To run XST synthesis, either right-click on the Synthesis-XST icon in the Processes window and select **Run** or simply double-click **Synthesize-XST**.
- 16. After synthesis is complete, close the compare project.



Chapter 3

Implementing the System

PlanAhead Project Entry

Launch the PlanAhead Tool

To launch the 11.4 version of the PlanAhead[™] tool, select:

Start \rightarrow All Programs \rightarrow Xilinx ISE Design Suite 11 \rightarrow PlanAhead \rightarrow Xilinx PlanAhead

PlanAhead Project Creation

The PlanAhead tool works with any synthesized netlist (XST, Synplify, etc.). The regular guidelines are followed to generate a new project and import the netlist into the PlanAhead tool to create a floorplan for the design.

1. Set up a new PlanAhead project:

$\textbf{File} \rightarrow \textbf{New Project}$

- 2. Click **Next** and enter:
 - Project name: For this lab, the **Floorplan_SCC** project name is used.
 - Project location: \Xilinx_Design\PlanAhead



Figure 3-1: New Project

- 3. Click **Next**.
- 4. Select **Import a synthesized (EDIF or NGC) netlist** (see Figure 3-2) because the modules have already been synthesized and click **Next**.

| New Project 🔀 |
|--|
| Design Source Specify the type of sources for your design. You can start with RTL or a synthesized EDIF |
| Import RTL Sources You will be able to run RTL analysis, synthesis and implementation. Import synthesized (EDIF or NGC) netlist You will be able to run post-synthesis design analysis, planning, and implementation. Import ISE Place & Route results You will be able to do post-implementation analysis of your design. Do not import sources at this time You will be able to do pin planning now and import a netlist later. |
| < Back Next > Cancel |

Figure 3-2: New Project (Design Source)

5. Import the previously generated top level Netlist (see Figure 3-3).

| 🔂 New Project | X |
|--|-------|
| Import Netlist Specify the EDIF or NGC netlist that contains the top module, and optionally a list of directories to be used as a search path. | |
| Netlist file: C:\Xilinx_Design\synthesis\SCC_LAB_TOP\SCC_LAB_TOP.ngc | |
| C:\Xilinx_Design\synthesis\aes C:\Xilinx_Design\synthesis\aes_r C:\Xilinx_Design\synthesis\compare | |
| < Back Next > Ca | ancel |

Figure 3-3: New Project (Import Netlist)

6. Designate the top-level netlist and library directories. The netlist directories should include the static logic top.ngc and only one "version" of each partially reconfigurable (PR) or isolated (ISO) module.

Netlist File:

..\Xilinx_Design\synthesis\top\top.ngc

Netlist Directories:

- ..\Xilinx_Design\synthesis\aes
- ..\Xilinx_Design\synthesis\aes_r
- ..\Xilinx_Design\synthesis\compare
- 7. Click Next.

8. Choose the Product Family (see Figure 3-4). For this lab, Virtex5 is used.

| New Project | |
|---------------------------------|--|
| Choose a Par Enter a nam | rt and a Floorplan Name e for your floorplan and choose a Xilinx device |
| Product Family: Choose Part: | Virtex5 |
| Floorplan name: | fp_v5lx85_ff676-2_1 |
| | < Back Next > Cancel |



- 9. Choose the device:
 - a. Click the '...' icon to the right of the Choose Part field.
 - b. Navigate to the **LX85** device.
 - c. Select the **ff676** package
 - d. Select the **-2** speed grade
- 10. Set the Floorplan name to **fp_v5lx85_ff676-2_1**.
- 11. Click Next.
- 12. This lab creates the final UCF from scratch. Thus no UCF files need to be imported.

| New Project | |
|---|-------|
| Import Constraints Import physical and timing constraints from a UCF file. You can also import a UCF file later with the Import Constraints command. | |
| Constraints files | |
| Add Remove Up Down | |
| < Back Next > C | ancel |

Figure 3-5: New Project (Import Constraints)

13. Click **Next** and **Finish**.



| 1 T th (8) | | * × • | | |
|---|--|---------------------------|---------------------|---|
| (a, y346, ¥69-2,) ≩ ROOT | (a) 550 (J.M.) (b) (b) (b) (b) (b) (b) (b) (b) (b) (b) (b) | | | |
| sporter P ⊕ <mark>™</mark> A | | | | |
| Properties Stocton DPorts Tanam Dr Stock ports (1) Stock port | jij Netšat, 🍙 Contrantos Neg Off Par Locaton Sark. 10.95d D | E Effective (* Denke × | | |
| B Consile. Do 1/0 Ports J | CC - [C:\Xilinx_Des | ign\PlanAhead\Flooi | Plan_SCC\FloorPla | Post-Series Rive - set of n_SCC.ppr] - PlanAhead |
| File Edit View | v Tools Window S | ielect Layout Help | 1 (vc5vlv85ff576-2) | |
| Physical Hierarchy | | | Netlist | |
| o 💎 🚔 🖪 | | | | |
| P_vSlx85_ff67 | 6-2_1 | | | 6) aes) aes_r) compare) |
| | | | | |
| | | | | |



15. To turn the project into a PR project, select File \rightarrow Set PR Project. When the Set PR **Project** option is selected, it is grayed out to indicate this step has been completed.

Place I/Os with the PlanAhead Tool

When placing I/Os it is imperative to consider the physical location of I/Os in relation to the logic regions they interface with. For example, the clock input can be placed anywhere because it does not have to be isolated. Because the led pin is part of the COMPARE logic, it needs to be physically placed in that region. Similarly, the reset and push_button pins are owned by the AES logic, so they need to be physically placed inside that region.

To place the PlanAhead tool in Site Constraint Mode:

1. Select the symbol shown in Figure 3-7 that is on the vertical shortcut bar between the netlist window and the device window.



Figure 3-7: Select Site Constraint Mode Symbol

2. Select **Edit** \rightarrow **Find** (shortcut: **Ctrl-F**).

| 🔂 Find 🔀 |
|-----------------------|
| Find: Sites |
| |
| Name v matches V F14 |
| |
| |
| |
| |
| |
| More Fewer Match Case |
| OK Cancel |

Figure 3-8: Find

- 3. Select **Sites** from the Find pull-down menu.
- 4. Select Name and matches from the two pull-down menus under Criteria.
- 5. Type **F14** in the remaining field box and click **OK**.

The search results appear on a tab in the Find Results section at the bottom left of the PlanAhead tool window.

 Select the result and press F9 to zoom to the current selection, as shown in Figure 3-9. Alternatively, select View → Fit Selection from the top menu bar to perform the same task.



Figure 3-9: Find Results

7. Select $Edit \rightarrow Find$ (shortcut: Ctrl-F).

| 🔂 Find | × |
|----------------------|------------|
| Find: Ports | |
| Criteria | |
| Name 🔽 matches 🔽 clk | |
| | |
| | |
| | |
| | |
| More Fewer | Match Case |
| | OK Cancel |

Figure 3-10: Find

- 8. Select **Ports** from the Find: pull-down menu.
- 9. Select Name and matches from the two pull-down menus under Criteria.
- 10. Type **clk** in the field box, and click **OK**. The search results appear on a tab in the Find Results section at the bottom left of the PlanAhead tool window.
- 11. Click and drag the result **clk** to the site identified in step 2 through step 6, as shown in Figure 3-11.



Figure 3-11: Drag clk

- 12. Repeat step 1 to step 10 to place "reset" at D11.
- 13. Repeat step 1 to step 10 to place "push_button" at D10.
- 14. Repeat step 1 to step 10 to place "led" at P6.

Place DCM with the PlanAhead Tool

These steps describe how to place the DCM using the PlanAhead tool:

- 1. Select $Edit \rightarrow Find$ (shortcut: Ctrl-F) and select Sites from the Find: pull-down menu.
- 2. Select **Name** and **matches** from the two pull-down menus under Criteria.

| 诸 Find | | × |
|-------------|----------------------|-------------------|
| Find: Sites | ~ | |
| Criteria | |] |
| Name | winatches V DCM_ADV. | _X0Y8 |
| | | |
| | | |
| | | |
| | | |
| | | |
| More Fewer | | Match Case |
| | | OK Cancel |
| | | X1135_c2_12_04071 |



3. Type **DCM_ADV_X0Y8** in the field box, and click **OK**.

The search results appear on a tab in the Find Results section at the bottom left of the PlanAhead tool window.

 Select this result and press F9 to zoom to the current selection. The PlanAhead tool then zooms to the selected location (see Figure 3-13). Alternatively, select View → Fit Selection from the top menu bar to perform the same task.



Figure 3-13: Search Results

5. In the Netlist window, expand **Primitives**, then drag **TOP_DCM_ADV** to the DCM_ADV_X0Y8 box as shown in Figure 3-14.

| TioorPlan_SCC [C:Willinx_DesignPlanAheadViloorPl | an_SCCV loorPlan_SCC.ppr] - PlanAl | ead 11.4 | |
|--|--|--|--|
| File Edit View Tools Window Select Layout Help | | | |
| 周二月日月日日×年日のの⇒⇒米日 | 金属 化化化剂物 医长角 | ♦ X & X | |
| Project - FloorPlan_SCC Ploorplan - fp_vSlv85_ff676-2_1 | (xcSvbs858676-2)* × | | |
| Physical Herarchy | Netlet | ◆ □ 빛 · [| 별이 많이 문 별이 많이 문 별 🗥 |
| < | X B | | |
| 1 1p_v9x85_f1676-2_1 | SCC_LAB_TOP | · 다음의 · 다음의 · 다음의 · 다음의 · 다음의 · 다음의 · 다 · 다 · 다 · 다 · 다 | |
| | - Primbves (6) | | B 572 B 572 B |
| | TOP_CLKD_BUFG (BUFG) | ····································· | |
| | I I | ····································· | |
| | - CE TOP_DOM_ADV (DOM_ADV) | Land Carlo Carlo Carlo Carlo Carlo Carlo | |
| | - 2 XST_GND ((7/0) | 꽃님 같 수 있는 별 수 있는 별 수 있는 별 수 있는 별 수 있는 일 수 있는 것 같 수 있는 별 수 있는 일 수 있는 것 같 같 수 있는 것 같 수 있는 않 않 않 않 않 않 않 않 않 않 않 않 않 않 않 않 않 않 | |
| | in Tu Atst (ani) | | |
| | Nets (12124) | | |
| | 12_AES2 (000_7) | | 8.000.000 |
| | B-S US_Comp (compare) | ····································· | |
| | 224 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 | | |
| | | The second second second second second | Teles and the second |
| | | 그 다 밖 이 많이 잘 다 많이 잘 다 있다. 이 밖 이 많이 잘 다 있다. 이 밖 이 밖 이 많이 잘 다 있다. 이 밖 이 밖 이 있다. 이 밖 이 밖 이 있다. 이 있 있 있다. 이 있 있 있다. 이 있 있 있다. 이 있 있다. 이 있 있 있다. 이 있 있 있다. 이 있 있 있다. 이 있 있 있다. 이 있 있 있 있 있 있다. 이 있 있 있 있 있다. 이 있 있 있 있 있 있 있 있 있 있 있 있 있 있 있 있 있 있 | 물이들이를 물이들이를 물 |
| | | | |
| | | | 6 200 6 200 6 |
| | | | 0.000.000 |
| | | 이 변경에서는 변경에서 변경에서 변경에서 변경에서 비용하는 것이 있다. | |
| mmm | | 김 활이 영양을 불이 영양을 불이 있는 불이 영양을 불이 않았는 것이 같다. | 법 다아이 밤 다아이 밥 |
| Instance Properties | | | |
| | | | |
| U TOP DOM ADV | | | |
| FullName: TOP_DON_ADV | | | |
| Cell: DOM ADV | | | B P P B P P B |
| Test Child | | 다 활성(학교 - 발성)학교 물성(학교 - 발성)학교 물성(학교 | 월 드라이스템 드레이스템 |
| | | | |
| Ster BOM_ADV_X018 Fixed | | Republic R | They make a first state of the second state of |
| | | 0.0.510 0.510 0.510 0.510 0.510 | 0.510.0.510.0 |
| | | 이 집중주었다. 집중주었다 집중주었다. 집중중하다 집중주했다 | |
| | | | |
| | | Grand G G Same G State G Same G State G Same G State G Same G State G Sta | DATA BALLER BALL |
| | | 김 많다. 동네님 : 많다. 동네는 이 같다. 이 많다. 이 많다. 이 많다. 이 가 있다. 이 가 있는 것이 같다. 이 가 있는 않는 것이 같다. 이 가 있는 것이 같이 않는 것이 않는 것이 않는 것이 같이 않는 것이 않 않은 않아? 것이 않는 것 | 방수 등관 등 방수 등관 등 방 |
| | | | |
| | | | |
| | | | 8-5-2 8-5-2 8- |
| General Statistics Pins Children Attributes Connectivity | | C C | 2 |
| Properties 👌 Selection | 📰 Netlist 🚔 Constraints | Padkage Service × | () II |
| Find Results - Sites - Name matches "DCM_ADV_30V6" (1) | | | ⊡∂e× world ⊡∂e× |
| A 1d Name Type 100 Alias Instances | | | |
| I DOM_ADV_30HE DOM_ADV | 1 | | |
| • | | | |
| | | | |
| | | | |
| | | | |
| JO Ports - Name matches "reset" (1) 🔯 Sites - Name matches "Di | (0" (1) D- UO Ports - Name matches "push_b | ator" (1) 🗐 Sites - Name matches "P6" (1) 🗇 1/0 Ports - Name matches "Ied" (1) 🗐 Sites - Name matches "DCM_ADV | _X0Y0"(1) × 4 > = |
| Gorsole D- UO Ports # Find Results | | | |
| TOP DOM ADV | | | Post-Surthesis Flow 101M of 118M |
| | | | X1135 c2 14 031810 |

Figure 3-14: DCM_ADV_X0Y8

- 6. Repeat step 1 through step 5 to place **TOP_CLK0_BUFG** at BUFGCTRL_X0Y31.
- 7. Repeat step 1 through step 5 to place the **TOP_CLKDEV_BUFG** at BUFGCTRL_X0Y30.

Setting Up Timing Constraints with the PlanAhead Tool

This section shows that timing constraints can be applied to each module and the top-level design easily with the PlanAhead tool.

 Select the **Constraints** tab in the NETLIST window, and click the **Create New Constraint** button located at the top of the Netlist / Constraints window (see Figure 3-15).



Figure 3-15: Create New Constraint

2. Add a basic period constraint of 20 ns to the design by entering the value as shown in Figure 3-16.

| 🐧 New Timing Constrain | t 🔀 |
|--|--|
| Constraint Types Basic period Circle Timespec period Circle Derived period Circle To circle offset Circle to output pad offset Circle to output pad offset Circle To circle offset Circle Offset Circ | Basic period Clock Nets: ck Period Specification Width: 20 ns Duty Cycle: OK Cancel |
| | X1135 c2 16 040710 |

Figure 3-16: New Timing Constraint

- 3. Click **OK**.
- 4. Check the Delay Value box and add a Global Input constraint of 6 ns to the design by entering the value as shown in Figure 3-17.

| ሽ Create New Timing Cor | nstraint | |
|---|--|--------------------|
| Constraint Types | Input pad to clk offset | |
| Basic period Timespec period Derived period Input pad to clk offset Clk to output pad offset Path delay (FROM-TO) Basic group (TNM) Multi group (TIMEGRP) Object false path Group false path Feedback | Data arrival: Before clock Clock: clk O Delay value: 6 ns Valid: 0 Valid: 0 None Pad selection | |
| | ОК | Cancel |
| | | X1135 c2 17 040710 |

Figure 3-17: New Timing Constraint

5. Click **OK**.

6. Check the Delay Value box and add a Global Output constraint of 6 ns to the design by entering the value as shown in Figure 3-18.

Figure 3-18: New Timing Constraint (Delay Value)

7. Click **OK**.

Defining RP or ISO Partitions with the PlanAhead Tool

Next each partition must be turned into either a Reconfigurable Partition (RP) or an Isolated Partition (ISO) as follows:

- n_SCC.ppr] PlanAhead 11. CI F le 【1993年1999年)。(1995年)(1995年)||1995年||1995年||1995年||1995年||1995年||1995年||1995年||1995年||1995年||1995年||1995年||1995年||1995年||1995年||1995年||1995年||1995年||1995年||1995年||1995年||1995年||1995年||1995年||1995年||1995年||1995年||1995年||1995年||1995年||1995年||1995年||1995年||19 Protect - FloorPlan SCC 🔄 Floorplan - fp vSbr85 ff676-2 1 (xcSvbr85ff676-2)* 🗴 Physical Herarchy **X** 🗉 4 SOC_LAB_TOP 1 1 AC
 1 AC
 1 U2_ACS
 1 U2_Cor 🥥 insta Export Statistics. Unempi
 Draw Pblod New Pblock Set Parts Select Primitives 1 Highlight DØØX Mark ++ UL AESI PUIN UI_AESI Highlight Frind Show Connectivity Chil+1 Those Henarchy Package Device ties 1 Se 1/O Std Drive St gth Slew Type Pull Type De Neg Diff Pair Los -Input Input Input Output 3 LVCMOS25 12 9.04 12 SLOW 12 SLOW 12 SLOW 12 SLOW 16 LVCM0525 16 LVCH0525 12 LVCH0525 D. 810 C Ce ole D 1/0 Ports @ Find Results Set Reconfigurable Part
- Right-click on **U1_AES (aes)** in the Netlist tab and select **Set Reconfigurable** 1. **Partition** from the pull-down menu (see Figure 3-19).

Figure 3-19: Set Reconfigurable Partition

- Keep the default name at **module_1**. If there were to be multiple configurations for 2. this block, then the user would change the field to a more descriptive name for that specific configuration.
- Repeat step 1 through step 2 for the **U2_AES2 (aes_r)** partition. 3.
- Repeat step 1 through step 2 for the **U3_Comp (compare)** partition with the following 4. exception:

Select Set Partition instead of Set Reconfigurable Partition. The compare block is an ISO block, not an RP block. This menu option is three lines below the Set Reconfigurable Partition option.

Partial P.

X1135_c2_19_022610

Defining Attributes for each RP or ISO Partition with the PlanAhead Tool

SCC rules have many additional constraint rules as opposed to a traditional Partial Reconfiguration flow. It is necessary to define several attributes for each partition as follows:

- 1. Under the Physical Hierarchy pane, select **pblock_U1_AES1**.
- 2. Under the Pblock Properties window, select the **Attributes** tab for *pblock_U1_AES1*.
- 3. Select the **Define New Attribute** button (a green + symbol).
- 4. Select the **BOUNDARYCROSS**, **CONTAINED**, and **PRIVATE** attributes by clicking each attribute while holding the **Ctrl** button (see Figure 3-20).

| 🖸 Define Attribute | < |
|-----------------------|-----|
| Search: Q- | |
| BOUNDARYCROSS | |
| COMPRESSION | |
| CONTAINED | |
| EXPAND | |
| GROUP | |
| IMPLEMENT | |
| INCLUSIVE | |
| PRIVATE | |
| RANGE_CAPTURE | |
| RANGE_DCIRESET | |
| RANGE_EFUSE_USR | |
| RANGE_FRAME_ECC | |
| RANGE_JTAGPPC | |
| RANGE_KEY_CLEAR | |
| RANGE_PMV | |
| RANGE_STARTUP | |
| RANGE_USR_ACCESS_SITE | |
| | - |
| OK Cancel |] |
| X1135 c2 20 050 | 700 |

Figure 3-20: **Define Attribute**

5. Click **OK.**

6. In the Pblock Properties window (see Figure 3-21), uncheck **BOUNDARYCROSS**, set CONTAINED to **ROUTE** and set PRIVATE to **NONE**.

| Pbloc | k Properties | | | 00 | ф × |
|-------|---------------------|----------|------------|------------|-------------|
| 4 | 🔶 🛐 📐 | | | | |
| 💽 pb | lock_U1_AES1 | | | | |
| + | Name | Туре | Value | | |
| × | BOUNDARYCROSS | Boolean | | | |
| | CONTAINED | Enum | ROUTE | ~ | |
| | PRIVATE | Enum | NONE | ~ | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| Ger | neral Statistics Ir | nstances | Rectangles | Attributes | |
| | | Apply | 🛛 🧝 Cance | el | |
| I | 🗿 Properties 📃 | Selectio | in | X1135_c2 | 2_21_022610 |

Figure 3-21: Pblock Properties

- 7. Click Apply.
- 8. Repeat step 1 through step 7 for **pblock_U2_AES2**.
- 9. Repeat step 1 through step 7 for **pblock_U3_Comp**.

Set Up the Area Groups for the RP Regions with the PlanAhead Tool

These steps configure the area groups for the RP regions:

- 1. Under the Physical Hierarchy pane, select the block **pblock_U1_AES1**.
- 2. Right-click on **pblock_U1_AES1** and select **Set Pblock Size** from the pull-down menu.
- 3. Draw a rectangle as shown in Figure 3-22 in the upper left-hand corner of the device window. (It does not have to be 100% accurate it will be resized shortly.)



Figure 3-22: Pblock Layout

Note: The PlanAhead tool might resize the rectangles to many different combinations of smaller rectangles. The user needs only to ensure that there is one CLB of separation between the Pblocks AES1, AES2, and COMPARE.

4. A dialog box (see Figure 3-23) pops up with various attributes and associated checkboxes. All boxes must be checked, including (if listed) DCM_ADV, PLL_ADV, BUFGCTRL, BUFR, and BUFIO. Even though most of these blocks are not in the design, it is important to select them to take advantage of their routing resources. All used components must be included.

Note: Trusted Routing requires that all clocking components be assigned to the AREA GROUP that physically contains the component, even though the component is not logically instantiated in the HDL of that module.

| 🔂 Set Pblock | \mathbf{X} |
|--------------------------------------|------------------------|
| Description | |
| Which resources do you constrain? | wish pblock_U1_AES1 to |
| Grids | |
| SLICE | |
| BUFIO | |
| UFR BUFR | |
| DCI | |
| DSP48 | |
| IDELAYCTRL | |
| ILOGIC | |
| IOB IOB | |
| IODELAY | |
| OLOGIC | |
| PMVBRAM | |
| RAMB36 | |
| | |
| | OK Cancel |
| | X1135_c2_23_022610 |

Figure 3-23: **Set Pblock**

- 5. Click **OK**.
- 6. In the Choose LOC mode dialog box, select the action **Leave all location constraints** in their current position.
- 7. Click **OK**.
- 8. Ensure that **pblock_U1_AES1** is selected in the Physical Hierarchy pane.
- 9. Select the **Rectangles** tab in the Properties tab of the Pblock Properties window.

10. Adjust the rectangle graphically as necessary to match the following coordinates:

```
X Lo = 0
Y Lo = 1
```

- X Hi = 63
- Y Hi = 65

| Pblock Properties | ㅁ @ ₽ | × |
|---|------------|----------|
| 4 4 🔁 R | | |
| pblock_U1_AES1 | | |
| Id XLo YLo XHi YHi | | |
| 1 0 1 63 65 | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| General Statistics Instances Rectangles | Attributes | _ |
| Selection @ Properties | | |
| - Selection - Fropercies | X1135 c2 3 | 4 050800 |

Figure 3-24: Pblock Properties

- 11. Under the Physical Hierarchy tab, select the block **pblock_U1_AES1**.
- 12. Right-click on **pblock_U1_AES1** and select **Add Pblock Rectangle** from the pulldown menu.

- <image>
- 13. Draw a rectangle in the upper center of the device window. (It does not have to be 100% accurate—it will be resized shortly.)

Figure 3-25: Pblock Layout

14. A dialog box (see Figure 3-26) pops up with various attributes and associated checkboxes. All boxes must be checked, including (if listed) DCM_ADV, PLL_ADV, BUFGCTRL, BUFR, and BUFIO. Even though most of these blocks are not in the design, it is important to select them to take advantage of their routing resources. All used components must be included.

Note: Trusted Routing requires that all clocking components be assigned to the AREA GROUP that physically contains the component even though the component is not logically instantiated in the HDL of that module.

| 诸 Add Rectangle | × |
|--|-------------------|
| Description | |
| Which ranges should be added to pblock_U | J1_AES1? |
| Grids | |
| SLICE | |
| DCM_ADV | |
| PLL_ADV | |
| PMVBRAM | |
| RAMB36 | |
| | |
| | |
| ОК | Cancel |
| x | 1135_c2_26_022610 |

Figure 3-26: Add Rectangle

- 15. Click **OK**.
- 16. Ensure that **pblock_U1_AES1** is selected in the Physical Hierarchy pane.
- 17. Select the **Rectangles** tab in the Properties tab of the Pblock Properties window.
- 18. Adjust the rectangle graphically as necessary to match the following coordinates: X Lo = 65
 - Y Lo = 1

X Hi = 109

Y Hi = 31

| | | Jercies | | | | | D | 4 | |
|----------|--------|------------|--------|--------|----------|----------|----|---|--|
| • | • | 5 R | | | | | | | |
| 🖲 pb | lock_U | J1_AES1 | | | | | | | |
| | Id | X Lo | Y Lo | X Hi | Y Hi | | | | |
| | 1 | 65 | 1 | 109 | 31 | | | | |
| | 2 | 0 | 1 | 63 | 65 | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| Ger | neral | Statistics | Instan | ces Re | ctangles | Attribut | es | | |
| Ger | neral | Statistics | Instan | ces Re | ctangles | Attribut | es | 1 | |

Figure 3-27: Pblock Properties

- 19. Under the Physical Hierarchy tab, select the block **pblock_U1_AES1**.
- 20. Right-click on **pblock_U1_AES1** and select **Add Pblock Rectangle** from the pulldown menu.



21. Draw a rectangle in the upper right of the device window. (It does not have to be 100%accurate—it will be resized shortly.)

EXILINX.

X1135_c2_28_022610

Figure 3-28: Pblock Layout

22. A dialog box (see Figure 3-29) pops up with various attributes and associated checkboxes. All boxes must be checked, including (if listed) DCM_ADV, PLL_ADV, BUFGCTRL, BUFR, and BUFIO. Even though most of these blocks are not in the design, it is important to select them to take advantage of their routing resources. All used components must be included.

Note: Trusted Routing requires that all clocking components be assigned to the AREA GROUP that physically contains the component even though the component is not logically instantiated in the HDL of that module.

| 🐧 Add Rectangle 🛛 🔀 |
|---|
| Description |
| Which ranges should be added to pblock_U1_AES1? |
| Grids |
| SLICE |
| UFIO |
| UFR |
| ✓ DCI |
| IDELAYCTRL |
| ☑ ILOGIC |
| ☑ IOB |
| IODELAY |
| ☑ OLOGIC |
| RAMB36 |
| |
| |
| OK Cancel |
| X1135 c2 29 022610 |

Figure 3-29: Add Rectangle

- 23. Click **OK**.
- 24. Ensure that **pblock_U1_AES1** is selected in the Physical Hierarchy pane.
- 25. Select the **Rectangles** tab in the Properties tab of the Pblock Properties window.

26. Adjust the rectangle graphically as necessary to match the following coordinates:

```
X Lo = 112
Y Lo = 1
X Hi = 155
```

Y Hi = 53

| 1 | Id | XI.o. | VIa | Y Hi | V Hi | |
|---|----|-------|------|------|------|------|
| a | 10 | 112 | 1 10 | 155 | 53 | |
| ŏ | 2 | 65 | 1 | 109 | 31 | |
| õ | 3 | 0 | 1 | 63 | 65 | |
| | | | | | | |
| | | | | | | |

Figure 3-30: Pblock Properties

- 27. Under the Physical Hierarchy tab, select the block **pblock_U2_AES2**.
- 28. Right-click on **pblock_U2_AES2** and select **Set Pblock Size** from the pull-down menu.


29. Draw a rectangle in the lower left-hand corner of the device window. (It does not have to be 100% accurate—it will be resized shortly.)

Figure 3-31: **Pblock Layout**

30. A dialog box (see Figure 3-32) pops up with various attributes and associated checkboxes. All boxes must be checked, including (if listed) DCM_ADV, PLL_ADV, BUFGCTRL, BUFR, and BUFIO. Even though most of these blocks are not in the design, it is important to select them to take advantage of their routing resources. All used components must be included.

| 🐧 Set Pblock 🛛 🔀 |
|---|
| Description |
| Which resources do you wish pblock_U2_AES2 to constrain? |
| Grids |
| SLICE |
| UFIO |
| UFR |
| ✓ DCI |
| ☑ DSP48 |
| IDELAYCTRL |
| ILOGIC |
| ✓ IOB |
| V IODELAY |
| ○ OLOGIC |
| PMVBRAM |
| RAMB36 |
| |
| OK Cancel |
| X1135_c2_32_02261 |

Figure 3-32: Set Pblock

- 31. Click **OK**.
- 32. Ensure that **pblock_U2_AES2** is selected in the Physical Hierarchy pane.
- 33. Select the **Rectangles** tab in the Properties tab of the Pblock Properties window.

34. Adjust the rectangle graphically as necessary to match the following coordinates:

X Lo = 0

- Y Lo = 68
- X Hi = 63
- Y Hi = 131



Figure 3-33: Pblock Properties

- 35. Under the Physical Hierarchy tab, select the block **pblock_U2_AES2**.
- 36. Right-click on **pblock_U2_AES2** and select **Add Pblock Rectangle** from the pulldown menu.
- 37. Draw a rectangle in the lower center of the device window. (It does not have to be 100% accurate—it will be resized shortly.)



Figure 3-34: Pblock Layout

www.xilinx.com

38. A dialog box (see Figure 3-35) pops up with various attributes and associated checkboxes. All boxes must be checked, including (if listed) DCM_ADV, PLL_ADV, BUFGCTRL, BUFR, and BUFIO. Even though most of these blocks are not in the design, it is important to select them to take advantage of their routing resources. All used components must be included.

| ሸ Add Rectangle | X |
|---|---------|
| Description | |
| Which ranges should be added to pblock_U2_AES | 2? |
| Grids | |
| SLICE | |
| DCM_ADV | |
| PLL_ADV | |
| PMVBRAM | |
| RAMB36 | |
| | |
| | |
| OK Canc | el |
| X1135_c2_3 | 5_02261 |

Figure 3-35: Add Rectangle

- 39. Click **OK**.
- 40. Ensure that **pblock_U2_AES2** is selected in the Physical Hierarchy pane.
- 41. Select the **Rectangles** tab in the Properties tab of the Pblock Properties window.

- 42. Adjust the rectangle graphically as necessary to match the following coordinates:
 - X Lo = 65
 - Y Lo = 101
 - X Hi = 109
 - Y Hi = 131

| pt | olock_ | <u>U</u> 2 | _AES2 | | | | | | |
|----|--------|------------|-------|------|----|------|------|--|------|
| | Id | | X Lo | Y Lo | XH | li 🛛 | Y Hi | | |
| | | 1 | 65 | 10: | 1 | 109 | 131 | | |
| J | | 2 | 0 | 68 | 3 | 63 | 131 | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |

Figure 3-36: Pblock Properties

- 43. Under the Physical Hierarchy tab, select the block **pblock_U2_AES2**.
- 44. Right-click on **pblock_U2_AES2** and select **Add Pblock Rectangle** from the pulldown menu.

- <image>
- 45. Draw a rectangle in the lower right of the device window. (It does not have to be 100% accurate—it will be resized shortly.)

Figure 3-37: Pblock Layout

46. A dialog box (see Figure 3-38) pops up with various attributes and associated checkboxes. All boxes must be checked, including (if listed) DCM_ADV, PLL_ADV, BUFGCTRL, BUFR, and BUFIO. Even though most of these blocks are not in the design, it is important to select them to take advantage of their routing resources. All used components must be included.

| 试 Add Rectan | gle 🔀 |
|------------------|---------------------------------|
| Description | |
| Which ranges sho | uld be added to pblock_U2_AES2? |
| Grids | |
| SLICE | |
| BUFIO | |
| UFR BUFR | |
| DCI 🗹 | |
| IDELAYCTRL | |
| ILOGIC 🗹 | |
| IOB IOB | |
| IODELAY | |
| OLOGIC | |
| RAMB36 | |
| | |
| | |
| | OK Cancel |
| | X1135 c2 38 02261 |

Figure 3-38: Add Rectangle

- 47. Click **OK**.
- 48. Ensure that **pblock_U2_AES2** is selected in the Physical Hierarchy pane.
- 49. Select the **Rectangles** tab in the Properties tab of the Pblock Properties window.

50. Adjust the rectangle graphically as necessary to match the following coordinates:

```
X Lo =112
Y Lo = 79
```

- X Hi = 155
- Y Hi = 131

| Id | X Lo | Y Lo | $\rm XH^{-1}_{\rm c}$ | Y Hi | | |
|----|------|------|-----------------------|------|--|--|
| 1 | 112 | 79 | 155 | 131 | | |
| 2 | 65 | 101 | 109 | 131 | | |
| 3 | 0 | 68 | 63 | 131 | | |
| | | | | | | |

Figure 3-39: Pblock Properties

- 51. Under the Physical Hierarchy tab, select the block **pblock_U3_Comp**.
- 52. Right-click on **pblock_U3_Comp** and select **Set Pblock Size** from the pull-down menu.



53. Draw a rectangle in the center of the device window. (It does not have to be 100% accurate—it will be resized shortly.)

Figure 3-40: Pblock Layout

www.xilinx.com

54. A dialog box (see Figure 3-41) pops up with various attributes and associated checkboxes. All boxes must be checked, including (if listed) DCM_ADV, PLL_ADV, BUFGCTRL, BUFR, and BUFIO. Even though most of these blocks are not in the design, it is important to select them to take advantage of their routing resources. All used components must be included.

| 🚺 Set Pblock | |
|-----------------------------------|------------------------|
| Description | |
| Which resources do you constrain? | wish pblock_U3_Comp to |
| Grids | |
| | |
| SLICE | |
| BSCAN | |
| BUFGCTRL | |
| BUFIO | |
| DCI | |
| ICAP | |
| IDELAYCTRL | |
| ILOGIC | |
| ▼ IOB | |
| IODELAY | |
| IPAD | |
| OLOGIC | |
| PMVBRAM | |
| RAMB36 | |
| SYSMON | |
| | |
| | |
| | OK Cancel |

Figure 3-41: Set Pblock

- 55. Click **OK**.
- 56. In the Choose LOC mode dialog box, select the action **Leave all location constraints** in their current position.
- 57. Click **OK**.
- 58. Ensure that **pblock_U3_Comp** is selected in the Physical Hierarchy pane.
- 59. Select the **Rectangles** tab in the Properties tab of the Pblock Properties window.

- 60. Adjust the rectangle graphically as necessary to match the following coordinates:
 - X Lo = 67
 - Y Lo = 34
 - X Hi = 107
 - Y Hi = 98



Figure 3-42: Pblock Properties

- 61. Under the Physical Hierarchy tab, select the block **pblock_U3_Comp**.
- 62. Right-click on **pblock_U3_Comp** and select **Add Pblock Rectangle** from the pull-down menu.



63. Draw a rectangle in the center right of the device window. (It does not have to be 100% accurate—it will be resized shortly.)

Figure 3-43: Pblock Layout

64. A dialog box (see Figure 3-44) pops up with various attributes and associated checkboxes. All boxes must be checked, including (if listed) DCM_ADV, PLL_ADV, BUFGCTRL, BUFR, and BUFIO. Even though most of these blocks are not in the design, it is important to select them to take advantage of their routing resources. All used components must be included.

| 🗂 Add Rectangle 🛛 🔀 |
|---|
| Description |
| Which ranges should be added to pblock_U3_Comp? |
| Grids |
| SLICE |
| |
| UFR BUFR |
| DCI 🗹 |
| IDELAYCTRL |
| ☑ ILOGIC |
| ₩ IOB |
| V IODELAY |
| ✓ OLOGIC |
| RAMB36 |
| |
| |
| OK Cancel |
| X1135_c2_44_022610 |

Figure 3-44: Add Rectangle

- 65. Click **OK**.
- 66. Ensure that **pblock_U3_Comp** is selected in the Physical Hierarchy pane.
- 67. Select the **Rectangles** tab in the Properties tab of the Pblock Properties window.

- 68. Adjust the rectangle graphically as necessary to match the following coordinates:
 - X Lo = 109 Y Lo = 56
 - X Hi = 155
 - (III = 100
 - Y Hi = 76



Figure 3-45: Pblock Properties

69. The final layout is shown in Figure 3-46. Each block is separated by one CLB to ensure SCC isolation. A block RAM, DSP, IOB, or any other site type that contains a Global Switch Matrix (GSM) can be used for this isolation.



Figure 3-46: Final Layout

70. Ensure there is one CLB of spacing between each area group.

Running Design Rule Checks

Given the complexity of the SCC flow, it is highly recommended to run the built-in design rule checks (DRCs) of the PlanAhead tool. This step can be time saving because it can find many common mistakes early on in the design process.

1. From the Tools menu, select **Run DRC**. Alternatively, click the checkmark button on the top toolbar (see Figure 3-47).

| File Edit View Tools | Window Select Layout | : Help | | |
|---------------------------|-------------------------|----------------|-----------------|--------------------|
| i 💕 📑 🖿 📜 😰 | UT X 街 🥝 🕑 🕨 | 🌢 🗄 🎉 🌗 | × 🛛 🔍 🗸 🤇 | 🖎 🕵 🔍 👂 🖉 |
| 💽 Project - Floorplan_SCC | 😬 Floorplan - fp_v5lx85 | _ff676-2_1 (xc | :5vlx85ff676-2) | × |
| Physical Hierarchy | | ⊡ ₽ × | Netlist | |
| | | | | X1135_c2_57_022610 |

Figure 3-47: Checkmark Button

- 2. Due to incompatibilities between Commercial Partial Reconfiguration and SCC Trusted Routing, several DRC rules need to be disabled:
 - a. Bus macro between static logic and isolated region (ISNM)
 - b. PR static logic illegally placed (PRLL)
 - c. Area group tile alignment (FLBA).

3. Click **OK** (see Figure 3-48).

| 🖸 Run DRC | X | | | | | |
|--|---|--|--|--|--|--|
| Results Name: | results_1 | | | | | |
| Rules to Check | : 84 of 84 | | | | | |
| \$ | | | | | | |
| All Rule All Rule All Clo All Clo A | es (84) orplan (5) ck (5) nk (19) I (1) Buf (2) 3 (23) P48 (5) MB16 (1) lation (10) tial Reconfig (11) tilst (2) | | | | | |
| | | | | | | |
| | Select All Clear All | | | | | |
| | OK Cancel | | | | | |
| | ¥1135 c2 59 05010 | | | | | |

Figure 3-48: **Run DRC Dialog Box**

- 4. A new tab, DRC Results, appears at the bottom with a sub-tab named as specified in the Run DRC window (results_1 in this case). There should be two violations:
 - a. PRSC #1: This warning indicates that a configuration has not been defined for the U1_AES1 module. This warning goes away when at least one implementation has been completed.
 - b. PRSC #2: This warning indicates that a configuration has not been defined for the U2_AES2 module. This warning goes away when at least one implementation has been completed.

For a complete list of DRCs and their descriptions, consult the *PlanAhead User Guide* in the Help pull-down menu.

Running TimeAhead with the PlanAhead Tool

This section tests the timing constraints that were set up in "Setting Up Timing Constraints with the PlanAhead Tool," page 57.

1. Select **Tools** \rightarrow **Run TimeAhead** and click **OK** (see Figure 3-49).

| 🛱 Run TimeAhead 🛛 🛛 🔀 | | | | | |
|--|--------------------------|--------------------|--|--|--|
| Results Name: re | sults_1 | | | | |
| Delay Options | Delay Options | | | | |
| Interconnect: Estimated Speed Grade: -2 (default) | | | | | |
| Report Options | | | | | |
| From Pins/Instan | ices: | | | | |
| Through Pins/Ins | tances/Nets: | | | | |
| To Pins/Instance | s: | | | | |
| Sort Paths By: | | Slack 🔽 | | | |
| Transition: | Transition: Rise/Fall 🗸 | | | | |
| Setup/Hold Mode | Setup/Hold Mode: Setup 🗸 | | | | |
| Total Number of | Paths: | 10 📚 | | | |
| Number of Paths | per Endpoint: | 1 📚 | | | |
| | ОК | Cancel | | | |
| | | X1135 c2 53 050809 | | | |

Figure 3-49: **Run TimeAhead Dialog Box**

2. A new tab, Timing Results, appears at the bottom with a sub-tab named as specified in the Run TimeAhead window (results_1 in this case). As specified when launched, TimeAhead reports the 10 paths closest to missing timing.

Design Flow Progress

The DRC and Timing block of the SCC system design flow diagram is complete, as shown in Figure 3-50.





Exporting the Design

At this stage, the design can be exported. The user can export either the combined netlist in EDIF form, the top-level netlist, or both, if desired. For this lab, only the combined netlist need be exported for checking by the Isolation Verification Tool.

- 1. To export the netlist, select **File** \rightarrow **Export Netlist**.
- 2. Browse to the ... \Xilinx_Design directory and select **OK**.
- 3. Keep the default file name fp_v51x85_ff676-2_1.ucf
- 4. To export the constraints file, select File \rightarrow Export Constraints and browse to the desired location.
- 5. Click **OK**.



Chapter 4

Running the Isolation Verification Tool Against the UCF

The Xilinx Isolation Verification Tool (IVT) software verifies that an FPGA design that has been partitioned into isolated modules meets the stringent standards for a fail-safe design. IVT is a batch application with a command line and file-based user interface. While there is a graphical output there is no graphical user interface.

While not required, it is highly recommended that the user run IVT on the UCF. This can catch pin and area group isolation faults early in the design when changes are more easily integrated. The steps in this chapter guide the reader through the process. After implementation, the IVT NCD test (required for SCC designs) is run against the routed design.

Creating the File Used to Run the IVT UCF Test

These steps describe how to create the file used to run the IVT UCF test:

- 1. Open the directory ... \Xilinx_Design\ivt.
- 2. Notice the files:
 - ivt.exe the IVT executable for IVT, version 5.0
 - IVT_End_User_License_Agreement.pdf the IVT user license
- 3. Create a new text file and name the file SCC-LAB_ucf.ivt with the following contents:

-device xc5vlx85 -package ff676

| # Groups # | Isolation Group | Area Group |
|---|---------------------------|--|
| " -group -group -group | AES AES_r COMPARE | pblock_U1_AES1 pblock_U2_AES2 pblock_U3_Comp |
| <pre># Pin Isolation Groups -pig SCC-LAB.pig</pre> | | |
| <pre># User Constraint File\PlanAhead\Floorplan</pre> | _SCC\fp_v5lx85_ff676-2_1. | ucf |
| # Output file | | |

- -output SCC-LAB_ucf.rpt
- 4. Notice the instructions placed in the IVT command file:
 - a. The first line sets the target device and package for IVT to compare against

- b. Three isolation groups are assigned: AES, AES_r, and COMPARE
 Note: These names are arbitrary. If desired they can be named the same to denote RED and BLACK groups.
- c. Three Area groups are assigned: pblock_U1_AES1, pblock_U2_AES2, and pblock_U3_Comp

Note: These names must match the area groups in the UCF.

- d. IVT is pointed to the UCF
- e. IVT is told where to place the output file and it's associated name
- 5. Save and close the IVT command file.

Creating the Pin Isolation Group File

The pin isolation group (PIG) file is an IVT command file that defines which pins are associated with what isolation groups. These steps describe the process for creating a PIG file:

- 1. Open the directory .. \Xilinx_Design\ivt.
- 2. Create a new text file and name the file SCC-LAB.pig with the following contents:

```
# Place all Global (top level) signals here (each commented out)
# NET "clk" LOC = F14;
```

```
ISOLATION _GROUP AES BEGIN
NET "reset" LOC = D11;
NET "push_button" LOC = D10;
END ISOLATION_GROUP
```

```
ISOLATION_GROUP AES_r BEGIN
# There are no pins in AES_r
END ISOLATION_GROUP
```

```
ISOLATION_GROUP COMPARE BEGIN
NET "led" LOC = P6;
END ISOLATION_GROUP
```

- 3. Notice the instructions placed in the IVT PIG file:
 - a. The clock pin, clk, is commented out because it is not required to be isolated.
 - b. The three isolation groups that were created in the IVT UCF command file have their associated pins assigned to them.
- 4. The isolation group definitions must match the isolation group definitions from the IVT command created in "Creating the File Used to Run the IVT UCF Test."

Hint: The IVT PIG file uses UCF syntax for each pin definition. It is useful to copy the pins from the UCF and place them in the PIG file for start. From there the user can either comment out the lines at the top level or add isolation group definitions around the remaining pins to assign them to their specific isolation group.

Running the IVT UCF Test

These steps describe how to run the IVT UCF test:

- 1. Open a DOS command prompt (**Start** \rightarrow **Run** \rightarrow **cmd**).
- 2. Navigate to the ... \Xilinx_Design\ivt directory.
- 3. Run the UCF test by typing the following at the command prompt: ivt -f SCC-LAB_ucf.ivt
- 4. The output for a successful UCF test run will be "SUCCESS!".
- 5. For more detailed messages from IVT, add the **-verbose** switch to the IVT command file or at the command line.

Hint: It is useful to add this command line to a file with the name run_ivt_ucf.bat so that it can be double-clicked from a Windows Explorer window.

Examining the Output from the IVT UCF Test

The output report has five key sections:

- 1. Isolation Groups and Area Groups
- 2. Pin Isolation Summary

```
Die Pin Adjacency Violations: 0
Package Pin Adjacency Violations: 0
Bank Isolation Violations: 0
```

3. Area Fault Summary

Area Group Faults: 0

4. Isolation Verification Summary

```
UCF file contains 0 constraint violations.
Isolation analysis completed.
Elapsed time: 0:00:03
```



Chapter 5

Implementing the Design with the PlanAhead Tool

Generating and Running an Implementation

These steps describe how to generate and run a design implementation:

- 1. From the tools menu, select **Run Implementation**.
 - a. Select **Tools** \rightarrow **Run Implementation** (see Figure 5-1).

| 🖸 Run Implement | ation 🔀 |
|------------------------|--|
| Run Name: | impl_1 |
| Description | ISE Defaults, including packing registers in IOs off |
| Floorplan: | ■ fp_v5lx85_ff676-2_1 (xc5vlx85ff676-2) |
| Part: | < From floorplan > |
| Constraints: | < From floorplan > |
| Strategy: | ISE Defaults (ISE 11) |
| Launch Options: | Launch on local host |
| Specify Configuration: | U1_AES1=module_1 U2_AES2=module_1 |
| | OK Cancel |

Figure 5-1: Run Implementation

- b. Accept the defaults and click **OK**.
- A new tab named Design Runs is created and a single entry named "impl_1" is generated as specified in the Run Implementation window. It immediately starts the run: NGDBUILD → MAP → Place and Route.
- 3. The combined and routed design is placed in the following directory:

 $..\Xilinx_Design\PlanAhead\FloorPlan_SCC\FloorPlan_SCC.runs\impl\floorplans\fp_v5lx85_ff676-2_1\impl_1$

The file name is impl_1_routed.ncd and its view in FPGA Editor is shown in Figure 5-2.



Figure 5-2: FPGA Editor View

XILINX_®

Chapter 6

Verifying the Design with the NCD Isolation Verification Tool

Creating the File Used to Run the IVT NCD Test

These steps describe how to create the file used to run the IVT NCD test:

- Open the directory \Xilinx_Design\ivt\.
- Create a new text file in the \Xilinx_Design\ivt\ directory: SCC-LAB_ncd.ivt
- 3. Open text file SCC-LAB_ncd.ivt in a text editor.
- 4. Add the following to the SCC-LAB_ncd.ivt text test file.

| # C | omment | the next | line for | reduced det | ail in | the | repor | t file |
|------------|--------|--------------|----------|------------------|--------|------|--------|--------|
| # G # - | roups | Isolati | on Group | Instanc | e Name | in F | inal : | NCD |
| -gr -gr | oup | AES AES_r | | U1_AES U2_AES | 1 2 | | | |
| -group | | COMPARE | | U3_Com | ıp | | | |

```
# Combined design
..\PlanAhead\FloorPlan_SCC\FloorPlan_SCC.runs\impl\floorplans\
fp_v51x85_ff676-2_1\impl_1\impl_1_routed.ncd
```

```
# Output Report File
-output SCC-LAB_ncd.rpt
```

The NCD IVT command file sets the following options:

- Enables the verbose IVT switch
- Assigns three area groups to the three NCD files making up the project
- Points the IVT tool to the combined NCD file
- Tells IVT what to name the output report file and where to put the file

Note: The Isolation Group names are arbitrary but the instance name must match the actual design.

Running the IVT NCD Test

These steps describe how to run the IVT NCD test:

- 1. Open a command prompt (**Start** \rightarrow **Run** \rightarrow **cmd**).
- 2. Navigate to the \Xilinx_Design\ivt\ directory.
- 3. To run the IVT NCD test, type the following at the command prompt: ivt -f SCC-LAB_ncd.ivt
- 4. The output for a successful NCD test run will be "SUCCESS!".

Examining the Output from the IVT NCD Test

These steps describe how to read the output file from the IVT NCD test:

1. Open the IVT NCD test output file:

\Xilinx_Design\ivt\SCC-LAB_ncd.rpt

All the groups are listed in the input designs section:

Input Designs

Group AES module: U1_AES1 Group AES_r module: U2_AES2 Group COMPARE module: U3_Comp

2. Ensure that Clocks and Resets are listed in the unidentified shared networks section. For SCC Trusted Routing designs, signals shared between isolated regions are expected and intended and will show up here. An example of global signals are the following Global Clock signals, which are expected to be shared outside of an isolated region:

Unidentified Shared Networks

The networks below are found in multiple isolation groups, therefore it is incumbent upon the user to prove these signals do not violate data isolation requirements. Only power, ground, bus macros, global resets or explicitly permitted control signals may be shared.

clk_ibufg clk0_buf clkdev_buf

3. Ensure all remaining Clocks are listed in the Networks Driven by Global Clock Sources section:

Networks Driven by Global Clock Sources (BUFG, DCM, PLL, and PMCD)

clk_fb_i clk_i

- 4. Notice that in the Identified Networks section, Shared Networks Attached to Bus Macros are listed. All networks passing through Bus macros are considered safe by IVT and are listed in this section.
- 5. The next section in the report contains Shared Networks Without General Routing (Including Clock Networks) information. As stated in the report file each of the following networks is either internal to a logic block (such as a CLB or an IOB) or is part of a global clock network, therefore they do not need to be considered for isolation analysis.

- 6. Ensure that only bus macros are listed in the Bus Macro Usage Summary.
- 7. Pay special attention to the Package Pins, I/O Buffers, and I/O Banks because the note states the user must verify that pins connected to ignored networks are correct.

Package Pins, I/O Buffers, and I/O Banks

Note: It is incumbent on the user to verify that pins connected to ignored networks are correct. For example, pins must not be directly connected to bus macros, but pins can be connected to clocks, power, and global resets.

| Pin(col, | row) | Bank | I/O Buffer | Isolation Group | Network |
|----------|------|------|------------|-----------------|--------------------------|
| | | | | | |
| D10(16, | 22) | 16 | IOB_X2Y198 | AES | U1_AES1/push_button_IBUF |
| D11(15, | 22) | 16 | IOB_X2Y199 | AES | U1_AES1/reset_out |
| P6(20, | 12) | 12 | IOB_X2Y121 | COMPARE | U3_Comp/led_OBUF |
| F14(12, | 20) | 3 | IOB_X1Y179 | ignored | clk |

The following output in the NCD report indicates that there are no faults in the NCD and lists the time it took to run the test:

Pin Isolation Summary

Die Pin Adjacency Violations: 0 Package Pin Adjacency Violations: 0 Bank Isolation Violations: 0

Network Isolation Summary

Net Isolation Faults Found: 0 Net Isolation Faults Reported: 0 Routes Examined: 389030

Isolation Verification Summary

Design contains 0 constraint violations.

Isolation analysis completed.

Elapsed time: 0:03:35

Design Flow Progress

The Single Chip Crypto lab is complete with the IVT on NCD File block of the flow diagram, as shown in Figure 6-1.



