# Single Chip Crypto Lab Using PR/ISO Flow with the Virtex-5 Family for ISE Design Suite 12.1

XAPP1105 (v1.1.2) June 19, 2013

XILINX®

**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 or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in Critical Applications: http://www.xilinx.com/warranty.htm#critapps.

## Revision History

The following table shows the revision history for this document.

| Date | Version | Revision |
|---|---|---|
| 07/20/10 | 1.0 | Initial Xilinx release. |
| 09/27/10 | 1.1 | Added description of **PRIVATE** attribute settings to Defining Attributes for each RP or ISO Partition with the PlanAhead Tool. |
| 08/29/11 | 1.1.1 | Removed EAR banner from first page. Updated disclaimers. |
| 06/19/13 | 1.1.2 | Replaced SCC lounge with IDF page throughout. Added URL to download reference design in Reference Design Files. |

# Table of Contents

## Chapter 5: Implementing the Design with the PlanAhead Tool

## Chapter 6: Verifying the Design with the NCD Isolation Verification Tool

## Appendix A: Modified Implementation of the PlanAhead Tool 12.1

# 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, Floorplanning the System, gives step-by-step instructions for implementing the SCC design.
- Chapter 4, Running the Isolation Verification Tool Against the UCF, covers running the Isolation Verification Tool (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.
- Appendix A, Modified Implementation of the PlanAhead Tool 12.1, describes a modified method for implementing the design using the PlanAhead™ tool 12.1.

## Additional Resources

To find additional documentation, see the Xilinx website at:

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

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

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

# Conventions

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

## Typographical

The following typographical conventions are used in this document:

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

## 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.<br>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. |

# Design Challenge

From the instructions in this chapter, a single chip crypto system can be created and implemented utilizing an isolated, redundant Advanced Encryption Standard (AES) module targeting a Virtex®-5 or Virtex-5Q FPGA utilizing the ISE® Design Suite 12.1 and the PlanAhead™ design tools 12.1. It is necessary to use the ISE design suite 12.1 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.

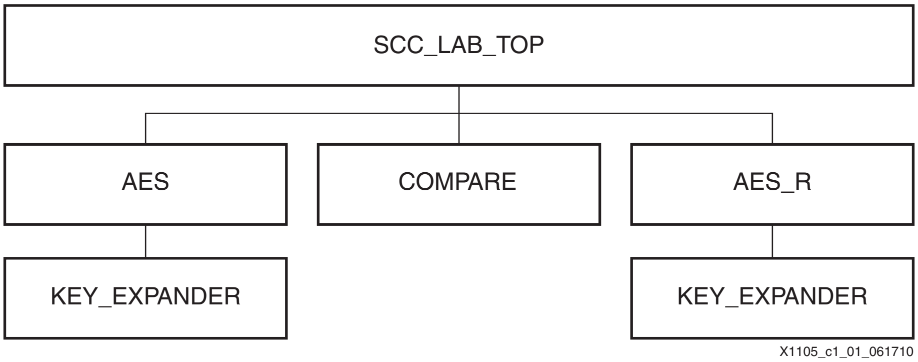


*Figure 1-1:* **VHDL Design Hierarchical Block Diagram**

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 1-2 shows the design flow used for the single chip crypto lab.

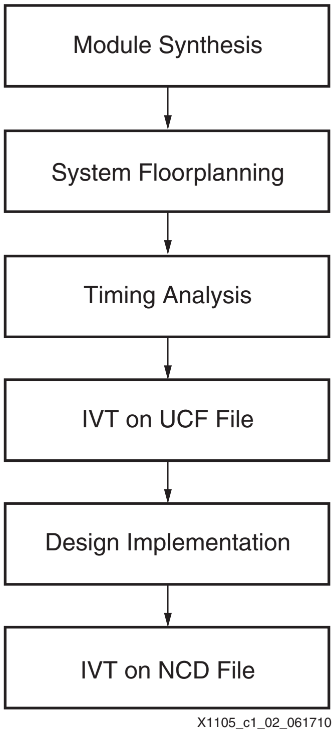


*Figure 1-2:* **SCC System Design Flow**

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

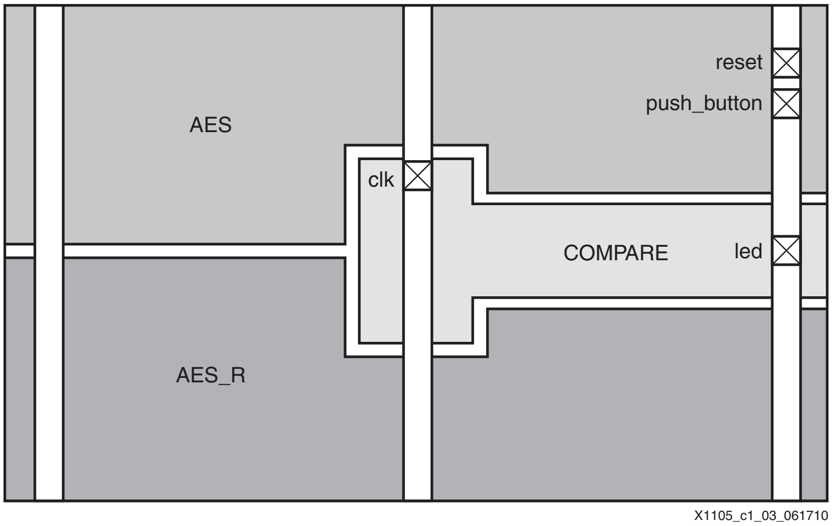
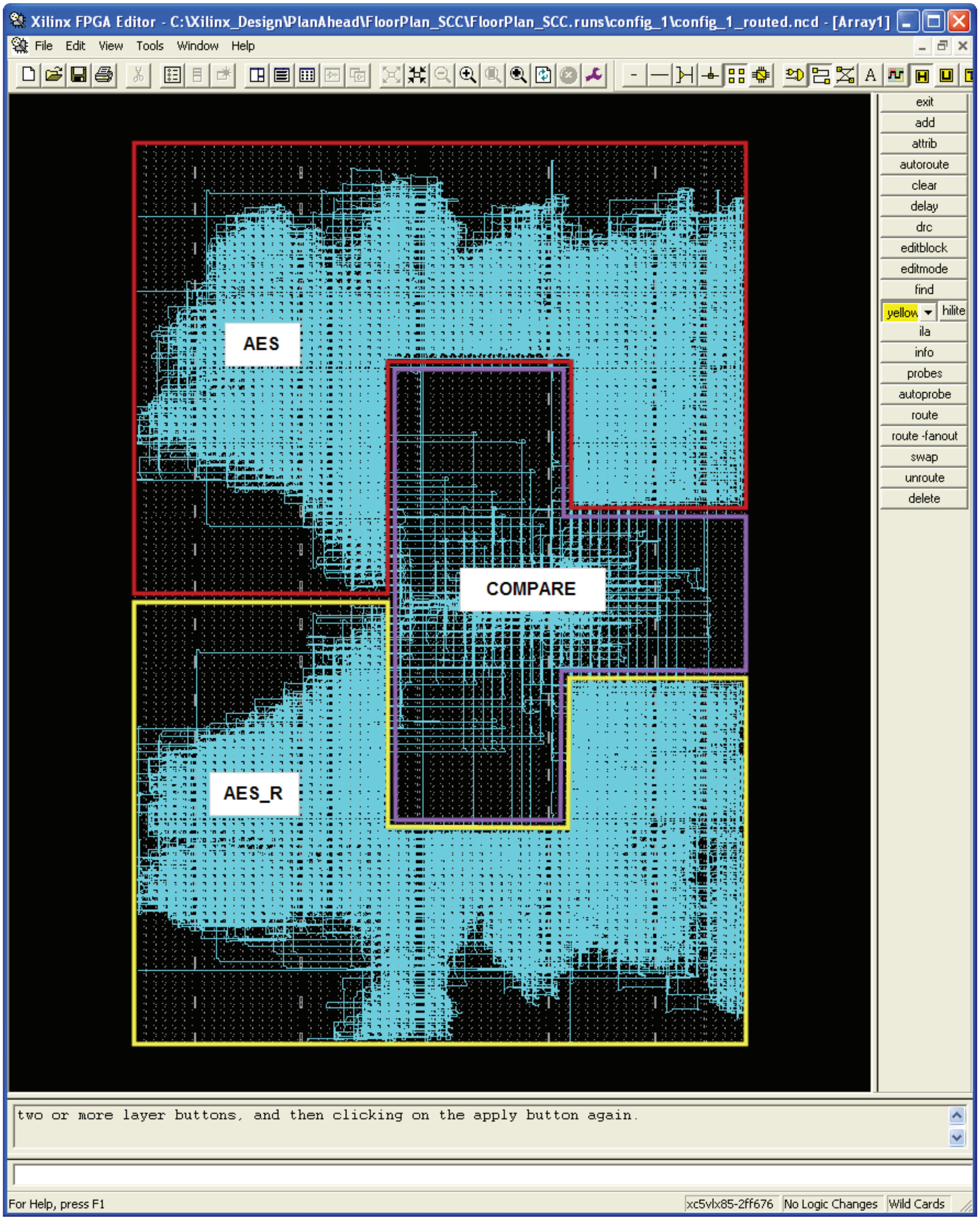


*Figure 1-3:* **Die View: Partition and I/O Map of XC5VLX85-2FF676**

Figure 1-4 is a view of the completed XSCC reference design 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.



```
two or more layer buttons, and then clicking on the apply button again.
```

X1105_c1_04_061810

*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=343383

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

*Table 1-1:* **Design Checklist**

| 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 third 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 design suite 12.1 |
| SPICE/IBIS simulations | N |
| **Implementation** | |
| Synthesis software tools/version used | XST 12.1 |
| Implementation software tools/versions used | ISE design suite 12.1 |
| Static timing analysis performed | Y |
| **Hardware Verification** | |
| Hardware verified | N |
| Hardware platform used for verification | N/A |

## Installing 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 `XAPP1105.zip` to the Windows desktop.
2. Double-click on `XAPP1105.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 (SCC_LAB_TOP) is synthesized first, followed by each of the lower-level modules (`aes`, `aes_r`, and `compare`).

## Synthesize the SCC_LAB_TOP Module

These steps describe how to synthesize the SCC_LAB_TOP module:

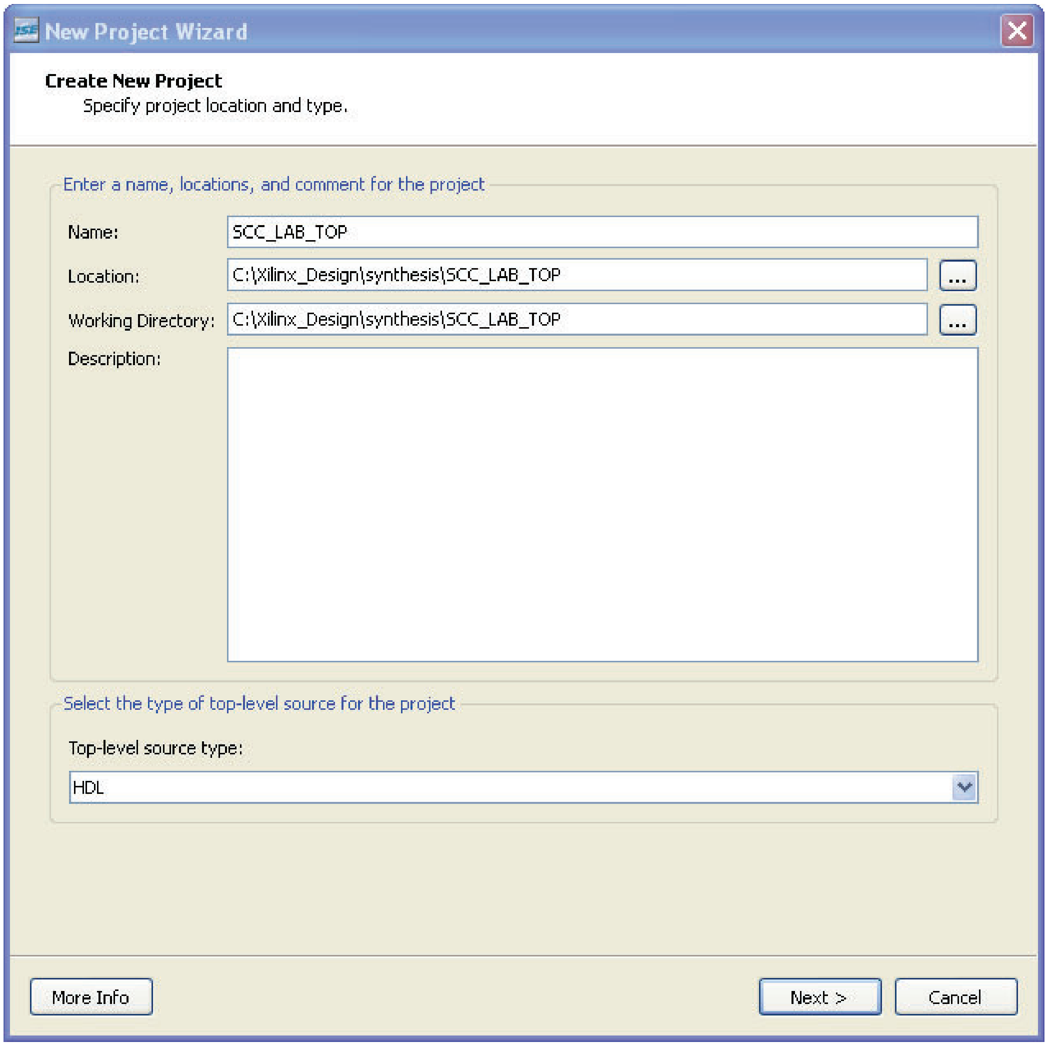
1. Start the ISE® design suite 12.1:

   **Start** → **All Programs** → **Xilinx ISE Design Suite 12.1** → **ISE Design Tools** → **Project Navigator**

2. Create a new ISE design suite 12.1 project:

   **File** → **New Project**

3. Set the project location to **`\Xilinx_Design\synthesis`**.
4. Set the project name to **`SCC_LAB_TOP`**.
5. Click **Next** (see Figure 2-1).



Create New Project
Specify project location and type.

Enter a name, locations, and comment for the project

Name: SCC_LAB_TOP

Location: C:\Xilinx_Design\synthesis\SCC_LAB_TOP

Working Directory: C:\Xilinx_Design\synthesis\SCC_LAB_TOP

Description:

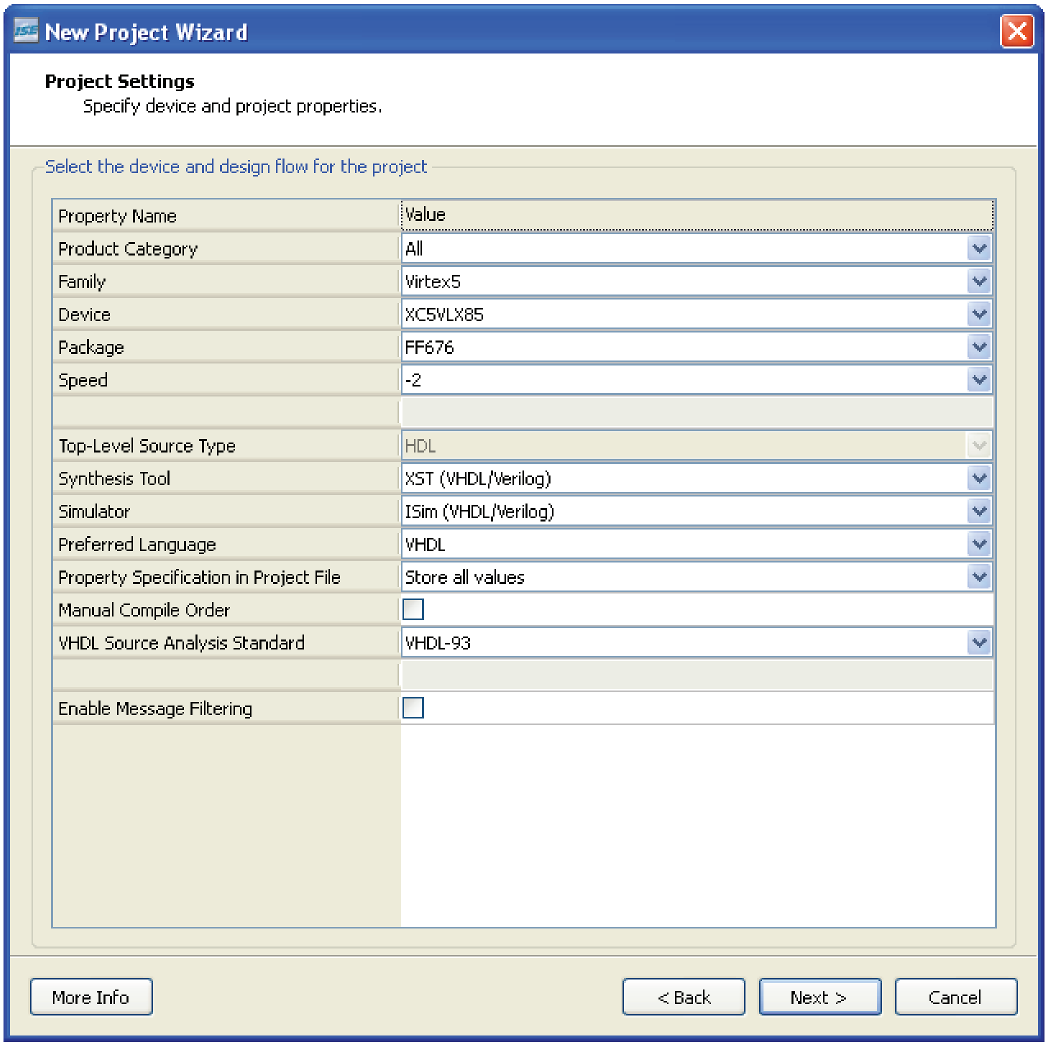
Select the type of top-level source for the project

Top-level source type: HDL

X1105_c2_01_061810

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

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



**Project Settings**

Specify device and project properties.

Select the device and design flow for the project

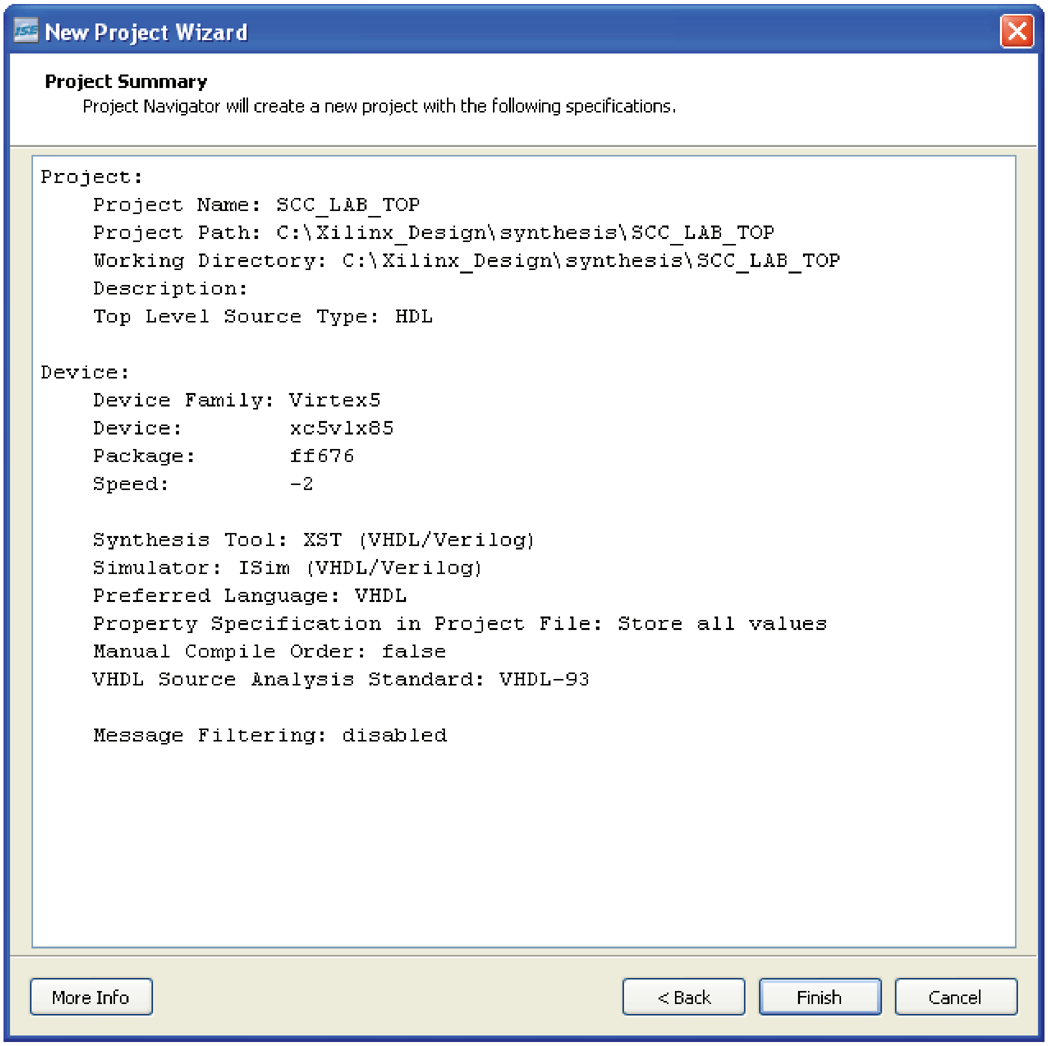
| Property Name | Value |
|---|---|
| Product Category | All |
| Family | Virtex5 |
| Device | XC5VLX85 |
| Package | FF676 |
| Speed | -2 |
| | |
| Top-Level Source Type | HDL |
| Synthesis Tool | XST (VHDL/Verilog) |
| Simulator | ISim (VHDL/Verilog) |
| Preferred Language | VHDL |
| Property Specification in Project File | Store all values |
| Manual Compile Order | □ |
| VHDL Source Analysis Standard | VHDL-93 |
| | |
| Enable Message Filtering | □ |

X1105_c2_02_061810

*Figure 2-2:* **New Project Wizard (Project Settings)**

7. Click **Next**.

8. Click **Finish** in the Project Summary window (see Figure 2-3).



**Project Summary**

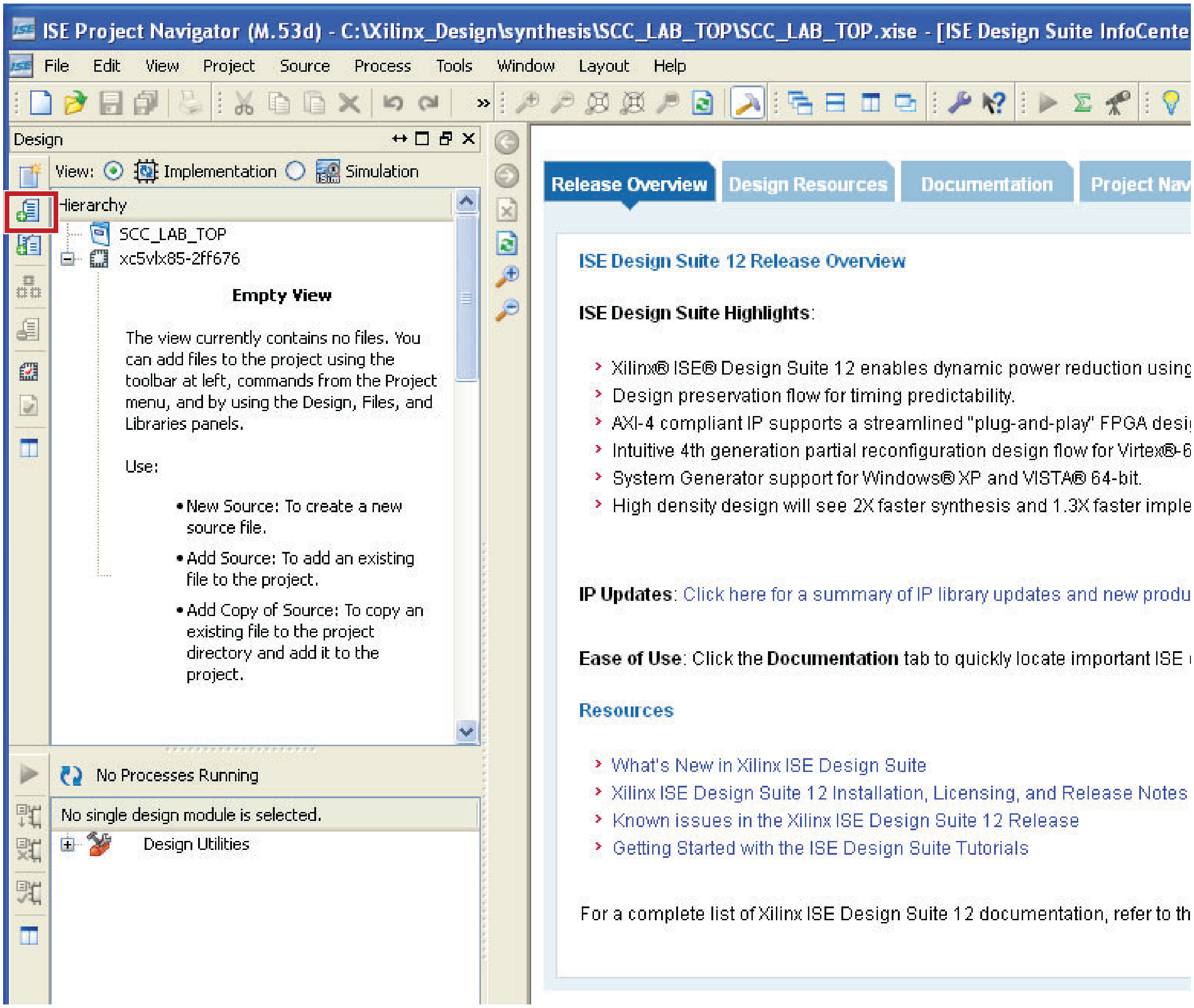
Project Navigator will create a new project with the following specifications.

```
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:        xc5vlx85
    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
    VHDL Source Analysis Standard: VHDL-93

    Message Filtering: disabled
```

X1105_c2_03_061810

*Figure 2-3:* **New Project Wizard (Project Summary)**

9. Select the **Add Source** button on the left side of the Design window in the Project Navigator of the ISE tool (see Figure 2-4).



ISE Project Navigator (M.53d) - C:\Xilinx_Design\synthesis\SCC_LAB_TOP\SCC_LAB_TOP.xise - [ISE Design Suite InfoCente
File Edit View Project Source Process Tools Window Layout Help
Design
View: ● Implementation ○ Simulation
Hierarchy
SCC_LAB_TOP
xc5vlx85-2ff676
Release Overview
Design Resources
Documentation
Project Nav
No Processes Running
Design Utilities

Empty View

The view currently contains no files. You can add files to the project using the toolbar at left, commands from the Project menu, and by using the Design, Files, and Libraries panels.

Use:

- New Source: To create a new source file.
- Add Source: To add an existing file to the project.
- Add Copy of Source: To copy an existing file to the project directory and add it to the project.

No single design module is selected.

ISE Design Suite 12 Release Overview

ISE Design Suite Highlights:

- Xilinx® ISE® Design Suite 12 enables dynamic power reduction using
- Design preservation flow for timing predictability.
- AXI-4 compliant IP supports a streamlined "plug-and-play" FPGA desi
- Intuitive 4th generation partial reconfiguration design flow for Virtex®-6
- System Generator support for Windows® XP and VISTA® 64-bit.
- High density design will see 2X faster synthesis and 1.3X faster imple

**IP Updates**: Click here for a summary of IP library updates and new produ

**Ease of Use**: Click the **Documentation** tab to quickly locate important ISE

Resources

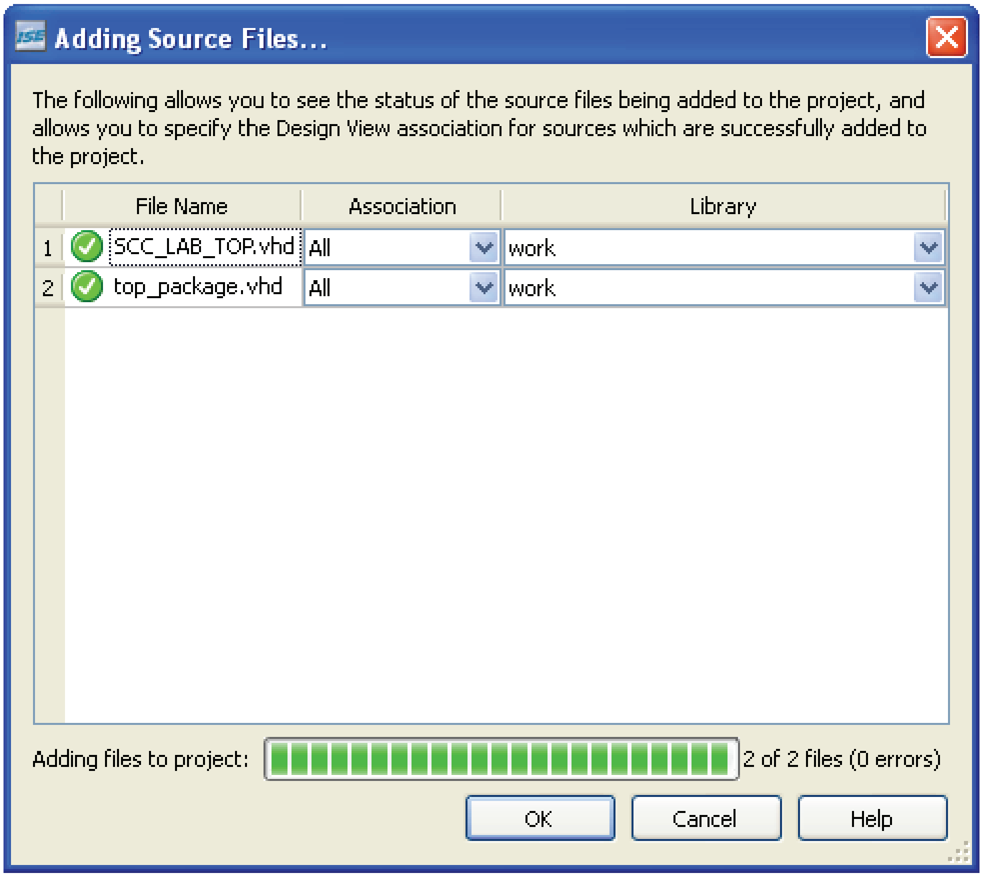
- What's New in Xilinx ISE Design Suite
- Xilinx ISE Design Suite 12 Installation, Licensing, and Release Notes
- Known issues in the Xilinx ISE Design Suite 12 Release
- Getting Started with the ISE Design Suite Tutorials

For a complete list of Xilinx ISE Design Suite 12 documentation, refer to th

X1105_c2_04_062210

*Figure 2-4:* **Project Navigator Window (Add Existing Sources)**

10. Navigate to the `source\design` directory, and add the `top_package.vhd` and `SCC_LAB_TOP.vhd` files to the project (see Figure 2-5). Click **OK**.



Adding Source Files...

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 | ✓ | SCC_LAB_TOP.vhd | All | work |
| 2 | ✓ | top_package.vhd | All | work |

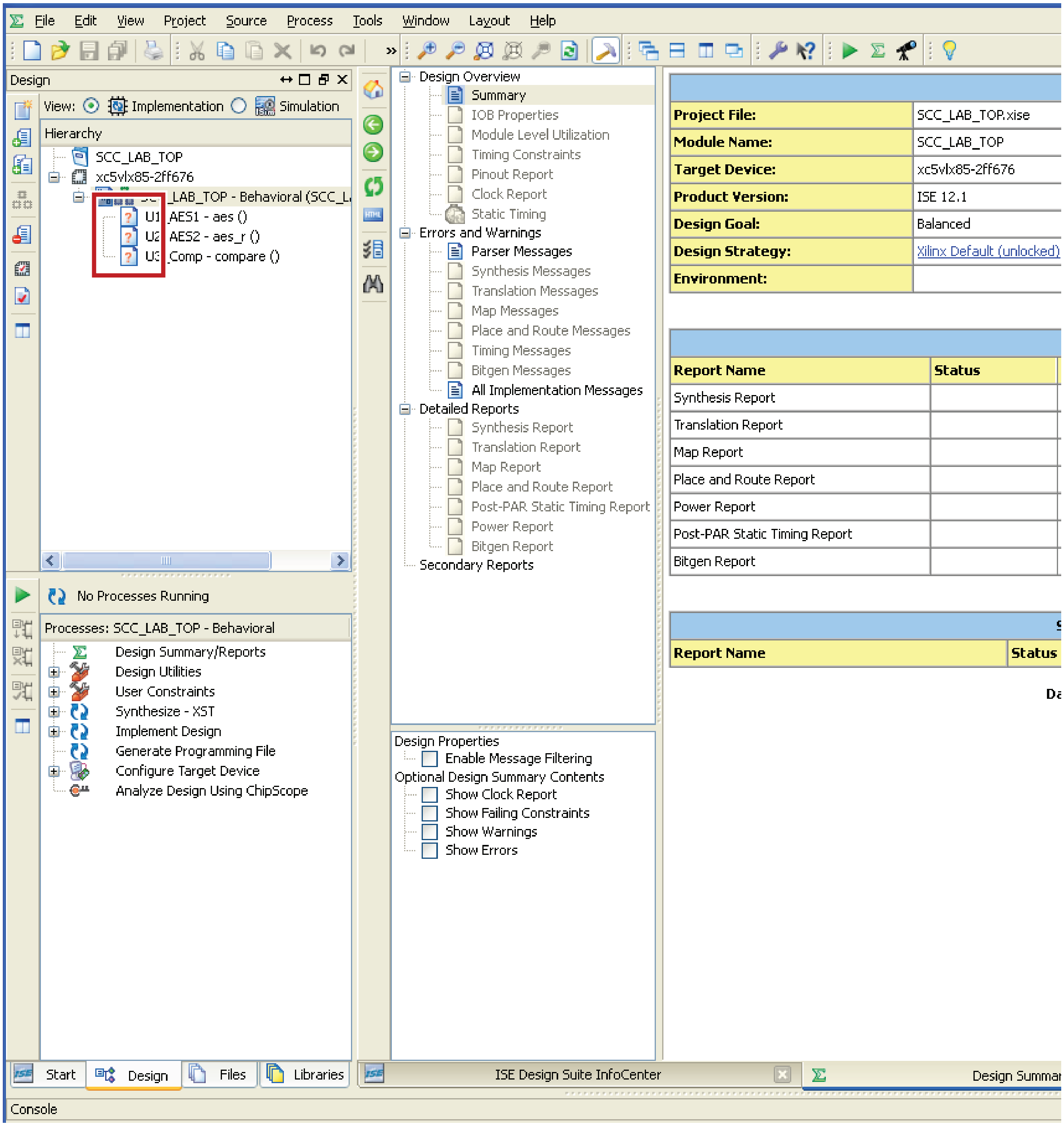
Adding files to project: 2 of 2 files (0 errors)

X1105_c2_05_061910

*Figure 2-5:* **Adding Source Files**

The Project Navigator window should look like the one 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.



| | |
|---|---|
| **Project File:** | SCC_LAB_TOP.xise |
| **Module Name:** | SCC_LAB_TOP |
| **Target Device:** | xc5vlx85-2ff676 |
| **Product Version:** | ISE 12.1 |
| **Design Goal:** | Balanced |
| **Design Strategy:** | Xilinx Default (unlocked) |
| **Environment:** | |

| **Report Name** | **Status** |
|---|---|
| Synthesis Report | |
| Translation Report | |
| Map Report | |
| Place and Route Report | |
| Power Report | |
| Post-PAR Static Timing Report | |
| Bitgen Report | |

| **Report Name** | **Status** |
|---|---|

X1105_c2_06_061910

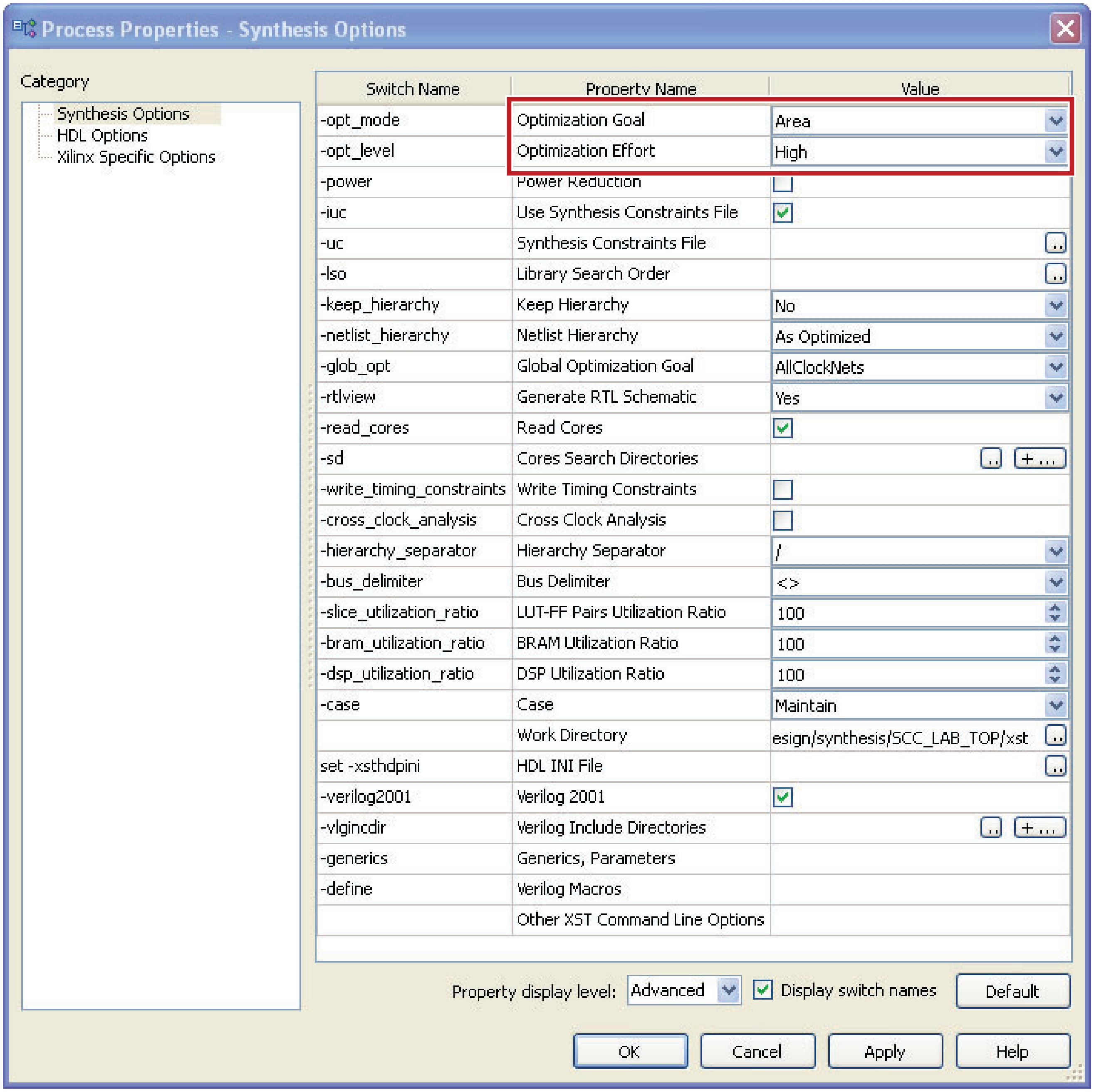
*Figure 2-6:* **Project Navigator Window**

11. 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 the Xilinx® Synthesis Technology (XST) synthesis tool to disable I/O insertion on these ports. The buffer_type attribute is necessary to guarantee that 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, this attribute should not be applied to the CLK pin of the top-level code.

12. 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 2-7). **Keep Hierarchy** does not have to be set on this module for this flow.



Process Properties - Synthesis Options

Category

- Synthesis Options
- HDL Options
- Xilinx Specific Options

| Switch Name | Property Name | Value |
|---|---|---|
| -opt_mode | Optimization Goal | Area |
| -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 | No |
| -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 | / |
| -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 |
| -case | Case | Maintain |
| | Work Directory | esign/synthesis/SCC_LAB_TOP/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 display level: Advanced ☑ Display switch names

X1105_c2_07_061910

*Figure 2-7:* **Process Properties (Synthesis Options)**

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

# Synthesize the aes Module

These steps describe how to synthesize the aes module:

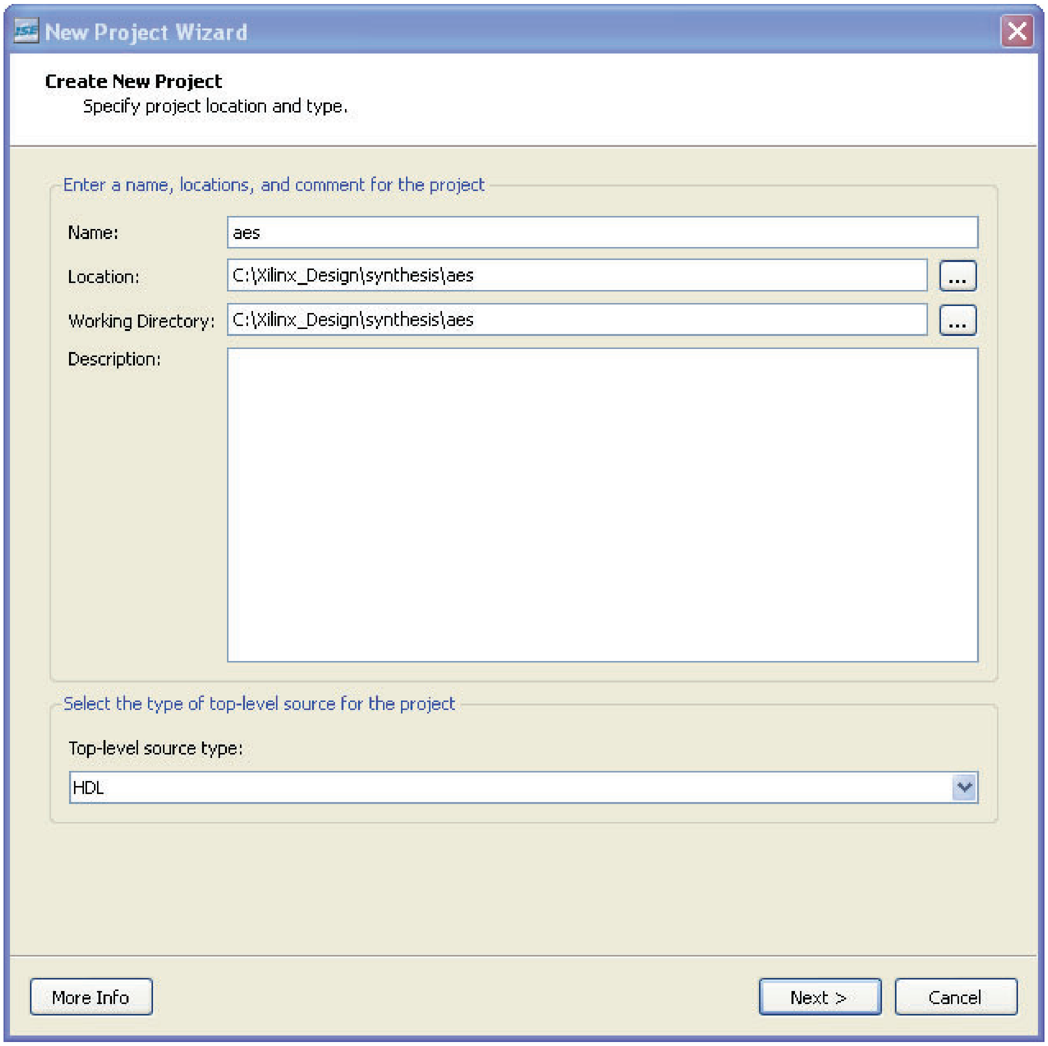
1. Start the ISE design suite 12.1.

   **Start** → **All Programs** → **Xilinx ISE Design Suite 12.1** → **ISE Design Tools** → **Project Navigator**

2. Create a new ISE design suite 12.1 project:

   **File** → **New Project**

3. Set the project location to **\Xilinx_Design\synthesis**.
4. Set the Project Name to **aes**.
5. Click **Next** (see Figure 2-8).



New Project Wizard

Create New Project
Specify project location and type.

Enter a name, locations, and comment for the project

Name: aes

Location: C:\Xilinx_Design\synthesis\aes

Working Directory: C:\Xilinx_Design\synthesis\aes

Description:

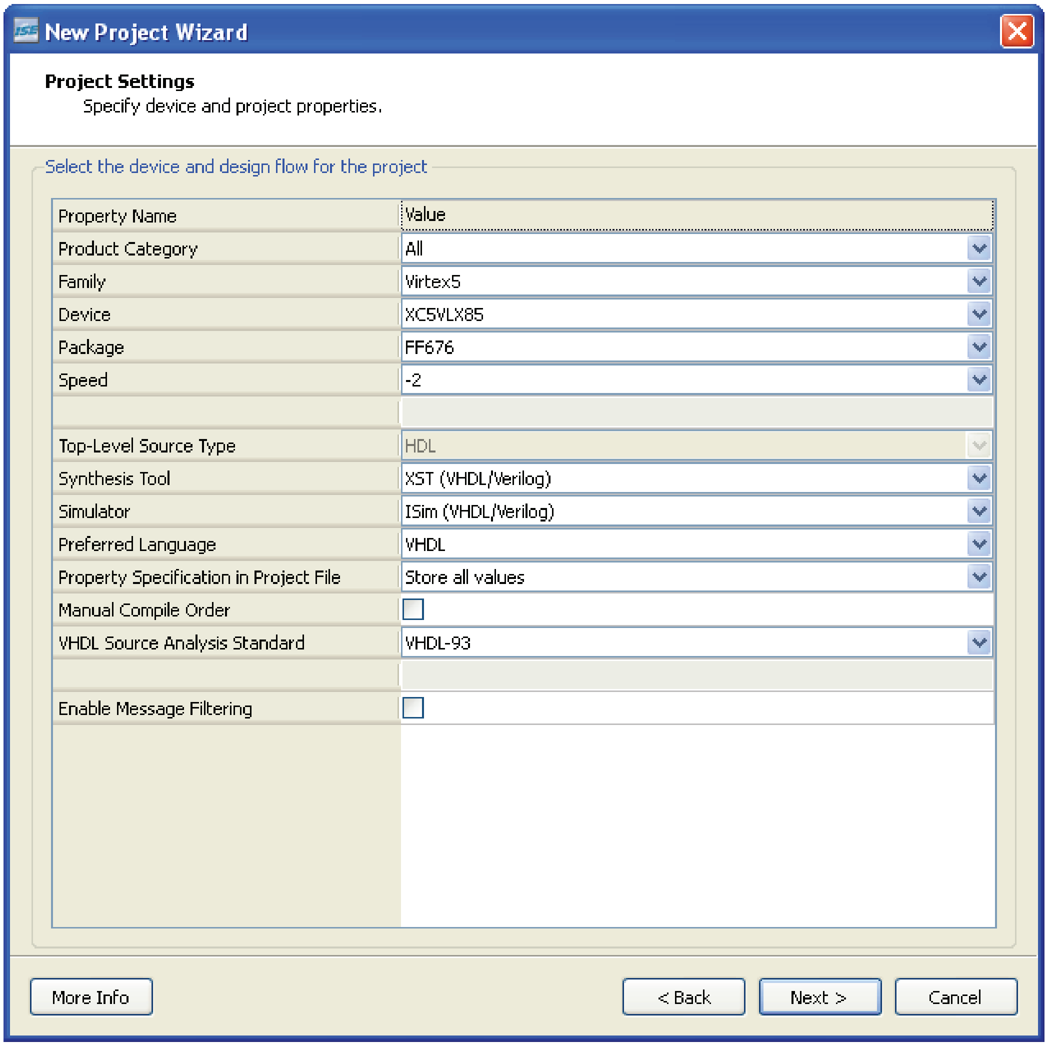
Select the type of top-level source for the project

Top-level source type: HDL

X1105_c2_08_061910

*Figure 2-8:* **New Project Wizard (Create New Project)**

6. Click **Next** (see Figure 2-9).



**Project Settings**

Specify device and project properties.

Select the device and design flow for the project

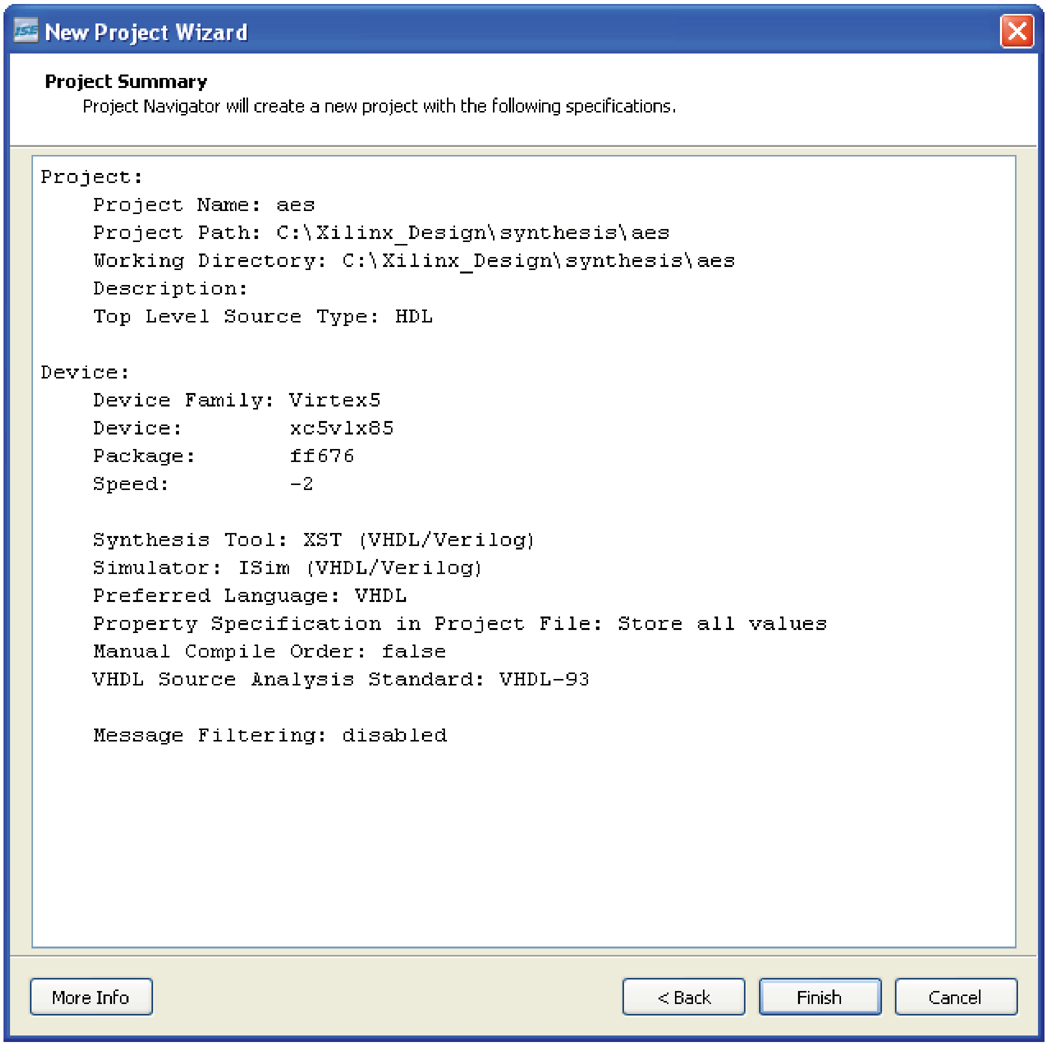
| Property Name | Value |
|---|---|
| Product Category | All |
| Family | Virtex5 |
| Device | XC5VLX85 |
| Package | FF676 |
| Speed | -2 |
| | |
| Top-Level Source Type | HDL |
| Synthesis Tool | XST (VHDL/Verilog) |
| Simulator | ISim (VHDL/Verilog) |
| Preferred Language | VHDL |
| Property Specification in Project File | Store all values |
| Manual Compile Order | □ |
| VHDL Source Analysis Standard | VHDL-93 |
| | |
| Enable Message Filtering | □ |

X1105_c2_09_061910

*Figure 2-9:* **New Project Wizard (Project Settings)**

7. Click **Next**.

8. Click **Finish** in the Project Summary window (see Figure 2-10).



**Project Summary**

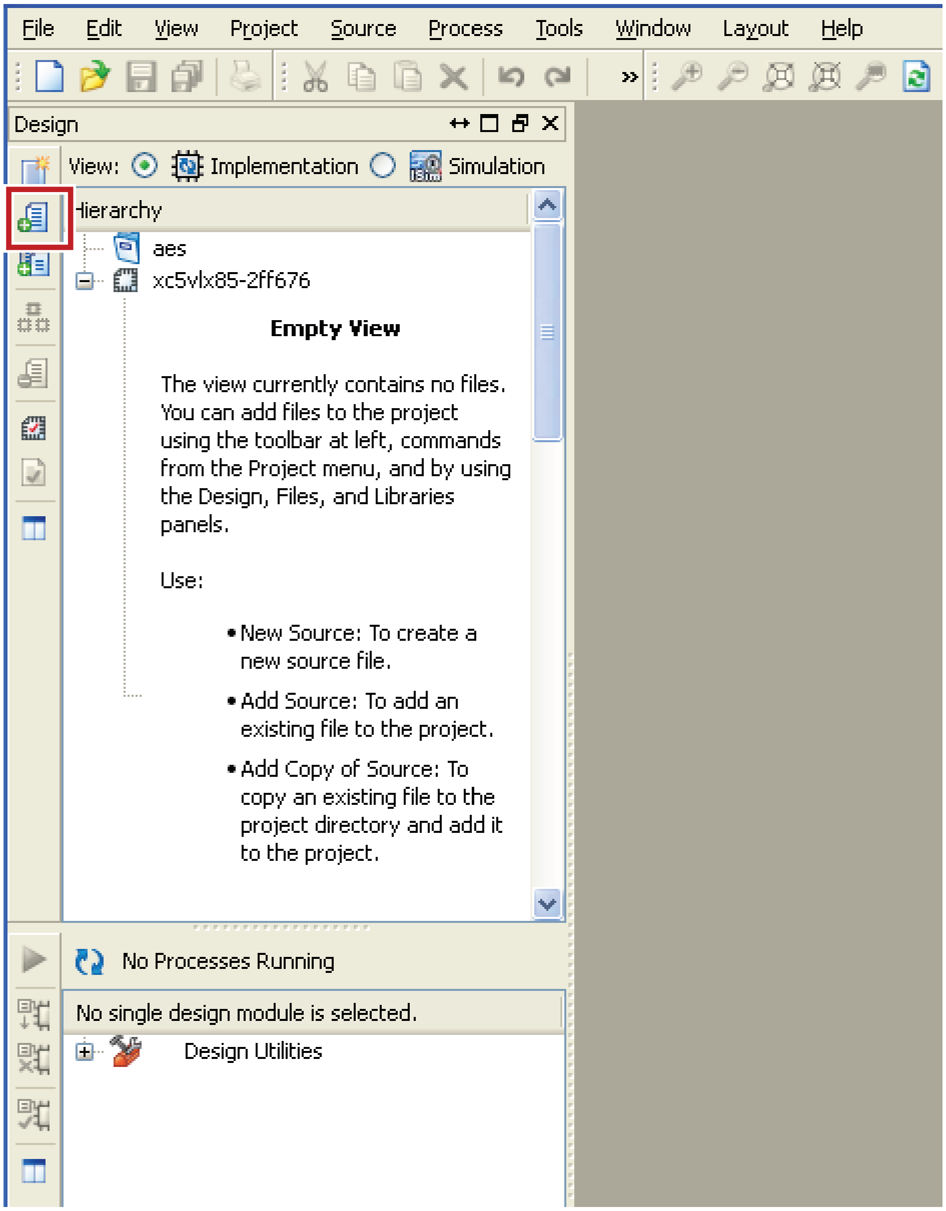
Project Navigator will create a new project with the following specifications.

```
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:         xc5vlx85
    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
    VHDL Source Analysis Standard: VHDL-93

    Message Filtering: disabled
```

X1105_c2_10_061910

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

9. Select the **Add Source** button on the left side of the Design window in the Project Navigator of the ISE tool (see Figure 2-11).


File Edit View Project Source Process Tools Window Layout Help
Design
View: Implementation Simulation
Hierarchy
aes
xc5vlx85-2ff676
No Processes Running
Design Utilities

**Empty View**

The view currently contains no files. You can add files to the project using the toolbar at left, commands from the Project menu, and by using the Design, Files, and Libraries panels.

Use:

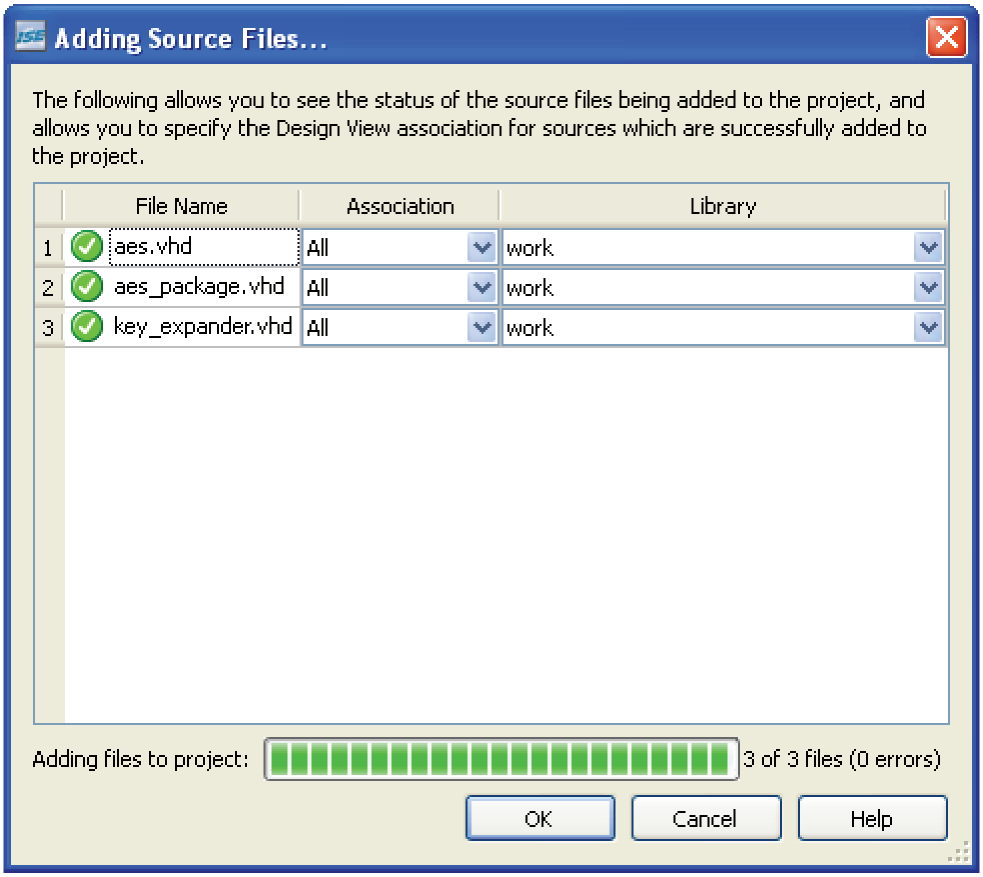
- New Source: To create a new source file.
- Add Source: To add an existing file to the project.
- Add Copy of Source: To copy an existing file to the project directory and add it to the project.

No single design module is selected.

X1105_c2_11_061910

*Figure 2-11:* **Project Navigator (Add Existing Sources)**

10. Navigate to the `source\design` directory and add the `aes.vhd`, `aes_package.vhd`, and `key_expander.vhd` files to the project (see Figure 2-12). Click **OK**.



Adding Source Files...

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 | ✓ aes.vhd | All | work |
| 2 | ✓ aes_package.vhd | All | work |
| 3 | ✓ key_expander.vhd | All | work |

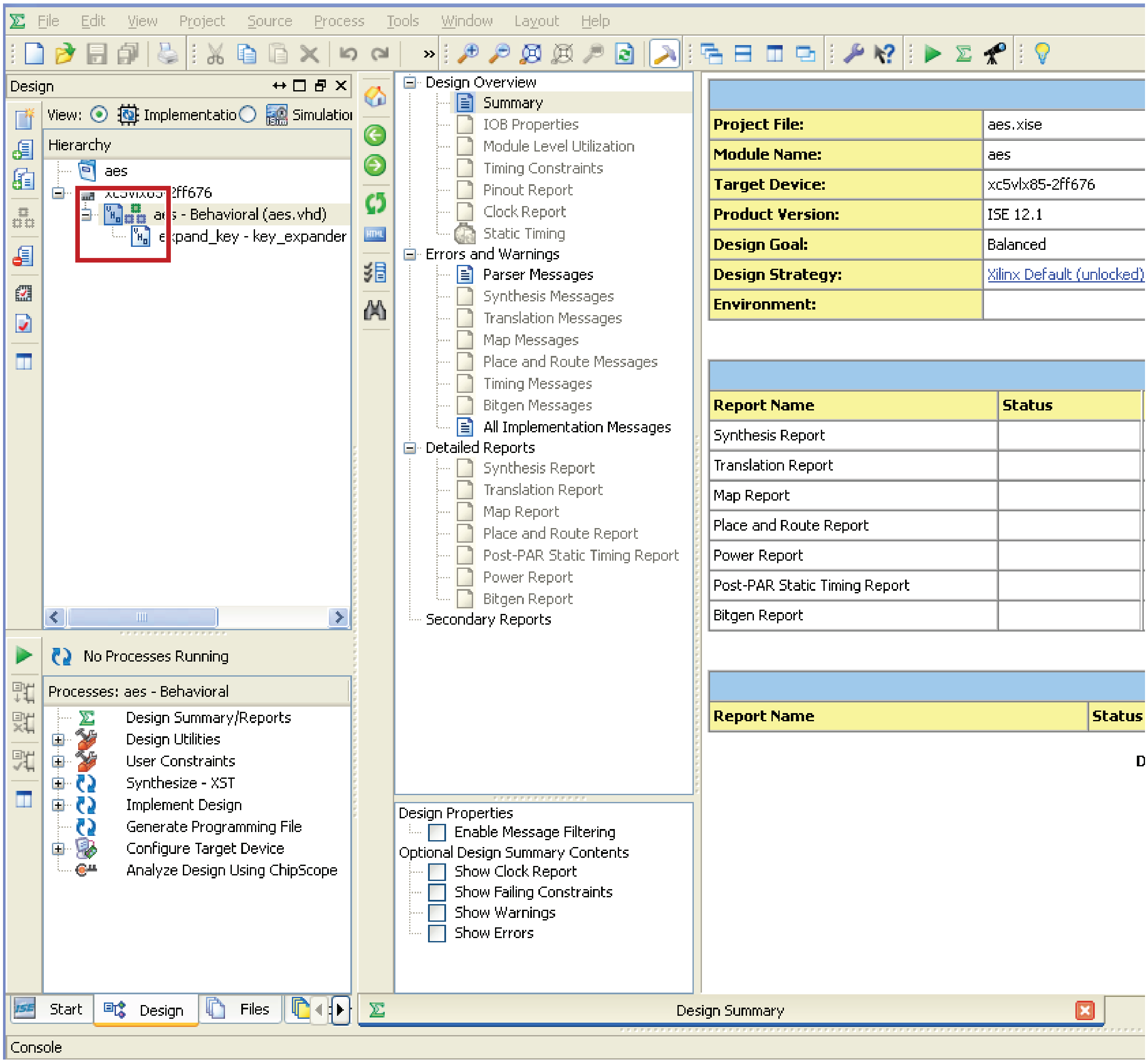
Adding files to project: 3 of 3 files (0 errors)

X1105_c2_12_061910

*Figure 2-12:* **Adding Source Files**

The Project Navigator window should look like the one shown in Figure 2-13.

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



| | |
|---|---|
| **Project File:** | aes.xise |
| **Module Name:** | aes |
| **Target Device:** | xc5vlx85-2ff676 |
| **Product Version:** | ISE 12.1 |
| **Design Goal:** | Balanced |
| **Design Strategy:** | Xilinx Default (unlocked) |
| **Environment:** | |

| Report Name | Status |
|---|---|
| Synthesis Report | |
| Translation Report | |
| Map Report | |
| Place and Route Report | |
| Power Report | |
| Post-PAR Static Timing Report | |
| Bitgen Report | |

| Report Name | Status |
|---|---|

X1105_c2_13_061910

*Figure 2-13:* **Project Navigator Window**

11. Open the aes.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 start      : signal is "none";
attribute buffer_type of mode       : signal is "none";
attribute buffer_type of load       : signal is "none";
attribute buffer_type of key        : signal is "none";
attribute buffer_type of data_in     : signal is "none";
attribute buffer_type of reset_out   : 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 XST 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.

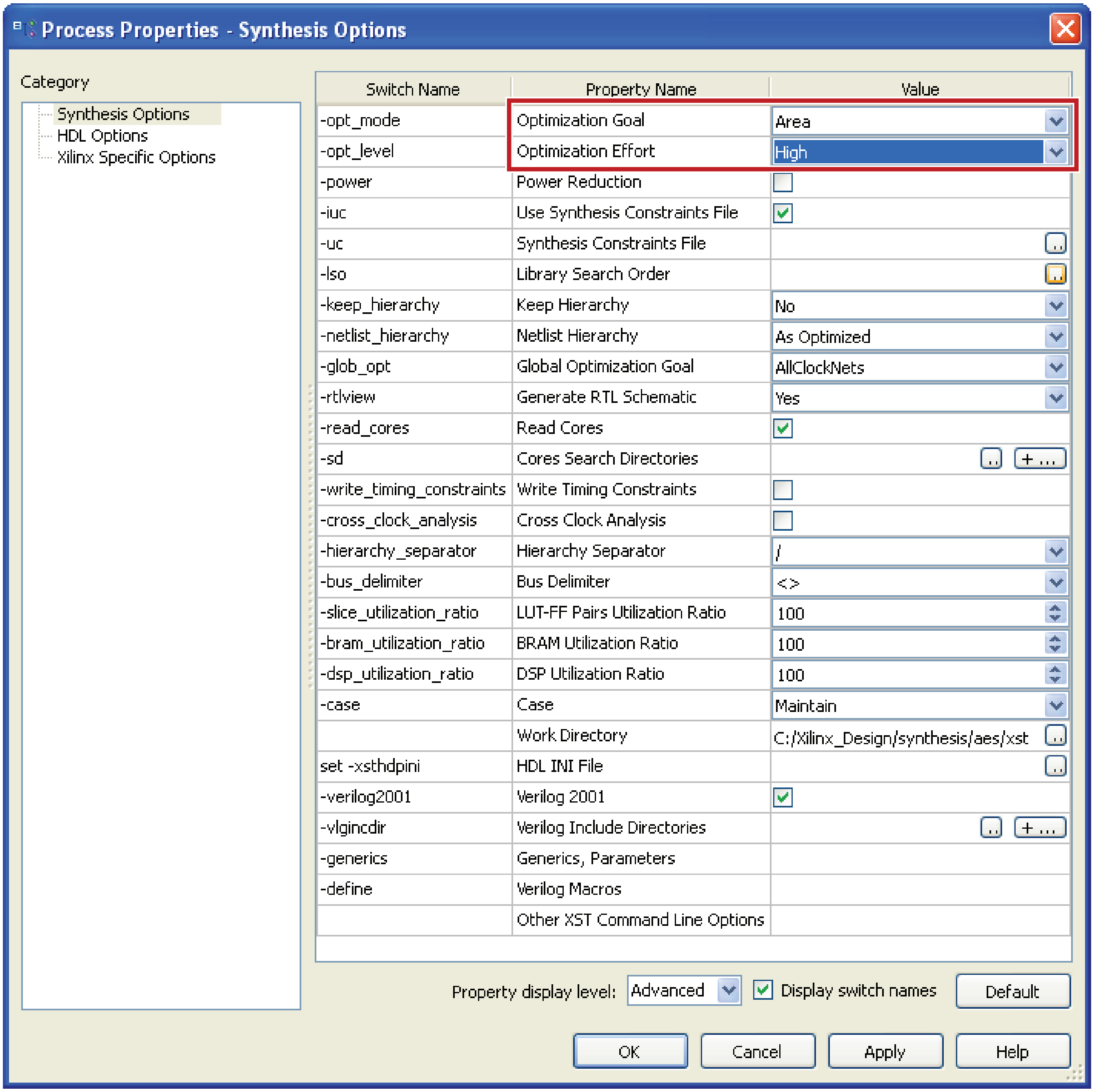
***Note:*** The reset and push_button signals do not have an attribute associated with them because they are "owned" by `aes` and therefore require an I/O to be inferred within the `aes` module rather than within SCC_LAB_TOP. The clk I/O was inferred when SCC_LAB_TOP was synthesized. The other signals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

12. 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, O => reset_out );
```

13. Right-click on the **Synthesize-XST** icon in the Processes window and select **Process Properties...**.

14. Set Optimization Goal to **Area**, set Optimization Effort to **High**, and click **OK** (see Figure 2-14).



Process Properties - Synthesis Options

Category

- Synthesis Options
- HDL Options
- Xilinx Specific Options

| Switch Name | Property Name | Value |
|---|---|---|
| -opt_mode | Optimization Goal | Area |
| -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 | No |
| -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 | / |
| -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 |
| -case | Case | Maintain |
| | Work Directory | C:/Xilinx_Design/synthesis/aes/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 display level: Advanced

☑ Display switch names

X1105_c2_14_061910

*Figure 2-14:* **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:

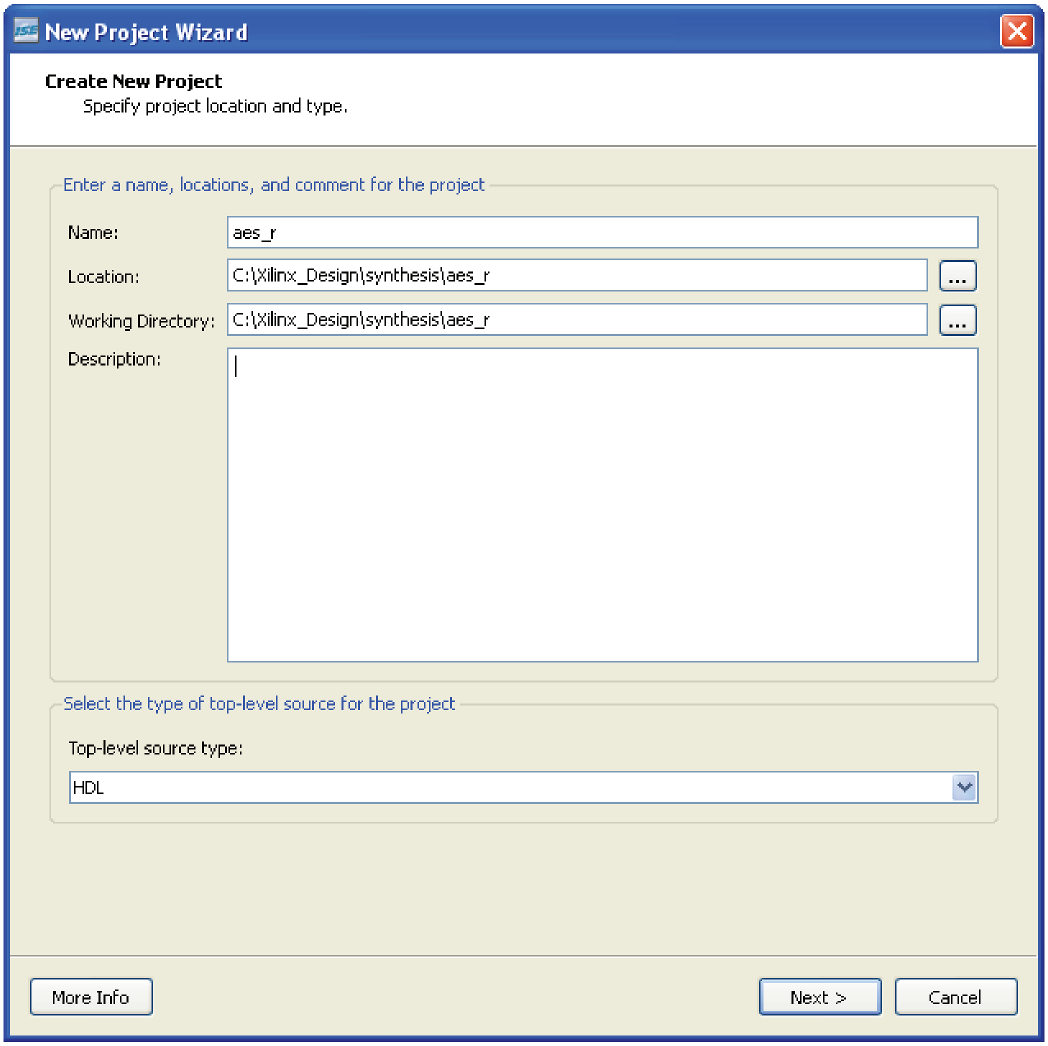
1. Start the ISE design suite 12.1:

   **Start** → **All Programs** → **Xilinx ISE Design Suite 12.1** → **ISE Design Tools** → **Project Navigator**

2. Create a new ISE 12.1 project:

   **File** → **New Project**

3. Set the Project Location to **`\Xilinx_Design\synthesis`**.
4. Set the Project Name to **aes_r**.
5. Click **Next** (see Figure 2-15).



New Project Wizard

**Create New Project**

Specify project location and type.

Enter a name, locations, and comment for the project

Name: aes_r

Location: C:\Xilinx_Design\synthesis\aes_r

Working Directory: C:\Xilinx_Design\synthesis\aes_r

Description:

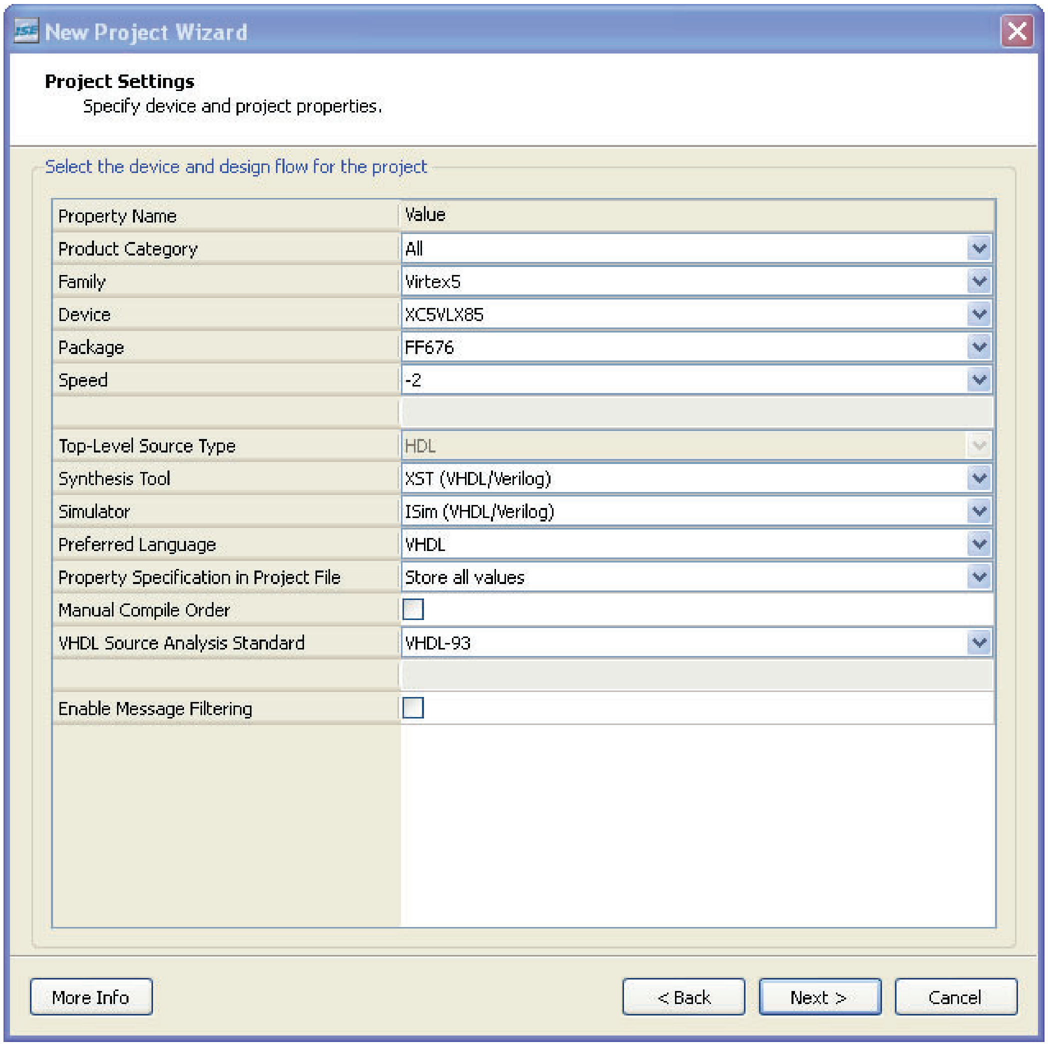
Select the type of top-level source for the project

Top-level source type: HDL

X1105_c2_15_061910

*Figure 2-15:* **New Project Wizard (Create New Project)**

6. Choose an XC5VLX85-FF676-2 as the target device (see Figure 2-16).

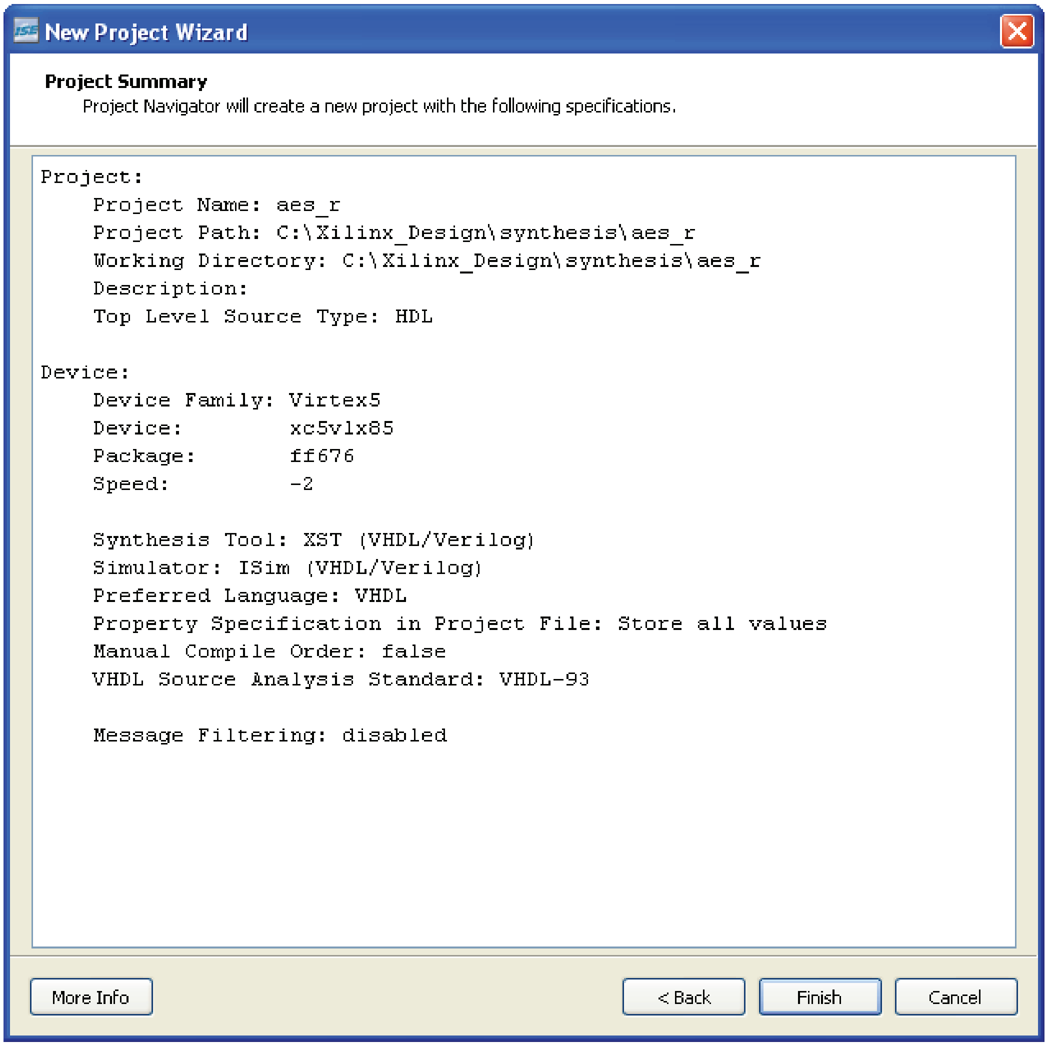

New Project Wizard
Project Settings
Specify device and project properties.
Select the device and design flow for the project
More Info
< Back
Next >
Cancel
X1105_c2_16_061910

| 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 | ☐ |
| VHDL Source Analysis Standard | VHDL-93 |
| | |
| Enable Message Filtering | ☐ |

*Figure 2-16:* **New Project Wizard (Project Settings)**

7. Click **Next**.

8. Click **Finish** in the Project Summary window (see Figure 2-17).

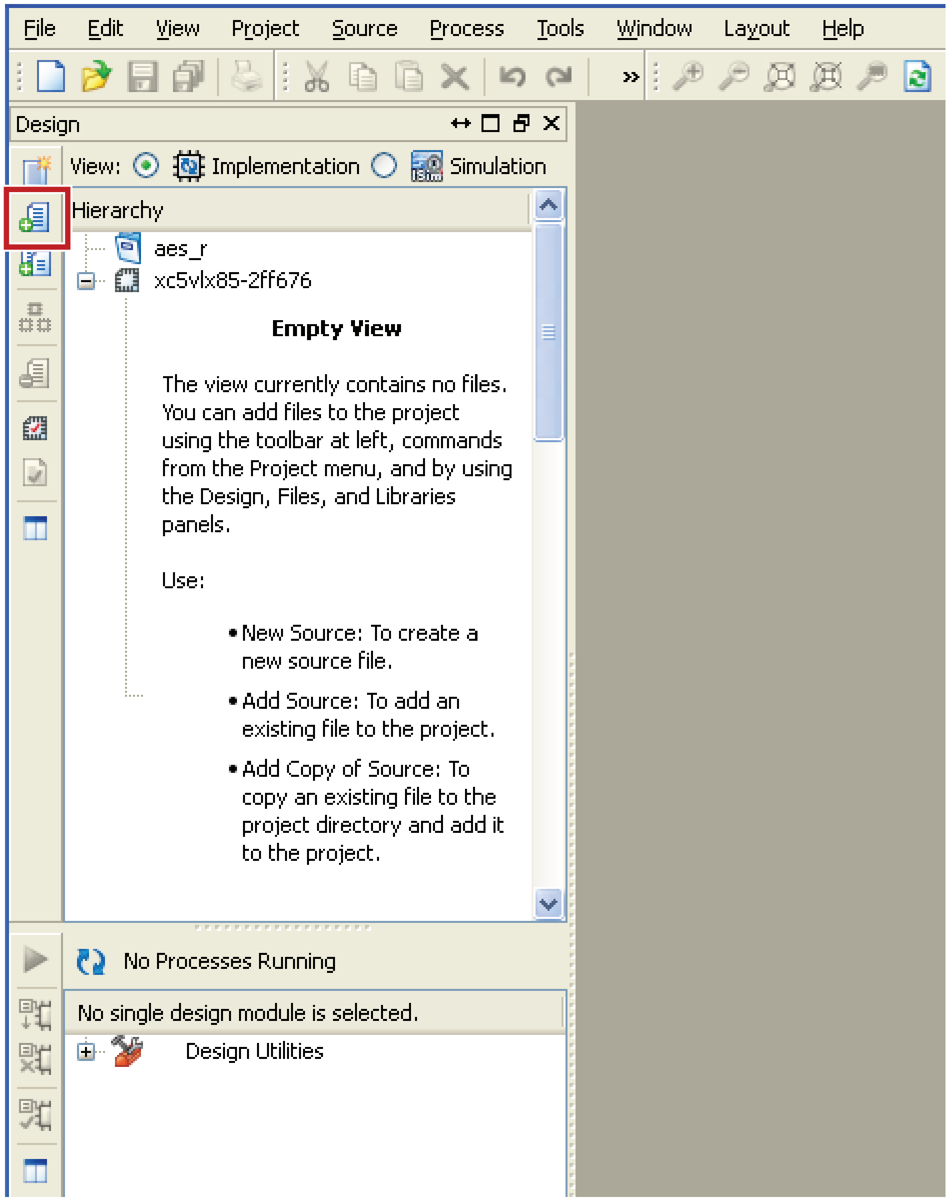

New Project Wizard
Project Summary
Project Navigator will create a new project with the following specifications.
More Info
< Back
Finish
Cancel

```
Project:
    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:        xc5vlx85
    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
    VHDL Source Analysis Standard: VHDL-93

    Message Filtering: disabled
```

X1105_c2_17_061910

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

9. Select the **Add Source** button on the left side of the Design window in the ISE software Project Navigator, as shown in Figure 2-18.


File Edit View Project Source Process Tools Window Layout Help
Design
View: ● Implementation ○ Simulation
Hierarchy
aes_r
xc5vlx85-2ff676
No Processes Running
Design Utilities

**Empty View**

The view currently contains no files. You can add files to the project using the toolbar at left, commands from the Project menu, and by using the Design, Files, and Libraries panels.

Use:

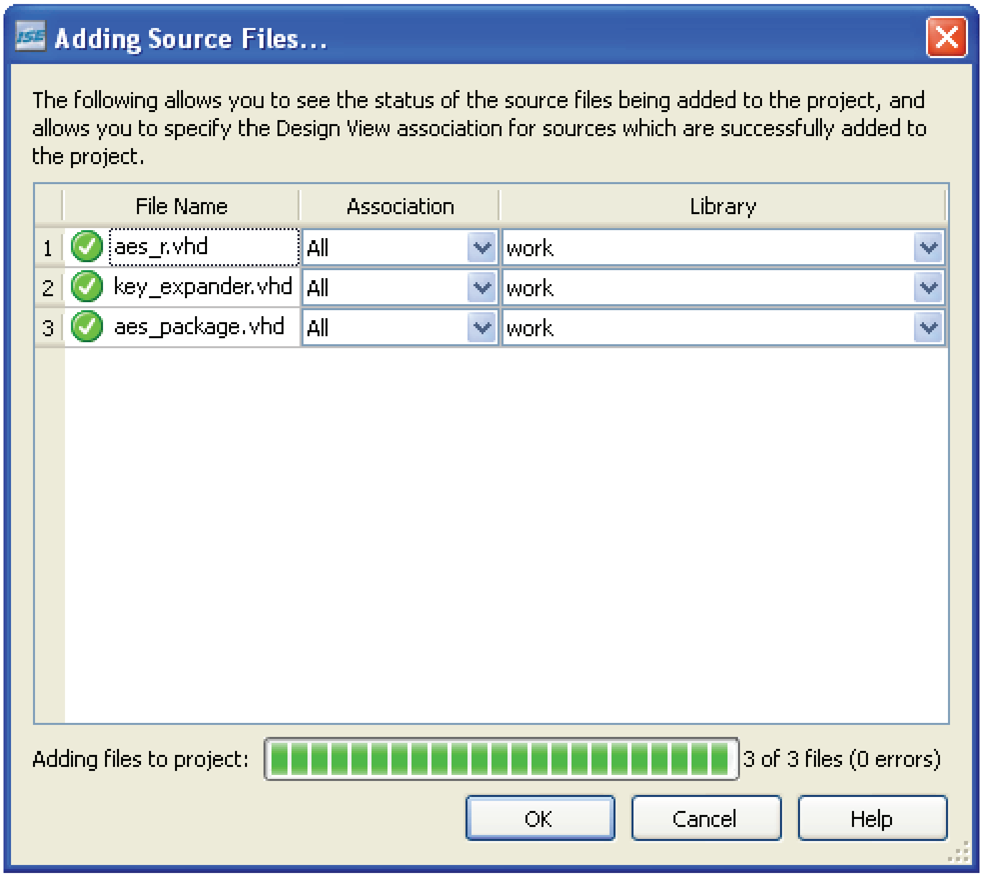
- New Source: To create a new source file.
- Add Source: To add an existing file to the project.
- Add Copy of Source: To copy an existing file to the project directory and add it to the project.

No single design module is selected.

X1105_c2_18_061910

*Figure 2-18:* **Project Navigator (Add Existing Sources)**

10. Navigate to the `source\design` directory, and add the `aes_r.vhd`, `aes_package.vhd`, and `key_expander.vhd` files to the project (see Figure 2-19). Click **OK**.



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 | ✓ aes_r.vhd | All | work |
| 2 | ✓ key_expander.vhd | All | work |
| 3 | ✓ aes_package.vhd | All | work |

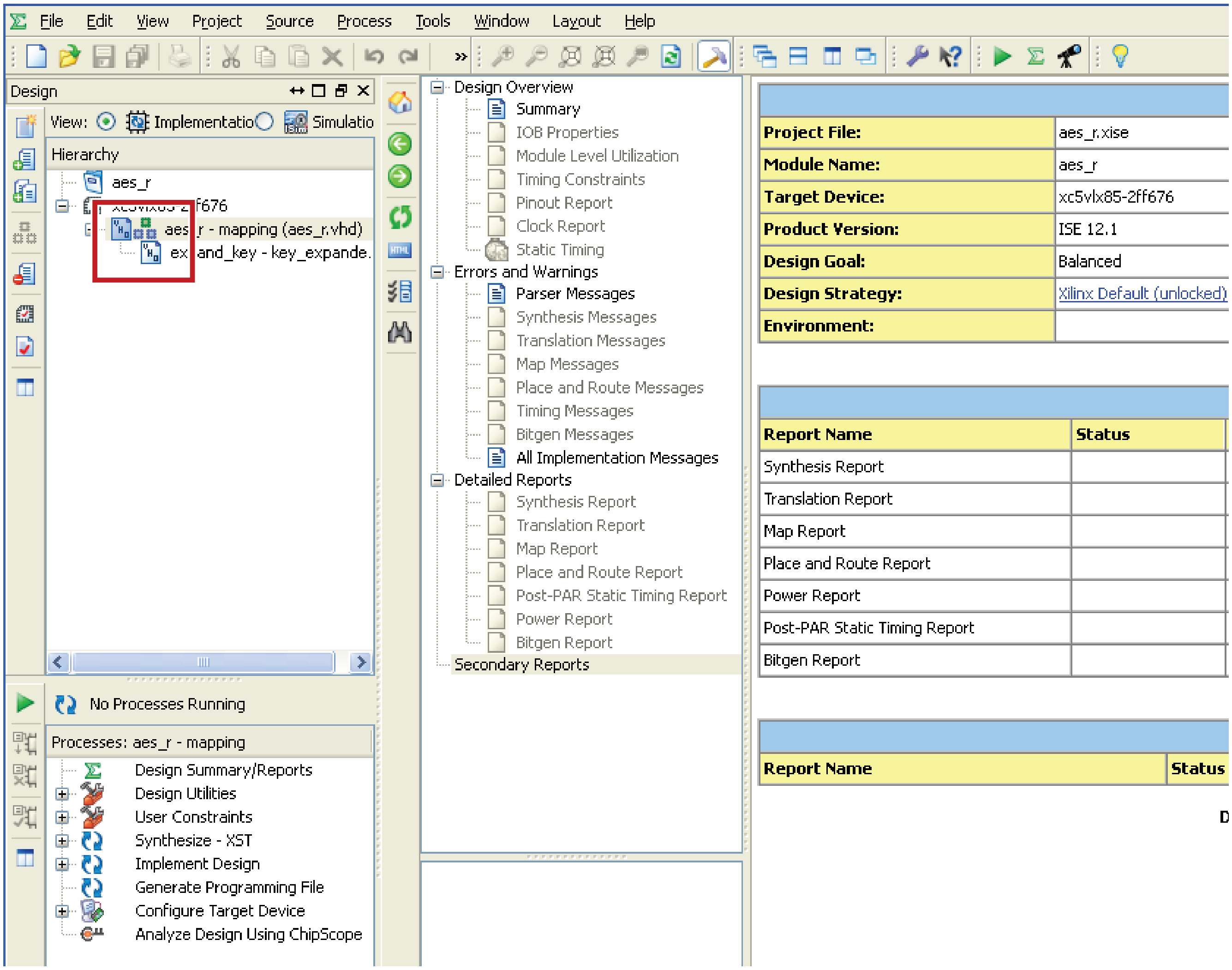
Adding files to project: 3 of 3 files (0 errors)

X1105_c2_19_061910

*Figure 2-19:* **Adding Source Files**

The ISE Project Navigator Window should look like the window shown in Figure 2-20.

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



| | |
|---|---|
| **Project File:** | aes_r.xise |
| **Module Name:** | aes_r |
| **Target Device:** | xc5vlx85-2ff676 |
| **Product Version:** | ISE 12.1 |
| **Design Goal:** | Balanced |
| **Design Strategy:** | Xilinx Default (unlocked) |
| **Environment:** | |

| **Report Name** | **Status** |
|---|---|
| Synthesis Report | |
| Translation Report | |
| Map Report | |
| Place and Route Report | |
| Power Report | |
| Post-PAR Static Timing Report | |
| Bitgen Report | |

| **Report Name** | **Status** |
|---|---|

X1105_c2_20_061910

*Figure 2-20:* **Project Navigator Window**

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

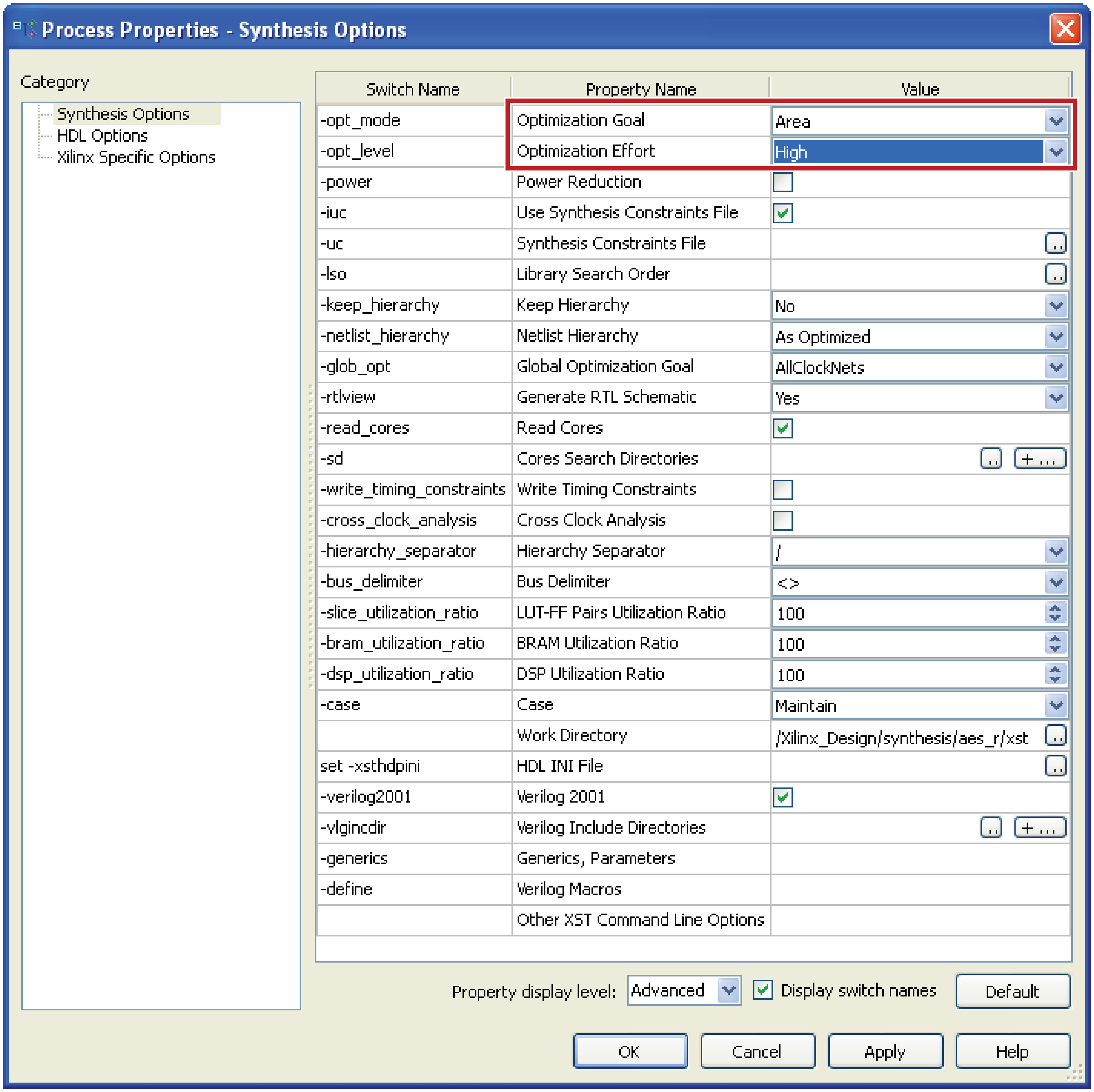
```
attribute buffer_type: string;
attribute buffer_type of clk         : signal is "none";
attribute buffer_type of reset       : signal is "none";
attribute buffer_type of push_button : signal is "none";
attribute buffer_type of start       : signal is "none";
attribute buffer_type of mode        : signal is "none";
attribute buffer_type of load        : signal is "none";
attribute buffer_type of key         : signal is "none";
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 XST 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 SCC_LAB_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 and select **Process Properties…**.

13. Set Optimization Goal to **Area**, set Optimization Effort to **High**, and click **OK** (see Figure 2-21).



Process Properties - Synthesis Options

Category

- Synthesis Options
- HDL Options
- Xilinx Specific Options

| Switch Name | Property Name | Value |
|---|---|---|
| -opt_mode | Optimization Goal | Area |
| -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 | No |
| -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 | / |
| -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 |
| -case | Case | Maintain |
| | Work Directory | /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 display level: Advanced ☑ Display switch names

*Figure 2-21:* **Process Properties**

14. To run XST synthesis, either right-click on the **Synthesize-XST** icon in the Processes window and select **Run**, or 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:

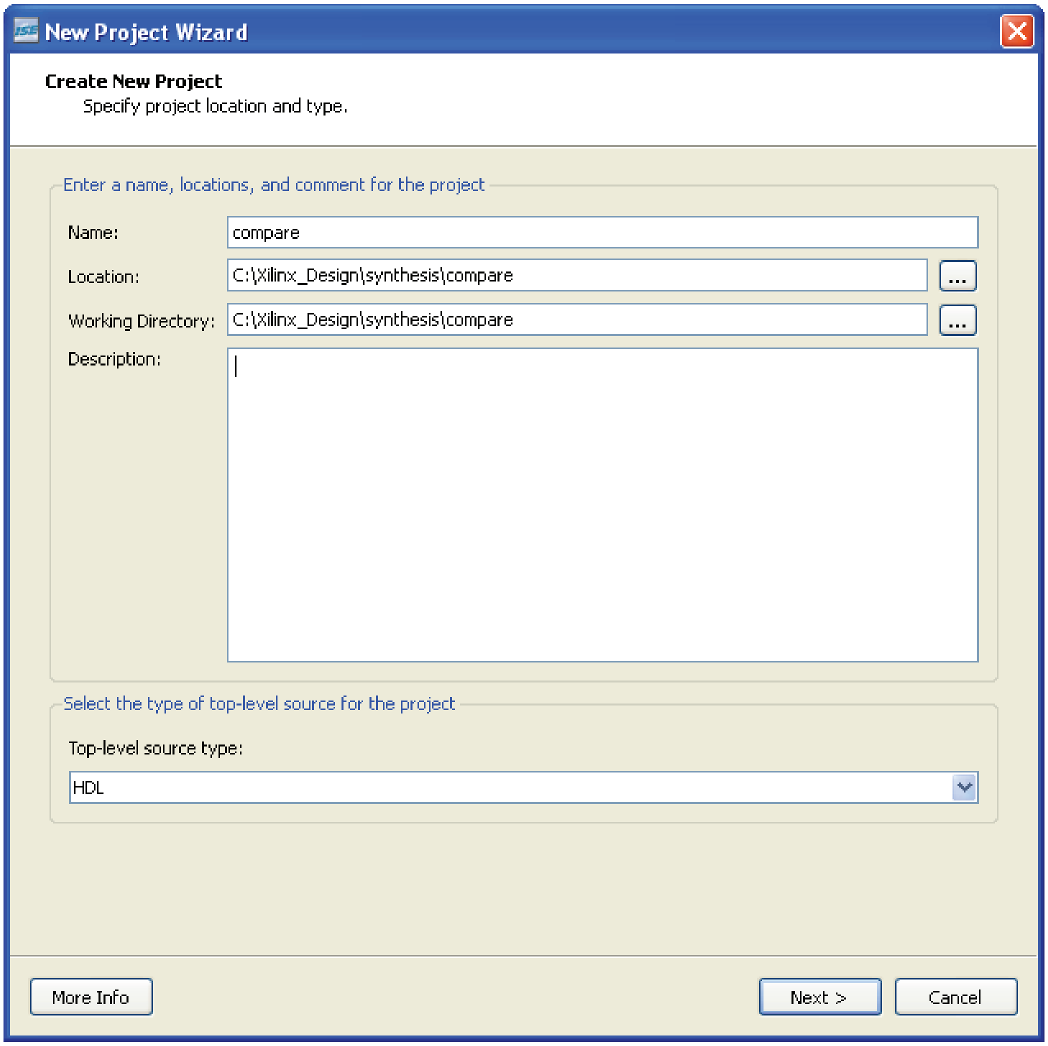
1. Start the ISE design suite 12.1:

   **Start** → **All Programs** → **Xilinx ISE Design Suite 12.1** → **ISE Design Tools** → **Project Navigator**

2. Create a new ISE design suite 12.1 project:

   **File** → **New Project**

3. Set the project location to **`\Xilinx_Design\synthesis`**.
4. Set the project name to **compare**.
5. Click **Next** (see Figure 2-22).



New Project Wizard

Create New Project
Specify project location and type.

Enter a name, locations, and comment for the project

Name: compare

Location: C:\Xilinx_Design\synthesis\compare

Working Directory: C:\Xilinx_Design\synthesis\compare

Description:

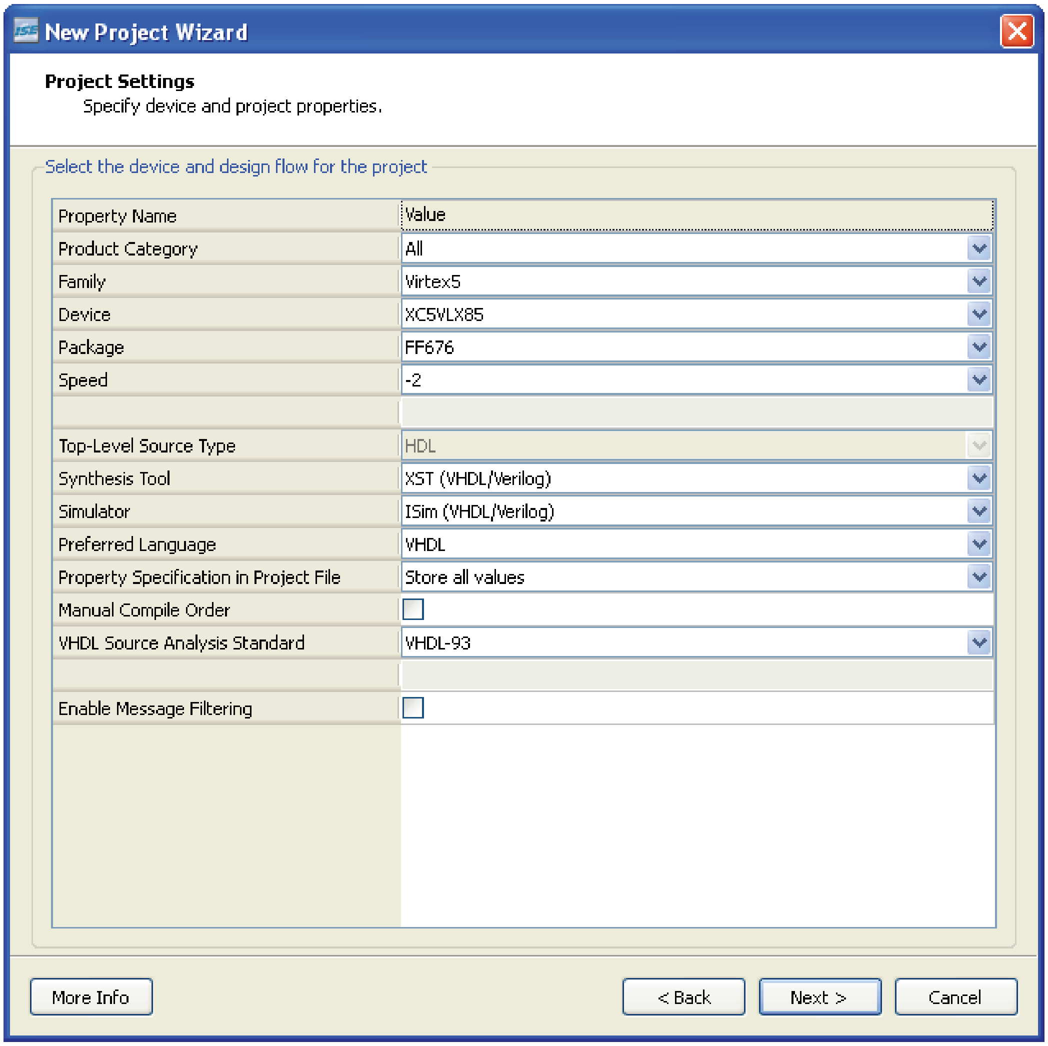
Select the type of top-level source for the project

Top-level source type: HDL

X1105_c2_22_061910

*Figure 2-22:* **New Project Wizard (Create New Project)**

6. Choose an XC5VLX85-FF676-2 as the target device (see Figure 2-23).



New Project Wizard
Project Settings
Specify device and project properties.
Select the device and design flow for the project
More Info
< Back
Next >
Cancel

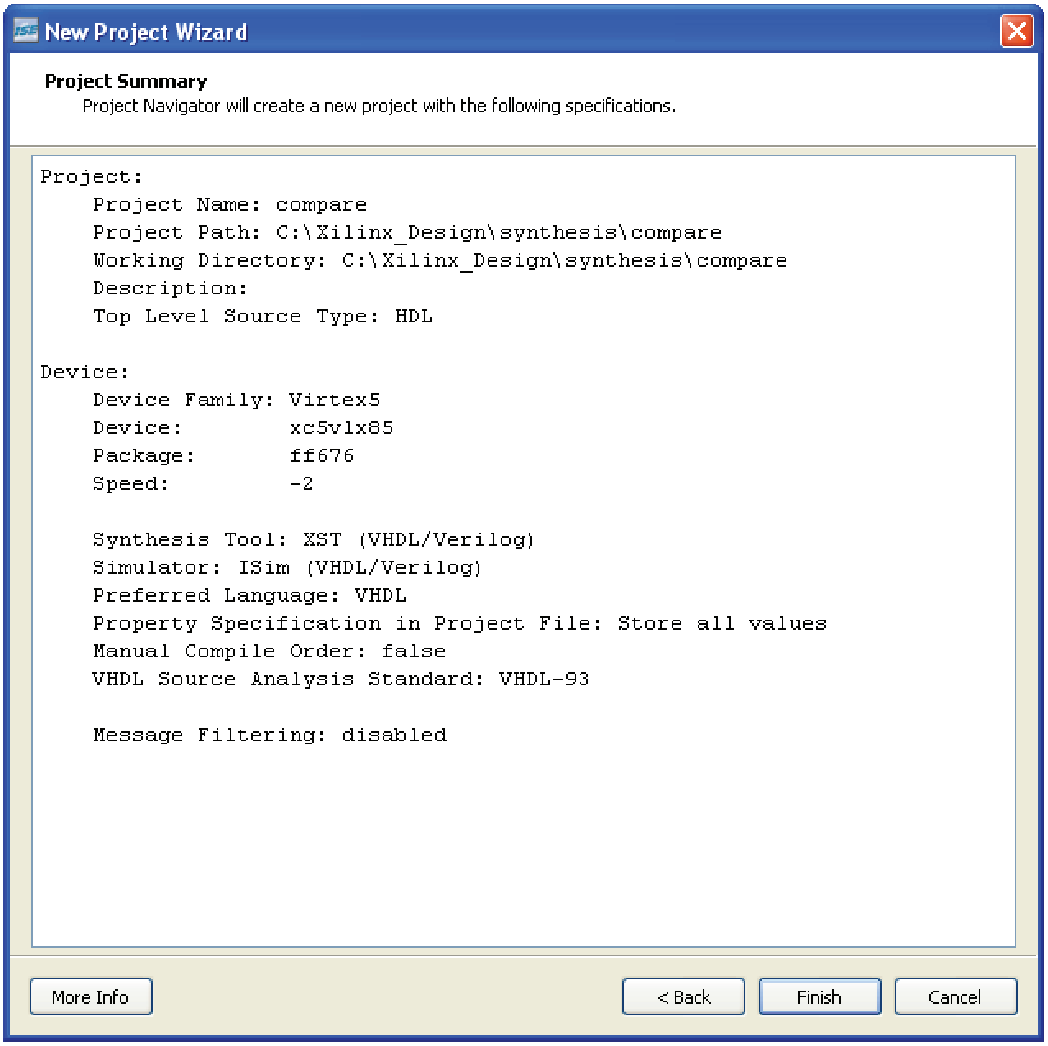
| Property Name | Value |
|---|---|
| Product Category | All |
| Family | Virtex5 |
| Device | XC5VLX85 |
| Package | FF676 |
| Speed | -2 |
| | |
| Top-Level Source Type | HDL |
| Synthesis Tool | XST (VHDL/Verilog) |
| Simulator | ISim (VHDL/Verilog) |
| Preferred Language | VHDL |
| Property Specification in Project File | Store all values |
| Manual Compile Order | □ |
| VHDL Source Analysis Standard | VHDL-93 |
| | |
| Enable Message Filtering | □ |

X1105_c2_23_061910

*Figure 2-23:* **New Project Wizard (Project Settings)**

7. Click **Next**.

8. Click **Finish** in the Project Summary window (see Figure 2-24).



**Project Summary**

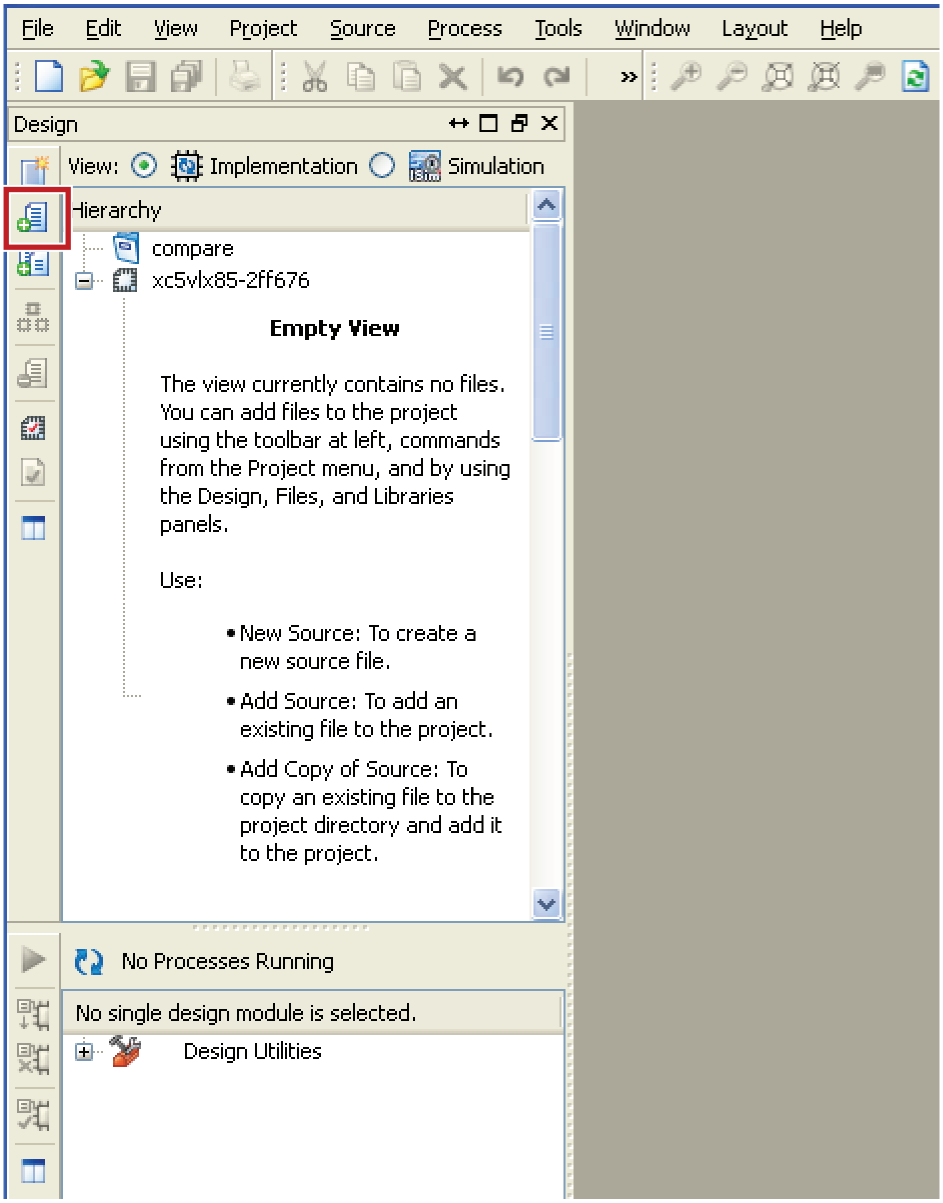
Project Navigator will create a new project with the following specifications.

```
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 Family: Virtex5
    Device:        xc5vlx85
    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
    VHDL Source Analysis Standard: VHDL-93

    Message Filtering: disabled
```

X1105_c2_24_061910

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

9. Select the **Add Source** button on the left side of the Design window in the Project Navigator of the ISE tool (see Figure 2-25).



**Empty View**

The view currently contains no files. You can add files to the project using the toolbar at left, commands from the Project menu, and by using the Design, Files, and Libraries panels.

Use:

- New Source: To create a new source file.
- Add Source: To add an existing file to the project.
- Add Copy of Source: To copy an existing file to the project directory and add it to the project.

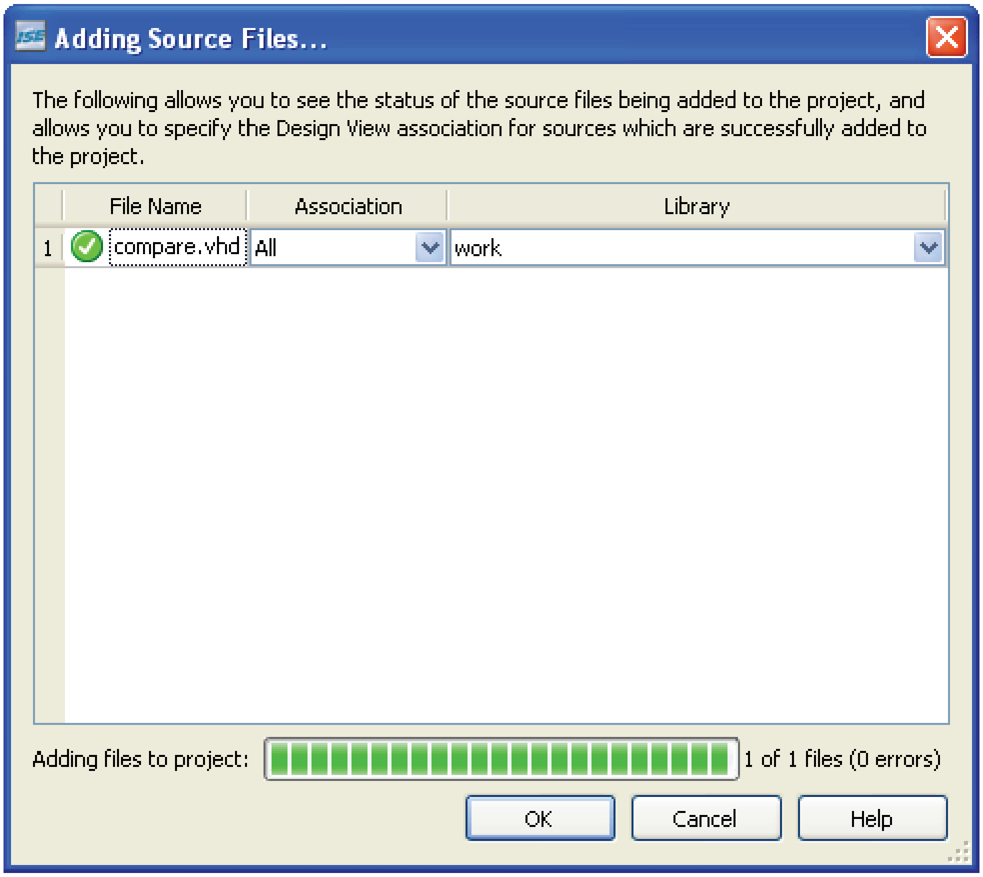
No Processes Running

No single design module is selected.

X1105_c2_25_061910

*Figure 2-25:* **Project Navigator (Add Existing Sources)**

10. Navigate to the `source\design` directory, and add the `compare.vhd` file to the project (see Figure 2-26). Click **OK**.



Adding Source Files...

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 | work |

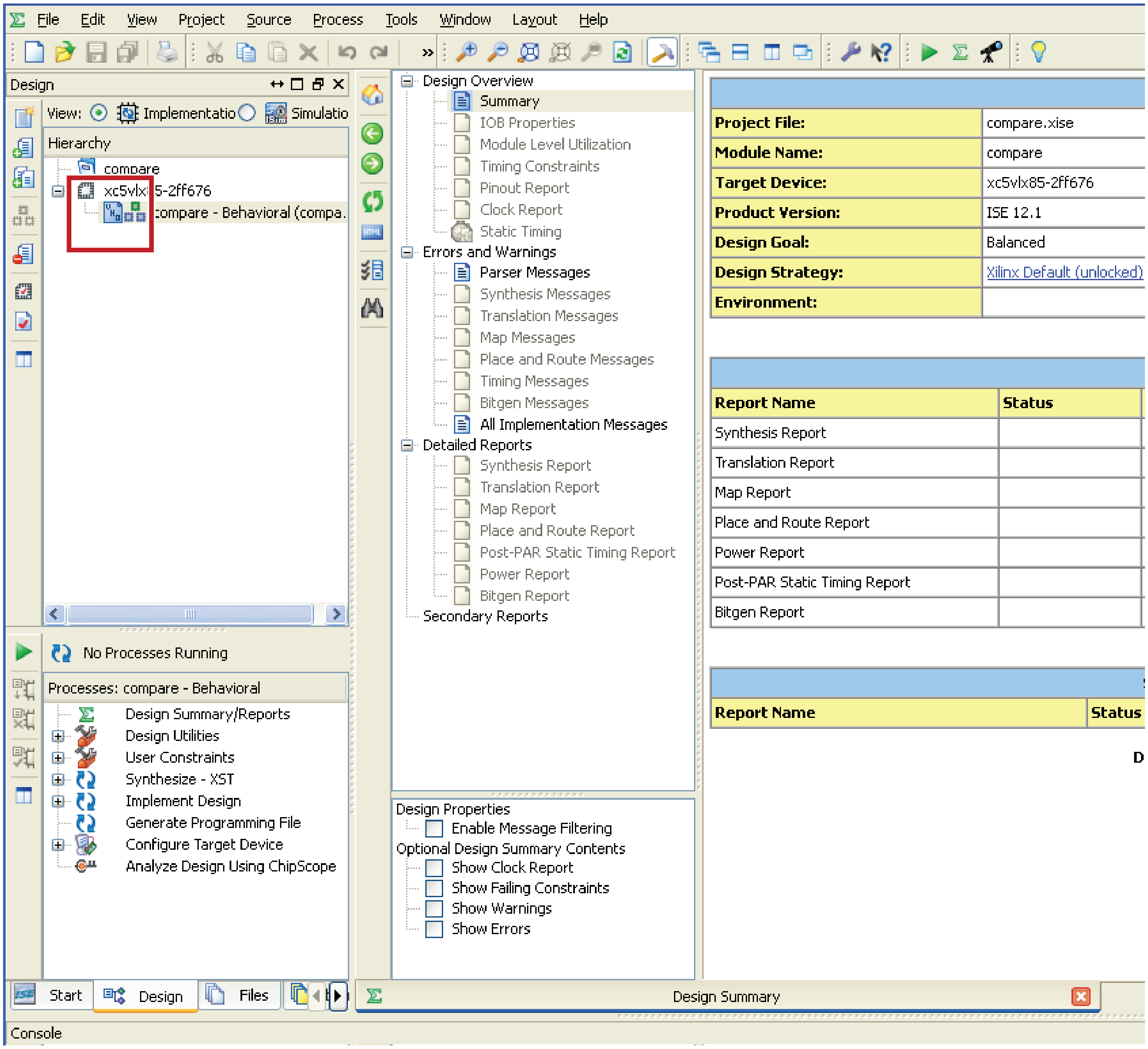
Adding files to project: 1 of 1 files (0 errors)

X1105_c2_26_061910

*Figure 2-26:* **Adding Source Files**

The ISE Project Navigator Window should look like the window shown in Figure 2-27.

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



| | |
|---|---|
| **Project File:** | compare.xise |
| **Module Name:** | compare |
| **Target Device:** | xc5vlx85-2ff676 |
| **Product Version:** | ISE 12.1 |
| **Design Goal:** | Balanced |
| **Design Strategy:** | Xilinx Default (unlocked) |
| **Environment:** | |

| **Report Name** | **Status** |
|---|---|
| Synthesis Report | |
| Translation Report | |
| Map Report | |
| Place and Route Report | |
| Power Report | |
| Post-PAR Static Timing Report | |
| Bitgen Report | |

| **Report Name** | **Status** |
|---|---|

X1105_c2_27_061910

*Figure 2-27:* **Project Navigator Window**

11. Open the compare.vhd file and locate the buffer 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 reset_out    : signal is "none";
attribute buffer_type of done1        : signal is "none";
attribute buffer_type of done2        : signal is "none";
attribute buffer_type 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";
```

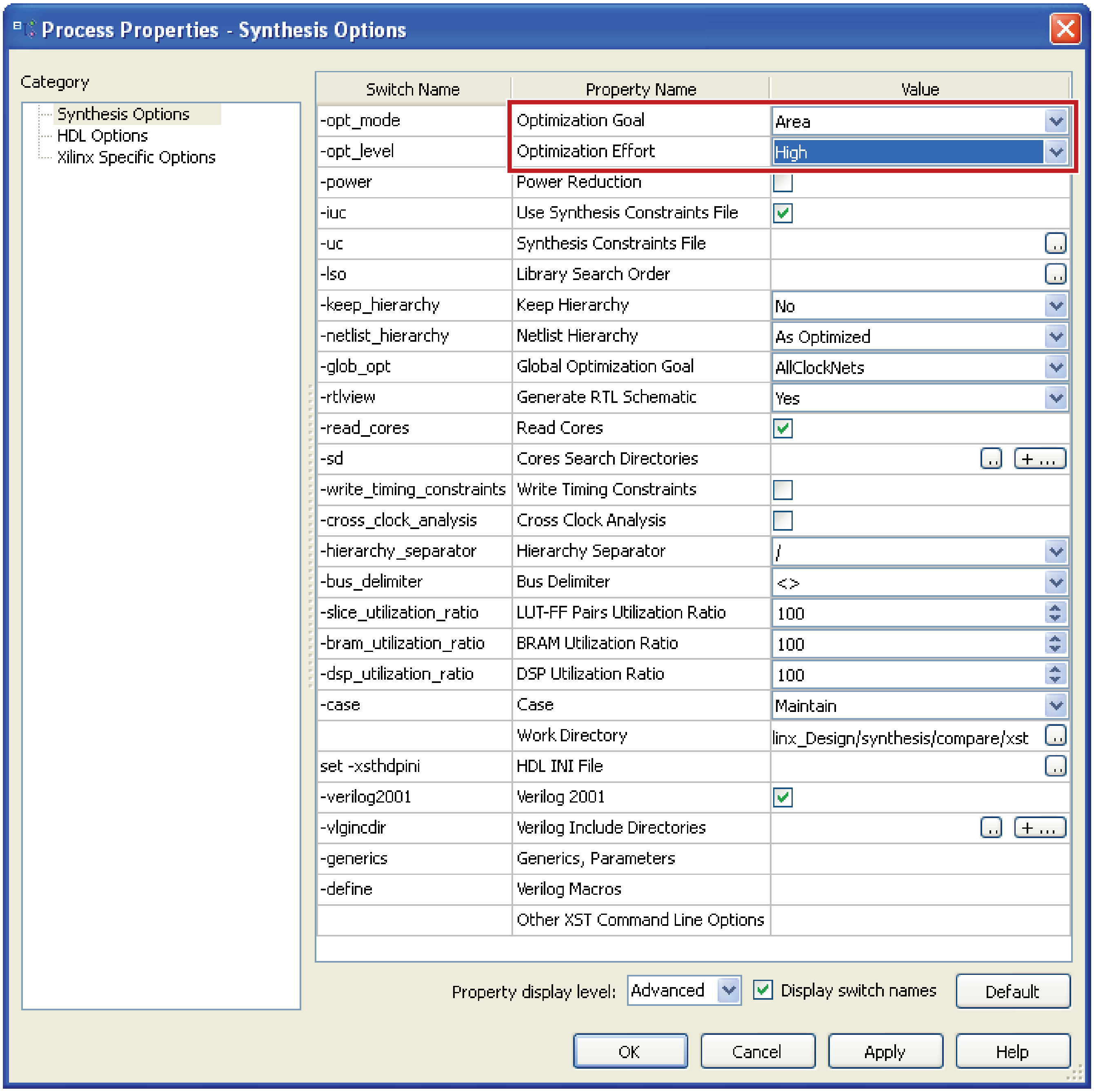
The buffer_type attribute directs XST to disable I/O insertion on the specified ports. By default, XST inserts an I/O on all ports. The buffer_type attribute is necessary to prevent this insertion at this level in the hierarchy either because the I/Os are placed at the top level (such as CLK) or because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

12. In the `compare.vhd` source code file, the LUT instantiation between certain inputs and outputs 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_i, O => reset_out );

  lut_start_aes1_out : LUT1
  GENERIC MAP (INIT => X"2")
  PORT MAP (I0 => start_i, O => start_aes1 );

  lut_start_aes2_out : LUT1
  GENERIC MAP (INIT => X"2")
  PORT MAP (I0 => start_i, O => start_aes2 );

  lut_load_aes1_out : LUT1
  GENERIC MAP (INIT => X"2")
  PORT MAP (I0 => load_i, O => load_aes1 );

  lut_load_aes2_out : LUT1
  GENERIC MAP (INIT => X"2")
  PORT MAP (I0 => load_i, O => load_aes2 );
```

13. Right-click on the **Synthesize-XST** icon in the Processes window and select **Process Properties** (see Figure 2-28).



| Switch Name | Property Name | Value |
|---|---|---|
| -opt_mode | Optimization Goal | Area |
| -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 | No |
| -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 | / |
| -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 |
| -case | Case | Maintain |
| | Work Directory | linx_Design/synthesis/compare/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 display level: Advanced ☑ Display switch names

X1105_c2_28_061910

*Figure 2-28:* **Process Properties**

14. Set the Optimization Goal to **Area** and the Optimization Effort to **High**.
15. Click **OK**.
16. To run XST synthesis, either right-click on the **Synthesize-XST** icon in the Processes window and select **Run**, or double-click **Synthesize-XST**.

17. After synthesis is complete, close the `compare` project.

## Design Flow Progress

The Module Synthesis block of the SCC system design flow diagram is complete, as shown in Figure 2-29.

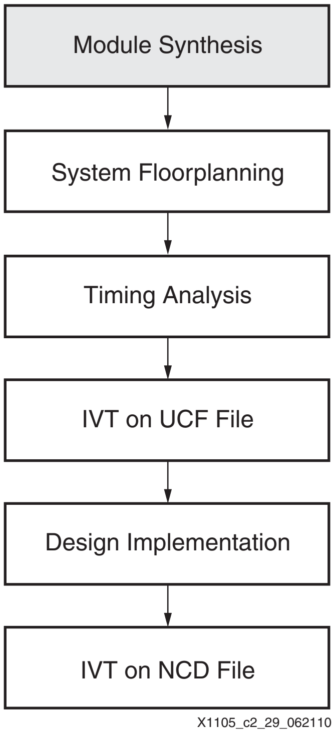


*Figure 2-29:* **SCC System Design Flow with Module Synthesis Block Complete**

*Chapter 3*

# *Floorplanning the System*

## Project Entry in PlanAhead Tool

### Launch the PlanAhead Tool

To launch the 12.1 version of the PlanAhead™ tool, select:

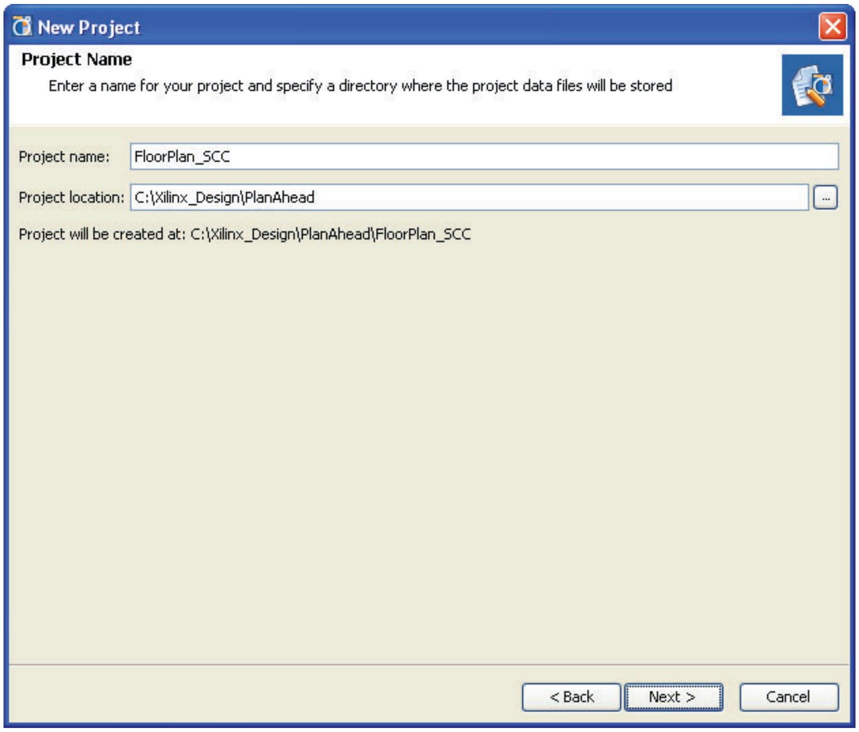
**Start** → **All Programs** → **Xilinx ISE Design Suite 12.1** → **PlanAhead** → **PlanAhead**

### 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 tool project (see Figure 3-1):

   **File** → **New Project**



Project Name

Enter a name for your project and specify a directory where the project data files will be stored

Project name: FloorPlan_SCC

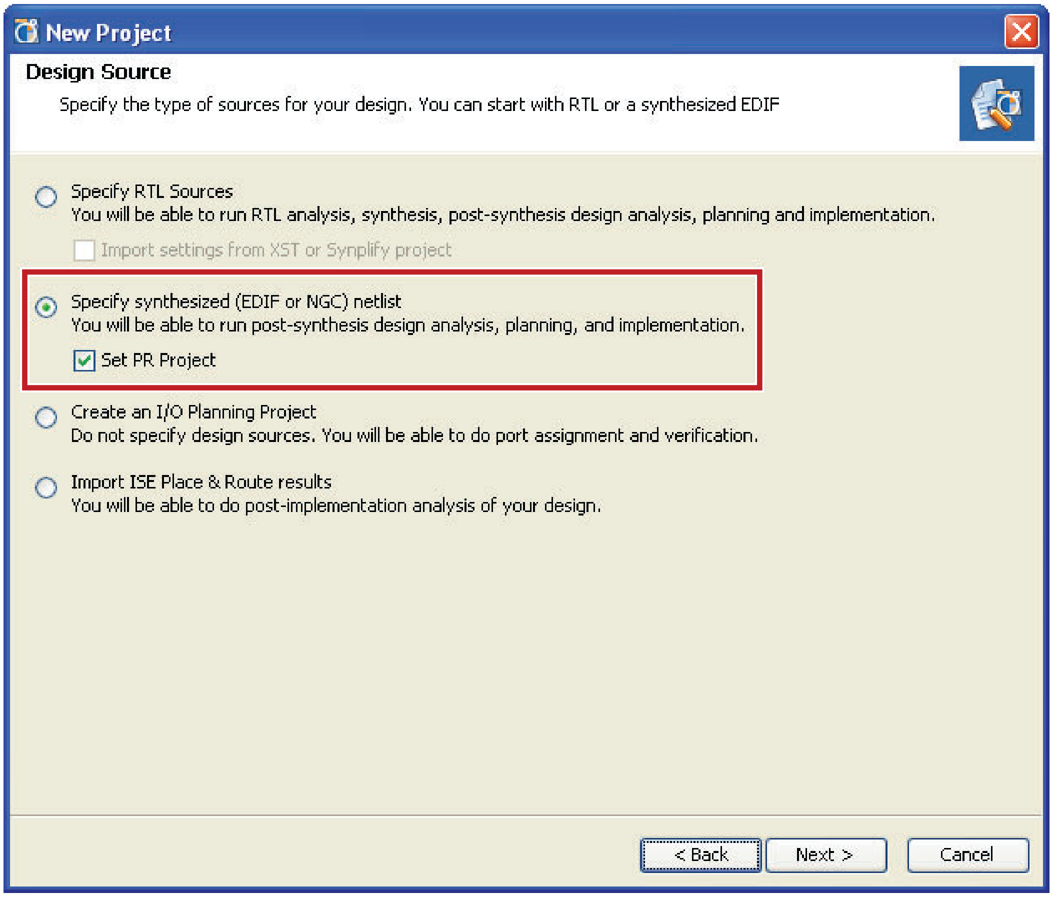
Project location: C:\Xilinx_Design\PlanAhead

Project will be created at: C:\Xilinx_Design\PlanAhead\FloorPlan_SCC

X1105_c3_01_062310

*Figure 3-1:* **New Project**

2. Click **Next** and enter:
    - Project name: For this lab, the **FloorPlan_SCC** project name is used.
    - Project location: `\Xilinx_Design\PlanAhead`
3. Click **Next**.
4. Select **Specify synthesized (EDIF or NGC) netlist** (see Figure 3-2) because the modules have already been synthesized. To turn the project into a partially reconfigurable (PR) project, select **Set PR Project**. Click **Next**.



New Project

Design Source

Specify the type of sources for your design. You can start with RTL or a synthesized EDIF

○ Specify RTL Sources
You will be able to run RTL analysis, synthesis, post-synthesis design analysis, planning and implementation.

□ Import settings from XST or Synplify project

● Specify synthesized (EDIF or NGC) netlist
You will be able to run post-synthesis design analysis, planning, and implementation.

☑ Set PR Project

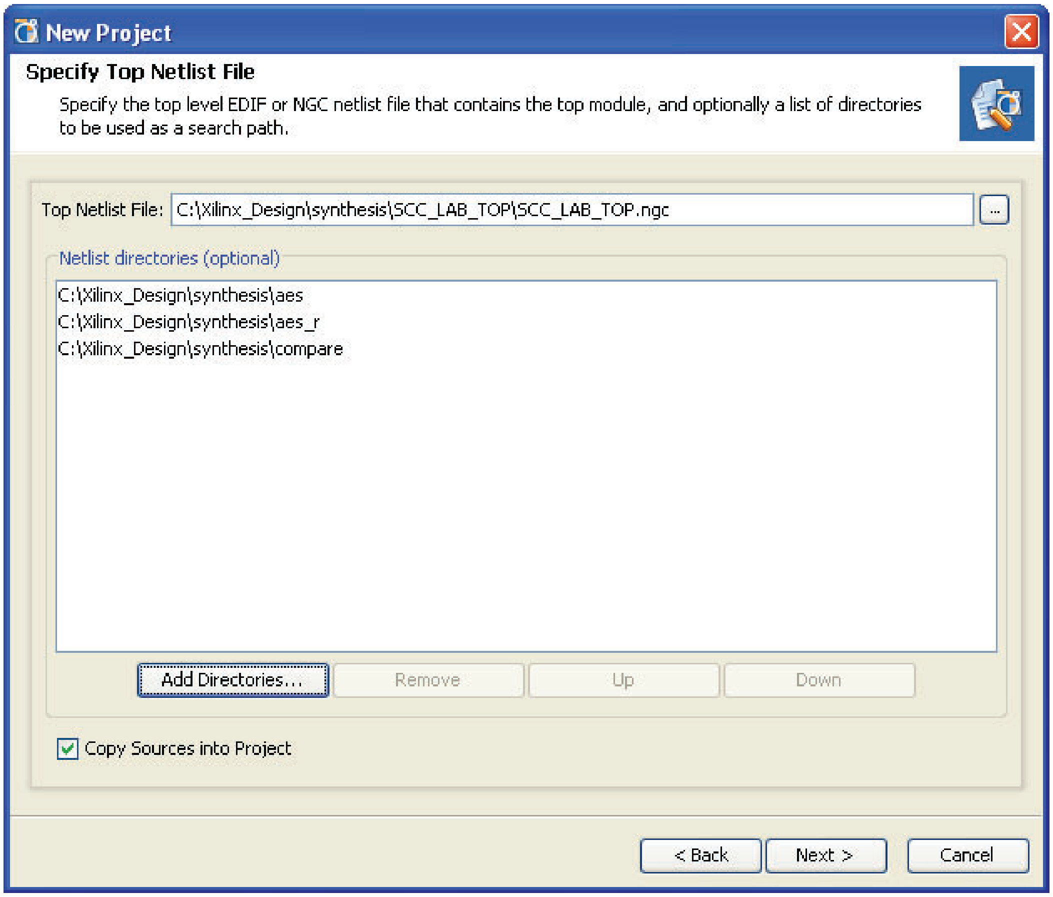
○ Create an I/O Planning Project
Do not specify design sources. You will be able to do port assignment and verification.

○ Import ISE Place & Route results
You will be able to do post-implementation analysis of your design.

X1105_c3_02_061910

*Figure 3-2:* **New Project (Design Source)**

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



New Project

**Specify Top Netlist File**

Specify the top level EDIF or NGC netlist file that contains the top module, and optionally a list of directories to be used as a search path.

Top Netlist File: C:\Xilinx_Design\synthesis\SCC_LAB_TOP\SCC_LAB_TOP.ngc

Netlist directories (optional)

C:\Xilinx_Design\synthesis\aes
C:\Xilinx_Design\synthesis\aes_r
C:\Xilinx_Design\synthesis\compare

☑ Copy Sources into Project

X1105_c3_03_061910

*Figure 3-3:* **New Project (Import Netlist)**

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

   **Netlist File:**

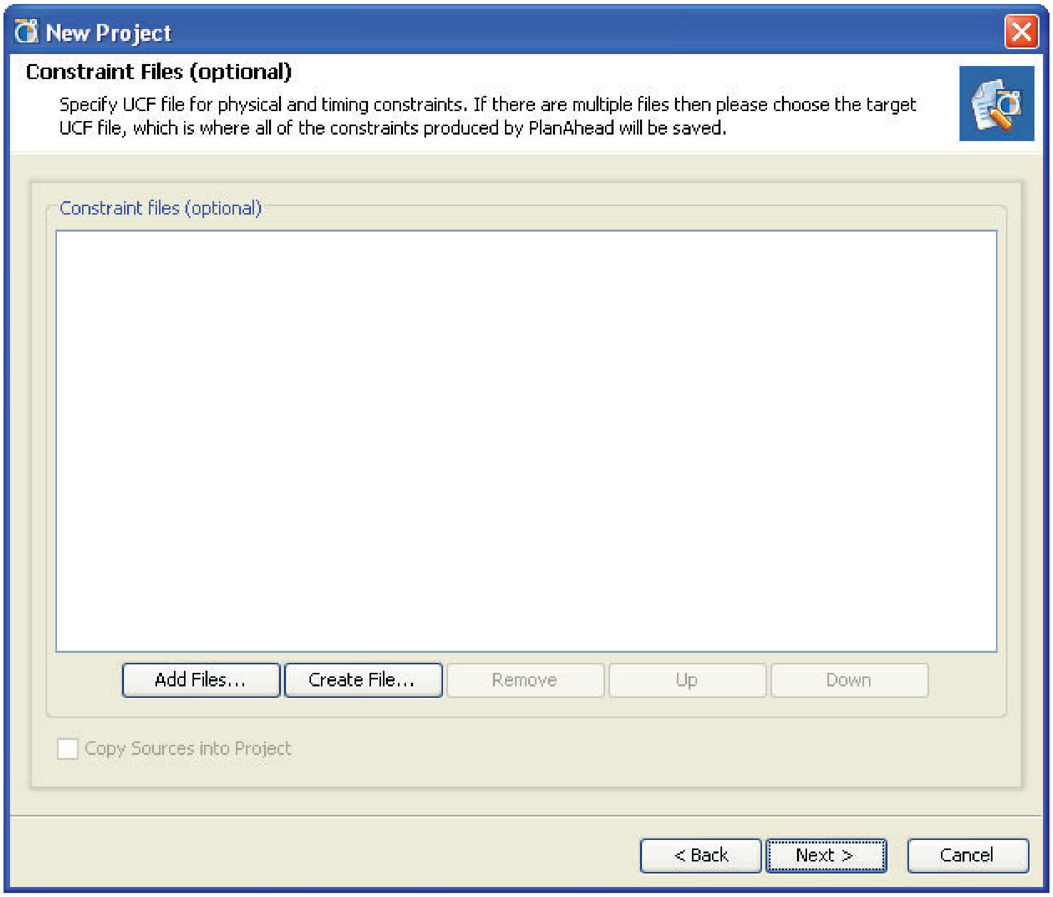
   ```
   ..\Xilinx_Design\synthesis\SCC_LAB_TOP\SCC_LAB_TOP.ngc
   ```

   **Netlist Directories:**

   ```
   ..\Xilinx_Design\synthesis\aes
   ..\Xilinx_Design\synthesis\aes_r
   ..\Xilinx_Design\synthesis\compare
   ```

7. Click **Next**.

8. This lab creates the final user constraints file (UCF) from the beginning. Thus no UCF files need to be imported, as shown in Figure 3-4. Click **Next**.



New Project

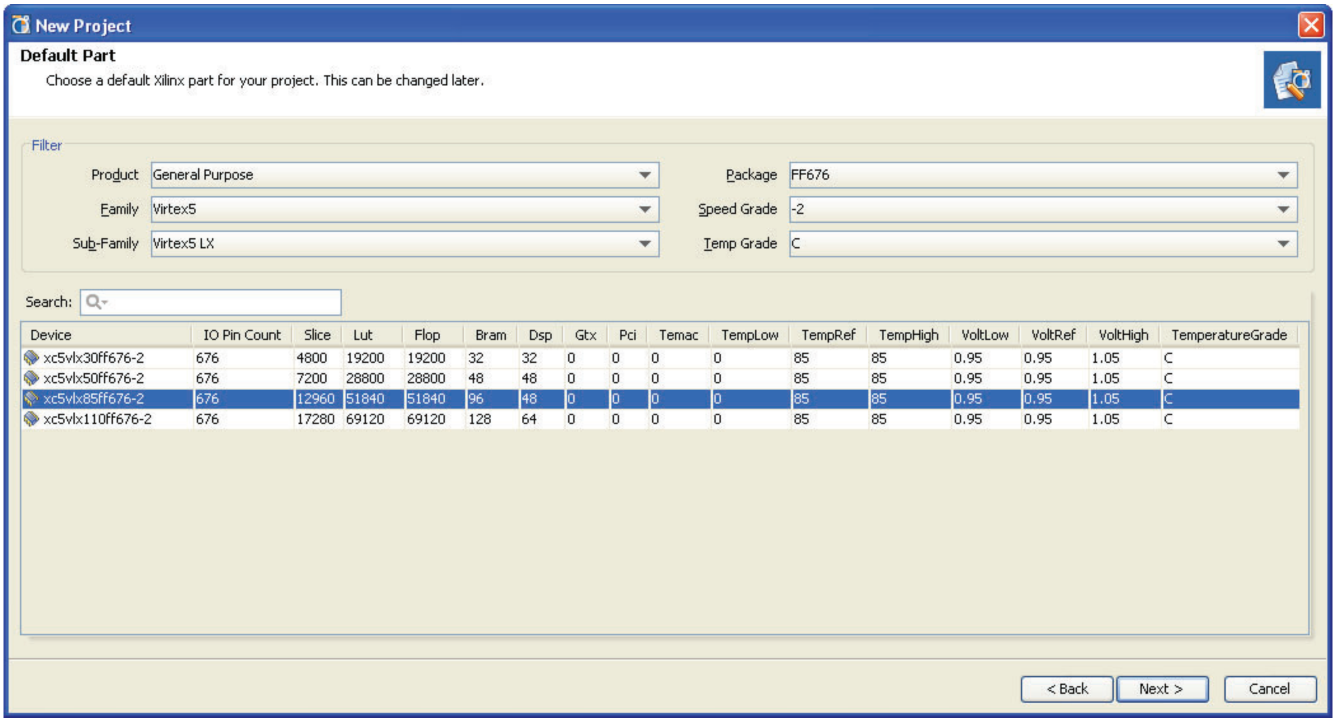
Constraint Files (optional)

Specify UCF file for physical and timing constraints. If there are multiple files then please choose the target UCF file, which is where all of the constraints produced by PlanAhead will be saved.

X1105_c3_04_061910

*Figure 3-4:* **New Project (Import Constraints)**

9. Choose the product, family, and sub-family for the default part (see Figure 3-5). For this lab, **Virtex5 LX** is used.



**Default Part**

Choose a default Xilinx part for your project. This can be changed later.

Filter

Product: General Purpose

Family: Virtex5

Sub-Family: Virtex5 LX

Package: FF676

Speed Grade: -2

Temp Grade: C

Search:

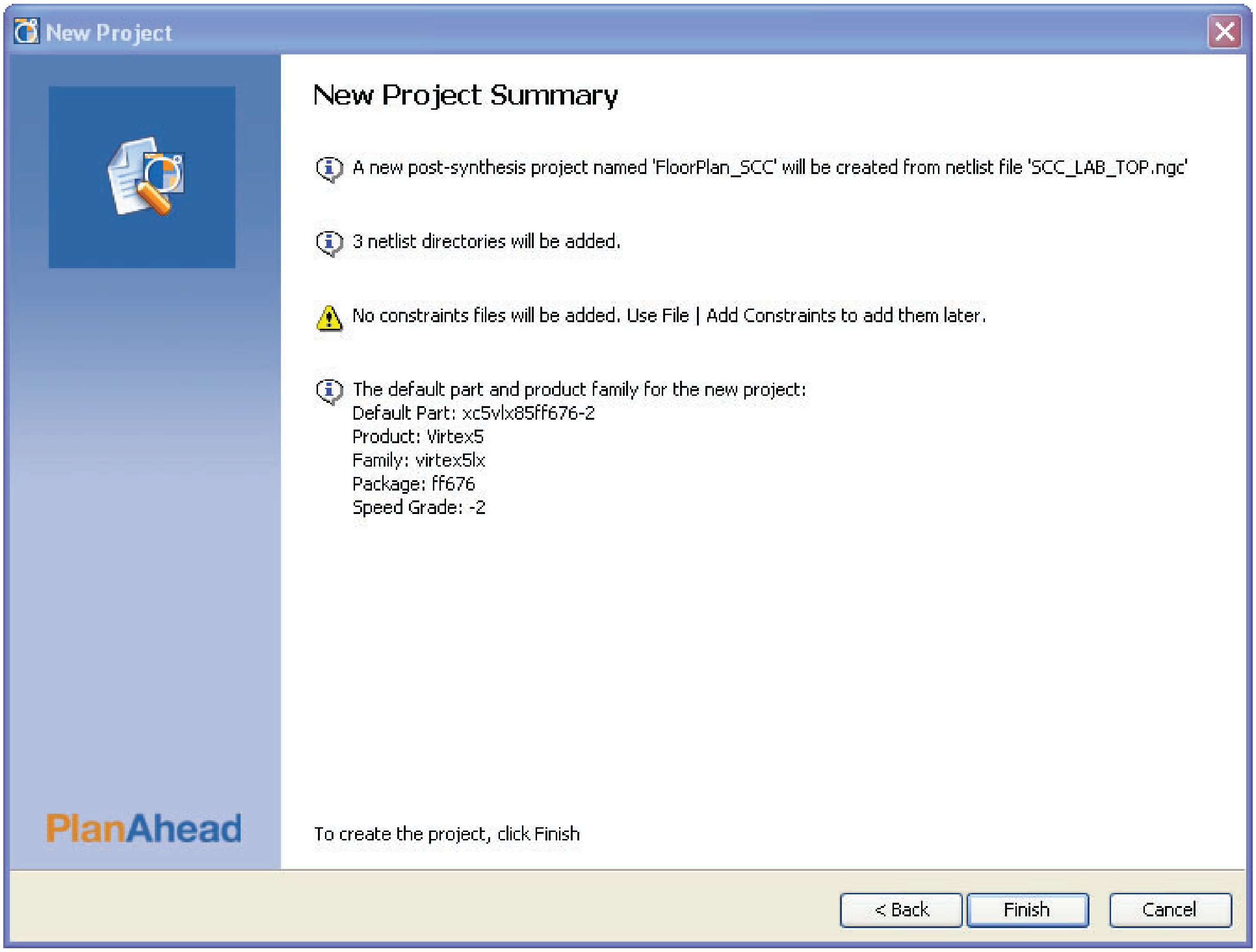
| Device | IO Pin Count | Slice | Lut | Flop | Bram | Dsp | Gtx | Pci | Temac | TempLow | TempRef | TempHigh | VoltLow | VoltRef | VoltHigh | TemperatureGrade |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| xc5vlx30ff676-2 | 676 | 4800 | 19200 | 19200 | 32 | 32 | 0 | 0 | 0 | 0 | 85 | 85 | 0.95 | 0.95 | 1.05 | C |
| xc5vlx50ff676-2 | 676 | 7200 | 28800 | 28800 | 48 | 48 | 0 | 0 | 0 | 0 | 85 | 85 | 0.95 | 0.95 | 1.05 | C |
| xc5vlx85ff676-2 | 676 | 12960 | 51840 | 51840 | 96 | 48 | 0 | 0 | 0 | 0 | 85 | 85 | 0.95 | 0.95 | 1.05 | C |
| xc5vlx110ff676-2 | 676 | 17280 | 69120 | 69120 | 128 | 64 | 0 | 0 | 0 | 0 | 85 | 85 | 0.95 | 0.95 | 1.05 | C |

X1105_c3_05_061910

*Figure 3-5:* **New Project (Product Family and Floorplan)**

10. Choose the device specifics:
    a. Select the **FF676** package
    b. Select the **-2** speed grade
    c. Select the **C** temperature grade
    d. Select the **xc5vlx85ff676-2** device
11. Click **Next**.

12. Figure 3-6 shows the New Project Summary screen, for review of the part and product family for the new project.



New Project Summary

A new post-synthesis project named 'FloorPlan_SCC' will be created from netlist file 'SCC_LAB_TOP.ngc'

3 netlist directories will be added.

No constraints files will be added. Use File | Add Constraints to add them later.

The default part and product family for the new project:
Default Part: xc5vlx85ff676-2
Product: Virtex5
Family: virtex5lx
Package: ff676
Speed Grade: -2

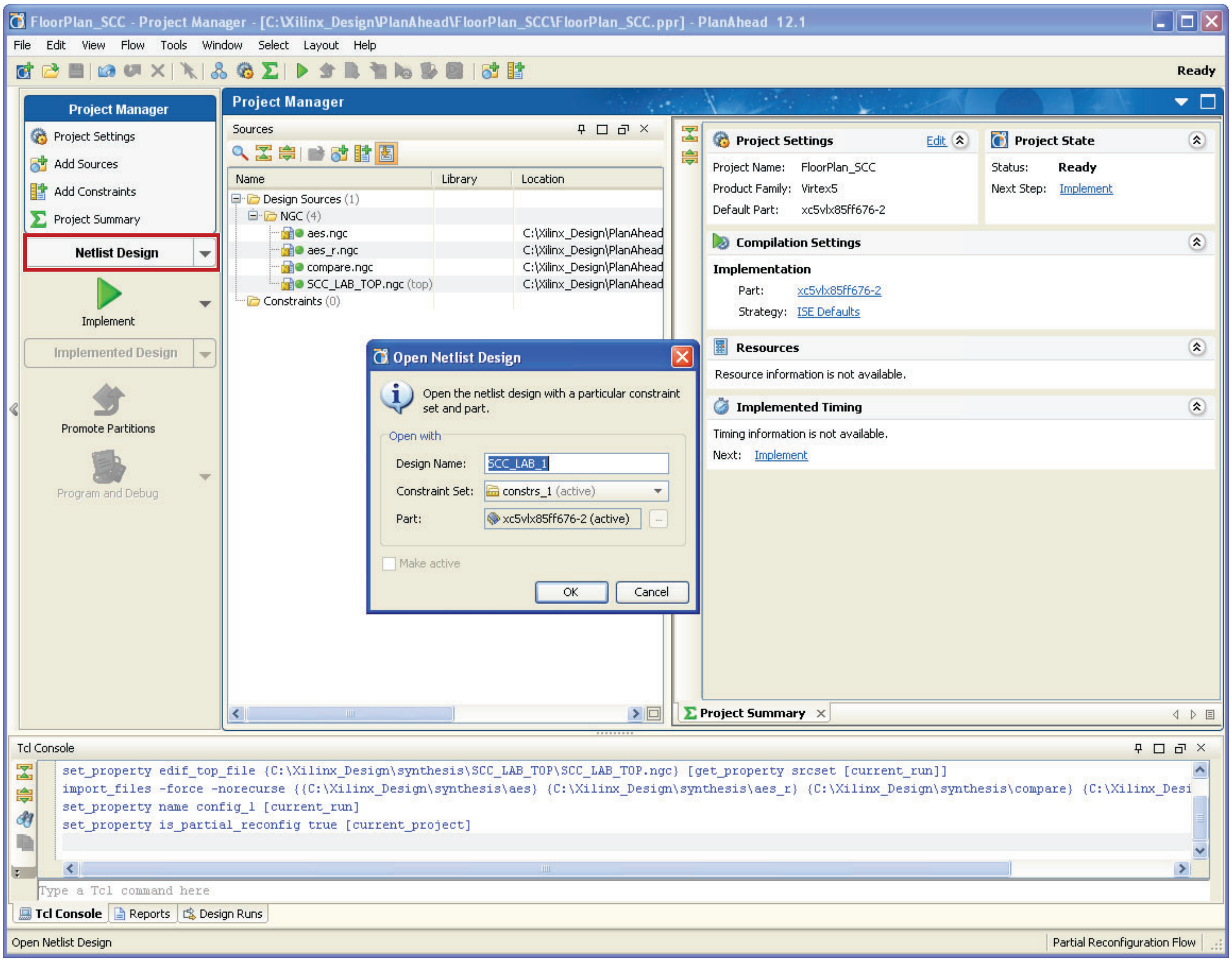
To create the project, click Finish

X1105_c3_06_061910

*Figure 3-6:* **New Project (New Project Summary)**

13. Click **Finish**.

14. The PlanAhead tool project is created. Figure 3-7 shows the PlanAhead tool Project Manager window and the Open Netlist Manager window for the FloorPlan_SCC project.



| Name | Library | Location |
|---|---|---|
| aes.ngc | | C:\Xilinx_Design\PlanAhead |
| aes_r.ngc | | C:\Xilinx_Design\PlanAhead |
| compare.ngc | | C:\Xilinx_Design\PlanAhead |
| SCC_LAB_TOP.ngc (top) | | C:\Xilinx_Design\PlanAhead |

Project Name: FloorPlan_SCC
Product Family: Virtex5
Default Part: xc5vlx85ff676-2

Status: Ready
Next Step: Implement

Part: xc5vlx85ff676-2
Strategy: ISE Defaults

Resource information is not available.

Timing information is not available.
Next: Implement

Open Netlist Design

Open the netlist design with a particular constraint set and part.

Open with
Design Name: SCC_LAB_1
Constraint Set: constrs_1 (active)
Part: xc5vlx85ff676-2 (active)

□ Make active

OK Cancel

```
set_property edif_top_file {C:\Xilinx_Design\synthesis\SCC_LAB_TOP\SCC_LAB_TOP.ngc} [get_property srcset [current_run]]
import_files -force -norecurse {{C:\Xilinx_Design\synthesis\aes} {C:\Xilinx_Design\synthesis\aes_r} {C:\Xilinx_Design\synthesis\compare} {C:\Xilinx_Desi
set_property name config_1 [current_run]
set_property is_partial_reconfig true [current_project]
```

Type a Tcl command here

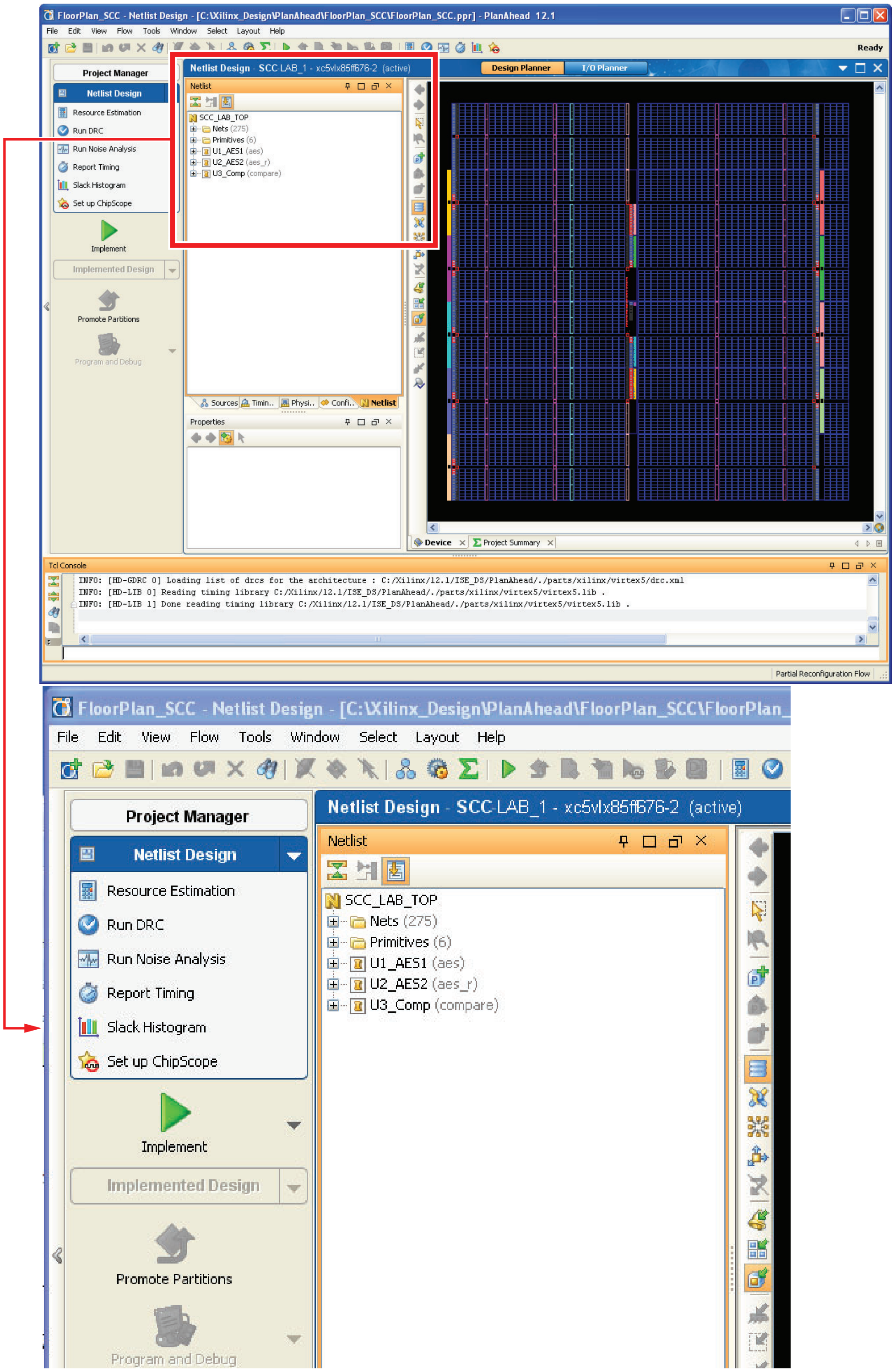
X1105_c3_07_061910

*Figure 3-7:* **PlanAhead Tool Project Manger Window (Open Netlist Design)**

15. To open the netlist design in the PlanAhead tool, under the Project Manager pane on the left, select **Netlist Design** → **Open Netlist Design...**.

16. Set the design name to **SCC_LAB_1**. Click **OK**.

17. The PlanAhead tool project now looks as shown in Figure 3-8.



```
INFO: [HD-GDRC 0] Loading list of drcs for the architecture : C:/Xilinx/12.1/ISE_DS/PlanAhead/./parts/xilinx/virtex5/drc.xml
INFO: [HD-LIB 0] Reading timing library C:/Xilinx/12.1/ISE_DS/PlanAhead/./parts/xilinx/virtex5/virtex5.lib .
INFO: [HD-LIB 1] Done reading timing library C:/Xilinx/12.1/ISE_DS/PlanAhead/./parts/xilinx/virtex5/virtex5.lib .
```

X1105_c3_08_061910

*Figure 3-8:* **PlanAhead Project View**

18. The project was turned into a PR project in step 4; therefore, the **File** → **Set PR Project** option is grayed out to indicate that this step has been completed.

# Placing 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 3-9 that is on the vertical shortcut bar between the netlist window and the device window.

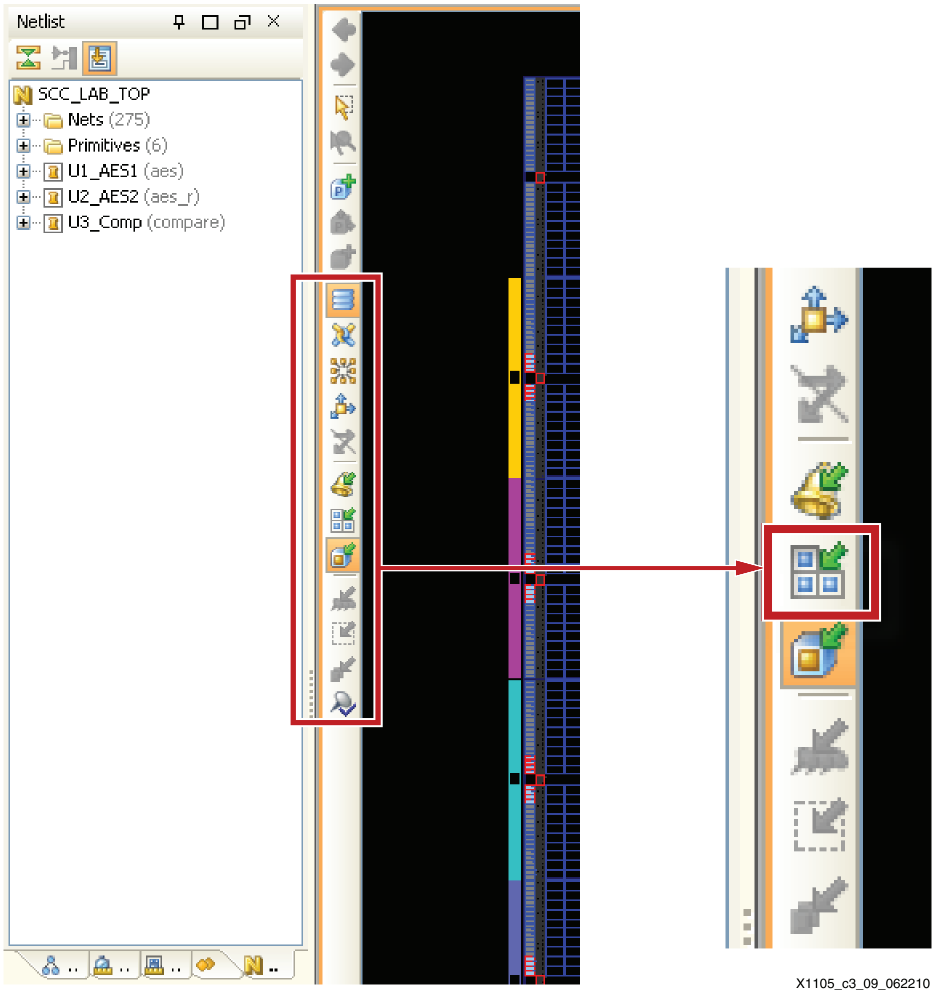
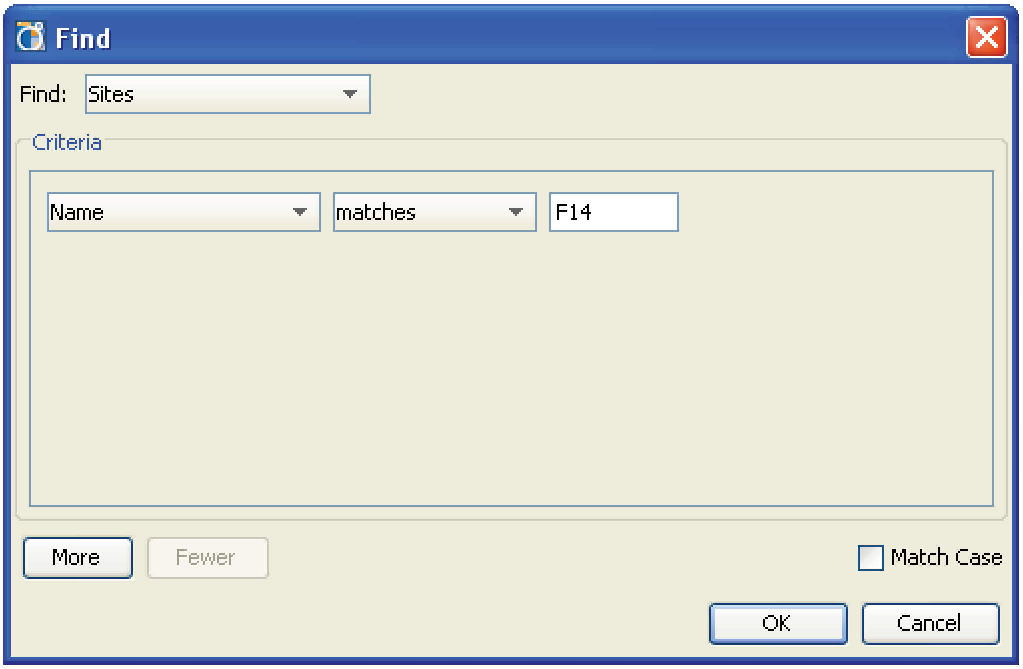


*Figure 3-9:* **Select Site Constraint Mode Symbol**

2. Select **Edit** → **Find** (shortcut: **Ctrl-F**). A Find window appears, as shown in Figure 3-10.



Find: Sites

Name | matches | F14

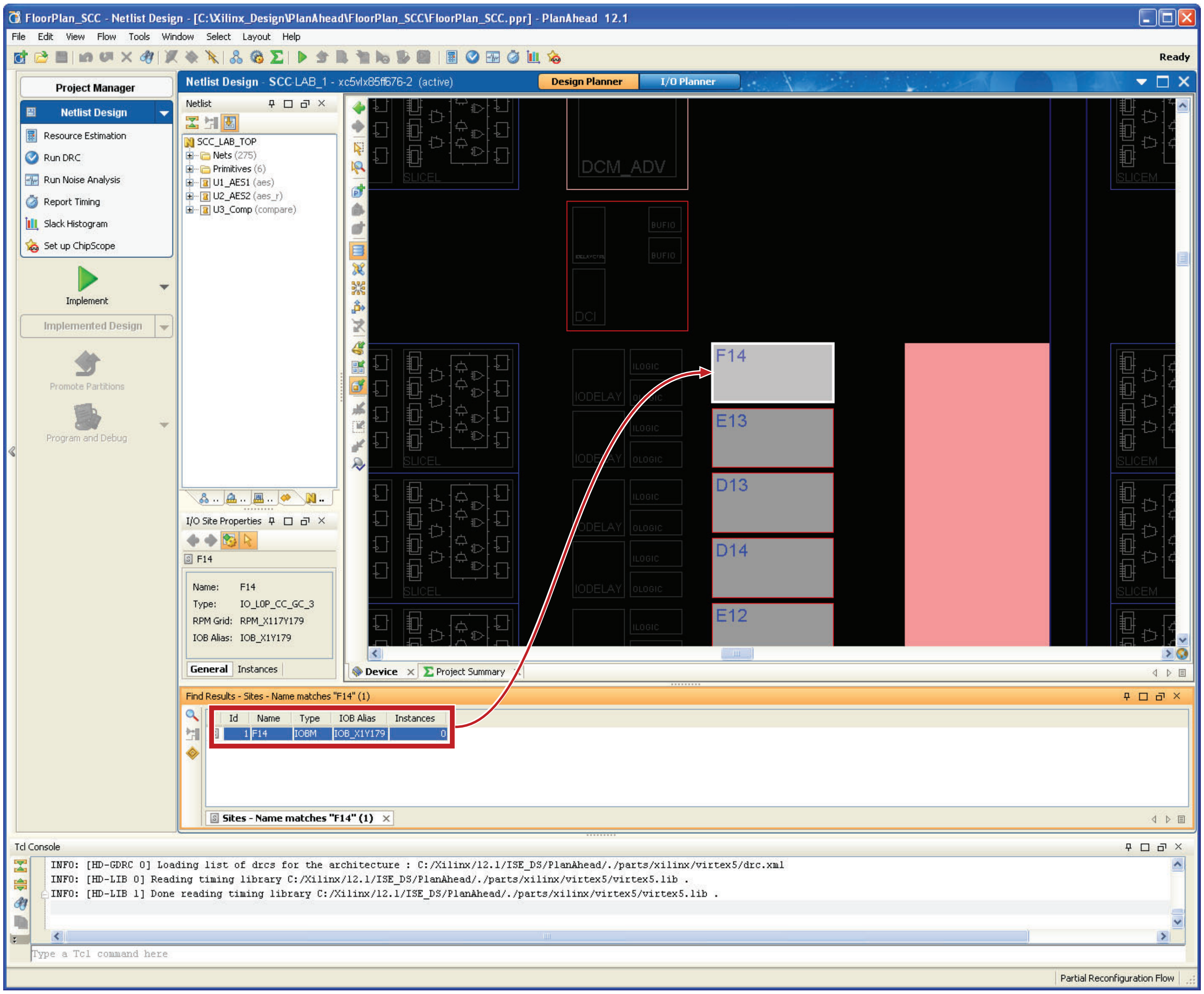
X1105_c3_10_061910

*Figure 3-10:* **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.

6. Select the result and press **F9** to zoom to the current selection, as shown in Figure 3-11. Alternatively, select **View** → **Fit Selection** from the top menu bar to perform the same task.



I/O Site Properties

F14

Name: F14
Type: IO_L0P_CC_GC_3
RPM Grid: RPM_X117Y179
IOB Alias: IOB_X1Y179

General | Instances

Find Results - Sites - Name matches "F14" (1)

| Id | Name | Type | IOB Alias | Instances |
|---|---|---|---|---|
| 1 | F14 | IOBM | IOB_X1Y179 | 0 |

Sites - Name matches "F14" (1)

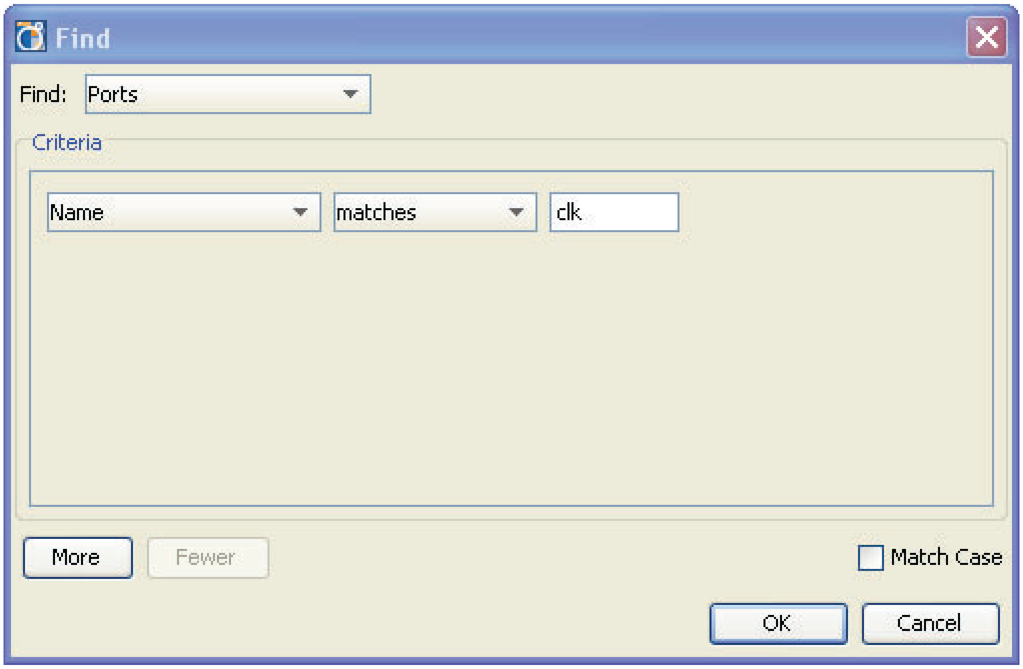
```
INFO: [HD-GDRC 0] Loading list of drcs for the architecture : C:/Xilinx/12.1/ISE_DS/PlanAhead/./parts/xilinx/virtex5/drc.xml
INFO: [HD-LIB 0] Reading timing library C:/Xilinx/12.1/ISE_DS/PlanAhead/./parts/xilinx/virtex5/virtex5.lib .
INFO: [HD-LIB 1] Done reading timing library C:/Xilinx/12.1/ISE_DS/PlanAhead/./parts/xilinx/virtex5/virtex5.lib .
```

Type a Tcl command here

X1105_c3_11_062310

*Figure 3-11:* **Find Results**

7. Select **Edit** → **Find** (shortcut: **Ctrl-F**). A Find window appears, as shown in Figure 3-12.



Find: Ports

Criteria

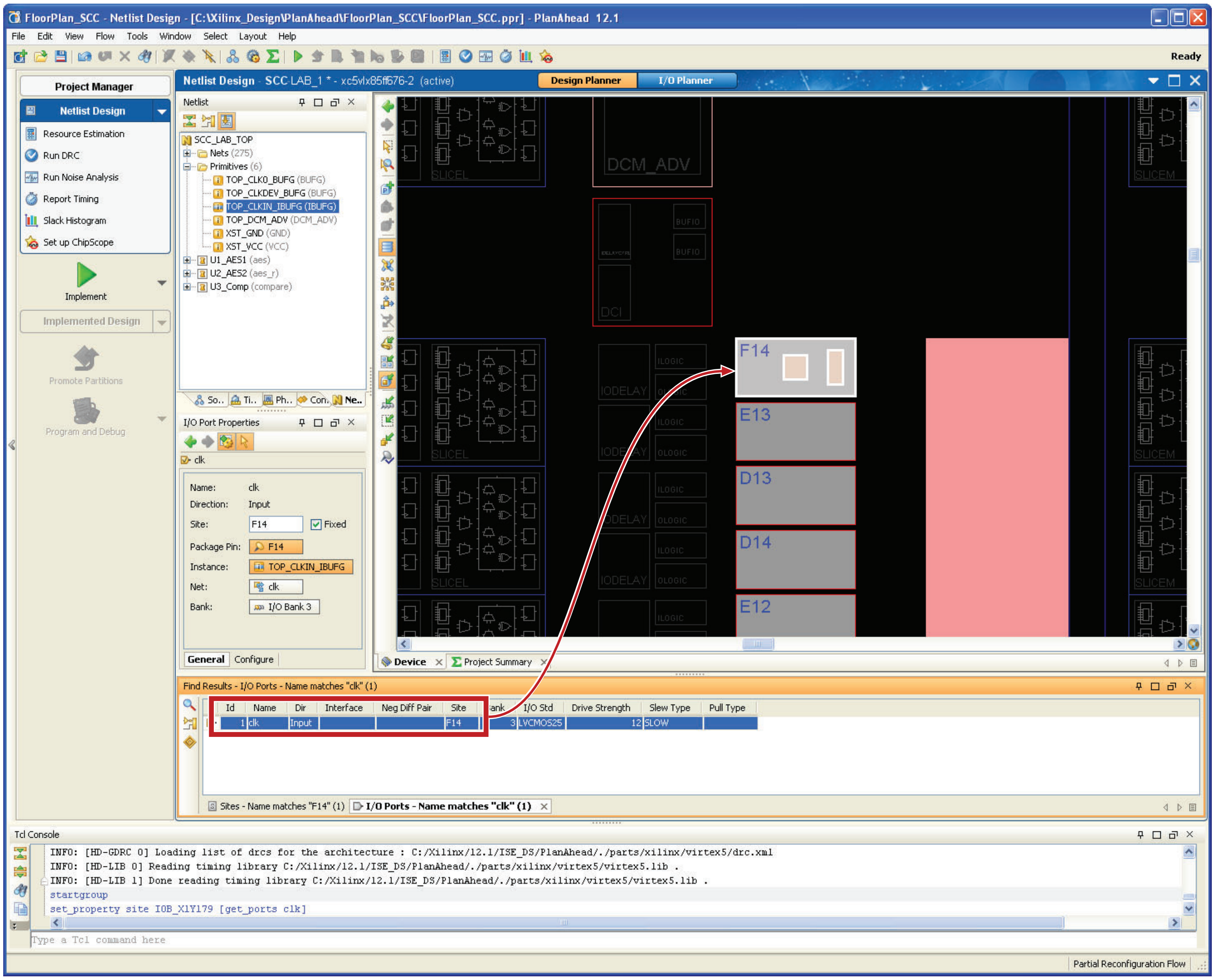
Name matches clk

X1105_c3_12_061910

*Figure 3-12:* **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 3-13.



I/O Port Properties

clk

Name: clk
Direction: Input
Site: F14 ☑ Fixed
Package Pin: F14
Instance: TOP_CLKIN_IBUFG
Net: clk
Bank: I/O Bank 3

General Configure

Find Results - I/O Ports - Name matches "clk" (1)

| Id | Name | Dir | Interface | Neg Diff Pair | Site | ank | I/O Std | Drive Strength | Slew Type | Pull Type |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | clk | Input | | | F14 | 3 | LVCMOS25 | 12 | SLOW | |

Sites - Name matches "F14" (1) | I/O Ports - Name matches "clk" (1)

```
INFO: [HD-GDRC 0] Loading list of drcs for the architecture : C:/Xilinx/12.1/ISE_DS/PlanAhead/./parts/xilinx/virtex5/drc.xml
INFO: [HD-LIB 0] Reading timing library C:/Xilinx/12.1/ISE_DS/PlanAhead/./parts/xilinx/virtex5/virtex5.lib .
INFO: [HD-LIB 1] Done reading timing library C:/Xilinx/12.1/ISE_DS/PlanAhead/./parts/xilinx/virtex5/virtex5.lib .
startgroup
set_property site IOB_X1Y179 [get_ports clk]
Type a Tcl command here
```

X1105_c3_13_062310

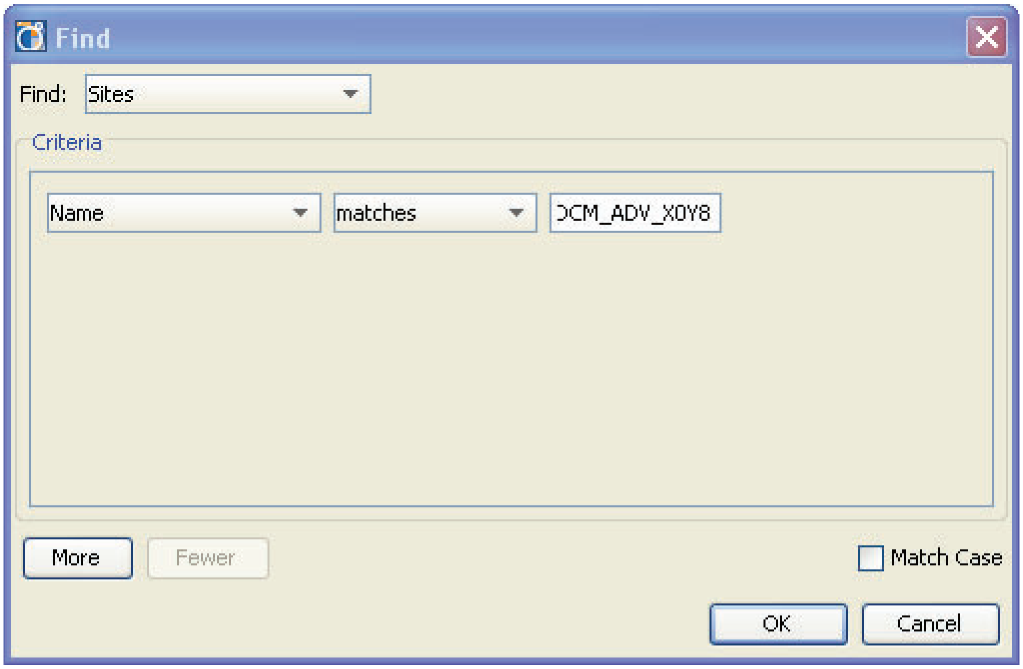
*Figure 3-13:* **Drag clk**

12. Repeat step 1 to step 11 to place “reset” at D11.
13. Repeat step 1 to step 11 to place “push_button” at D10.
14. Repeat step 1 to step 11 to place “led” at P6.

# Placing DCM with the PlanAhead Tool

These steps describe how to place the digital clock manager (DCM) using the PlanAhead tool:

1. Select **Edit** → **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 (see Figure 3-14).



Find: Sites

Name | matches | DCM_ADV_X0Y8

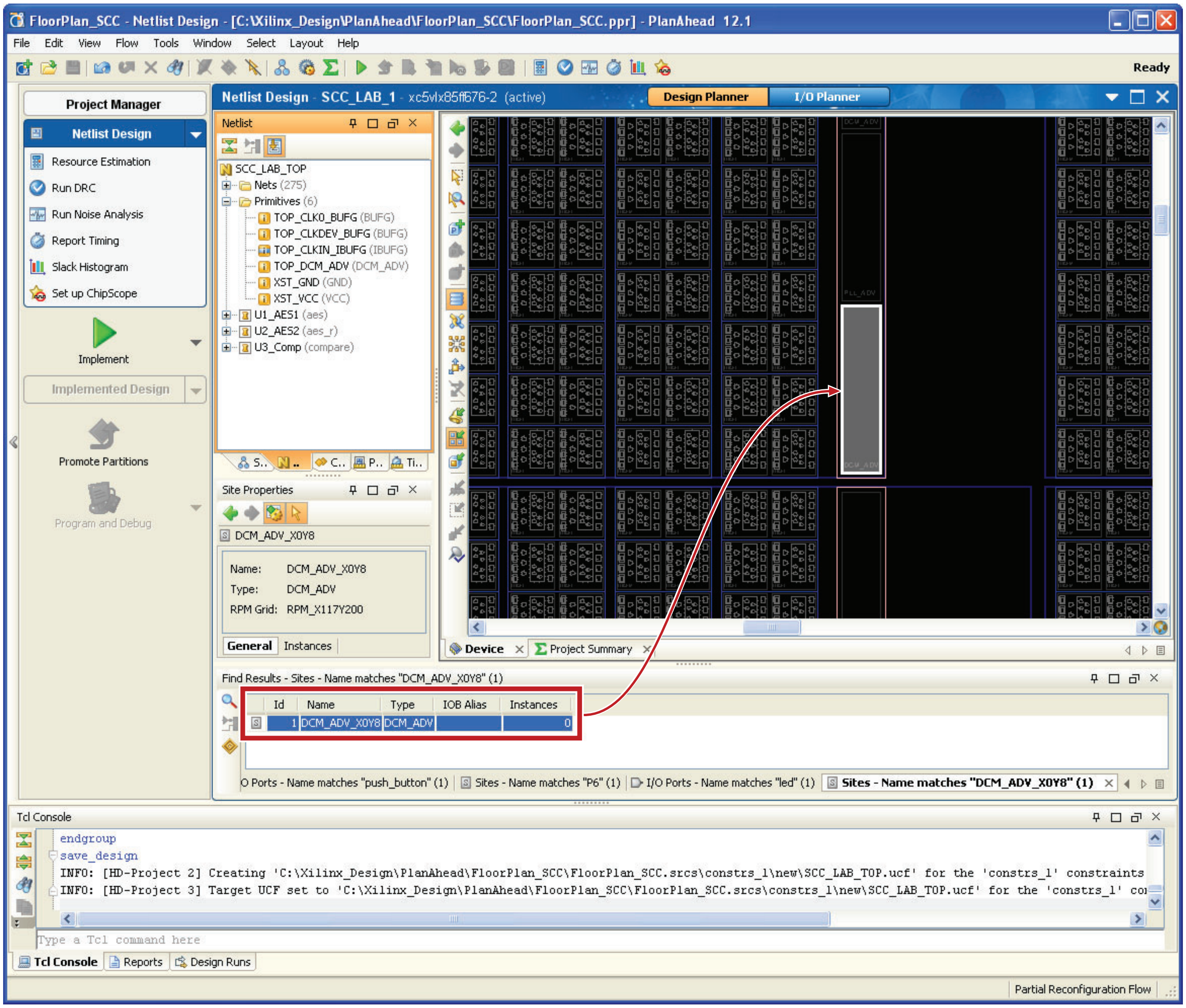
X1105_c3_14_062010

*Figure 3-14:* **Find**

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.

4. Select this result and press **F9** to zoom to the current selection. The PlanAhead tool then zooms to the selected location (see Figure 3-15). Alternatively, select **View** → **Fit Selection** from the top menu bar to perform the same task.



DCM_ADV_X0Y8

Name: DCM_ADV_X0Y8
Type: DCM_ADV
RPM Grid: RPM_X117Y200

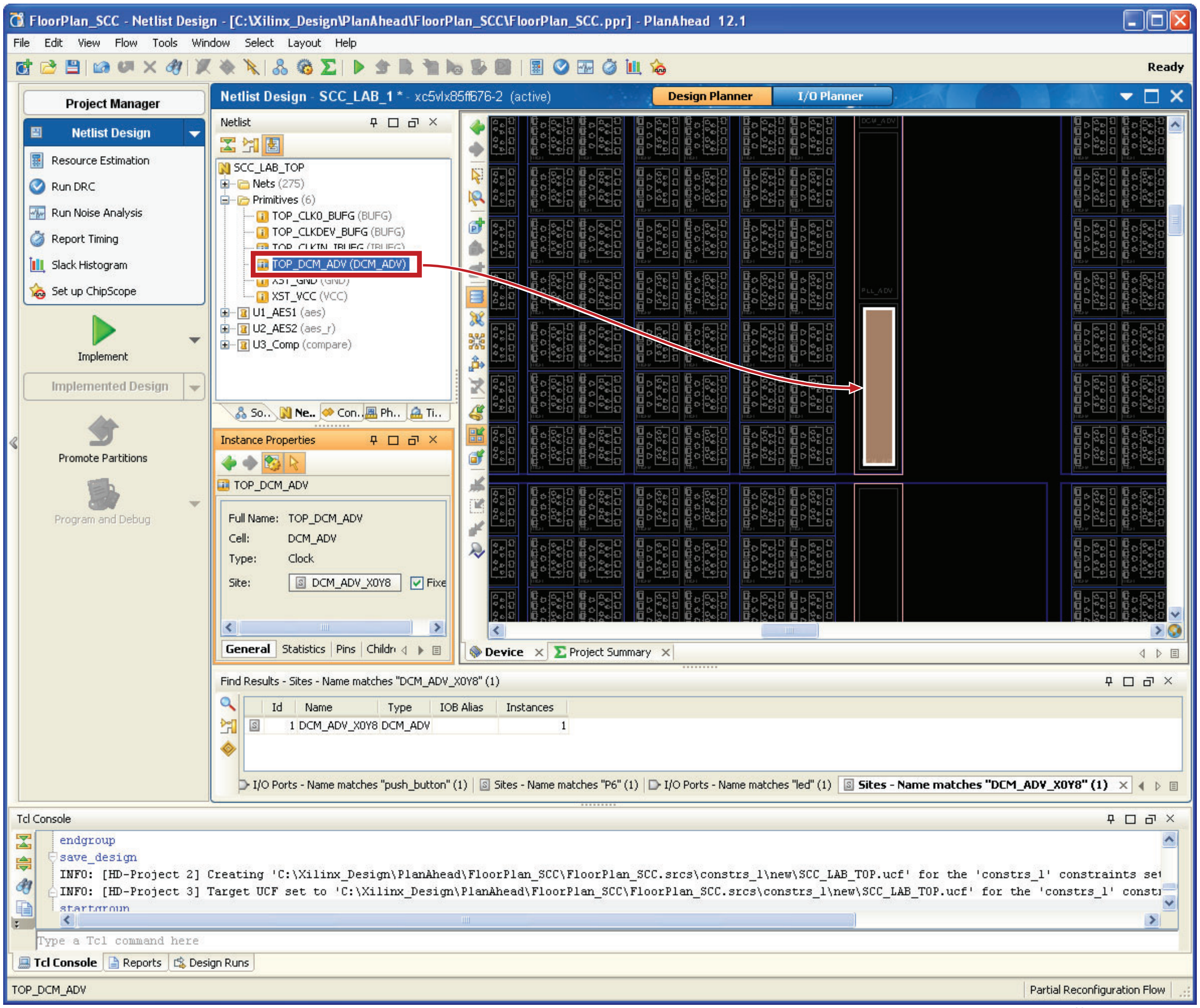
| Id | Name | Type | IOB Alias | Instances |
|---|---|---|---|---|
| 1 | DCM_ADV_X0Y8 | DCM_ADV | | 0 |

```
endgroup
save_design
INFO: [HD-Project 2] Creating 'C:\Xilinx_Design\PlanAhead\FloorPlan_SCC\FloorPlan_SCC.srcs\constrs_1\new\SCC_LAB_TOP.ucf' for the 'constrs_1' constraints
INFO: [HD-Project 3] Target UCF set to 'C:\Xilinx_Design\PlanAhead\FloorPlan_SCC\FloorPlan_SCC.srcs\constrs_1\new\SCC_LAB_TOP.ucf' for the 'constrs_1' co
```

X1105_c3_15_062310

*Figure 3-15:* **Search Results**

5. In the Netlist window, expand **Primitives**, then drag **TOP_DCM_ADV** to the DCM_ADV_X0Y8 box, as shown in Figure 3-16.



Full Name: TOP_DCM_ADV
Cell: DCM_ADV
Type: Clock
Site: DCM_ADV_X0Y8 ☑ Fixe

Find Results - Sites - Name matches "DCM_ADV_X0Y8" (1)

| Id | Name | Type | IOB Alias | Instances |
|---|---|---|---|---|
| 1 | DCM_ADV_X0Y8 | DCM_ADV | | 1 |

I/O Ports - Name matches "push_button" (1) | Sites - Name matches "P6" (1) | I/O Ports - Name matches "led" (1) | Sites - Name matches "DCM_ADV_X0Y8" (1)

```
endgroup
save_design
INFO: [HD-Project 2] Creating 'C:\Xilinx_Design\PlanAhead\FloorPlan_SCC\FloorPlan_SCC.srcs\constrs_1\new\SCC_LAB_TOP.ucf' for the 'constrs_1' constraints set
INFO: [HD-Project 3] Target UCF set to 'C:\Xilinx_Design\PlanAhead\FloorPlan_SCC\FloorPlan_SCC.srcs\constrs_1\new\SCC_LAB_TOP.ucf' for the 'constrs_1' constr
startgroup
```

Type a Tcl command here

TOP_DCM_ADV

X1105_c3_16_062310

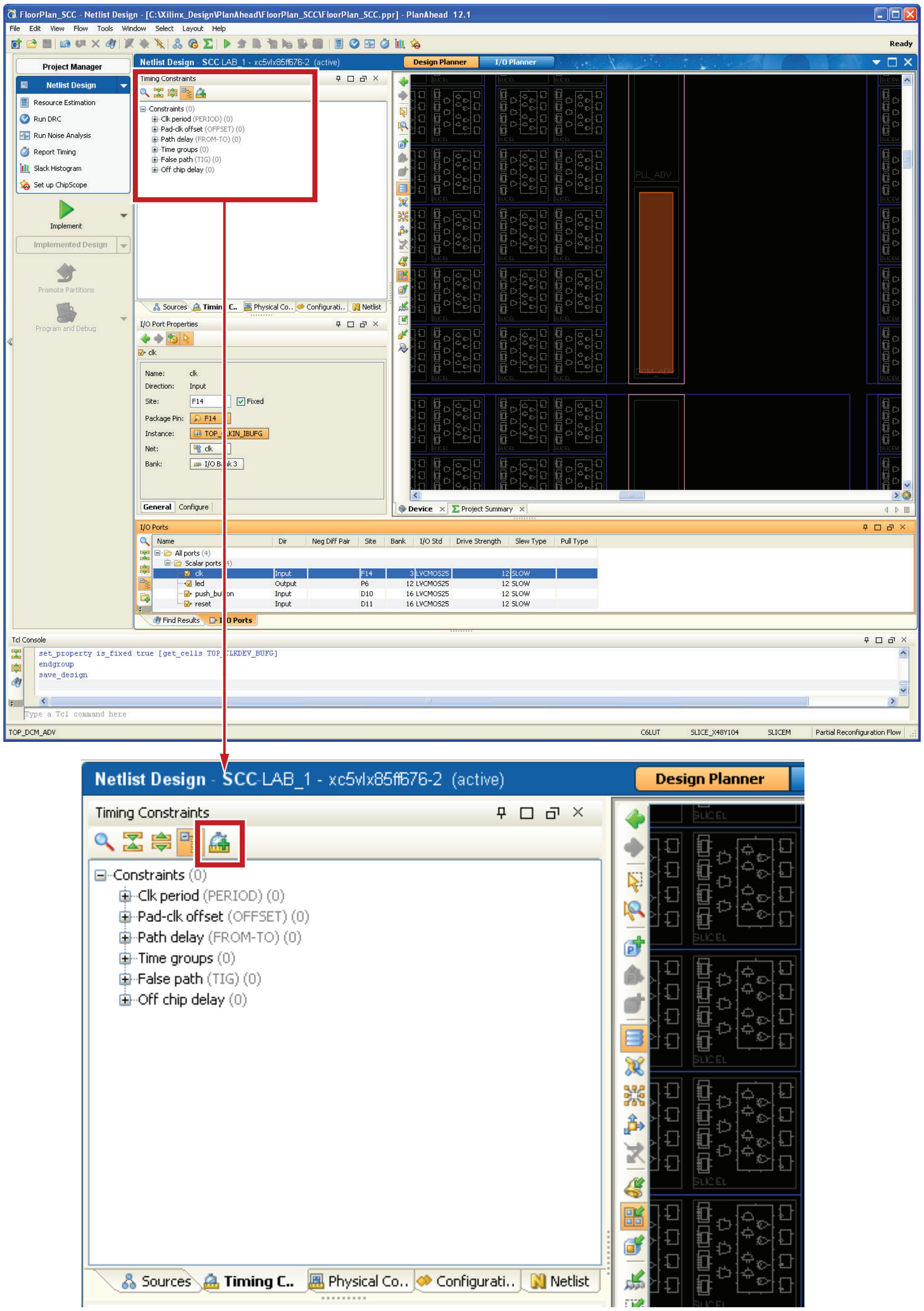
*Figure 3-16:* **DCM_ADV_X0Y8**

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.

1. Select the **Timing Constraints** tab in the Netlist Design window, and click the **Create New Constraint** button located at the top of the Netlist/Constraints window (see Figure 3-17).



I/O Port Properties: clk

Name: clk
Direction: Input
Site: F14 ☑ Fixed
Package Pin: F14
Instance: TOP_ KIN_IBUFG
Net: clk
Bank: I/O B k 3

I/O Ports

| Name | Dir | Neg Diff Pair | Site | Bank | I/O Std | Drive Strength | Slew Type | Pull Type |
|---|---|---|---|---|---|---|---|---|
| All ports (4) | | | | | | | | |
| Scalar ports 4) | | | | | | | | |
| clk | Input | | F14 | 3 | LVCMOS25 | 12 | SLOW | |
| led | Output | | P6 | 12 | LVCMOS25 | 12 | SLOW | |
| push_bu on | Input | | D10 | 16 | LVCMOS25 | 12 | SLOW | |
| reset | Input | | D11 | 16 | LVCMOS25 | 12 | SLOW | |

Tcl Console

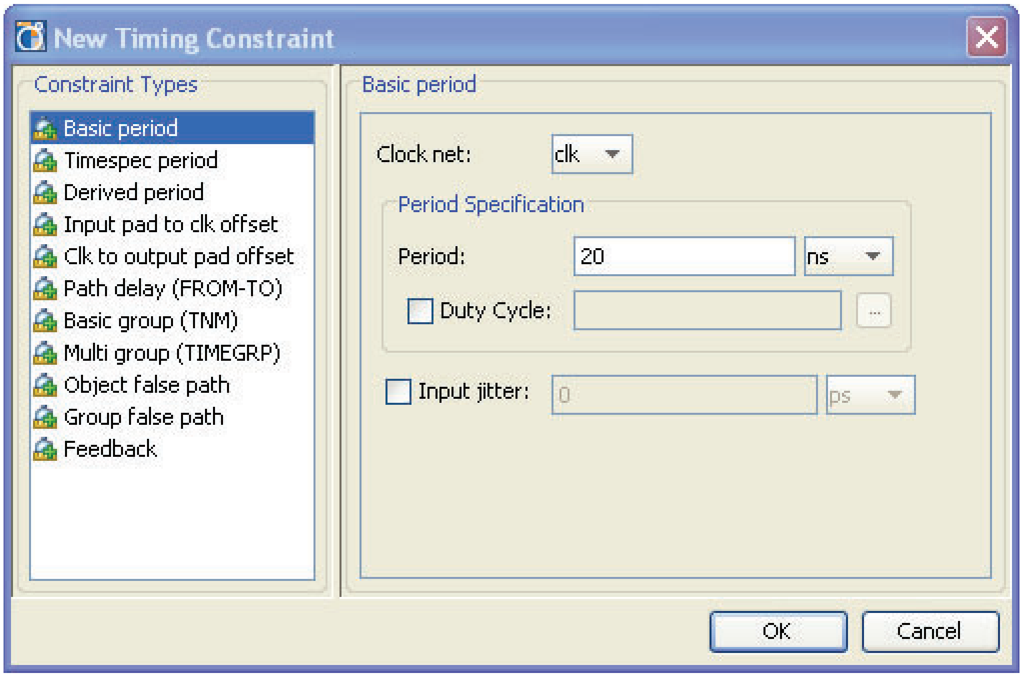
```
set_property is_fixed true [get_cells TOP_ LKDEV_BUFG]
endgroup
save_design
```

Type a Tcl command here

X1135_c3_17_062410

*Figure 3-17:* **Create New Constraint**

2. Add a basic period constraint of 20 ns to the design by entering the value as shown in Figure 3-18.



Clock net: clk

Period: 20 ns

□ Duty Cycle:

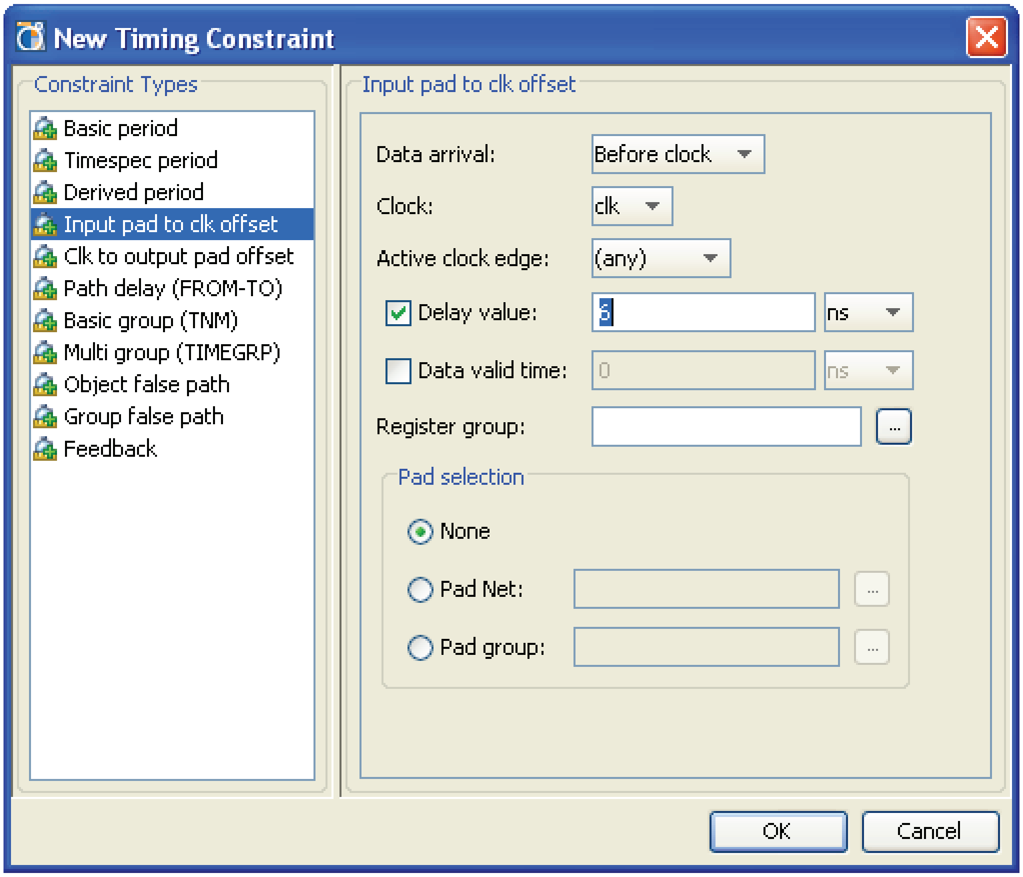
□ Input jitter: 0 ps

X1135_c3_18_062010

*Figure 3-18:* **New Timing Constraint**

3. Click **OK**.
4. Create a new timing constraint for Input pad to clk offset. 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 3-19.



Data arrival: Before clock

Clock: clk

Active clock edge: (any)

☑ Delay value: 6 ns

□ Data valid time: 0 ns

Register group:

● None

○ Pad Net:

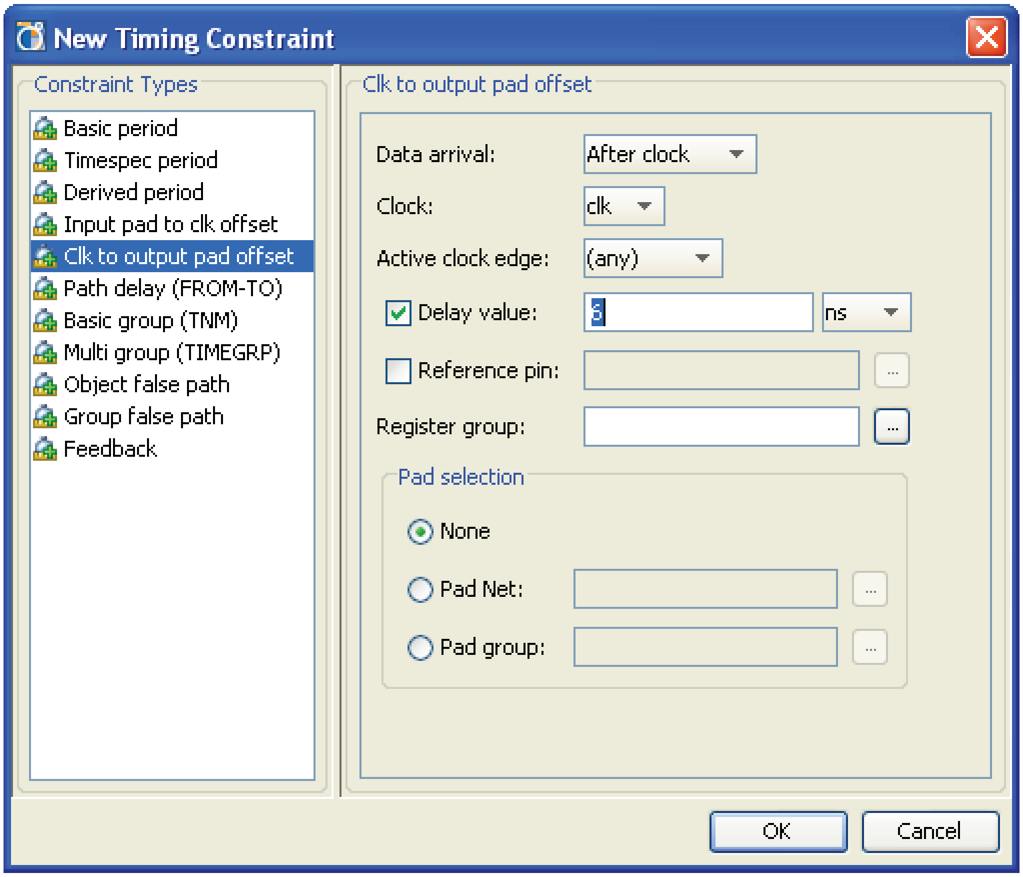
○ Pad group:

X1135_c3_19_062010

*Figure 3-19:* **New Timing Constraint**

5. Click **OK**.
6. Create a new timing constraint for Clk to output pad offset. 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 3-20.



Data arrival: After clock

Clock: clk

Active clock edge: (any)

☑ Delay value: 5 ns

□ Reference pin:

Register group:

● None

○ Pad Net:

○ Pad group:

X1135_c3_20_062010

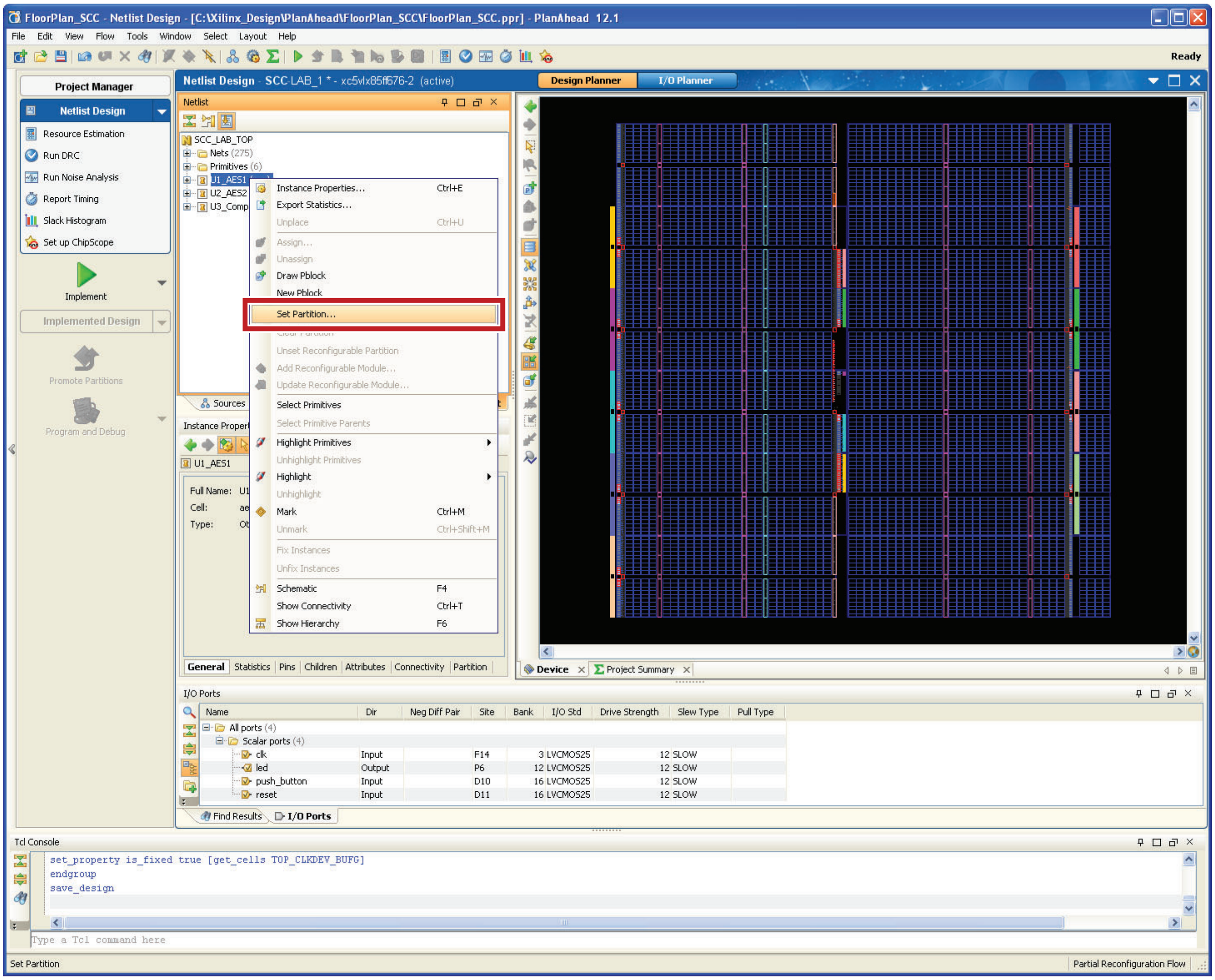
*Figure 3-20:* **New Timing Constraint (Delay Value)**

7. Click **OK**.

# Defining RP or ISO Partitions with the PlanAhead Tool

Each partition must be turned into either a reconfigurable partition (RP) or an ISO partition as follows:

1. Select and right-click on **U1_AES (aes)** in the Netlist tab and select **Set Partition** from the pull-down menu (see Figure 3-21).
2. The Set Partition wizard opens. Click **Next**.



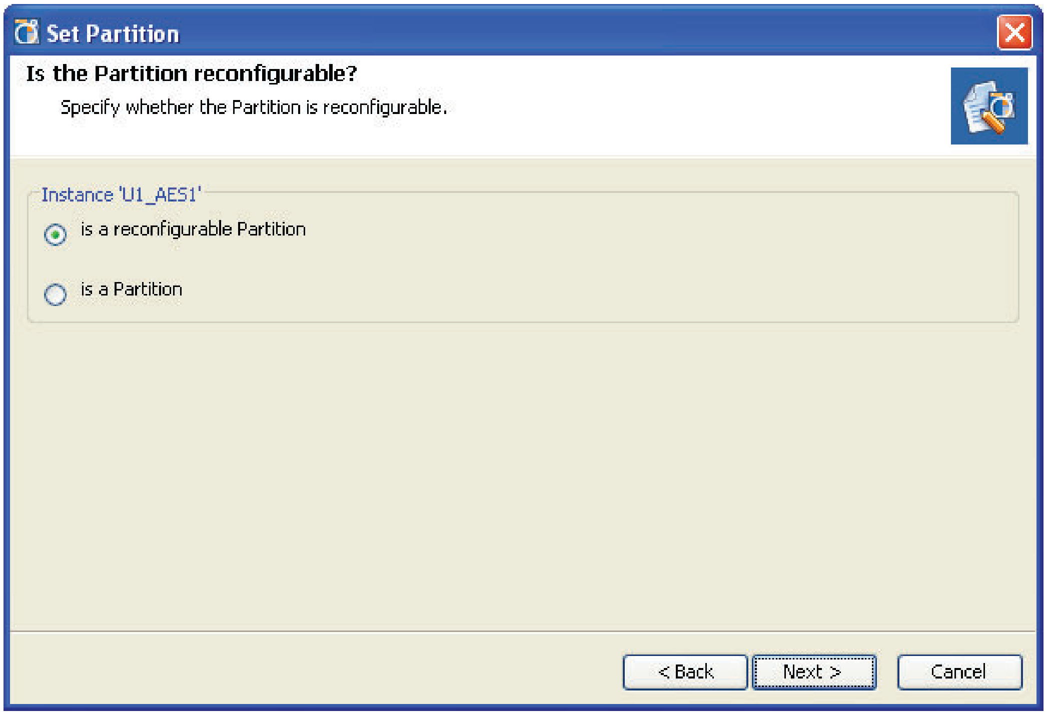
| Name | Dir | Neg Diff Pair | Site | Bank | I/O Std | Drive Strength | Slew Type | Pull Type |
|---|---|---|---|---|---|---|---|---|
| All ports (4) | | | | | | | | |
| Scalar ports (4) | | | | | | | | |
| clk | Input | | F14 | 3 | LVCMOS25 | 12 | SLOW | |
| led | Output | | P6 | 12 | LVCMOS25 | 12 | SLOW | |
| push_button | Input | | D10 | 16 | LVCMOS25 | 12 | SLOW | |
| reset | Input | | D11 | 16 | LVCMOS25 | 12 | SLOW | |

```
set_property is_fixed true [get_cells TOP_CLKDEV_BUFG]
endgroup
save_design
```

X1135_c3_21_062010

*Figure 3-21:* **Set Partition**

3. Select **is a reconfigurable Partition**, as shown in Figure 3-22. Click **Next**.



Set Partition

Is the Partition reconfigurable?

Specify whether the Partition is reconfigurable.

Instance 'U1_AES1'

● is a reconfigurable Partition

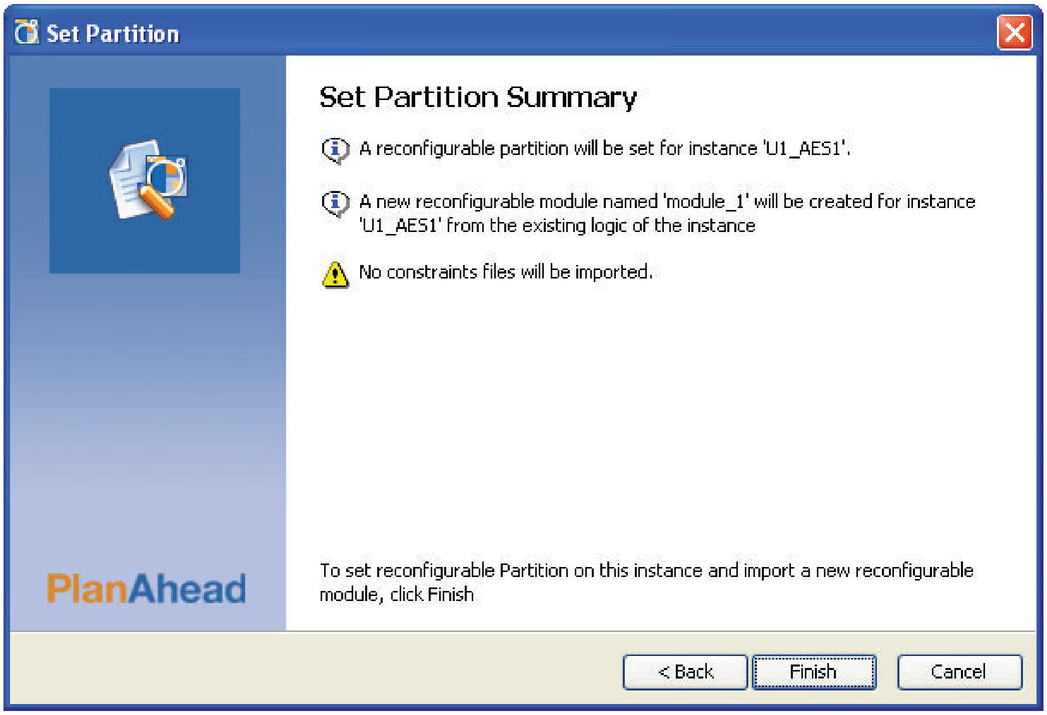
○ is a Partition

X1135_c3_22_062010

*Figure 3-22:* **Set Partition (Set Reconfigurable Partition)**

4. Keep the default name at **module_1**. If there are to be multiple configurations for this block, the user should change the field to a more descriptive name for that specific configuration. Click **Next**.

5. The Set Partition Summary screen is displayed, as shown in Figure 3-23. Click **Finish**.



Set Partition Summary

A reconfigurable partition will be set for instance 'U1_AES1'.

A new reconfigurable module named 'module_1' will be created for instance 'U1_AES1' from the existing logic of the instance

No constraints files will be imported.

To set reconfigurable Partition on this instance and import a new reconfigurable module, click Finish

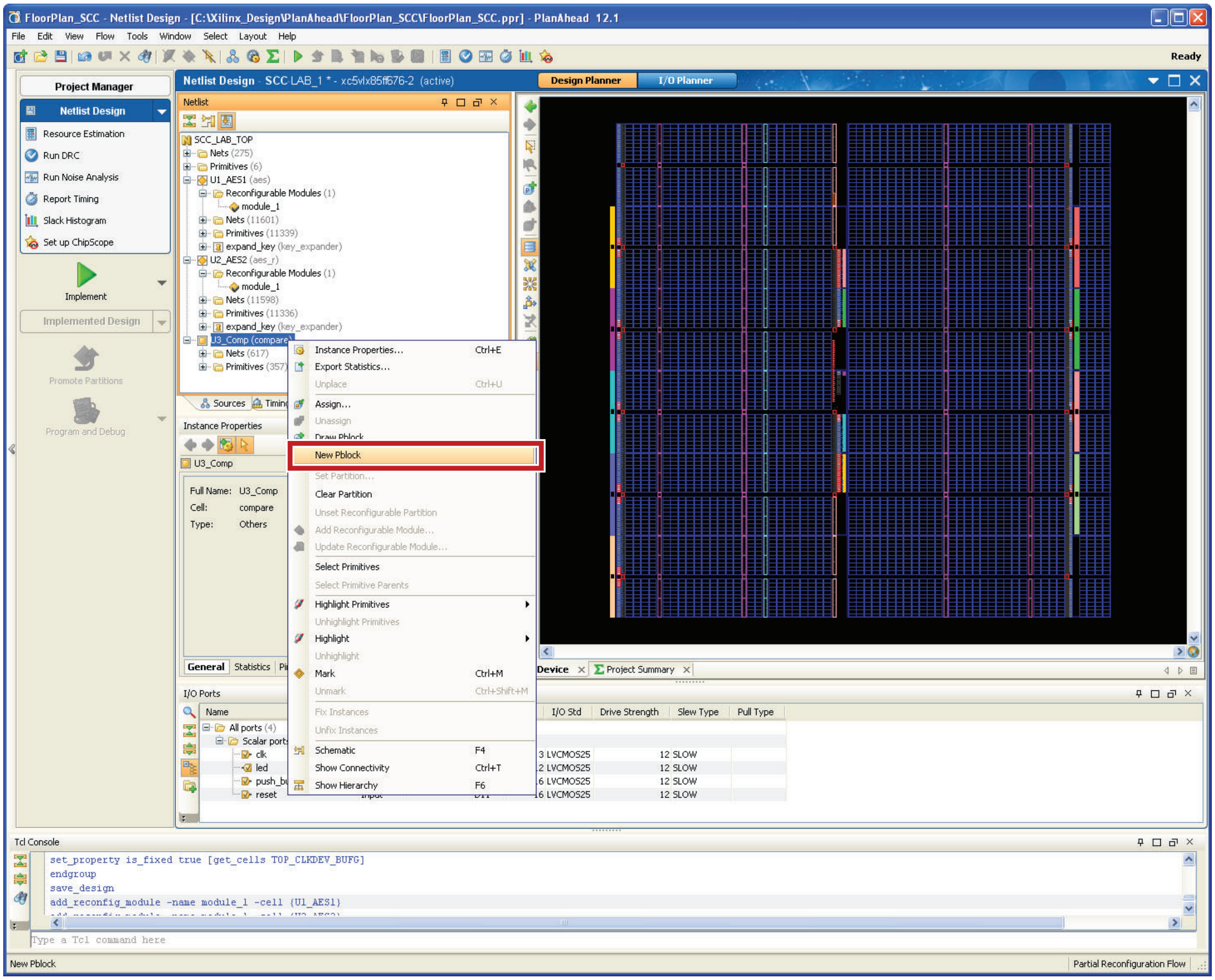
X1135_c3_23_062010

*Figure 3-23:* **Set Partition (Set Reconfigurable Partition)**

6. Repeat step 1 through step 4 for the **U2_AES2 (aes_r)** partition.
7. Repeat step 1 through step 4 for the **U3_Comp (compare)** partition with the following exception:

   Select the **is a Partition** option instead of the **is a reconfigurable Partition** option. The compare block is an ISO block, not an RP block. This option is selected in the Set Partition wizard screen (shown in Figure 3-22).

8. A new Pblock must be created for the compare block. A Pblock is automatically created for RPs but not for ISO partitions. Select and right-click on **U3_Comp (compare)** in the Netlist tab and select **New Pblock** from the pull-down menu (see Figure 3-24). Keep the default name, **pblock_U3_Comp**, and click **OK**.



Full Name: U3_Comp
Cell: compare
Type: Others

| I/O Std | Drive Strength | Slew Type | Pull Type |
|---|---|---|---|
| LVCMOS25 | 12 | SLOW | |
| LVCMOS25 | 12 | SLOW | |
| LVCMOS25 | 12 | SLOW | |
| LVCMOS25 | 12 | SLOW | |

```
set_property is_fixed true [get_cells TOP_CLKDEV_BUFG]
endgroup
save_design
add_reconfig_module -name module_1 -cell {U1_AES1}
```

Type a Tcl command here

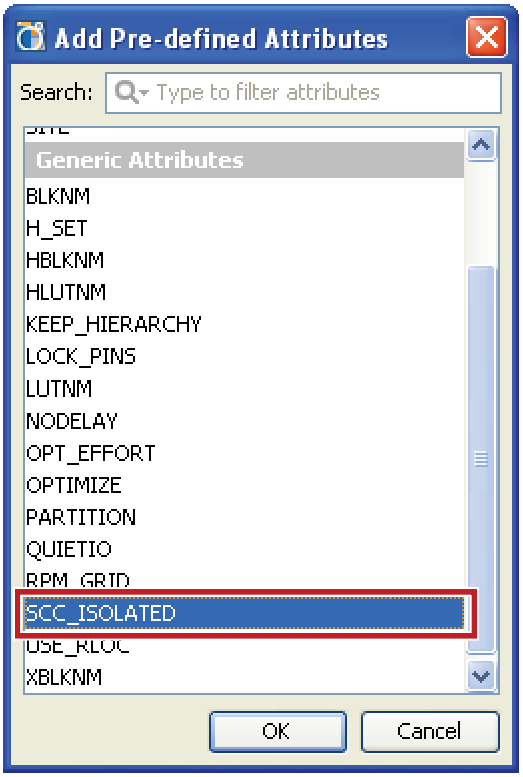
X1105_c3_24_062010

*Figure 3-24:* **Create New Pblock for the U3_Comp (compare) Partition**

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

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

1. Under the Netlist tab of the Netlist Design pane, select the **U1_AES1 (aes)** instance.
2. Under the Instance Properties window, select the **Attributes** tab for *U1_AES1*.
3. Select the **Add pre-defined attributes...** button (a green "+" symbol).
4. Select the **SCC_ISOLATED** attribute (see Figure 3-25).

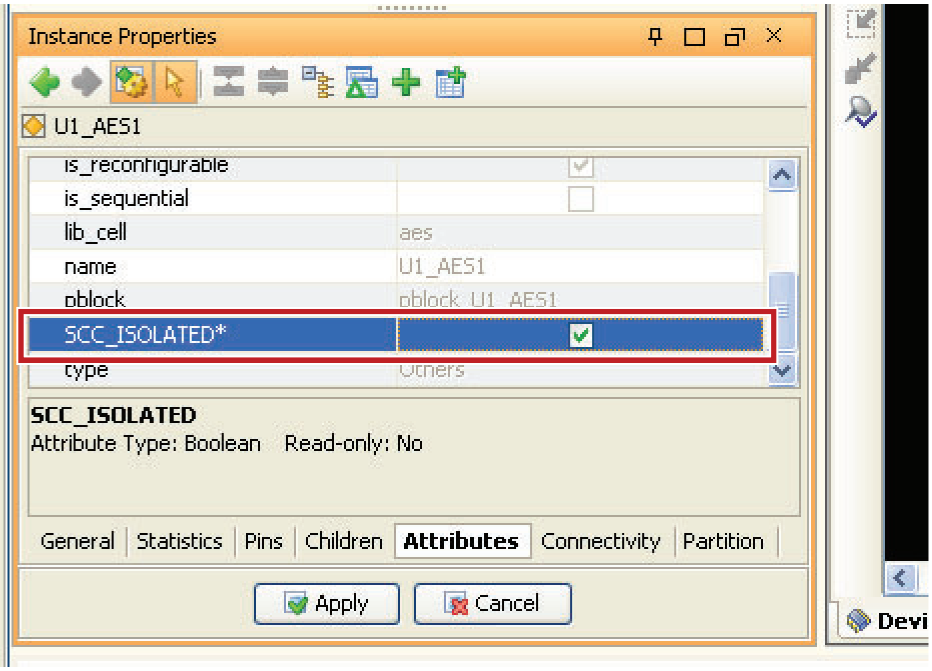


X1105_c3_25_062210

*Figure 3-25:* **Add Pre-defined Attributes (SCC_ISOLATED)**

5. Click **OK.**

6. In the **Instance Properties** window, scroll down to and check the box for the newly added **SCC_ISOLATED** attribute. Click **Apply** (see Figure 3-26).



Instance Properties

U1_AES1

| | |
|---|---|
| is_reconfigurable | ☑ |
| is_sequential | □ |
| lib_cell | aes |
| name | U1_AES1 |
| pblock | pblock_U1_AES1 |
| SCC_ISOLATED* | ☑ |
| type | Others |

**SCC_ISOLATED**
Attribute Type: Boolean Read-only: No

X1105_c3_26_062210

*Figure 3-26:* **Instance Properties (SCC_ISOLATED)**

7. Repeat step 1 through step 6 for the **U2_AES2 (aes_r)** instance.
8. Repeat step 1 through step 6 for the **U3_Comp (compare)** instance.

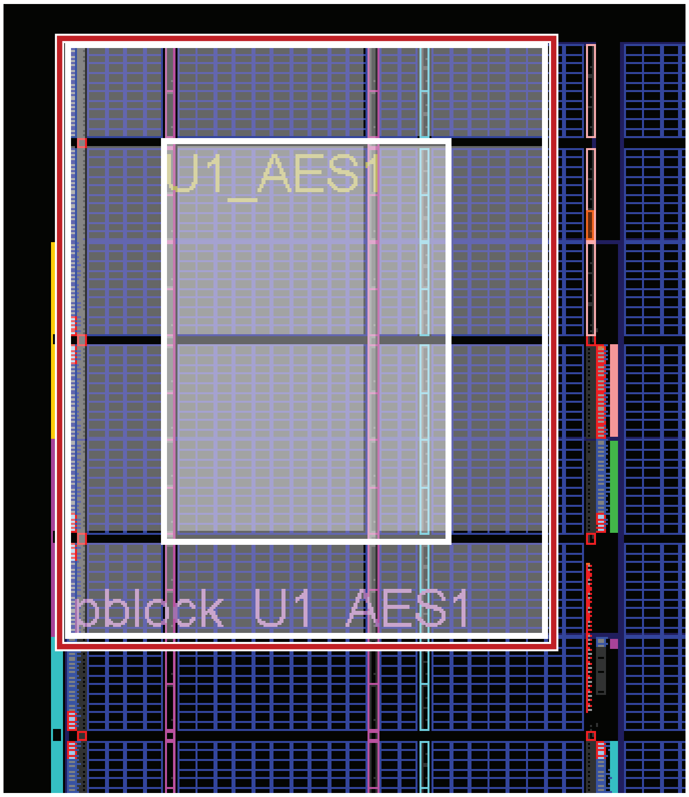
The **PRIVATE** attribute should be set for each pblock in the PlanAhead tool GUI as follows:

9. Under the **Physical Constraints** tab, select the **pblock_U1_AES** pblock, right-click on the selection, and click on **Pblock Properties** in the resulting menu.
10. Click on the **Attributes** tab of the **Pblock Properties** window, and click on the green "+" button to add pre-defined attributes.
11. Click on the **PRIVATE** attribute to select it, and click **OK**.
12. Select the attribute **PRIVATE** in the general attributes list in the **Pblock Properties** window, and select **NONE**.
13. Click **Apply**.
14. Repeat step 9 through step 13 for **pblock_U2_AES2** and set **PRIVATE** to **NONE**.
15. Repeat step 9 through step 13 for **pblock_U3_Comp** and set **PRIVATE** to **NONE**.

# Setting 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 Constraints 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 in the upper-left corner of the device window, as shown in Figure 3-27. (The rectangle does not have to be 100% accurate—it will be resized shortly.)



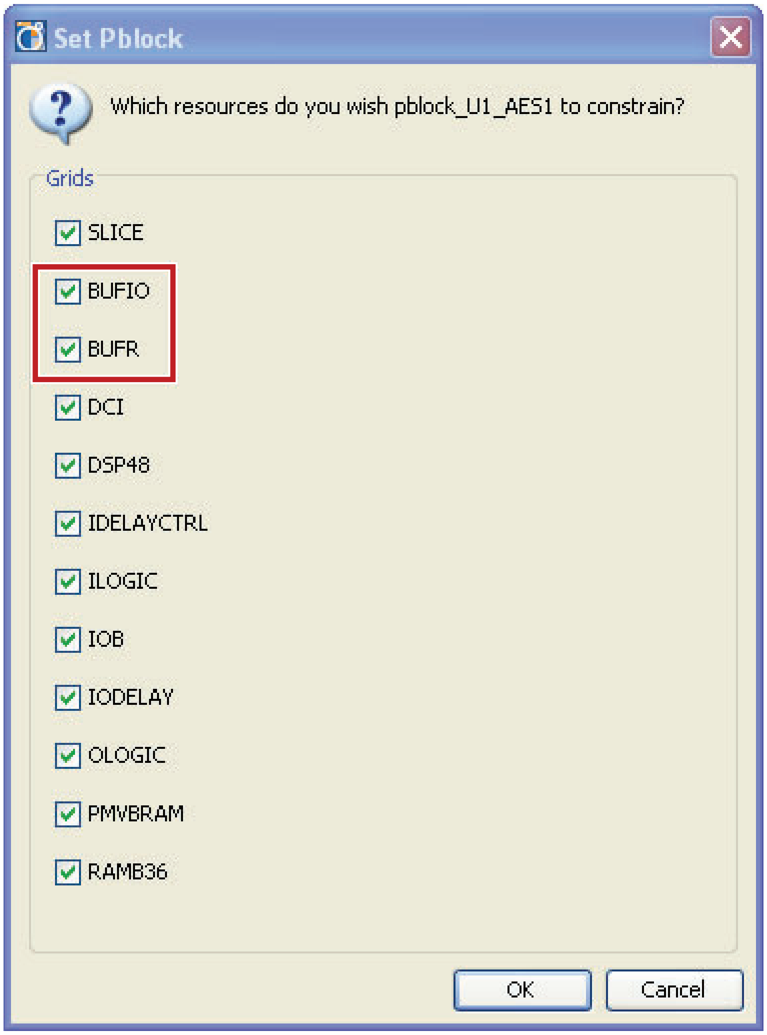
X1105_c3_27_062210

*Figure 3-27:* **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 configurable logic block (CLB) of separation between the Pblocks AES1, AES2, and COMPARE.

4. A dialog box (see Figure 3-28) appears 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

Which resources do you wish pblock_U1_AES1 to constrain?

Grids

☑ SLICE

☑ BUFIO

☑ BUFR

☑ DCI

☑ DSP48

☑ IDELAYCTRL

☑ ILOGIC

☑ IOB

☑ IODELAY

☑ OLOGIC

☑ PMVBRAM

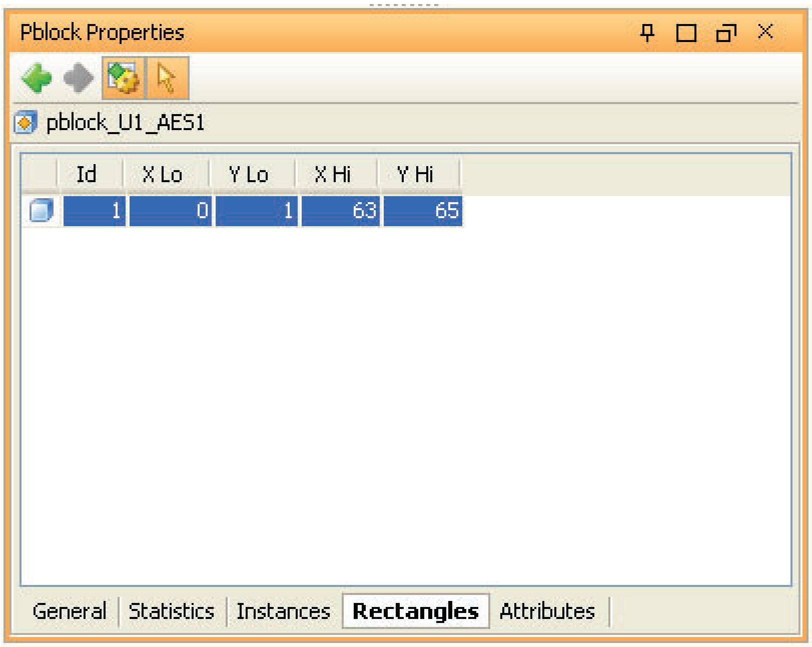
☑ RAMB36

X1105_c3_28_062210

*Figure 3-28:* **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 Constraints pane.

9. Select the **Rectangles** tab in the Pblock Properties window (see Figure 3-29).



pblock_U1_AES1

| Id | X Lo | Y Lo | X Hi | Y Hi |
|---|---|---|---|---|
| 1 | 0 | 1 | 63 | 65 |

X1105_c3_29_062210

*Figure 3-29:* **Pblock Properties**

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

    X Lo = 0

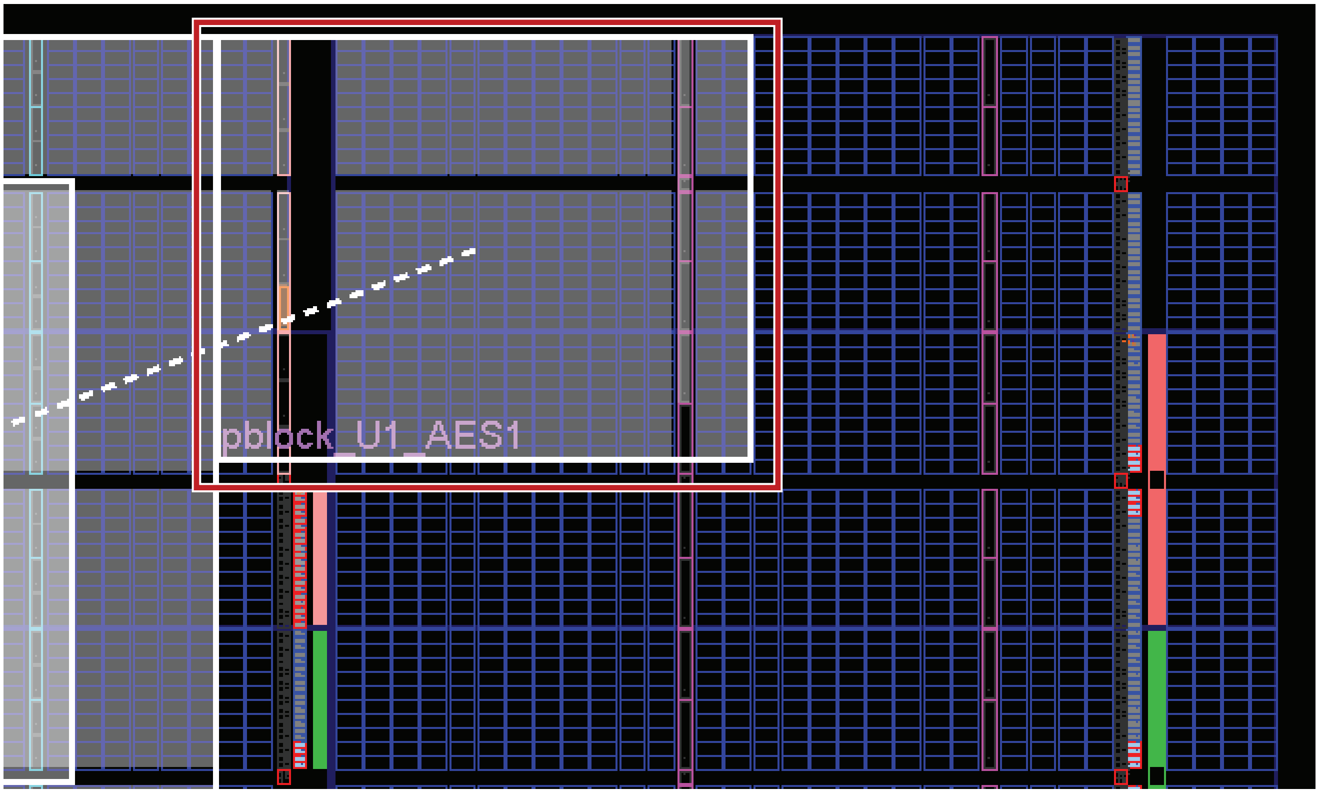
    Y Lo = 1

    X Hi = 63

    Y Hi = 65

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

13. Draw a rectangle in the upper center of the device window, as shown in Figure 3-30. (The rectangle does not have to be 100% accurate—it will be resized shortly.)

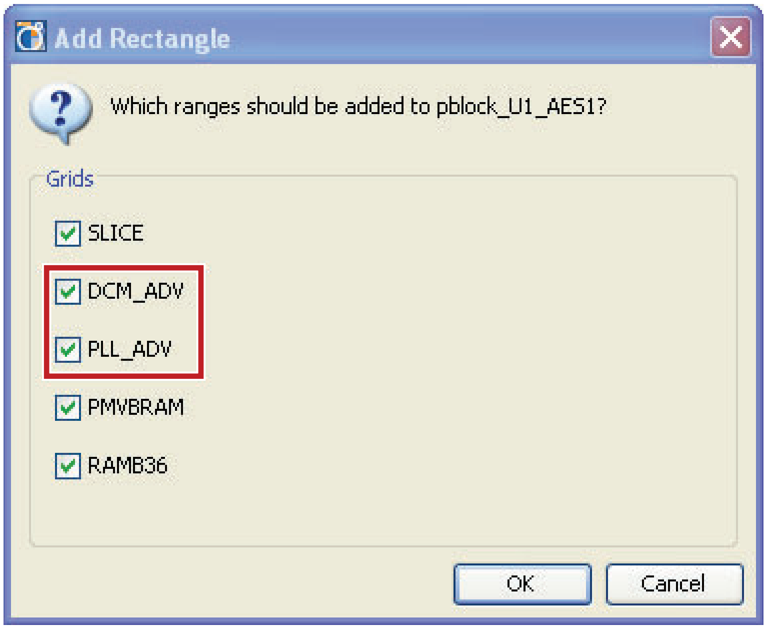


X1105_c3_30_062210

*Figure 3-30:* **Pblock Layout**

14. A dialog box (see Figure 3-31) appears 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

Which ranges should be added to pblock_U1_AES1?

Grids

☑ SLICE

☑ DCM_ADV

☑ PLL_ADV

☑ PMVBRAM

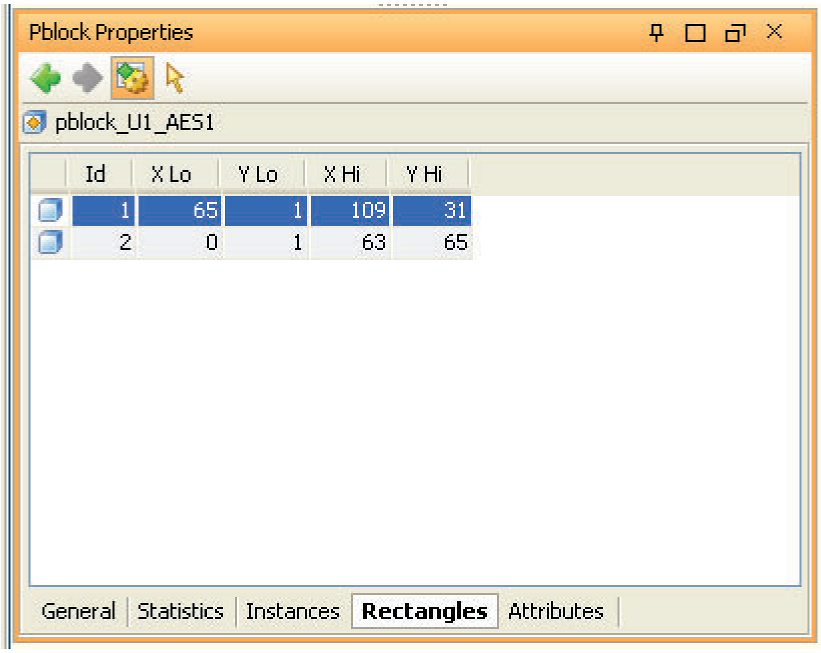
☑ RAMB36

X1105_c3_31_062210

*Figure 3-31:* **Add Rectangle**

15. Click **OK**.
16. Ensure that **pblock_U1_AES1** is selected in the Physical Constraints pane.

17. Select the **Rectangles** tab in the Pblock Properties window (see Figure 3-32).



Pblock Properties

pblock_U1_AES1

| Id | X Lo | Y Lo | X Hi | Y Hi |
|---|---|---|---|---|
| 1 | 65 | 1 | 109 | 31 |
| 2 | 0 | 1 | 63 | 65 |

X1105_c3_32_062210

*Figure 3-32:* **Pblock Properties**

18. Adjust the rectangle graphically as necessary to match these coordinates:

    X Lo = 65

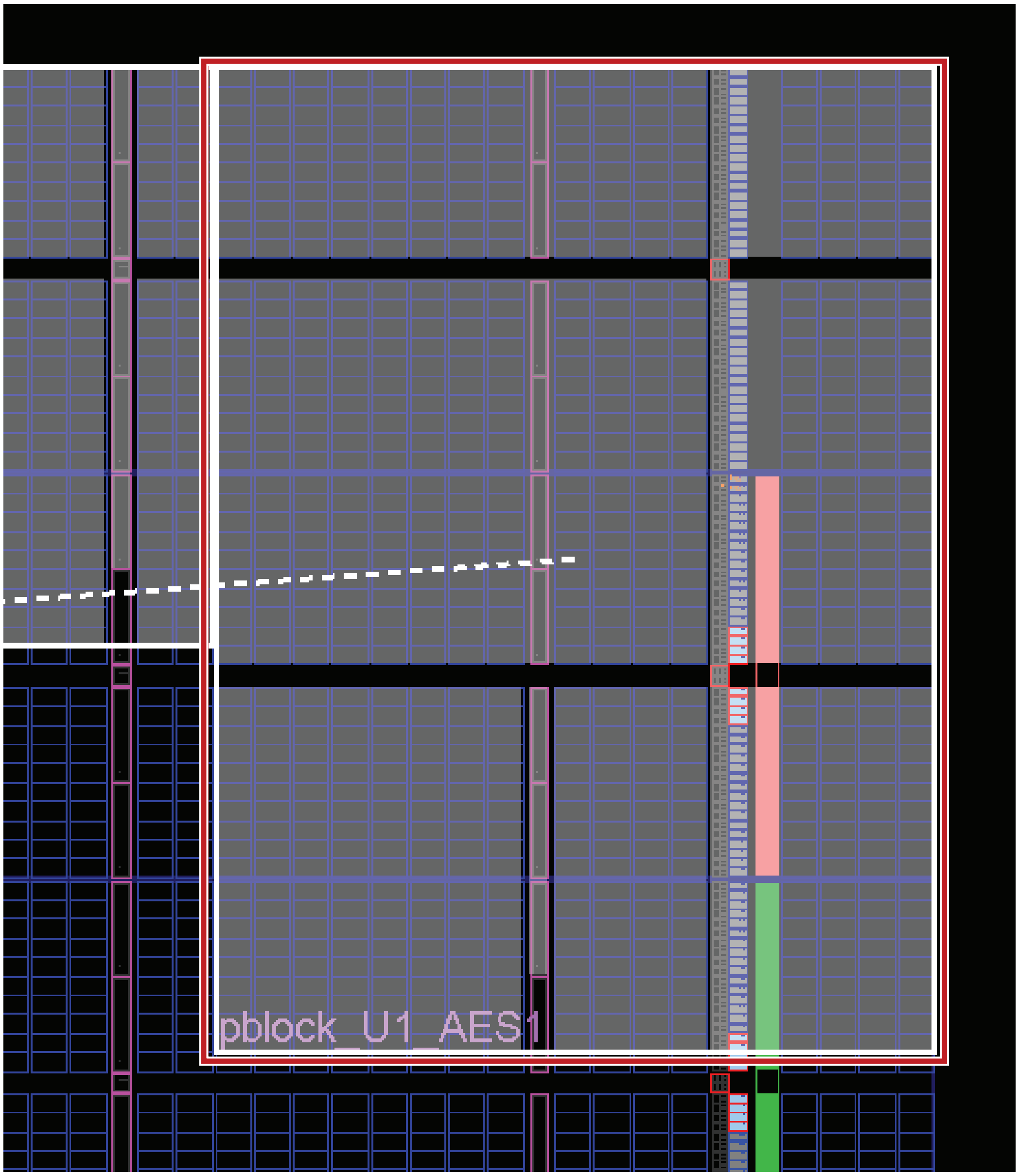
    Y Lo = 1

    X Hi = 109

    Y Hi = 31

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

21. Draw a rectangle in the upper right of the device window, as shown in Figure 3-33. (The rectangle does not have to be 100% accurate—it will be resized shortly.)

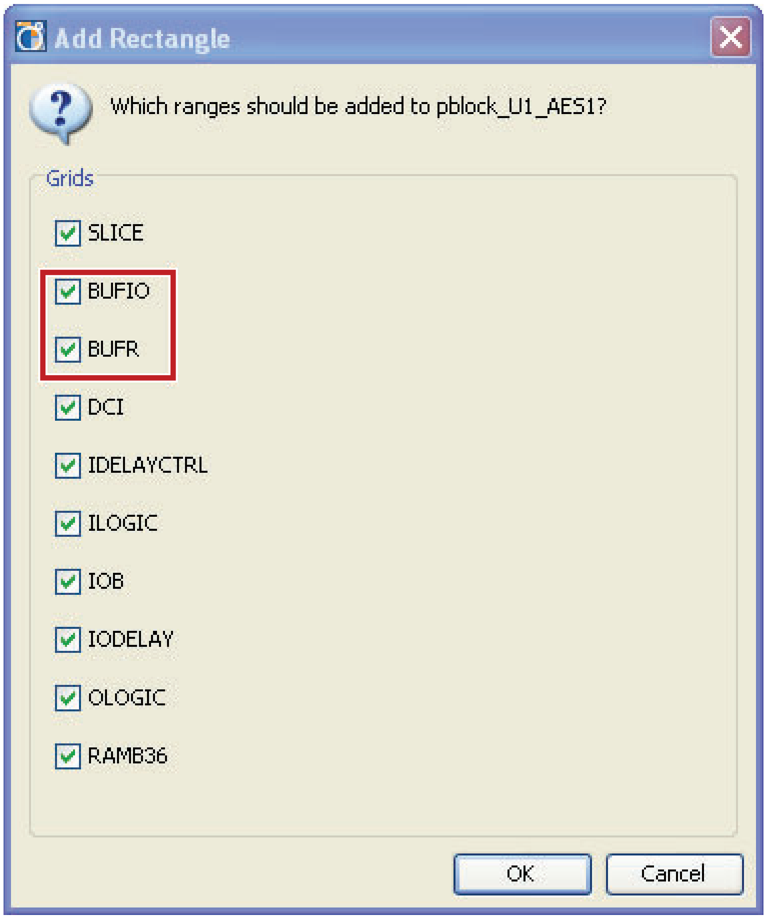


X1105_c3_33_062210

*Figure 3-33:* **Pblock Layout**

22. A dialog box (see Figure 3-34) appears 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

Which ranges should be added to pblock_U1_AES1?

Grids

☑ SLICE

☑ BUFIO

☑ BUFR

☑ DCI

☑ IDELAYCTRL

☑ ILOGIC

☑ IOB

☑ IODELAY

☑ OLOGIC

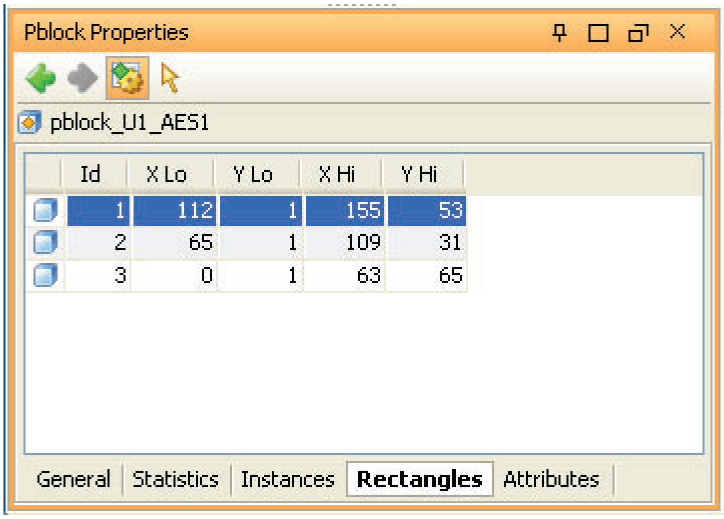
☑ RAMB36

X1105_c3_27_062210

*Figure 3-34:* **Add Rectangle**

23. Click **OK**.
24. Ensure that **pblock_U1_AES1** is selected in the Physical Constraints pane.

25. Select the **Rectangles** tab in the Pblock Properties window (see Figure 3-35).



Pblock Properties

pblock_U1_AES1

| Id | X Lo | Y Lo | X Hi | Y Hi |
|---|---|---|---|---|
| 1 | 112 | 1 | 155 | 53 |
| 2 | 65 | 1 | 109 | 31 |
| 3 | 0 | 1 | 63 | 65 |

X1105_c3_35_062210

*Figure 3-35:* **Pblock Properties**

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

X Lo = 112

Y Lo = 1

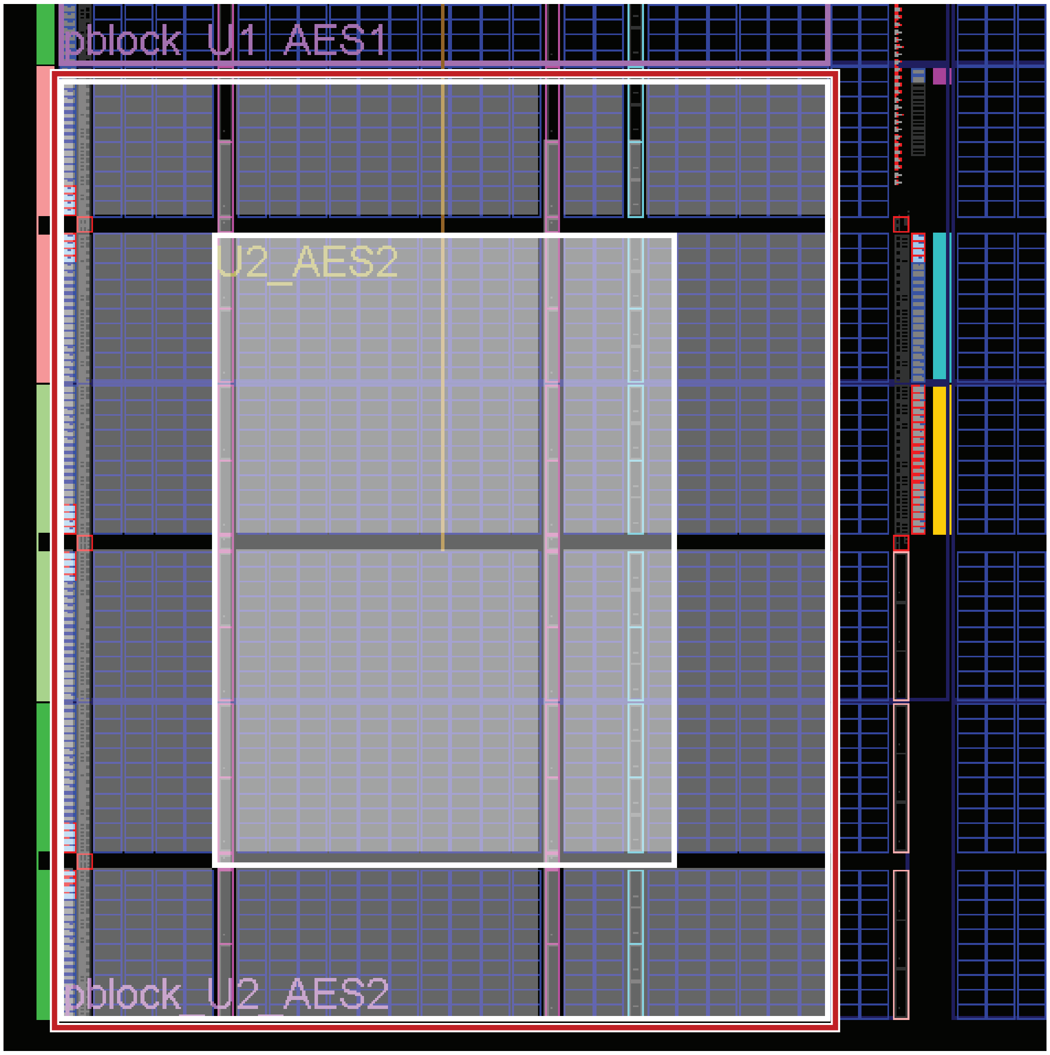
X Hi = 155

Y Hi = 53

27. Under the Physical Constraints 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, as shown in Figure 3-36. (The rectangle does not have to be 100% accurate—it will be resized shortly.)

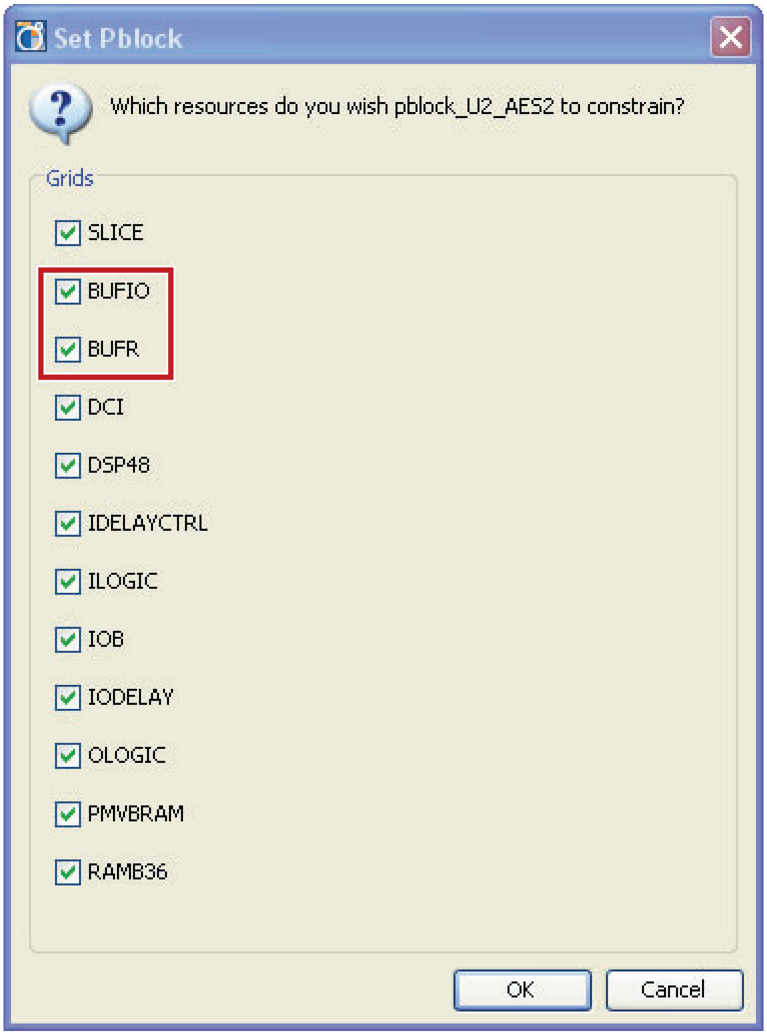


X1105_c3_36_062210

*Figure 3-36:* **Pblock Layout**

30. A dialog box (see Figure 3-37) appears 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

Which resources do you wish pblock_U2_AES2 to constrain?

Grids

☑ SLICE

☑ BUFIO

☑ BUFR

☑ DCI

☑ DSP48

☑ IDELAYCTRL

☑ ILOGIC

☑ IOB

☑ IODELAY

☑ OLOGIC

☑ PMVBRAM

☑ RAMB36

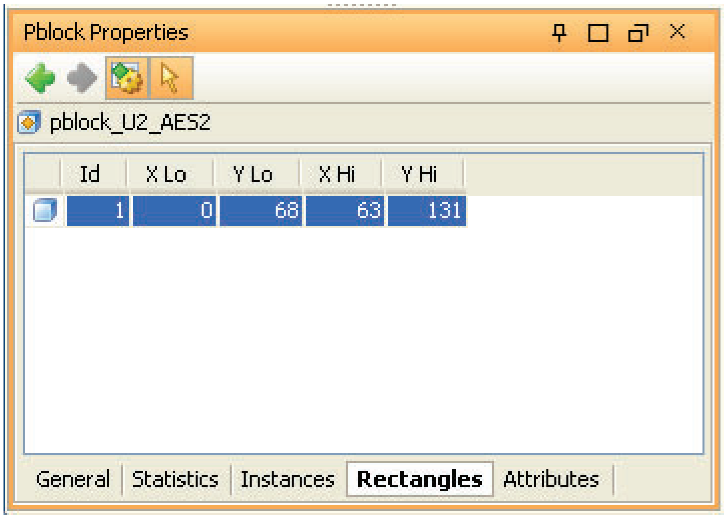
X1105_c3_37_06221 0

*Figure 3-37:* **Set Pblock**

31. Click **OK**.

32. Ensure that **pblock_U2_AES2** is selected in the Physical Constraints pane.

33. Select the **Rectangles** tab in the Pblock Properties window (see Figure 3-38).



| Id | X Lo | Y Lo | X Hi | Y Hi |
|---|---|---|---|---|
| 1 | 0 | 68 | 63 | 131 |

X1105_c3_38_062210

*Figure 3-38:* **Pblock Properties**

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

X Lo = 0

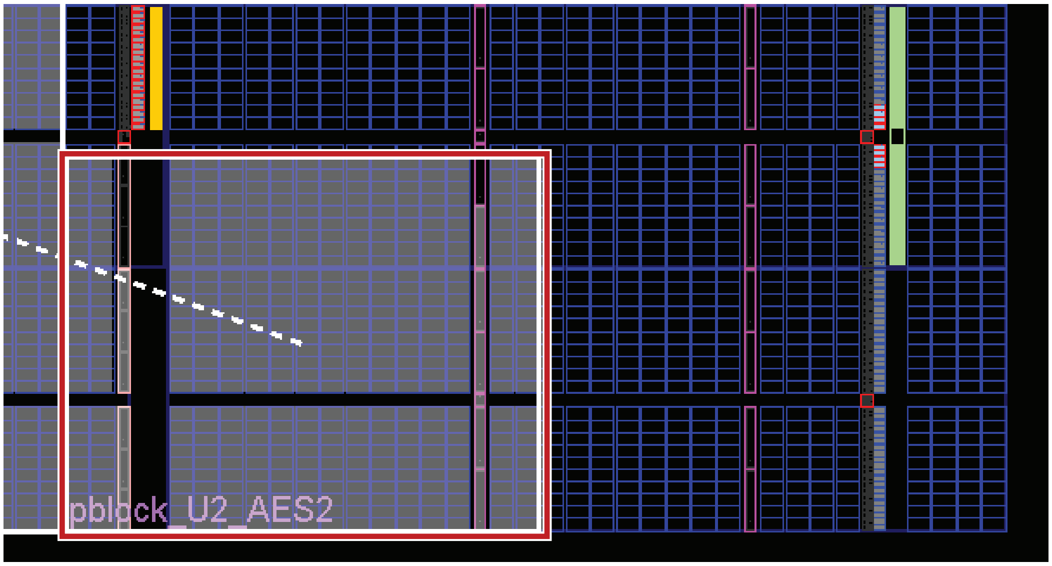
Y Lo = 68

X Hi = 63

Y Hi = 131

35. Under the Physical Constraints tab, select the block **pblock_U2_AES2**.
36. Right-click on **pblock_U2_AES2** and select **Add Pblock Rectangle** from the pull-down menu.

37. Draw a rectangle in the lower center of the device window, as shown in Figure 3-39. (The rectangle does not have to be 100% accurate—it will be resized shortly.)

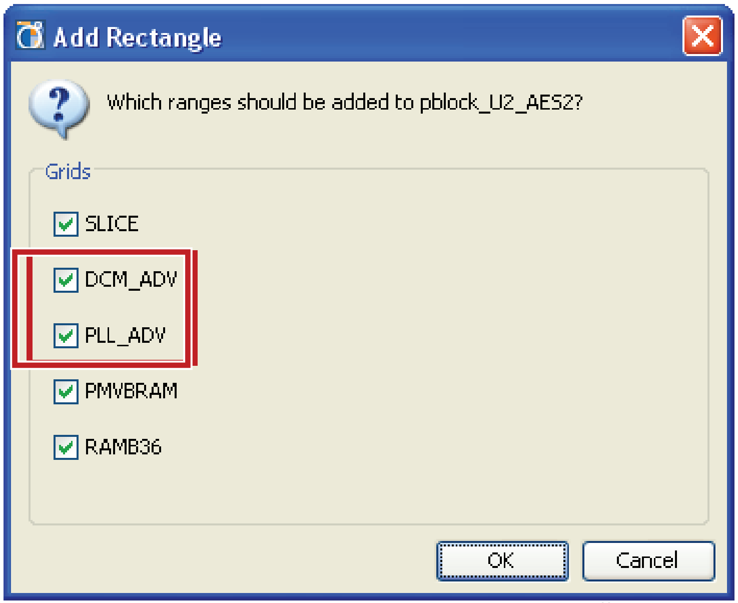


X1105_c3_39_062210

*Figure 3-39:* **Pblock Layout**

38. A dialog box (see Figure 3-40) appears 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.



Which ranges should be added to pblock_U2_AES2?

Grids

☑ SLICE

☑ DCM_ADV

☑ PLL_ADV

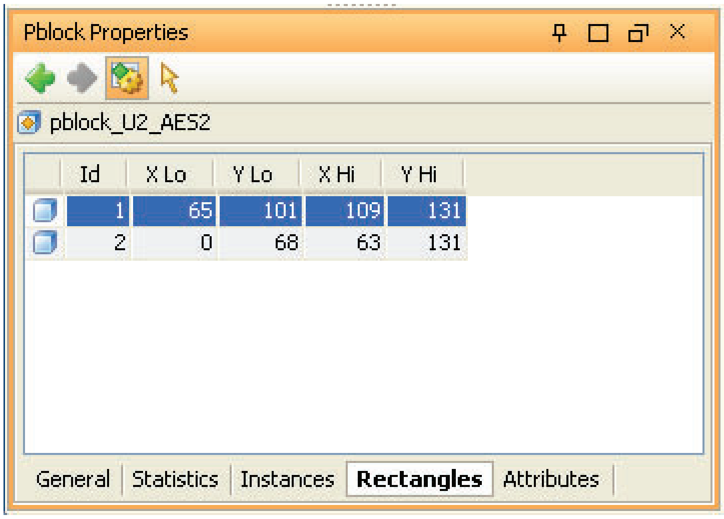
☑ PMVBRAM

☑ RAMB36

X1105_c3_40_062510

*Figure 3-40:* **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 Pblock Properties window (see Figure 3-41).



| Id | X Lo | Y Lo | X Hi | Y Hi |
|---|---|---|---|---|
| 1 | 65 | 101 | 109 | 131 |
| 2 | 0 | 68 | 63 | 131 |

X1105_c3_41_062210

*Figure 3-41:* **Pblock Properties**

42. Adjust the rectangle graphically as necessary to match these coordinates:

    X Lo = 65

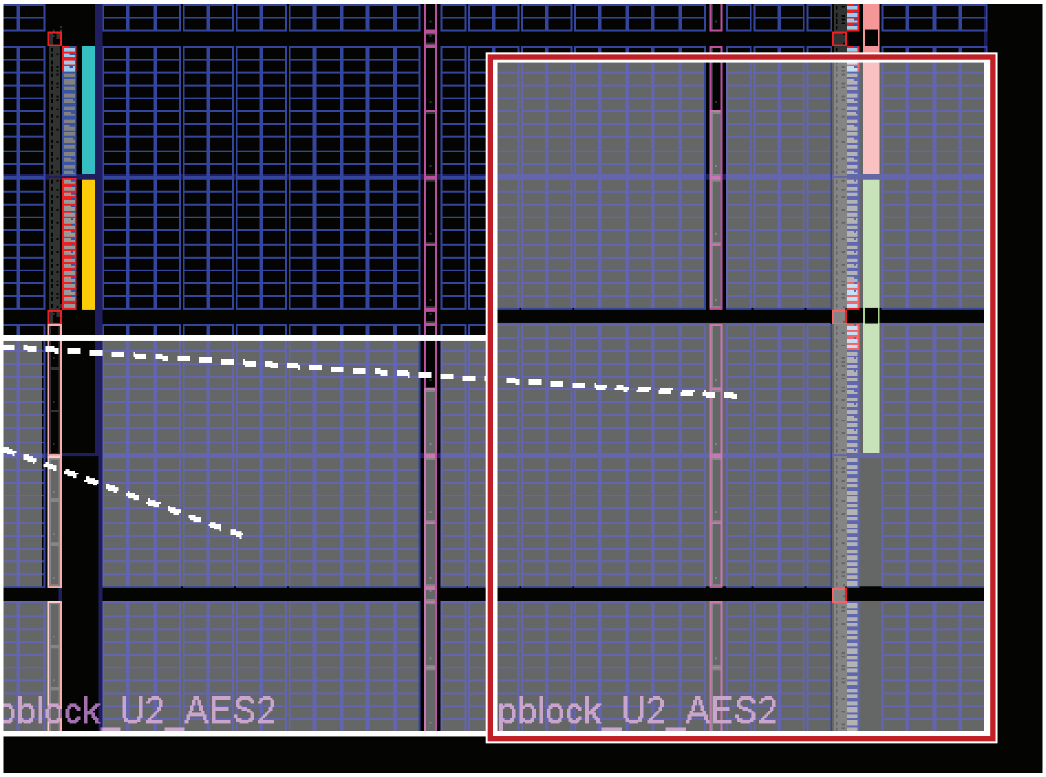
    Y Lo = 101

    X Hi = 109

    Y Hi = 131

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

45. Draw a rectangle in the lower right of the device window, as shown in Figure 3-42. (The rectangle does not have to be 100% accurate—it will be resized shortly.)

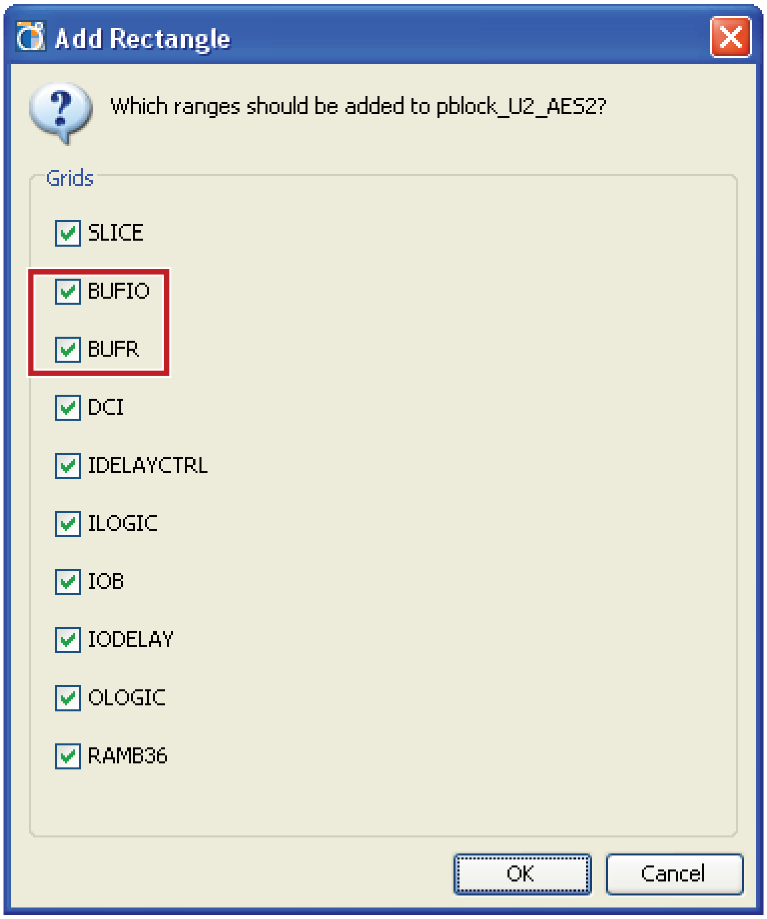


X1105_c3_42_062210

*Figure 3-42:* **Pblock Layout**

46. A dialog box (see Figure 3-43) appears 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

Which ranges should be added to pblock_U2_AES2?

Grids

☑ SLICE

☑ BUFIO

☑ BUFR

☑ DCI

☑ IDELAYCTRL

☑ ILOGIC

☑ IOB

☑ IODELAY

☑ OLOGIC

☑ RAMB36

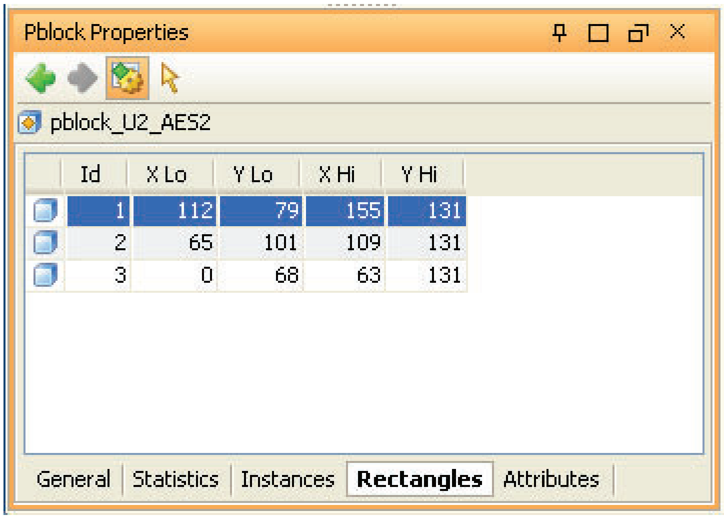
X1105_c3_43_062210

*Figure 3-43:* **Add Rectangle**

47. Click **OK**.

48. Ensure that **pblock_U2_AES2** is selected in the Physical Constraints pane.

49. Select the **Rectangles** tab in the Pblock Properties window (see Figure 3-44).



Pblock Properties

pblock_U2_AES2

| Id | X Lo | Y Lo | X Hi | Y Hi |
|---|---|---|---|---|
| 1 | 112 | 79 | 155 | 131 |
| 2 | 65 | 101 | 109 | 131 |
| 3 | 0 | 68 | 63 | 131 |

X1105_c3_44_062210

*Figure 3-44:* **Pblock Properties**

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

    X Lo =112

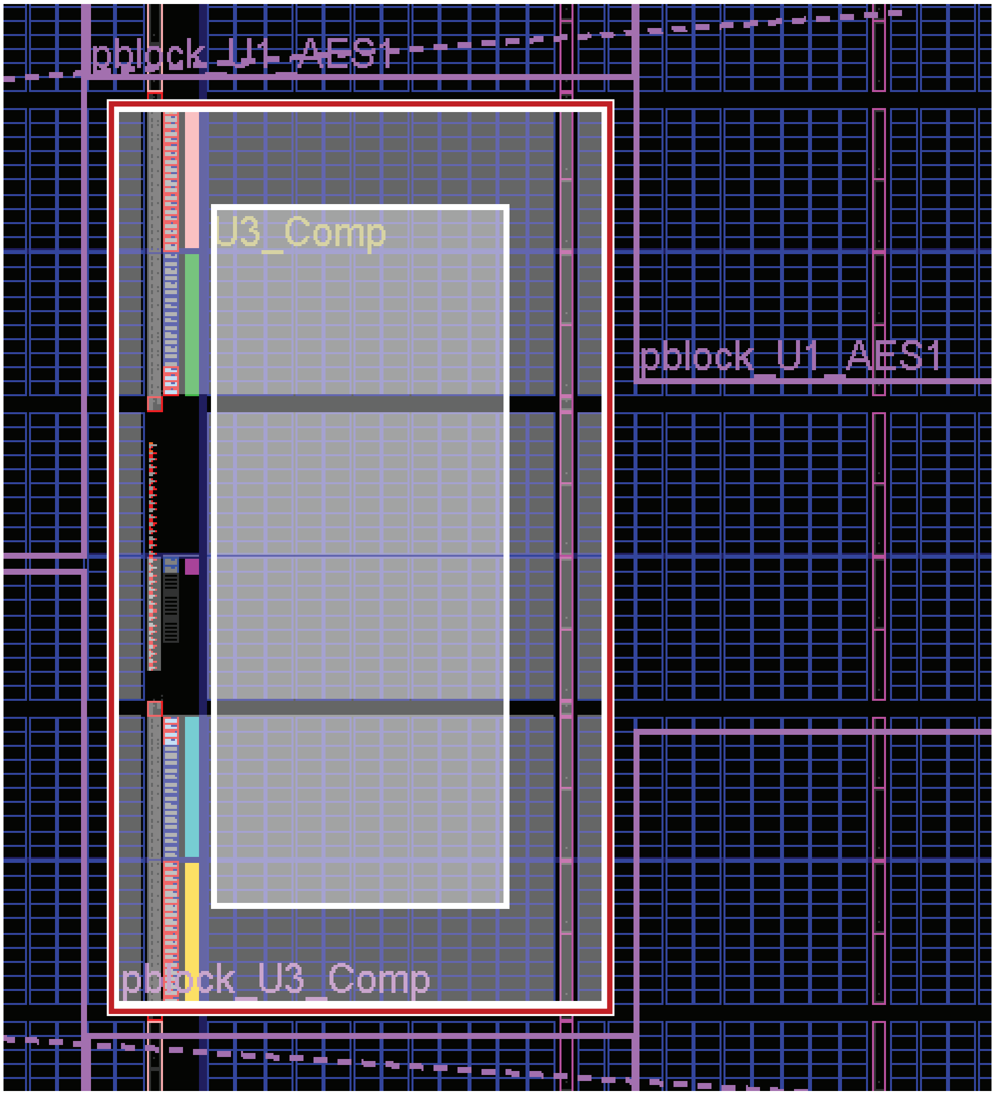
    Y Lo = 79

    X Hi = 155

    Y Hi = 131

51. Under the Physical Constraints 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, as shown in Figure 3-45. (The rectangle does not have to be 100% accurate—it will be resized shortly.)

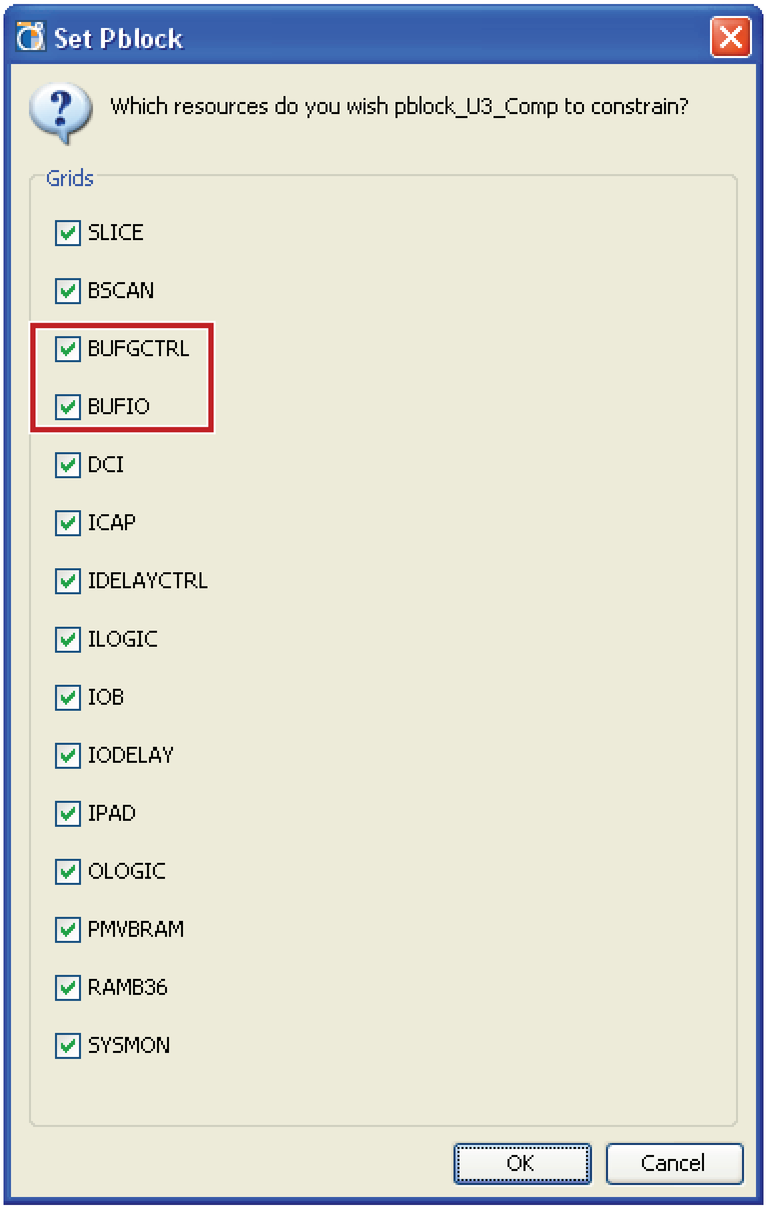


X1105_c3_45_062210

*Figure 3-45:* **Pblock Layout**

54. A dialog box (see Figure 3-46) appears 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

Which resources do you wish pblock_U3_Comp to constrain?

Grids

☑ SLICE
☑ BSCAN
☑ BUFGCTRL
☑ BUFIO
☑ DCI
☑ ICAP
☑ IDELAYCTRL
☑ ILOGIC
☑ IOB
☑ IODELAY
☑ IPAD
☑ OLOGIC
☑ PMVBRAM
☑ RAMB36
☑ SYSMON

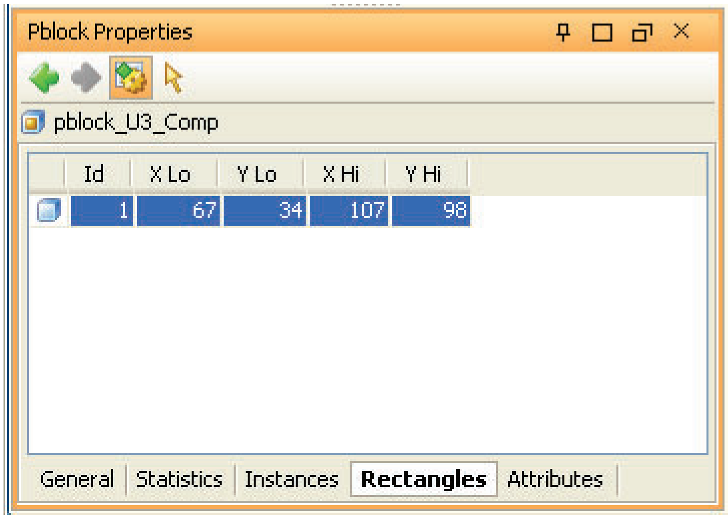
X1105_c3_46_062210

*Figure 3-46:* **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 Constraints pane.
59. Select the **Rectangles** tab in the Pblock Properties window (see Figure 3-47).



pblock_U3_Comp

| Id | X Lo | Y Lo | X Hi | Y Hi |
|---|---|---|---|---|
| 1 | 67 | 34 | 107 | 98 |

X1105_c3_47_062210

*Figure 3-47:* **Pblock Properties**

60. Adjust the rectangle graphically as necessary to match these coordinates:

    X Lo = 67

    Y Lo = 34

    X Hi = 107

    Y Hi = 98

61. Under the Physical Constraints 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, as shown in Figure 3-48. (The rectangle does not have to be 100% accurate—it will be resized shortly.)

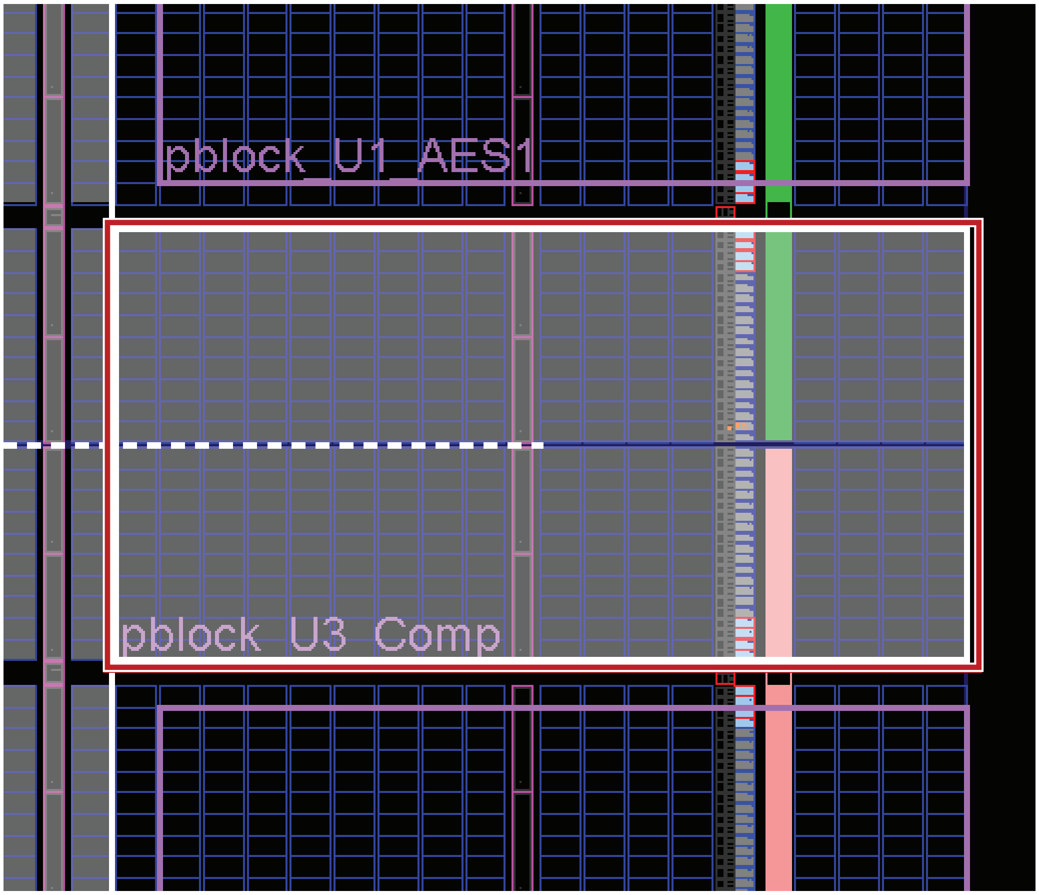
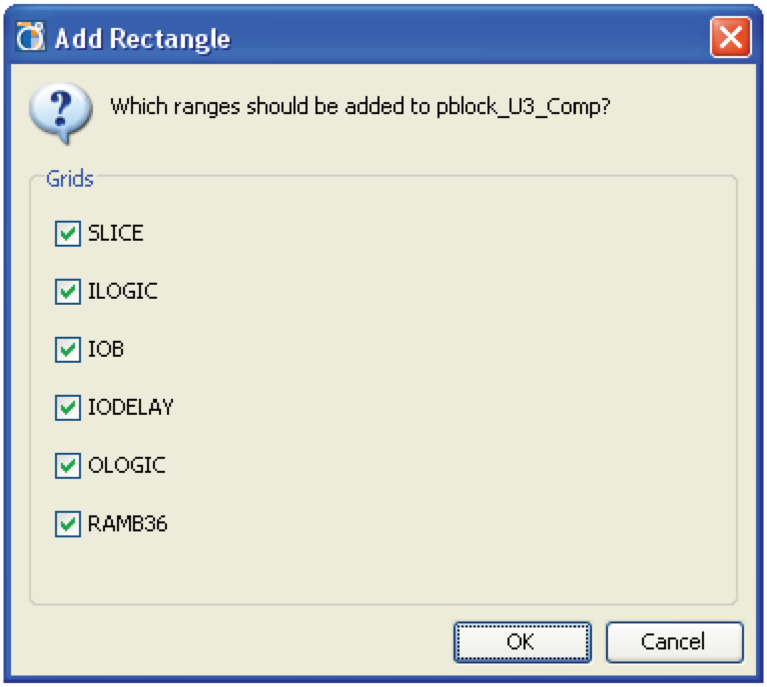


*Figure 3-48:* **Pblock Layout**

64. A dialog box (see Figure 3-49) appears 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

Which ranges should be added to pblock_U3_Comp?

Grids

☑ SLICE

☑ ILOGIC

☑ IOB

☑ IODELAY

☑ OLOGIC

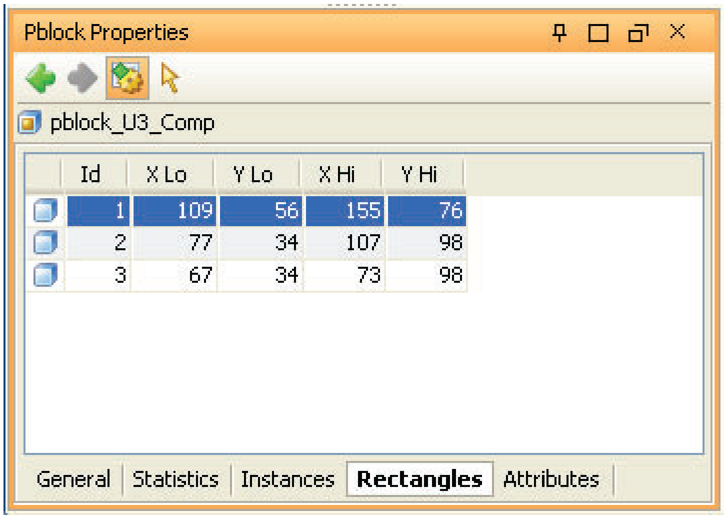
☑ RAMB36

X1105_c3_49_062210

*Figure 3-49:* **Add Rectangle**

65. Click **OK**.
66. Ensure that **pblock_U3_Comp** is selected in the Physical Constraints pane.

67. Select the **Rectangles** tab in the Pblock Properties window (see Figure 3-50).



pblock_U3_Comp

| Id | X Lo | Y Lo | X Hi | Y Hi |
|---|---|---|---|---|
| 1 | 109 | 56 | 155 | 76 |
| 2 | 77 | 34 | 107 | 98 |
| 3 | 67 | 34 | 73 | 98 |

X1105_c3_50_062210

*Figure 3-50:* **Pblock Properties**

68. Adjust the rectangle graphically as necessary to match these coordinates:

X Lo = 109

Y Lo = 56

X Hi = 155

Y Hi = 76

69. The final layout is shown in Figure 3-51. 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.

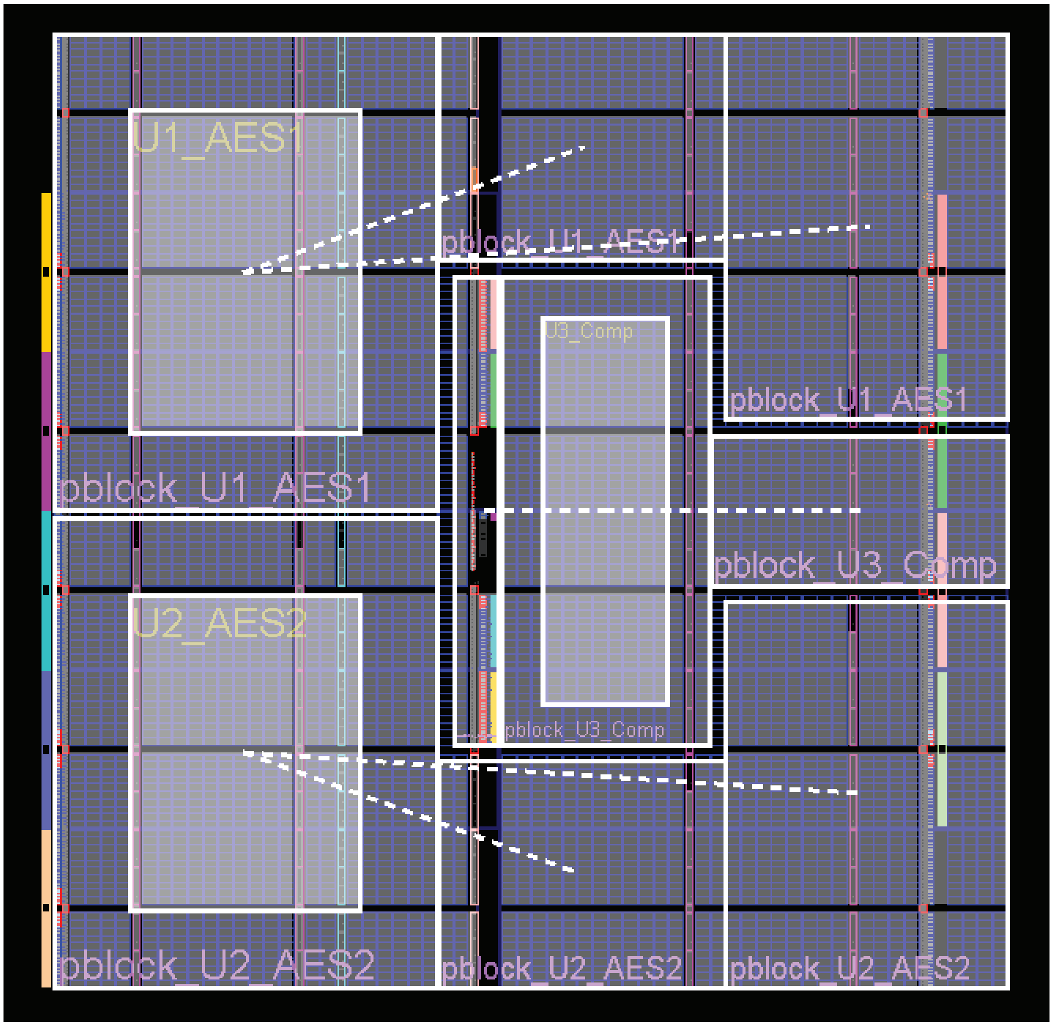


*Figure 3-51:* **Final Layout**

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

# Design Flow Progress

The System Floorplanning block of the SCC system design flow diagram is complete, as shown in Figure 3-52.

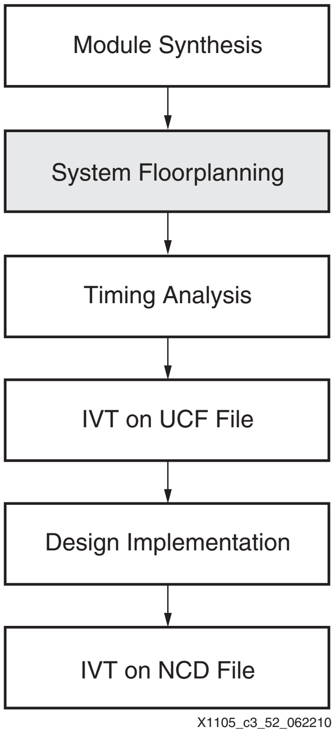
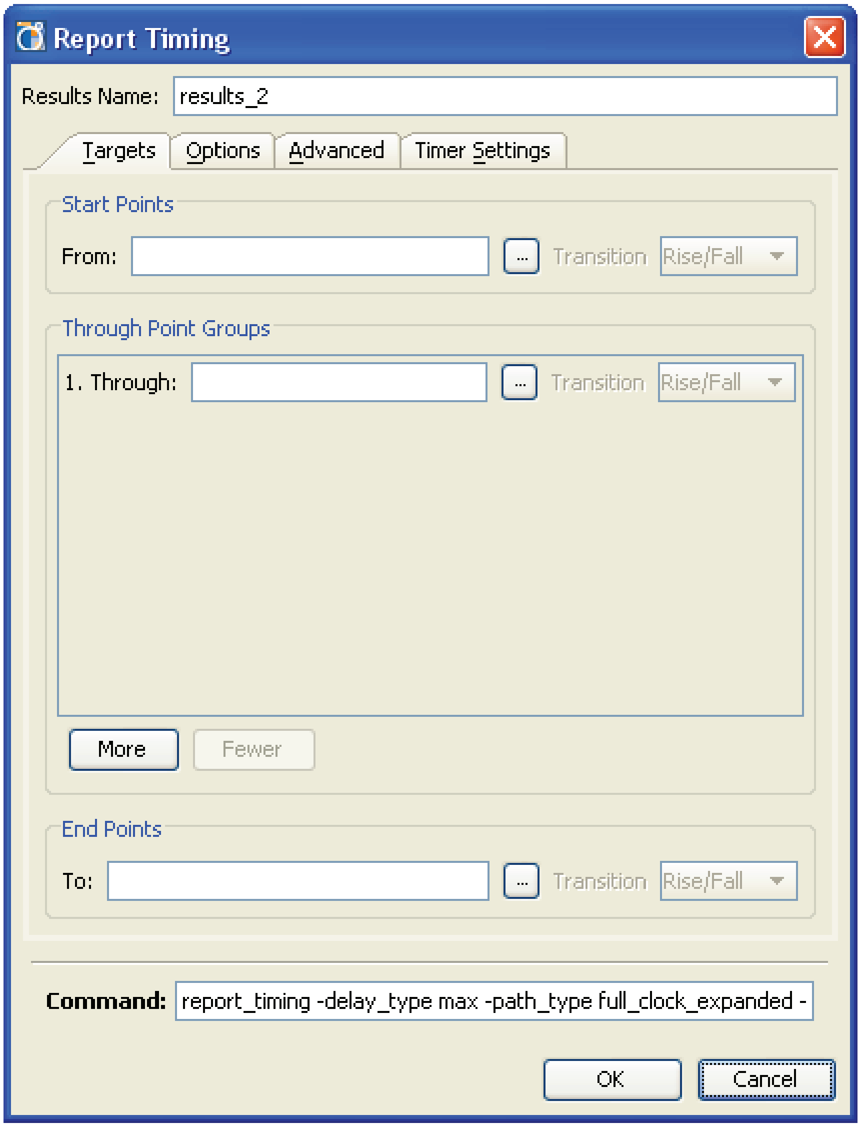


*Figure 3-52:* **SCC System Design Flow with System Floorplanning Block Complete**

# Running Report Timing with the PlanAhead Tool

This section tests the timing constraints that were set up in Setting Up Timing Constraints with the PlanAhead Tool, page 63.

1. Select **Tools** → **Report Timing** and click **OK** (see Figure 3-53).


Results Name: results_2

Command: report_timing -delay_type max -path_type full_clock_expanded -

X1105_c3_53_062210

*Figure 3-53:* **Report Timing 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 Timing Analysis block of the SCC system design flow diagram is complete, as shown in Figure 3-54.

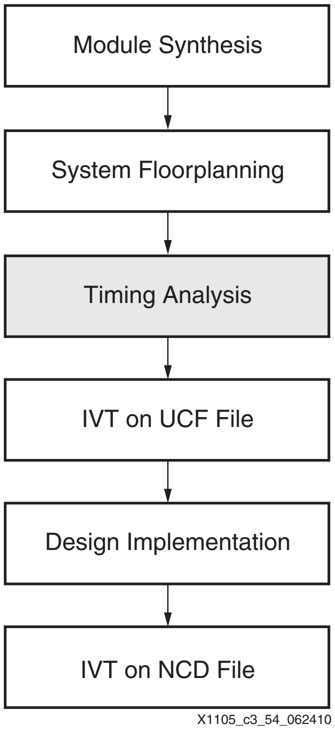


*Figure 3-54:* **SCC System Design Flow with Timing Analysis Block Complete**

# 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 UCF, or both, if desired. For this lab, only the UCF need be exported for checking by the Isolation Verification Tool.

1. To export the netlist, select **File** → **Export Netlist**.
2. Browse to the `..\Xilinx_Design\PlanAhead` directory and keep the default filename `FloorPlan_SCC.edf`.
3. Click **OK**.
4. To export the constraints file, select **File** → **Export Constraints** and browse to the desired location or choose the default location `..\Xilinx_Design\PlanAhead\FloorPlan_SCC\SCC_Lab_1.ucf`.
5. Click **OK**.

# 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 native circuit description (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. The `ivt.zip` file posted on the Isolation Design Flow page contains the files needed for IVT:
   - `ivt.exe` – the IVT executable for IVT
   - `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 these contents:

```
-device xc5vlx85 -package ff676

# Groups                 Isolation Group            Area Group
# -------              --------------------       ---------------------
-group                  AES                       pblock_U1_AES1
-group                  AES_r                     pblock_U2_AES2
-group                  COMPARE                   pblock_U3_Comp

# Pin Isolation Groups
-pig SCC-LAB.pig

# User Constraint File
..\PlanAhead\Floorplan_SCC\SCC_LAB.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 points to the UCF.

e. IVT is told where to place the output file and its 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 or modify a text file and name the file `SCC-LAB.pig` with these 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 as a starting point. 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** → **Run** → **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 is "`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. A sample of this file can be found in the `../ivt` directory.

## 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
   ```

## Design Flow Progress

The IVT on UCF File block of the SCC system design flow diagram is complete, as shown in Figure 4-1.

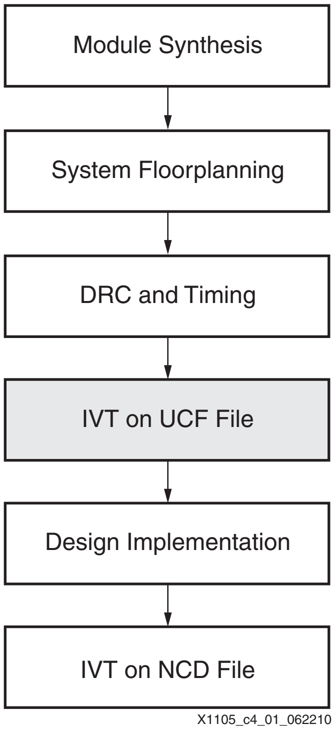


*Figure 4-1:* **SCC System Design Flow with IVT on UCF File Block Complete**

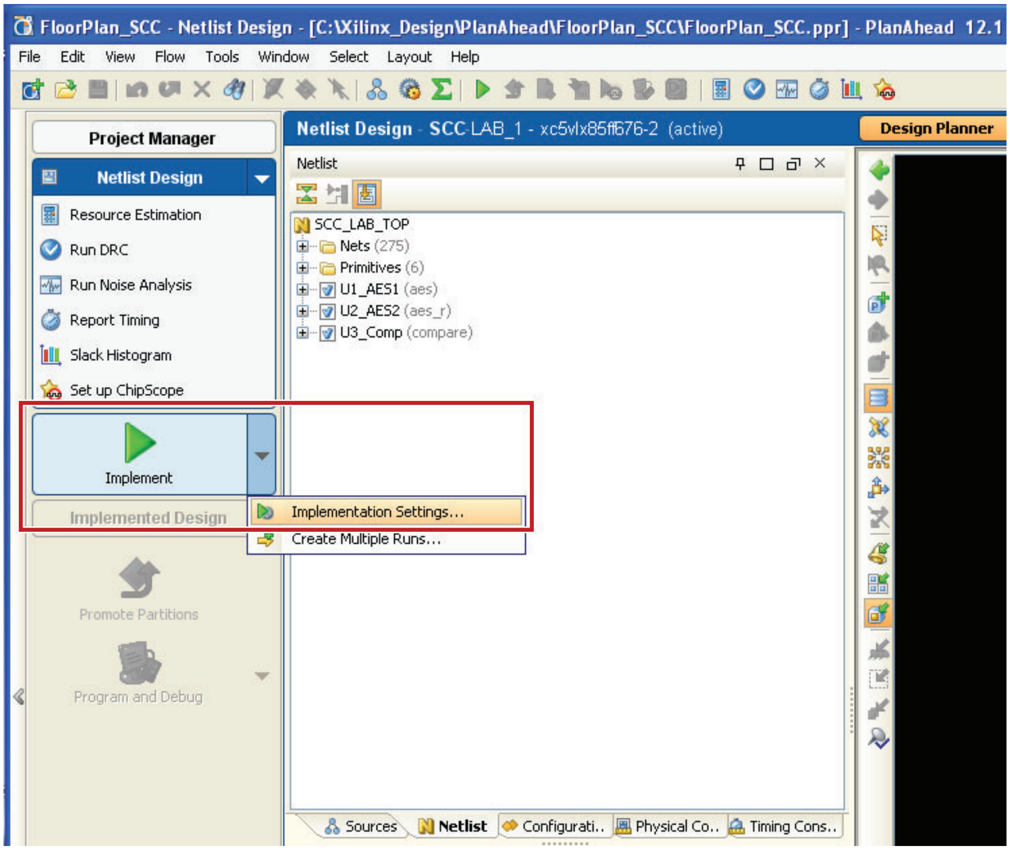
*Chapter 5*

# *Implementing the Design with the PlanAhead Tool*

## Generating and Running an Implementation

These steps describe how to generate and run a design implementation[1]:

1. From the Project Manager pane in the PlanAhead™ tool:
   a. Select **Implement** → **Implementation Settings...** (see Figure 5-1).

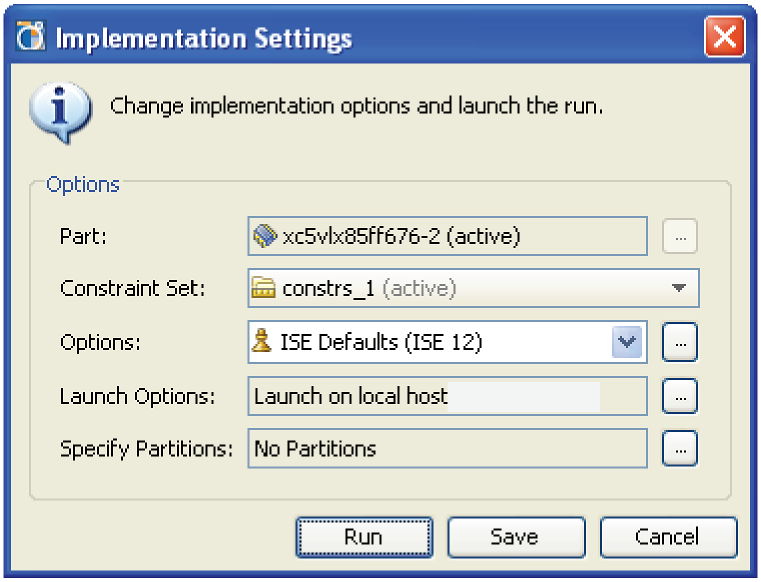


X1105_c5_01_062310

*Figure 5-1:* **Implement Design (Implementation Settings)**

[1] To implement a design using the PlanAhead tool 12.1, refer to the process detailed in Appendix A, Modified Implementation of the PlanAhead Tool 12.1.

b. An Implementation Settings dialog box appears (see Figure 5-2). To launch a run, accept the defaults and click **Run**.



Implementation Settings

Change implementation options and launch the run.

Options

Part: xc5vlx85ff676-2 (active)

Constraint Set: constrs_1 (active)

Options: ISE Defaults (ISE 12)

Launch Options: Launch on local host

Specify Partitions: No Partitions

X1105_c5_02_062210

*Figure 5-2:* **Implementation Settings**

2. A new tab named Design Runs is created, and a single entry named config_1 is generated. It immediately starts the run: **NGDBUILD** → **MAP** → **Place and Route**.
3. An Implementation Completed dialog box appears after the design run is completed. Select the **Open Implemented Design** option and click **OK**.

4. The Device tab in the Design Planner pane shows the placed, routed, and partitioned design (see Figure 5-3).

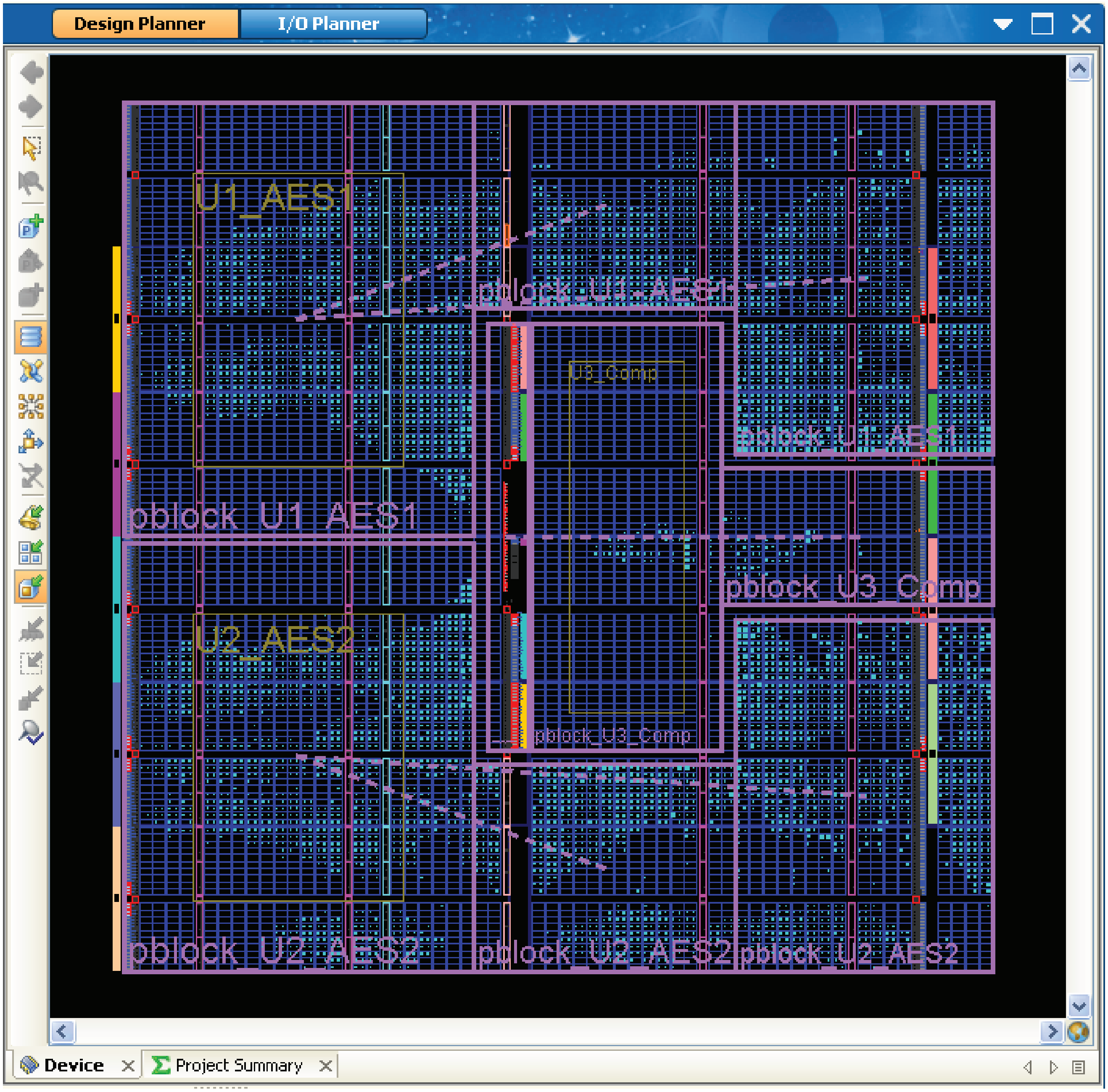
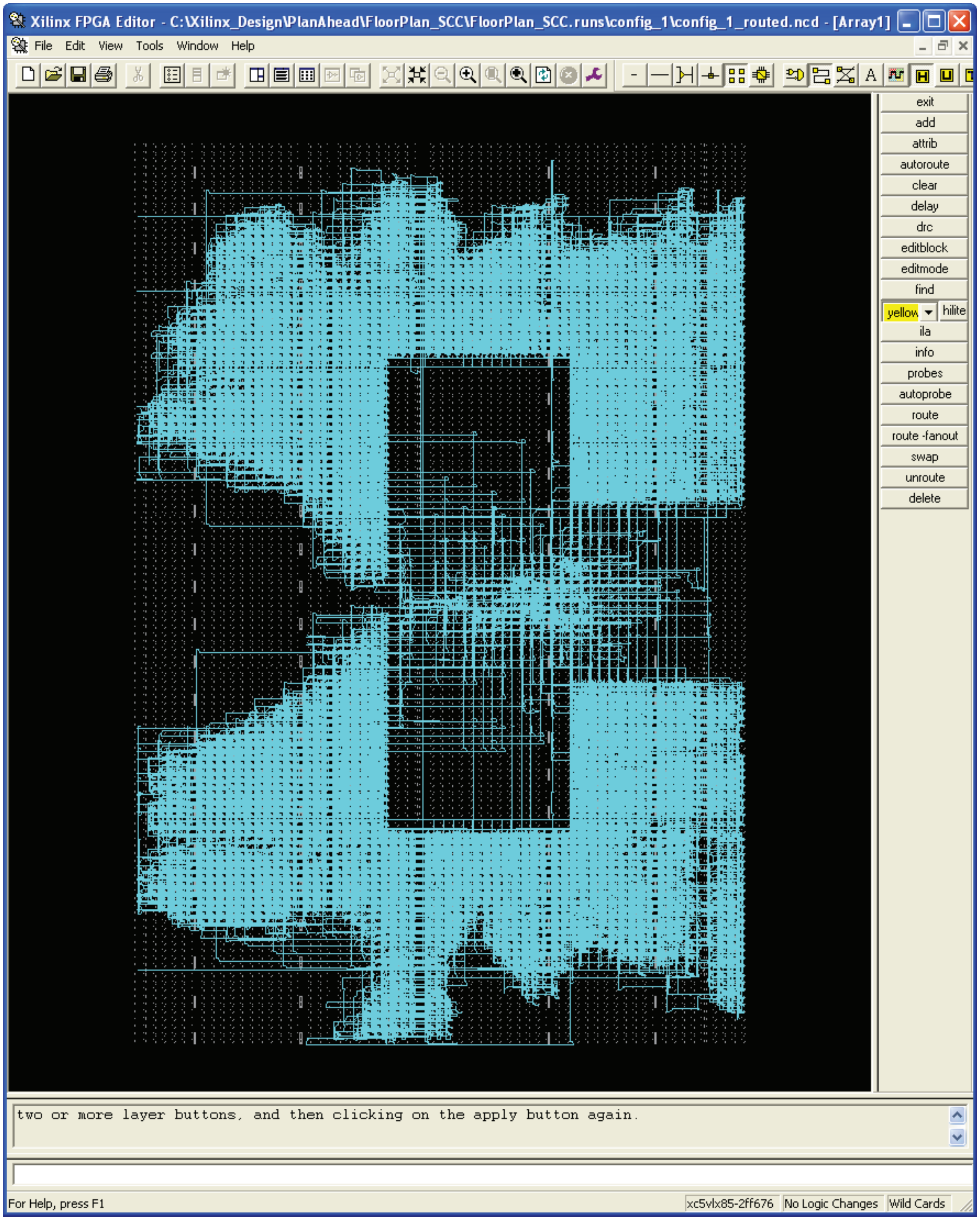


*Figure 5-3:* **Implemented and Floorplanned Design**

5. The combined and routed design is placed in this directory:

   `..\Xilinx_Design\PlanAhead\FloorPlan_SCC\FloorPlan_SCC.runs\config_1`

   The file name is `config_1_routed.ncd` and its view in FPGA Editor is shown in Figure 5-4.



```
two or more layer buttons, and then clicking on the apply button again.
```

X1105_c5_04_062210

*Figure 5-4:* **FPGA Editor View**

# Design Flow Progress

The Design Implementation block of the SCC system design flow diagram is complete, as shown in Figure 5-5.

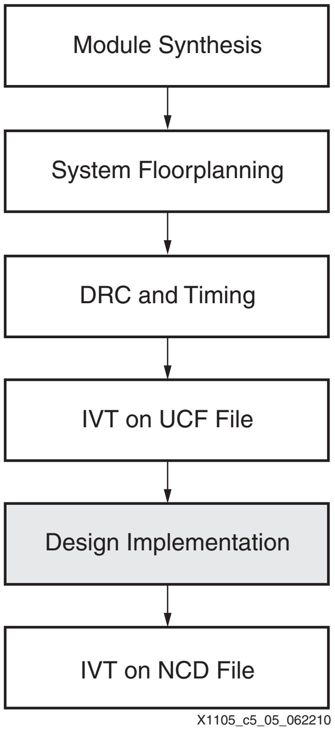


*Figure 5-5:* **SCC System Design Flow with Design Implementation Block Complete**

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:

1. Open the directory `\Xilinx_Design\ivt\`.
2. Create a new text file in the `\Xilinx_Design\ivt\` directory:

   `SCC-LAB_ncd.ivt`

3. Open the text file `SCC-LAB_ncd.ivt` in a text editor.
4. Add these lines to the `SCC-LAB_ncd.ivt` text test file.

```
# comment the next line for reduced detail in the report file.
-verbose
# Groups    Isolation Group     Instance Name in Final NCD
# --------  ----------------    --------------------------
-group       AES                 U1_AES1
-group       AES_r               U2_AES2
-group       COMPARE             U3_Comp

# Combined design
..\PlanAhead\FloorPlan_SCC\FloorPlan_SCC.runs\impl\floorplans\
fp_v5lx85_ff676-2_1\impl_1\impl_1_routed.ncd

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

The NCD IVT command file sets these 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** → **Run** → **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 is "`SUCCESS!`".

# Examining the Output from the IVT NCD Test

IVT creates two types of output files. The standard RPT file is a text report file. The SVG output file is a graphical view of the Virtex®-5 device with colored tiles denoting the ownership of the tiles by each isolated or PR area.

## IVT RPT Output File

These steps describe how to read the various sections of the RPT 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. In the Identified Networks section, ensure that 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, where applicable. All networks passing through Bus macros are considered safe by IVT and are listed in this section.
5. Ensure that only bus macros are listed in the bus macro usage summary.
6. Pay special attention to the package pins, I/O buffers, and I/O banks because the note states that 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.

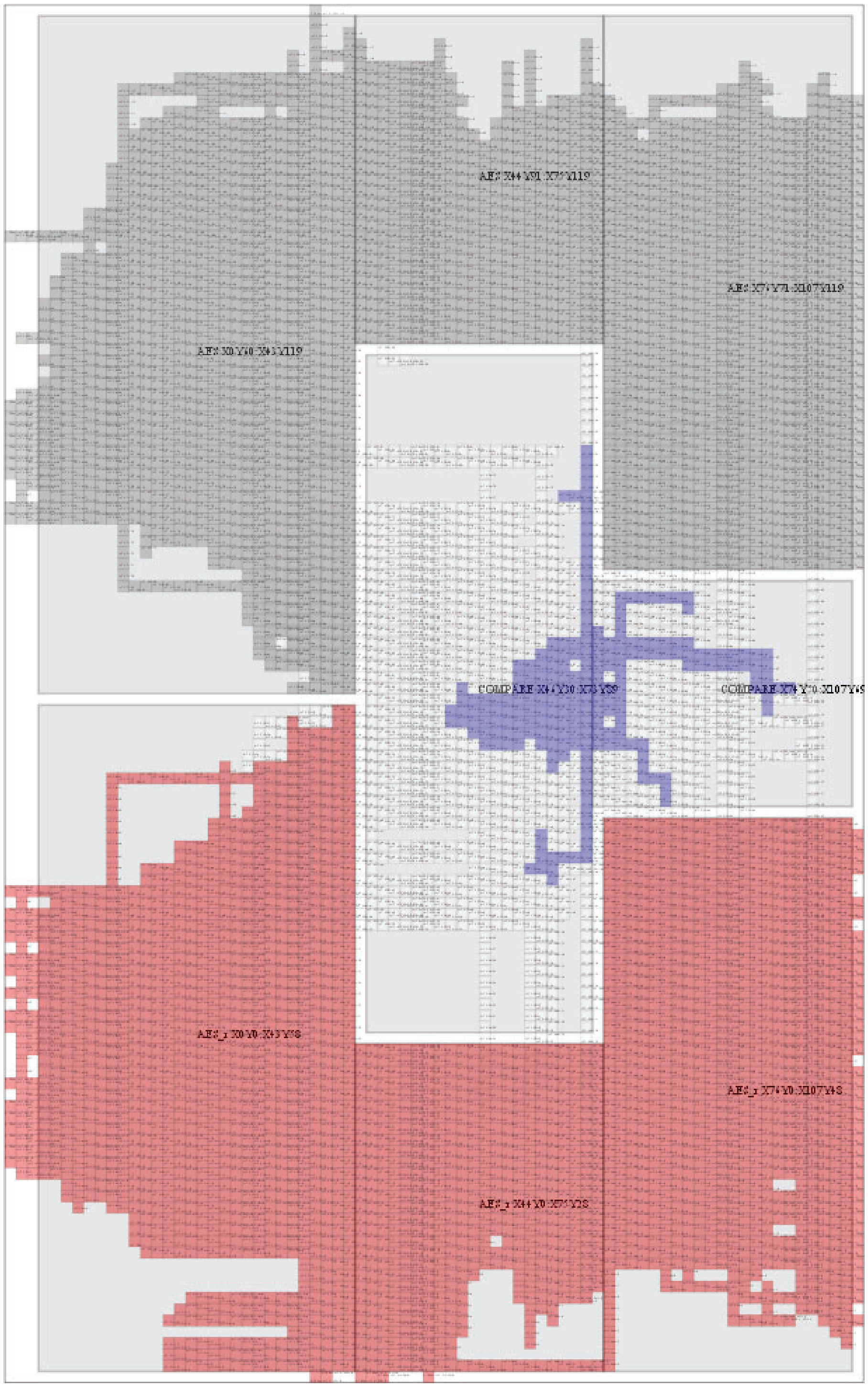
```
Pin(col, row)   Bank  I/O Buffer    Isolation Group          Network
-------------   ----  ------------  ---------------   ------------------------
D10( 16,  22)    16     IOB_X2Y198      AES           U1_AES1/push_button_IBUF
D11( 15,  22)    16     IOB_X2Y199      AES           U1_AES1/reset_out
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

Isolation Verification Summary

  Design contains 0 constraint violations.

  Isolation analysis completed.

  Elapsed time: 0:03:35
```

## IVT SVG Output File

The SVG file created by IVT gives a graphical view of the Virtex-5 device with colored tiles denoting the ownership of the tiles by each isolated or PR region. The SVG file also visually highlights that there is a proper fence isolating each of the regions (uncolored tiles). Figure 6-1 shows the SVG output for the Virtex-5 FPGA design used in the SCC lab.



X1105_c6_01_062510

*Figure 6-1:* **IVT SVG File Graphical Output**

The separate isolated regions can be easily recognized in the SVG graphical output and compared to the FPGA Editor screen capture shown in Figure 1-4, page 9, or the PlanAhead™ tool floorplan screen capture shown in Figure 5-3, page 107. The SVG format was chosen because it is a standard format that can be read by a web browser (such as Microsoft Internet Explorer or Mozilla Firefox) or by Microsoft Visio.

## Design Flow Progress

The SCC lab is complete with the IVT on NCD File block of the flow diagram, as shown in Figure 6-2.

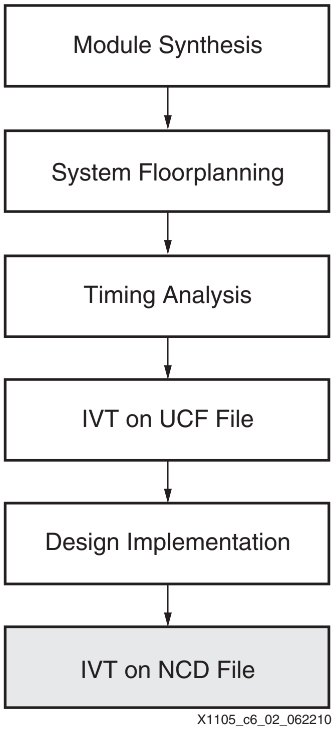


*Figure 6-2:* **SCC System Design Flow with IVT on NCD File Block Complete**

# Modified Implementation of the PlanAhead Tool 12.1

Due to a software issue in the PlanAhead™ tool 12.1, a workaround is required for modifying the UCF and then running implementation. The PlanAhead tool, which is used to generate the batch file for implementation, erroneously puts three SCC_ISOLATED instances into the generated UCF file during implementation. The user thus has to modify the UCF file generated by the PlanAhead tool. The generated batch script is run to implement the design. (The design should not be implemented using the PlanAhead tool because the UCF file is regenerated with the SCC_ISOLATED instances put back in, and implementation thus fails at ngdbuild.)

# Implementing the Design Using the PlanAhead Tool 12.1 and the Generated Batch Script File

This section describes the steps required to implement the reference design using the PlanAhead tool 12.1.

1. From the Project Manager pane in the PlanAhead tool, select **Implement** → **Implementation Settings...** (see Figure A-1).

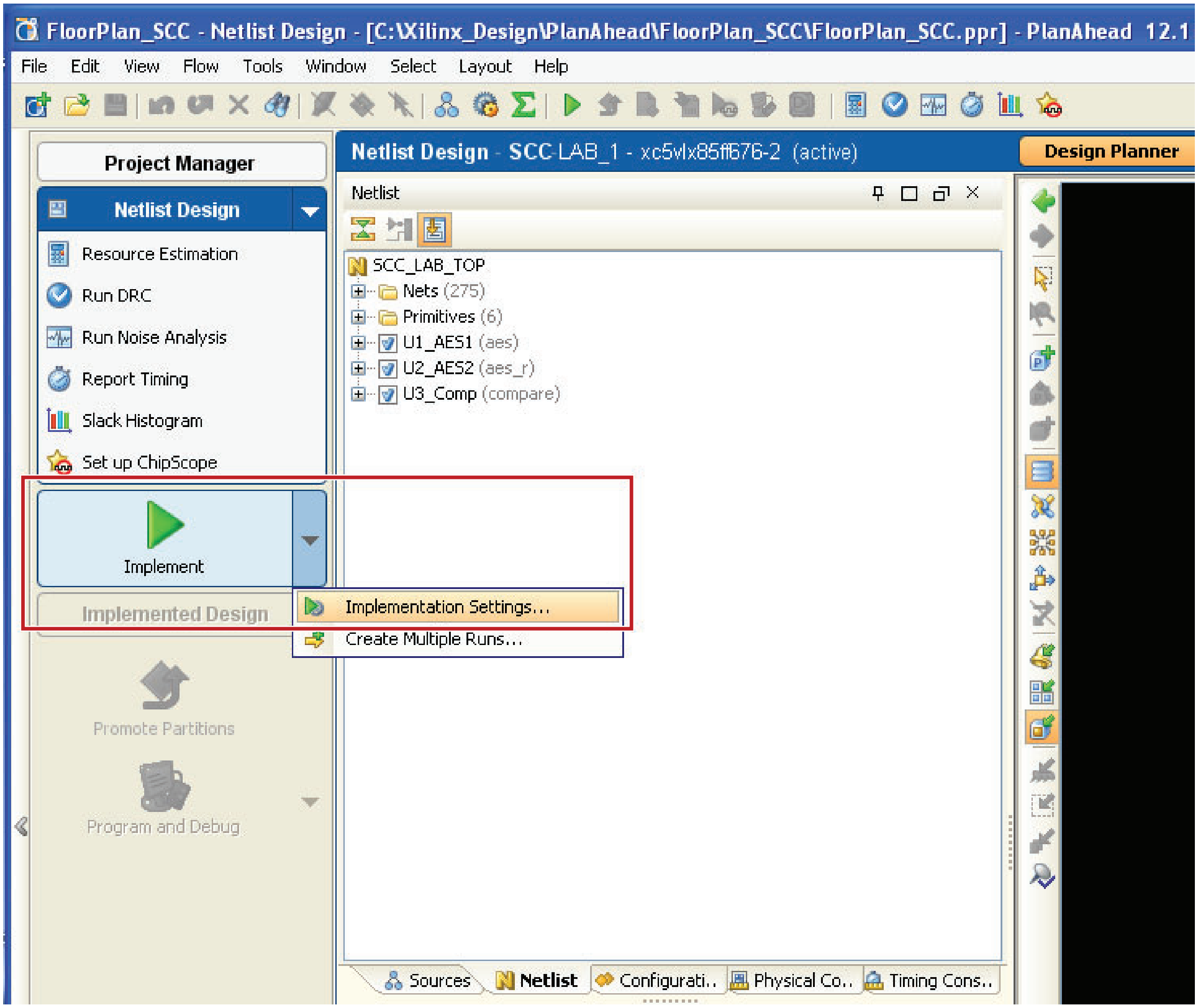
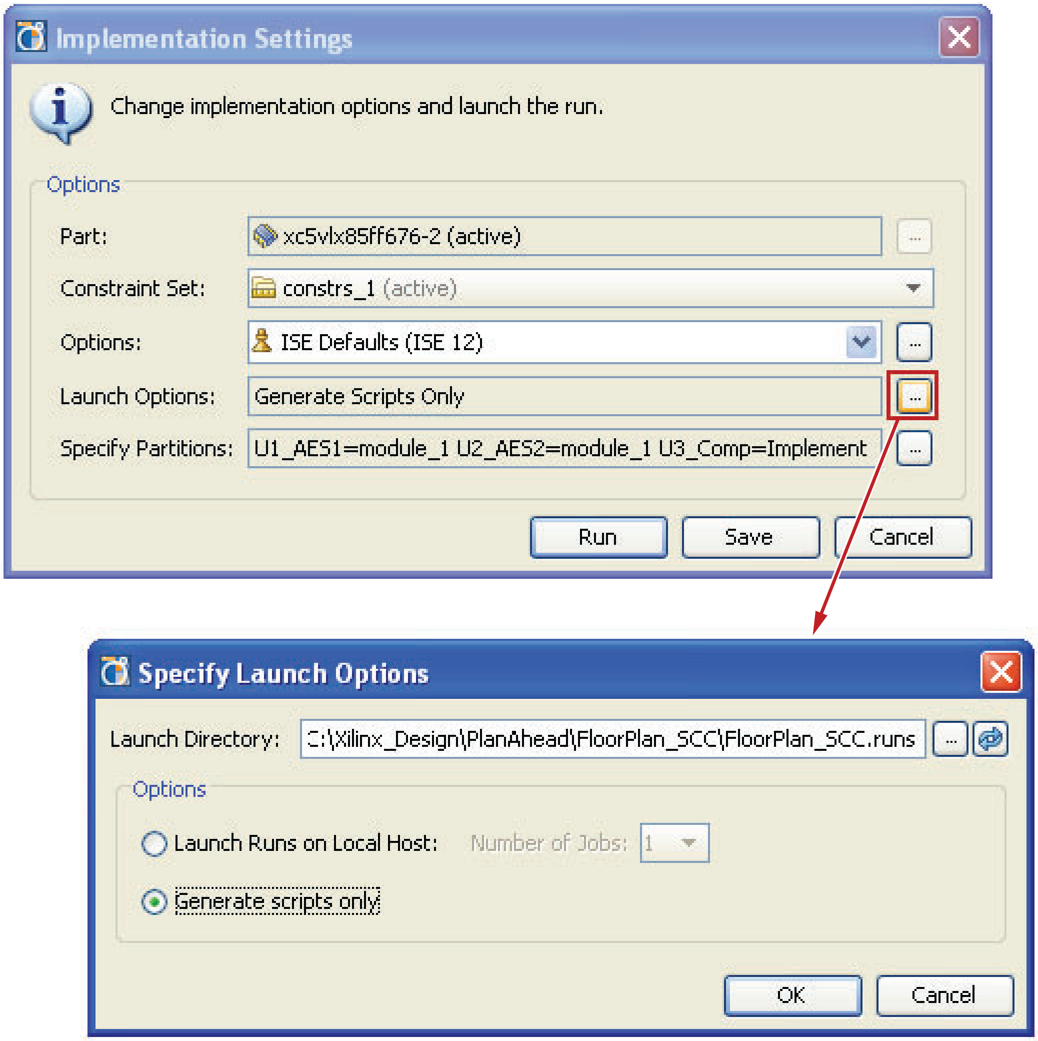


*Figure A-1:* **Implement Design (Implementation Settings)**

2. The Implementation Settings dialog box appears (see Figure A-2). This is not used to launch a run, but is only used to generate the batch script files. In the Implementation Settings dialog box, browse the Launch Options menu. In the Specify Launch Options dialog box, select the **Generate scripts only** option.



Implementation Settings

Change implementation options and launch the run.

Options

Part: xc5vlx85ff676-2 (active)

Constraint Set: constrs_1 (active)

Options: ISE Defaults (ISE 12)

Launch Options: Generate Scripts Only

Specify Partitions: U1_AES1=module_1 U2_AES2=module_1 U3_Comp=Implement

Specify Launch Options

Launch Directory: C:\Xilinx_Design\PlanAhead\FloorPlan_SCC\FloorPlan_SCC.runs

Options

○ Launch Runs on Local Host: Number of Jobs: 1

● Generate scripts only

X1105_aA_02_061810

*Figure A-2:* **Implementation Settings (Specify Launch Options)**

3. Click **OK** on the Specify Launch Options dialog box and then click **Run** on the Implementation Settings dialog box.

4. Modify the UCF file generated by the PlanAhead tool in the `..\PlanAhead\FloorPlan_SCC\FloorPlan_SCC.runs\config_1` directory. Open the UCF in a text editor and remove these three instances:

```
INST "U1_AES1" SCC_ISOLATED = "TRUE";
INST "U2_AES2" SCC_ISOLATED = "TRUE";
INST "U3_Comp" SCC_ISOLATED = "TRUE";
```

5. Save and close the UCF file.

6. Navigate to the `..\PlanAhead\FloorPlan_SCC\FloorPlan_SCC.runs\config_1` directory.

7. Run the batch script file created by the PlanAhead tool. Launch the runme file (`.bat` or `.sh`, depending on the platform).

   ***Note:*** If the project is open in the PlanAhead tool when the batch file is run, the implementation process can be monitored through the PlanAhead tool as if the implementation were initiated via the PlanAhead tool.

8. The routed NCD file `config_1_routed.ncd` that is used to run IVT on NCD File can be found in the `..\PlanAhead\FloorPlan_SCC\FloorPlan_SCC.runs\config_1` directory.
9. If the implementation is being monitored in the PlanAhead tool, an Implementation Completed dialog box appears after the design run is completed. Select the **Open Implemented Design** option and click **OK**.
10. The Device tab in the Design Planner pane shows the placed, routed, and partitioned design (see Figure 5-3, page 107).