7 Series Isolation Design Flow Lab Using ISE Design Suite 14.4

XAPP1085 (v1.0.2) July 31, 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
02/27/2013	1.0	Initial Xilinx release.
06/20/2013	1.0.1	Replaced SCC lounge with IDF page throughout. Added URL to download reference design in Reference Design Files.
07/31/2013	1.0.2	Replaced SCC lounge with IDF page in Lab Design Overview and Further Resources. Replaced Secure Solutions site with IDF page in References.

Table of Contents

Chapter 6: Verifying Design with NCD Isolation Verification Tool

Appendix A: Additional Resources

Chapter 1

Isolated Design Flow Overview

Isolation Design Flow Overview

The Isolation Design Flow (IDF) is the software methodology that allows Single Chip Crypto (SCC) implementations, or any other application requiring module isolation. This methodology is backed by significant schematic analysis and software verification (Isolation Verification Tool) to ensure elimination of single points of failure. SCC is an application allowing the implementation of a multi-chip crypto system in a single FPGA.

Single Chip Crypto

This chapter describes how an SCC system is created using a Kintex™-7 FPGA, ISE® Design Suite 14.4, and PlanAhead™ 14.4 development tools.

SCC systems are implemented using the isolation design flow (IDF). The IDF permits SCC applications that include redundant type 1 encryptors, resident red and black data, and functions that operate on multiple levels of security within the same Xilinx FPGA.

At the core of the IDF is physical and electrical isolation of chosen logic from other areas of the design. This is accomplished using:

- Conventional, best practices, modular Xilinx design entry where each function isolated by the designer must be at its own level of hierarchy
- Modular synthesis of each isolated region of the design
- (Mandatory) manual floorplanning to isolate the desired regions of the design on the FPGA while providing user-specified intra-region communication through the use of trusted routing
- Automated verification of isolation using the Xilinx Isolation Verification Tool (IVT) software

Lab Design Overview

The Kintex-7 FPGA IDF rules are outlined in the *Developing Secure and Reliable Single FPGA Designs with Xilinx 7 Series FPGAs Using the Isolation Design Flow* (XAPP1086). This companion application note describes how functions are isolated; specific differences between a normal partition flow and an IDF partition flow; IDF-specific HDL code mnemonics; and trusted routing rules. This application note is accessible from the Isolation Design Flow page on Xilinx.com [Ref 1].

To illustrate the IDF and its capabilities, this design implements isolated, redundant advanced encryption standard (AES) modules in a single 7 series FPGA, targeting a Kintex-7 device. Figure 1-1 is a hierarchical diagram of the various VHDL sub-blocks used in the implementation of the design. Type 1 crypto applications using 7 series FPGAs require defense-grade (XQ) devices for mask control. The design example used in this application note was created using a non-defense grade Kintex-7 XC7K325T-2FFG900 device.

Note: This lab requires the ISE software, version 14.4 or later.

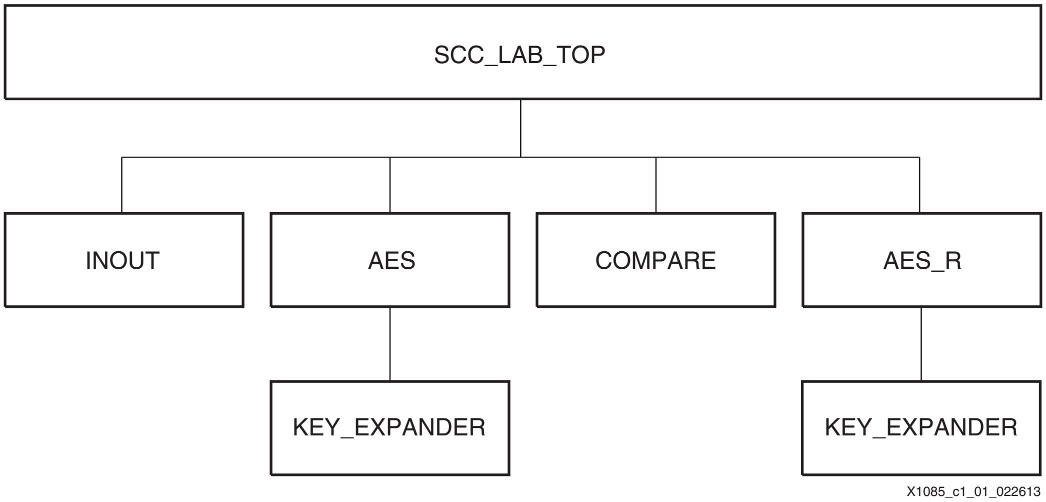


Figure 1-1: **VHDL Design Hierarchical Block Diagram**

Figure 1-2 shows the IDF that is used for the single-chip crypto lab.

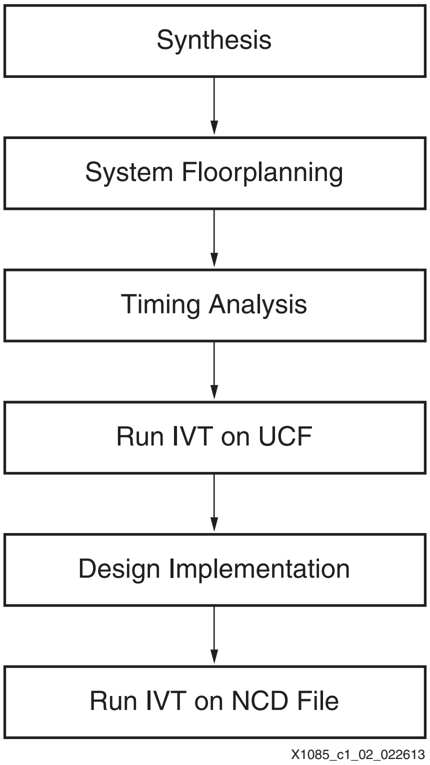


Figure 1-2: **IDF System Design Flow**

Figure 1-3 shows the partitioning and I/O mapping for the SCC lab design as implemented in an XC7K325T-2FFG900 device. The relative placement of the I/O pins is shown along the left and right sides of the device. All of the global clocking resources must physically reside in one of the isolated regions for them to be routed (as a condition of the isolated rules). The BUFGs and BUFHCEs are shown in the diagram as part of the AES partition.

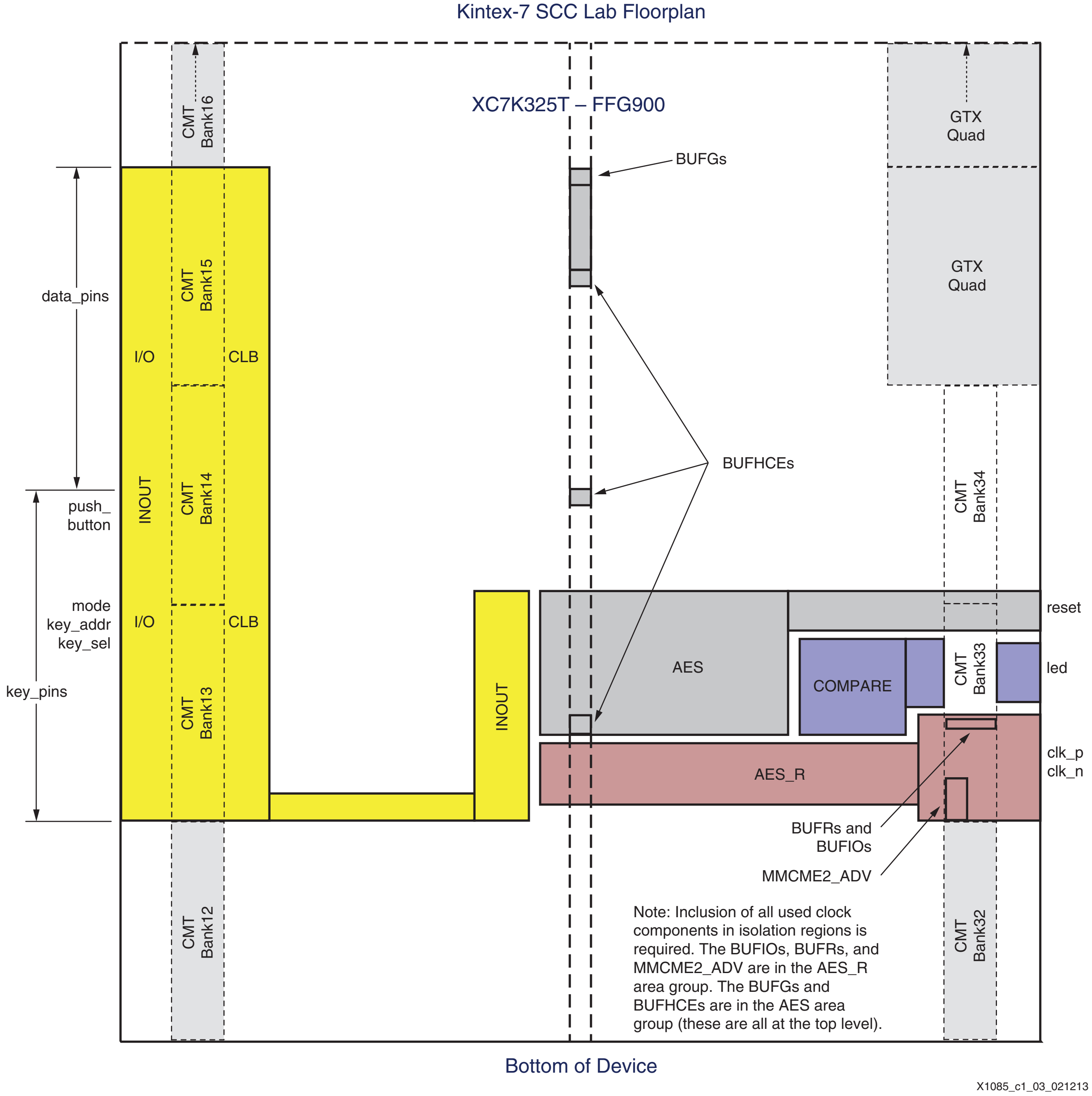


Figure 1-3: **Die View: Partitions and I/O Map in XC7K325T-2FFG900 Device**

Figure 1-4 is a detailed block diagram of the SCC lab design showing all of the isolated functions (modules), the I/O pins, and the internal signals connecting the isolated functions.

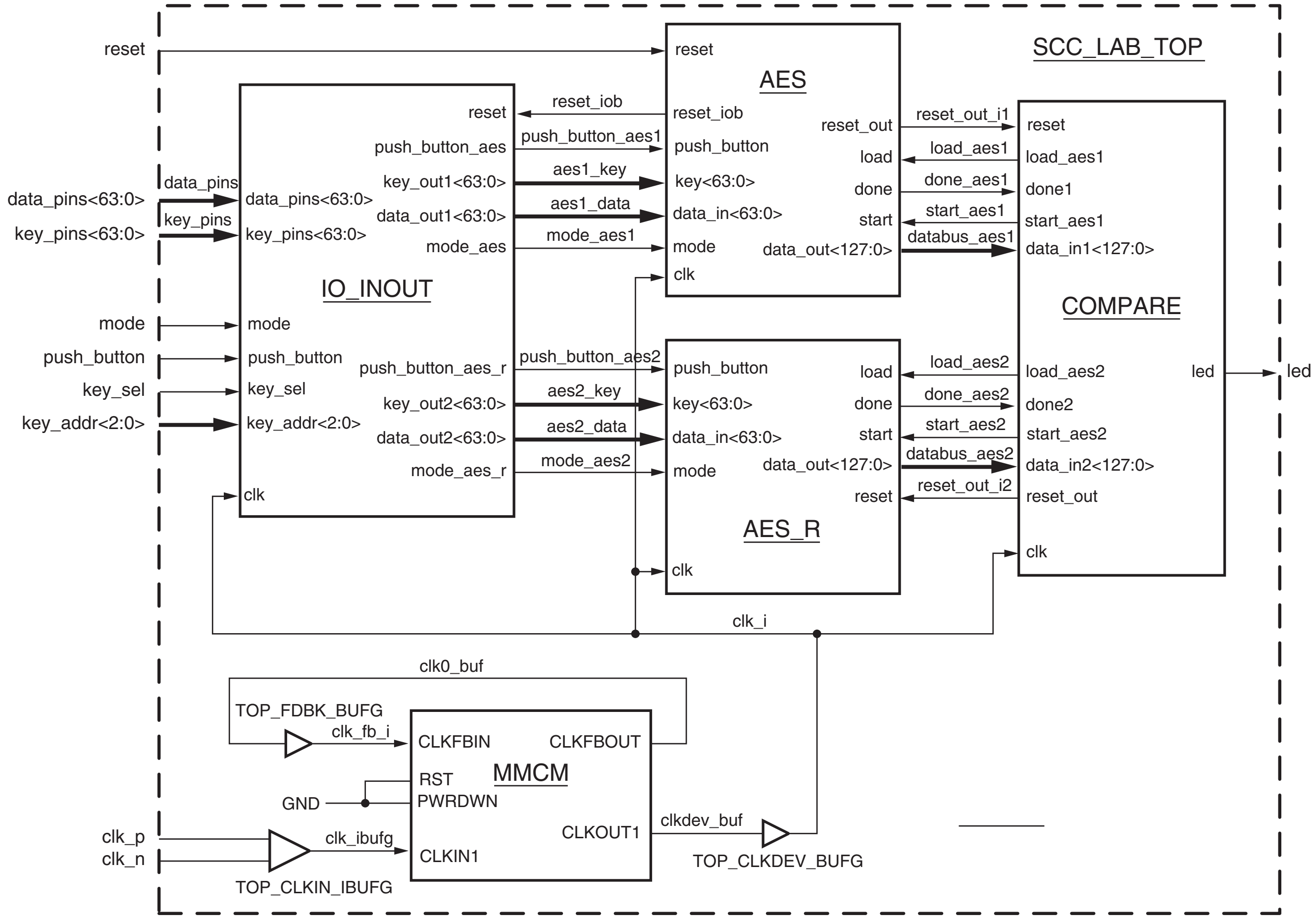


Figure 1-4: **Detailed Block Diagram of SCC Lab Modules**

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. 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. The fourth block shown is an I/O distribution module. It contains I/O and LUT buffers used for splitting the pin data to the redundant AES modules. A 64-bit data input bus and a 64-bit key input bus drive the AES blocks. The push_button pin, and the data and key input buses can be driven externally or left open because they have internal pull-downs in the INOUT HDL code. The mode pin has an internal pull-up. The key input to the encryption blocks can be selected from the key input external pins or from an internal ROM with eight sample 64-bit keys. This lab design gives an example of how a key might be stored internal to the FPGA in a ROM, potentially making a more secure design.

Figure 1-5 is a view of the completed SCC lab design displayed in the Xilinx FPGA Editor (FED) tool. The image to the left is the FED image of the entire XC7K325T FPGA. The magnified image to the right highlights the four isolated regions of the SCC lab design.

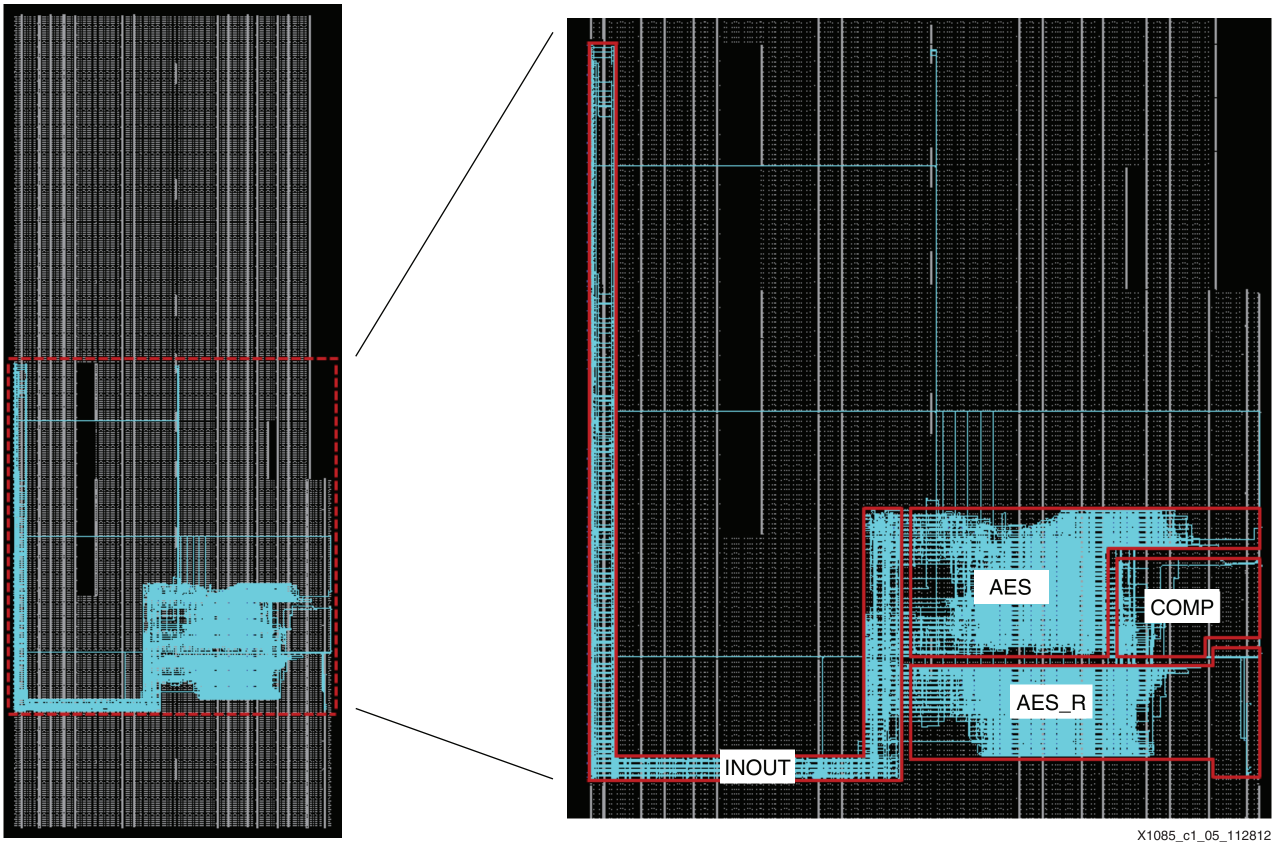


Figure 1-5: **FPGA Editor View: Implementation of Single-Chip Crypto Lab Design**

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=343444

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

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

Parameter	Description
General	
Developer Name	Xilinx
Target devices	Kintex-7 XC7K325T FPGA
Source code provided	Y

Table 1-1: **Reference Design Checklist *(Cont'd)***

Parameter	Description
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 14.4
SPICE/IBIS simulations	N
Implementation	
Synthesis software tools/version used	XST
Implementation software tools/versions used	ISE Design Suite 14.4
Static timing analysis performed	Y
Hardware Verification	
Hardware verified	N
Hardware platform used for verification	N/A

Table 1-2 shows the reference design utilization information for the targeted device.

Table 1-2: **Device Utilization Table**

Device	Speed Grade	Package	Pre-Map	Post-Route	Slices
XC7K325T	-2	FFG900	243.30 MHz	78.46 MHz	5861 (2%)

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 into a directory without any spaces in the path.

1. Copy `xapp1085.zip` to the Windows desktop.
2. Double-click **xapp1085.zip** and unzip the contents to the selected location.
3. The project files are placed in these directories:

```
\Xilinx_Design\buildscripts

\Xilinx_Design\ivt

\Xilinx_Design\results\

\Xilinx_Design\source\
```

Note: When the zip file is extracted, the `Xilinx_Design` directory is created, which contains the SCC lab project design files. The directory structure shown applies to this project.

Chapter 2

Synthesizing Design with PlanAhead Tool

This chapter describes the steps in a bottom-up synthesis flow using the PlanAhead™ tool. The bottom-up synthesis flow is used for IDF designs to maintain isolation. The PlanAhead tool v14.4 allows the user to import RTL source code directly.

Lab Design Source VHDL Files

The VHDL source files for the top-level and the individual isolated functions or modules are:

```
scc_lab_top.vhd
   top_package.vhd
aes.vhd
   aes_package.vhd
   key_expander.vhd
aes_r.vhd
   aes_package.vhd
   key_expander.vhd
compare.vhd
io_inout.vhd
```

In the flow in previous SCC/IDF application notes, each of the isolated functions are synthesized independently (out of context). The I/O buffers are inferred by default for each of the I/O pins in a module unless specifically inhibited via the buffer_type is none attribute.

The PlanAhead tool, version 14.4 provides the option to use a hierarchical or in-context RTL source import flow. This flow is used in this application note so the I/O buffers are specifically instantiated as needed in each isolated function at the HDL level. The global clock input buffers are instantiated in the top-level block (`scc_lab_top.vhdl`).

The Add I/O Buffers (-iobuf) synthesis option enables or disables I/O buffer insertion. The option is left at the default (which is *Yes*), so that I/O buffers are automatically inserted during synthesis, provided they are not already instantiated in the isolated function. XST does not place two I/O buffers on an I/O pin.

This is an example snippet of the I/O block (IOB) instantiation VHDL code for an input and an output buffer:

```
-- Instantiate Input Buffer for PUSH_BUTTON
        PushButton_IBUF : IBUF
         port map(
           O => push_button_inbuf,
           I => push_button
         );
-- Instantiate Output Buffer for LED
        Led_OBUF : OBUF
         port map(
           O => led,
           I => led_outbuf
         );
```

For this design, all of the inputs are registered and are packed into the IOBs to improve input/output path timing. The Map option for packing I/O registers must be set as:

> **`Set Pack internal flops/latches into IOBs`** (**`-pr`**) to **`Both`** (**`b`**).

The Pack I/O Registers Into IOBs (-iob) synthesis option is left at the default (which is *Auto)*. The input and output registers are automatically packed in the IOBs during synthesis, depending on the optimization setting.

This is an example snippet of VHDL code for the input register instantiation in the INOUT isolated module:

```
-- Input Register Instantiations
      process (clk)
      begin
         if rising_edge(clk)then
            if reset = '1' then
              data_inreg           <= (others => '0');
              key_inreg            <= (others => '0');
              mode_inreg           <= '0';
              push_button_inreg    <= '0';
              key_sel_inreg        <= '0';
              key_addr_inreg       <= (others => '0');
            else
              data_inreg           <= data_inbuf;
              key_inreg            <= key_inbuf;
              mode_inreg           <= mode_inbuf;
              push_button_inreg    <= push_button_inbuf;
              key_sel_inreg        <= key_sel_inbuf;
              key_addr_inreg       <= key_addr_inbuf;
            end if;
         end if;
      end process;
```

Entering Project in PlanAhead Tool

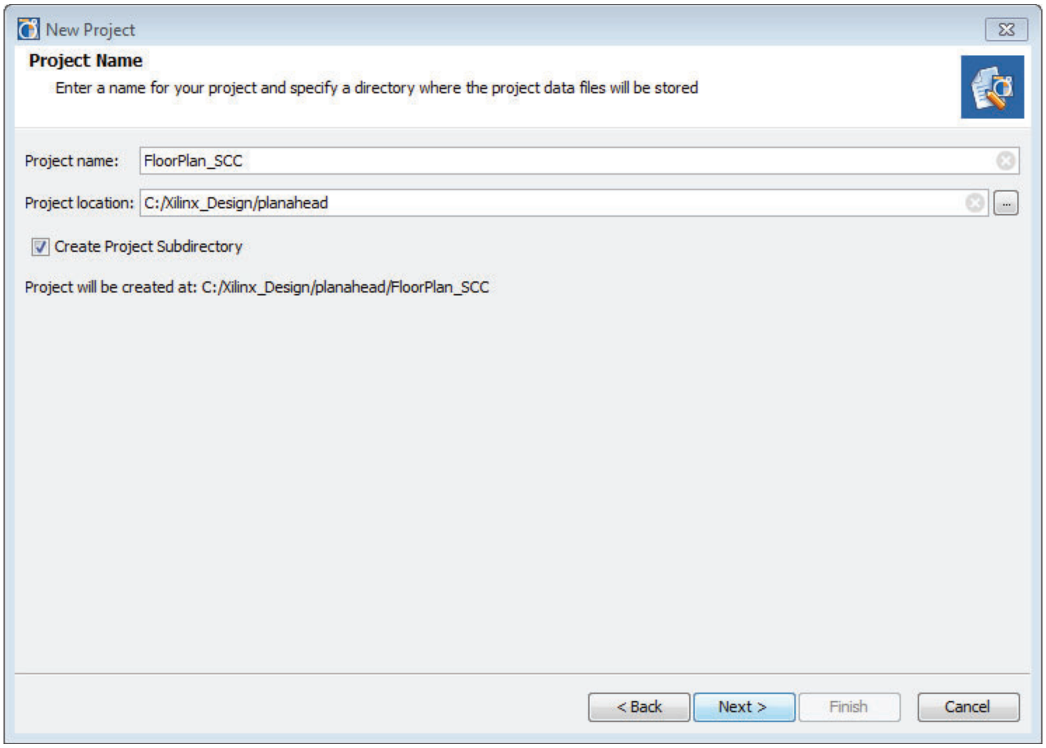
Launching PlanAhead Tool

To launch the PlanAhead tool v14.4, select **Start > All Programs > Xilinx Design Tools > ISE Design Suite 14.4 > PlanAhead > PlanAhead**.

Creating Project

The PlanAhead tool works with any standard RTL source file. The standard guidelines are followed to generate a new project and import the RTL source files into the PlanAhead tool to create a floorplan for the design.

1. Set up a new PlanAhead project. From the Getting Started menu, select **Create New Project** and click **Next** (Figure 2-1).



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

☑ Create Project Subdirectory

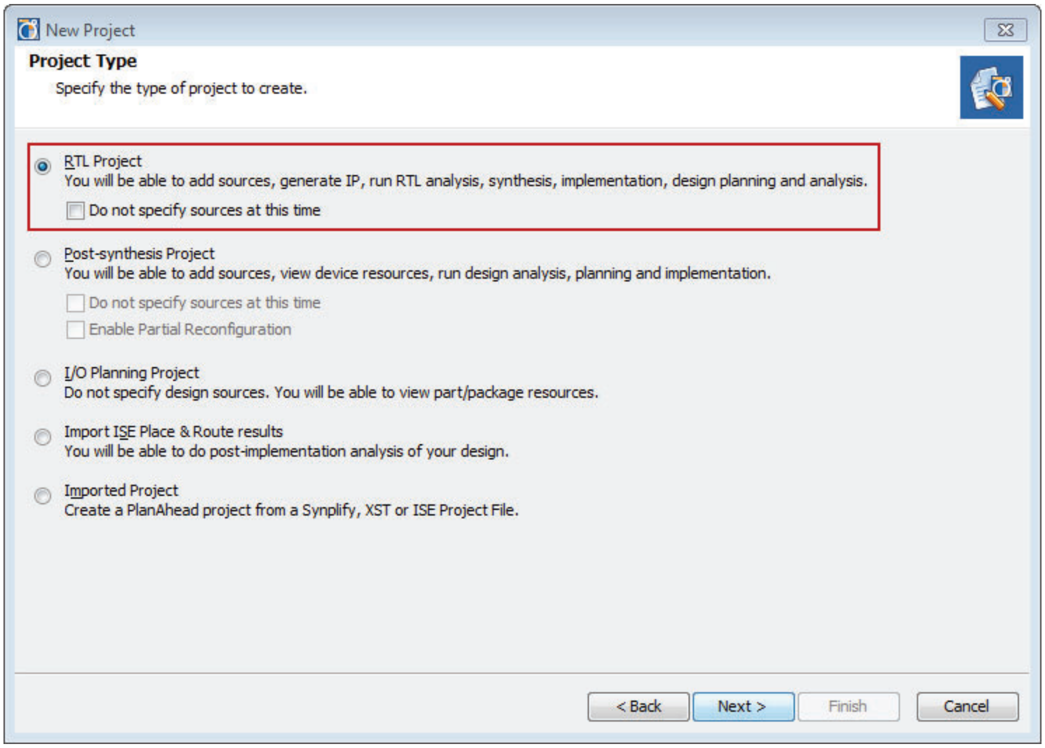
Project will be created at: C:/Xilinx_Design/planahead/FloorPlan_SCC

X1085_c2_01_112812

Figure 2-1: **New Project**

2. Enter:
   - Project name: **FloorPlan_SCC**
   - Project location: **../Xilinx_Design/planahead**
   - Create Project Subdirectory: checked
3. Click **Next**.

4. Select **RTL Project** (Figure 2-2) to specify and add the RTL source files.
5. Click **Next**.



New Project

Project Type

Specify the type of project to create.

● RTL Project
You will be able to add sources, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.
□ Do not specify sources at this time

○ Post-synthesis Project
You will be able to add sources, view device resources, run design analysis, planning and implementation.
□ Do not specify sources at this time
□ Enable Partial Reconfiguration

○ I/O Planning Project
Do not specify design sources. You will be able to view part/package resources.

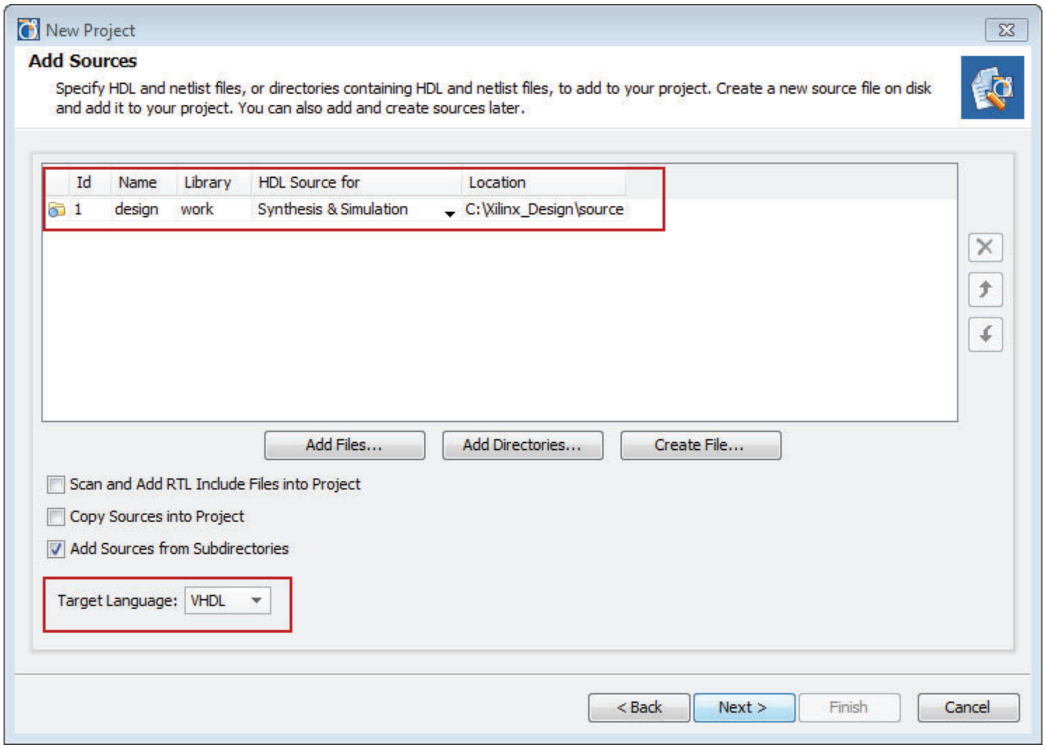
○ Import ISE Place & Route results
You will be able to do post-implementation analysis of your design.

○ Imported Project
Create a PlanAhead project from a Synplify, XST or ISE Project File.

X1085_c2_02_022613

Figure 2-2: **New Project (Project Type - RTL)**

6. Import the RTL source files, directories, or both. In this case, all of the design HDL files are located in the directory `..\Xilinx_Design\source\design`. Click **Add Directories**, browse to the directory shown, and click **Select** as shown in Figure 2-3. The HDL source file directory is shown in the Name field.



New Project

Add Sources

Specify HDL and netlist files, or directories containing HDL and netlist files, to add to your project. Create a new source file on disk and add it to your project. You can also add and create sources later.

Id	Name	Library	HDL Source for	Location
1	design	work	Synthesis & Simulation	C:\Xilinx_Design\source

☐ Scan and Add RTL Include Files into Project

☐ Copy Sources into Project

☑ Add Sources from Subdirectories

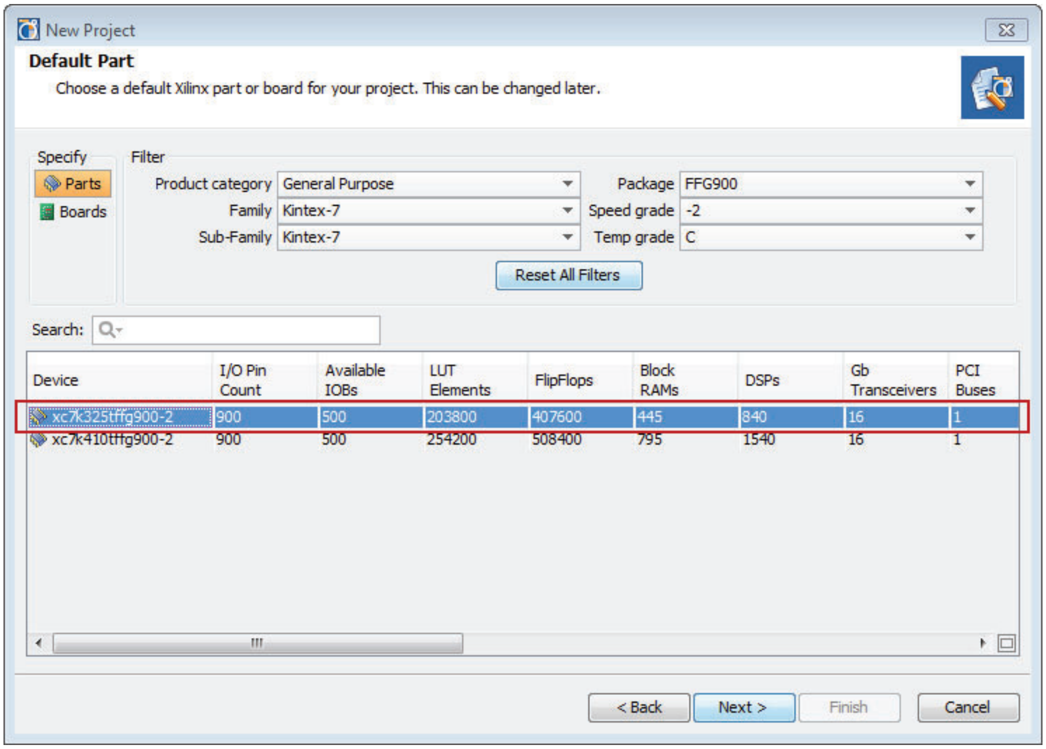
Target Language: VHDL

X1085_c2_03_112812

Figure 2-3: **New Project (Add Directories– Add Sources)**

7. Do not copy sources into the project for this lab. Uncheck **Copy Sources into Project**. Check **Add Sources from Subdirectories**. For Target Language, select **VHDL**.
8. Click **Next**.
9. This lab does not add existing IP, so click **Next** again.
10. This lab creates the final UCF from the beginning. Because no UCF files need to be imported, click **Next** again.

11. Select the product, family, and sub-family for the default part (see Figure 2-4). For this lab, make these selections:
    - Product: **General Purpose**
    - Family: **Kintex-7**
    - Sub-Family: **Kintex-7**
    - Package: **FFG900**
    - Speed Grade: **-2**
    - Temperature Grade: **C**



Default Part

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

Product category: General Purpose
Family: Kintex-7
Sub-Family: Kintex-7
Package: FFG900
Speed grade: -2
Temp grade: C

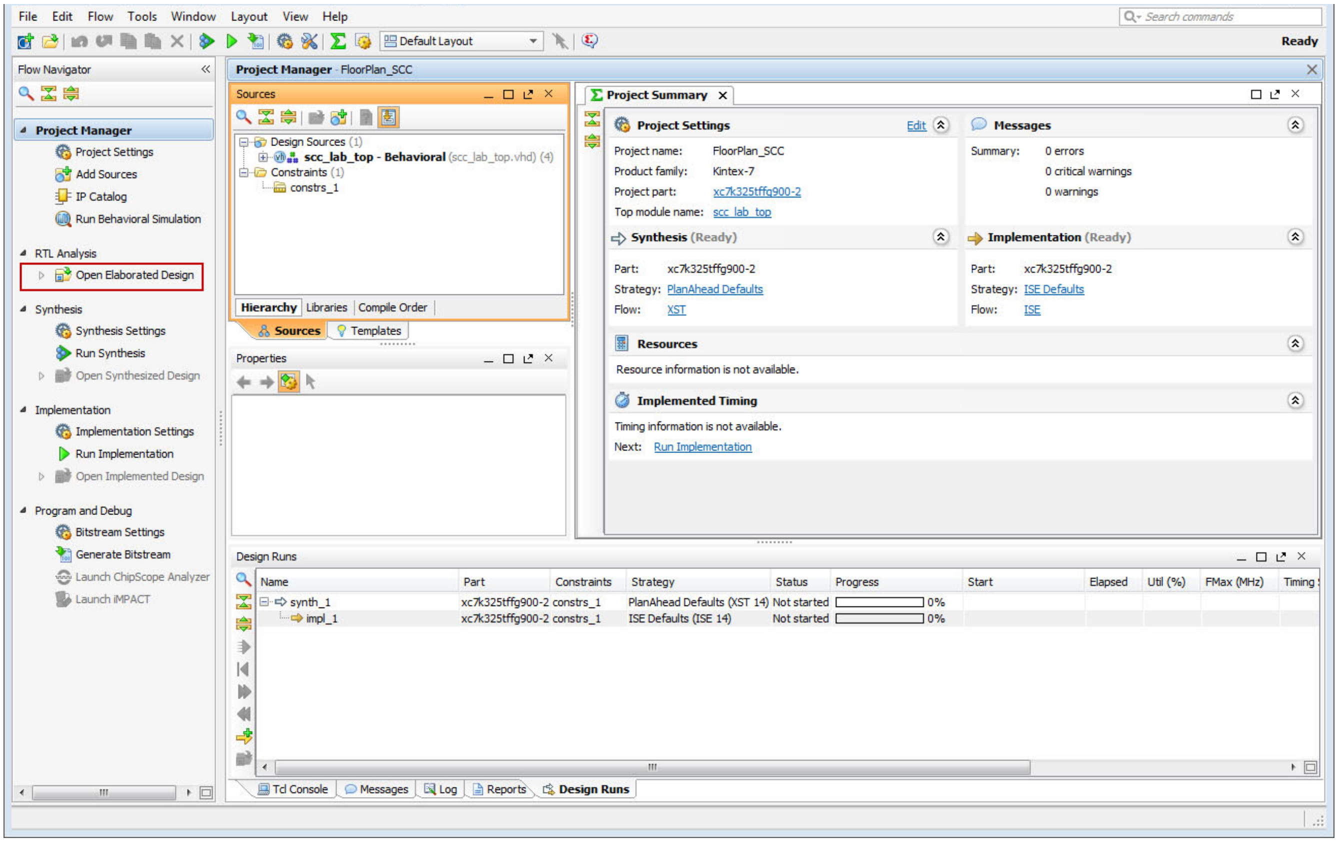
Device	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs	DSPs	Gb Transceivers	PCI Buses
xc7k325tffg900-2	900	500	203800	407600	445	840	16	1
xc7k410tffg900-2	900	500	254200	508400	795	1540	16	1

X1085_c2_04_112812

Figure 2-4: **New Project (Default Part)**

12. Choose the **xc7k325tffg900-2** device.
13. Click **Next** and then click **Finish**.

14. The PlanAhead tool project is created. Figure 2-5 shows the PlanAhead tool Project Manager window for the FloorPlan_SCC project.



Project name: FloorPlan_SCC
Product family: Kintex-7
Project part: xc7k325tffg900-2
Top module name: scc_lab_top

Summary: 0 errors
0 critical warnings
0 warnings

Synthesis (Ready)
Part: xc7k325tffg900-2
Strategy: PlanAhead Defaults
Flow: XST

Implementation (Ready)
Part: xc7k325tffg900-2
Strategy: ISE Defaults
Flow: ISE

Resource information is not available.

Timing information is not available.
Next: Run Implementation

Name	Part	Constraints	Strategy	Status	Progress	Start	Elapsed	Util (%)	FMax (MHz)	Timing
synth_1	xc7k325tffg900-2	constrs_1	PlanAhead Defaults (XST 14)	Not started	0%					
impl_1	xc7k325tffg900-2	constrs_1	ISE Defaults (ISE 14)	Not started	0%					

X1085_c2_05_021513

Figure 2-5: **PlanAhead Tool Project Manager (Open Elaborated Design)**

Building Netlists (Elaborated Design)

The next step is to generate the RTL netlist (compiled netlist of the design). From the RTL netlist, the partitions can be specified. Setting partitions in the PlanAhead tools builds the `xpartition.pxml` file that is used by the implementation tools to specify the partitions and the partition attributes. The partitions define the isolated regions and the isolated functions in the PlanAhead tool.

15. To open the elaborated design in the PlanAhead tool, under the Project Manager view on the left, select **RTL Analysis > Open Elaborated Design**. See Figure 2-5.

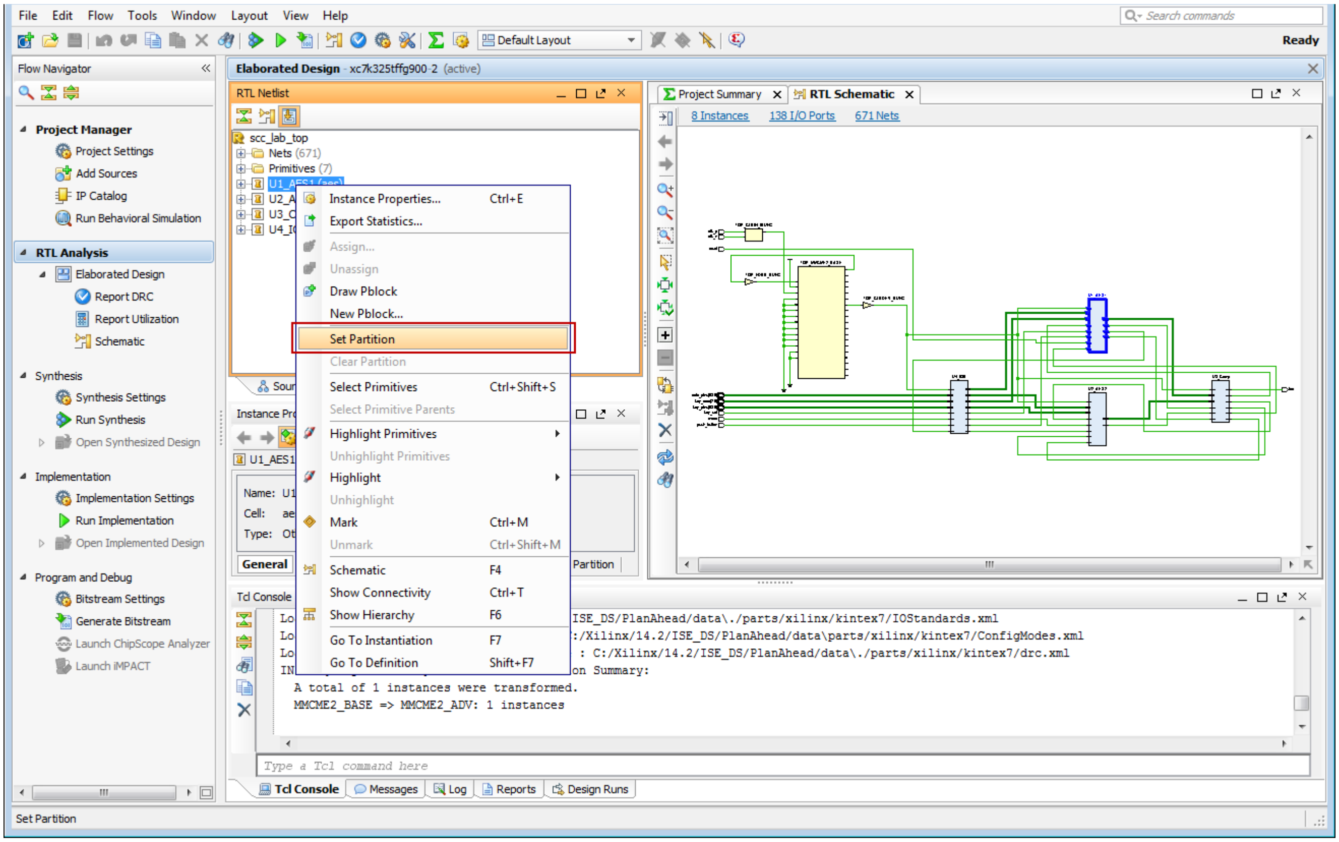
    The RTL Netlist tab is displayed in the PlanAhead tool along with the RTL schematic.

    Note: If at some time during the process of creating it, the lab must be saved and continued at a later time, the project must either be re-elaborated or the synthesized design must be re-opened.

Defining ISO Partitions

Each partition (isolated function) must be converted to an isolated partition (ISO). The square to the left of the instance name, in the RTL Netlist tab, turns solid yellow after the partition is set. This operation sets the is_partition property to true for each partition.

1. Right-click **U1_AES** (AES) in the Netlist tab and select **Set Partition** (see Figure 2-6).



```
Lo                                    ISE_DS/PlanAhead/data\./parts/xilinx/kintex7/IOStandards.xml
Lo                                    :/Xilinx/14.2/ISE_DS/PlanAhead/data\parts/xilinx/kintex7/ConfigModes.xml
Lo                                     : C:/Xilinx/14.2/ISE_DS/PlanAhead/data\./parts/xilinx/kintex7/drc.xml
IN                                    on Summary:
  A total of 1 instances were transformed.
  MMCME2_BASE => MMCME2_ADV: 1 instances
```

X1085_c2_06_021513

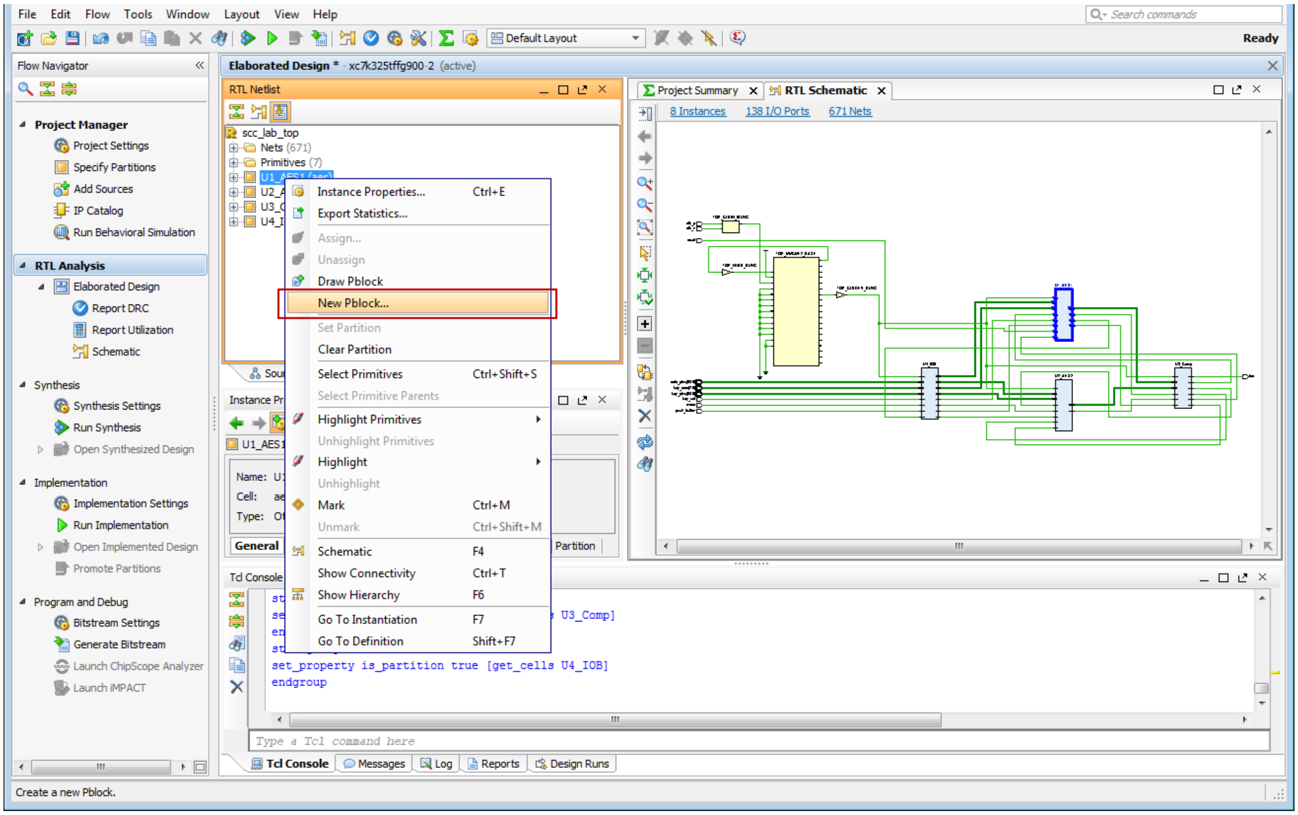
Figure 2-6: **Set Partition**

2. Repeat step 1 for the U2_AES2 (AES_R) partition.
3. Repeat step 1 for the U3_Comp (COMPARE) partition.
4. Repeat step 1 for the U4_IOB (IO_INOUT) partition.

The PlanAhead tool uses pblocks to define the area group definition for an isolated region. Area group constraints specify the components that are reserved for implementation of a partition or isolated function. The Physical Constraints tab defines the partitions by identifying pblocks.

A new pblock must be created for each partition. The solid yellow square to the left of the instance name in the RTL Netlist tab has a blue check mark after New Pblock is selected. Later in the process, the pblock sizes are actually set for each partition.

5. In the RTL_Netlist tab, right-click the **U1_AES1** (AES) partition and select **New Pblock** (see Figure 2-7). Keep the default name, pblock_U1_AES1. Make sure that **Assign selected instance** is checked to map the pblock to the isolated function and click **OK**.



```
st
se                                   U3_Comp]
en
st
set_property is_partition true [get_cells U4_IOB]
endgroup
```

X1085_c2_07_021513

Figure 2-7: **New Pblock**

6. Repeat step 5 for the U2_AES2 (AES_R) partition while keeping the default name, pblock_U2_AES2.
7. Repeat step 5 for the U3_Comp (COMPARE) partition while keeping the default name, pblock_U3_Comp.
8. Repeat step 5 for the U4_IOB (IO_INOUT) partition while keeping the default name, pblock_U4_IOB.

Synthesizing the Design

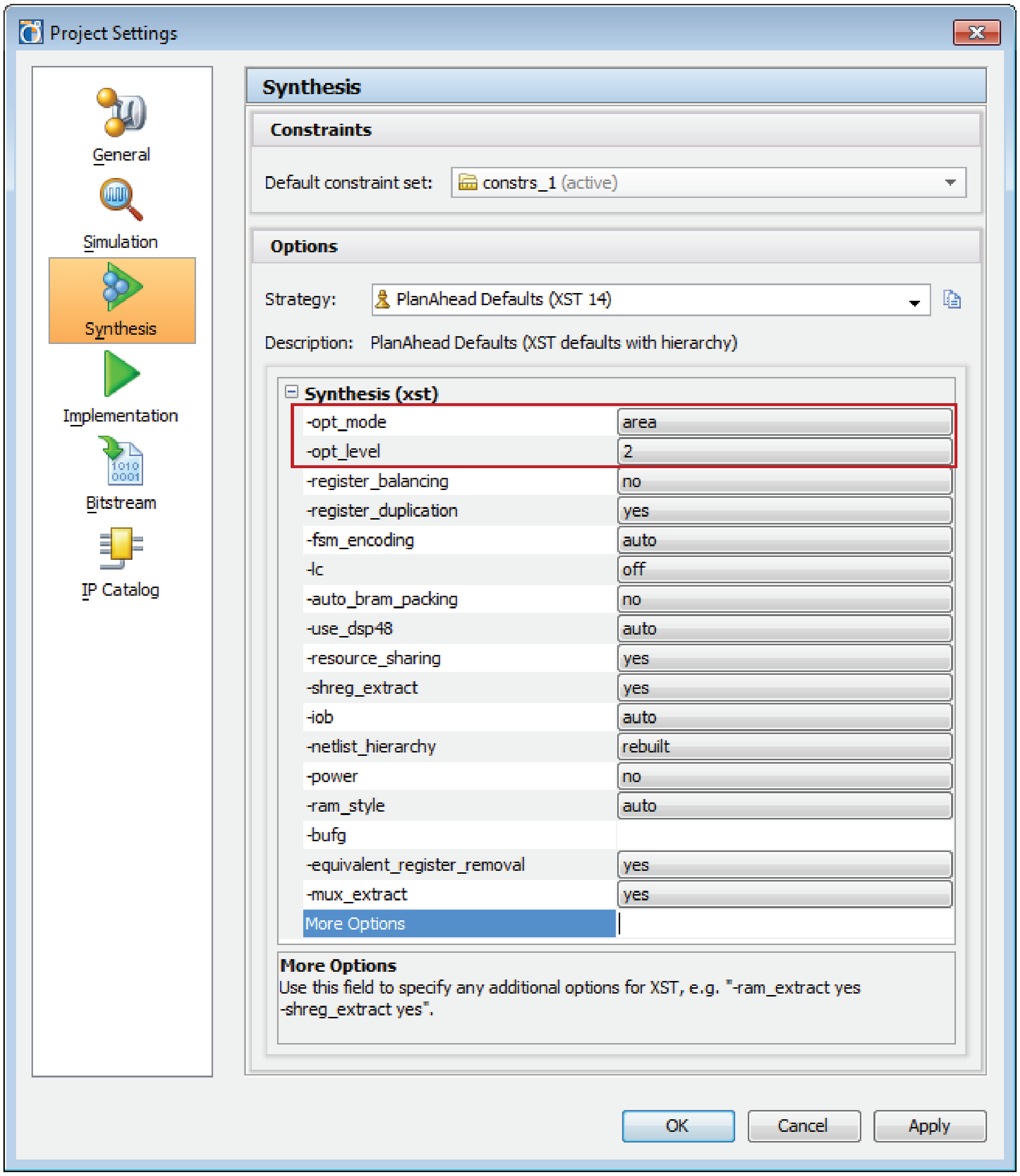
Setting Synthesis Options and Synthesizing the Design

The PlanAhead tool now has enough information to automatically execute a bottom-up synthesis of the design.

Before running the synthesis, the synthesis options need to be specified. These steps describe how to synthesize the design:

1. Set the synthesis options by bringing up the XST Synthesis Options window. Click **Synthesis > Synthesis Settings** in the Flow Navigator window. Set Optimization

Goal (-opt_mode) to **Area**, set Optimization Effort Level (-opt_level) to **2**. Refer to the Synthesis window in the Project Settings dialog shown in Figure 2-8. The remaining synthesis settings can use the default setting. Click **OK**.



Synthesis

Constraints

Default constraint set: constrs_1 (active)

Options

Strategy: PlanAhead Defaults (XST 14)

Description: PlanAhead Defaults (XST defaults with hierarchy)

Synthesis (xst)

-opt_mode	area
-opt_level	2
-register_balancing	no
-register_duplication	yes
-fsm_encoding	auto
-lc	off
-auto_bram_packing	no
-use_dsp48	auto
-resource_sharing	yes
-shreg_extract	yes
-iob	auto
-netlist_hierarchy	rebuilt
-power	no
-ram_style	auto
-bufg	
-equivalent_register_removal	yes
-mux_extract	yes
More Options	

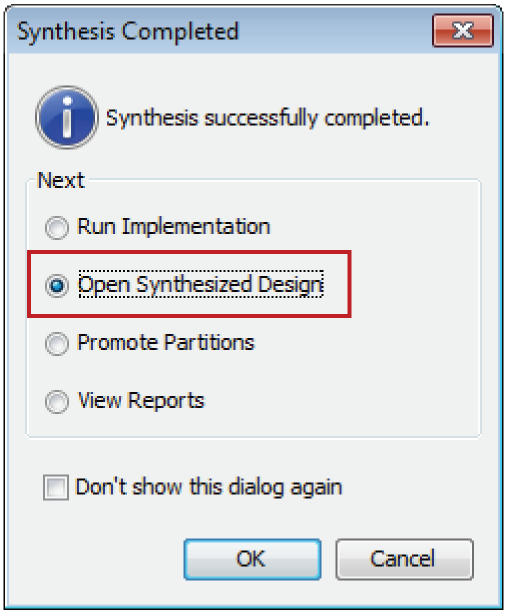
More Options
Use this field to specify any additional options for XST, e.g. "-ram_extract yes -shreg_extract yes".

X1085_c2_08_112812

Figure 2-8: **Project Settings (Synthesis Options)**

2. To run synthesis, click **Synthesis > Run Synthesis** in the Flow Navigator window. Click **Save** if prompted to save the elaborated design.

3. A Synthesis Completed window, shown in Figure 2-9, appears after synthesis is complete. The synthesized design requires more settings before running implementation. Select the **Open Synthesized Design** option and click **OK** to open the netlist or synthesized design.
4. Close the elaborated design, if prompted.



Synthesis Completed

Synthesis successfully completed.

Next

○ Run Implementation

● Open Synthesized Design

○ Promote Partitions

○ View Reports

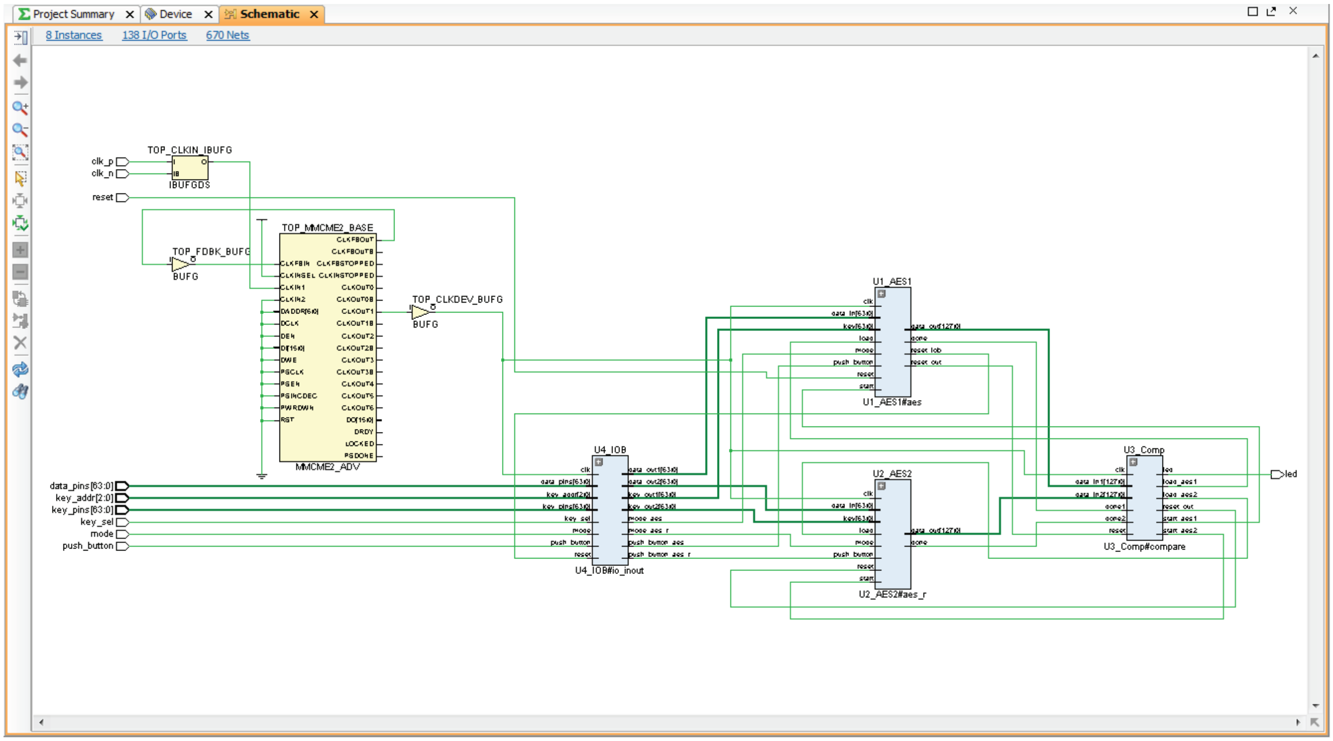
□ Don't show this dialog again

X1085_c2_09_112812

Figure 2-9: **Synthesis Completed Window**

5. With the netlist loaded, a graphical view of the device is displayed in the PlanAhead tool Device window. Because area group constraints have not been specified, no area groups appear in the device planner.

6. To create the schematic for the synthesized design, select **Synthesis > Synthesized Design > Schematic** in the Flow Navigator window (shown in Figure 2-10).



X1085_c2_10_112812

Figure 2-10: **Synthesis Schematic**

7. Now that the project is created and a constraints file has been generated, the target constraint file can be assigned. In the Sources window under Constraints > constrs_1, right-click `scc_lab_top.ucf` and select **Set as Target Constraint File**.

Isolation Design Flow Progress – Synthesis Block

The Synthesis block of the IDF system design flow diagram is complete, as shown in Figure 2-11.

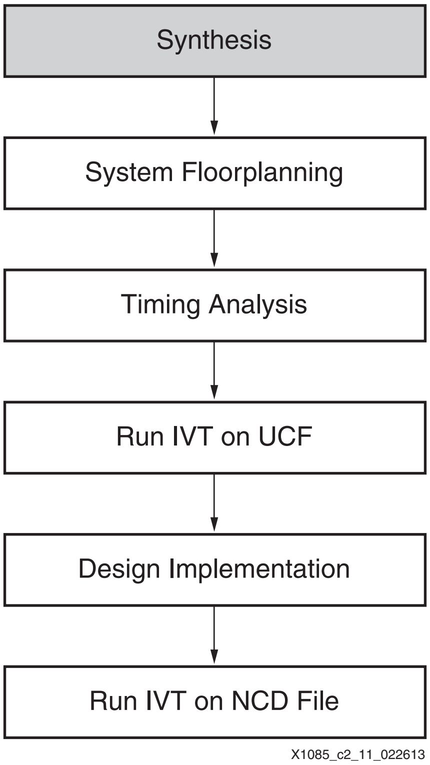


Figure 2-11: **IDF System Design Flow with Module Synthesis Block Completed**

Chapter 3

Floorplanning the System

Placing I/Os and Clocking Resources

Placing I/Os

When placing I/O buffers and pins, it is imperative to consider the physical location of I/Os in relation to the logic regions to which they interface. Initial placement of the I/O pins should be completed early in the floorplanning process to address any board dependencies.

Clocking components can be placed early in the floorplanning process or while the isolated regions are being defined. However, they should be placed before running implementation. All of the clocking resources in the design must be physically owned by one of the isolated functions (that is, the clocking resources must reside within the physical area group of one of the partitions (pblocks)). The clocking components are logically owned by the top-level module because they are global. The advantage of placing the clock components while the isolated partitions are being defined is that it is more obvious where to place them for the overall design. For this lab, the placing of the clocking components occurs in this stage because the general floorplan has already been defined.

While the clock input can be placed on any clock-capable pin, because it does not have to be isolated, the LED pin is part of the COMPARE logic so it does need to be physically placed within that region. Similarly, the reset pin is owned by the AES logic, so it needs to be physically placed inside the AES region. The remaining I/O pins are owned by the IO_INOUT (IOB) logic so they need to be placed within that region.

In general, all available resources should be included in each isolated partition, even if the logic is not used because excluding them also excludes using their respective routing resources. Failure to do so, while not an error, can produce designs that are difficult to route. All used components must be included. This includes the global, regional, and I/O clock buffers (BUFGCTRL, BUFHCE, BUFR, BUFIO, and BUFMRCE), even though such components are instantiated (logically owned) at the top level. The exceptions to this are the clock management tile (CMT) clocking components (MMCME2_ADV, PLLE2_ADV, PHASER_IN_PHY, PHASER_OUT_PHY, PHY_CONTROL, IN_FIFO, and OUT_FIFO). It is recommended to only include the CMT clocking resources in an isolated partition if they are used in the design.

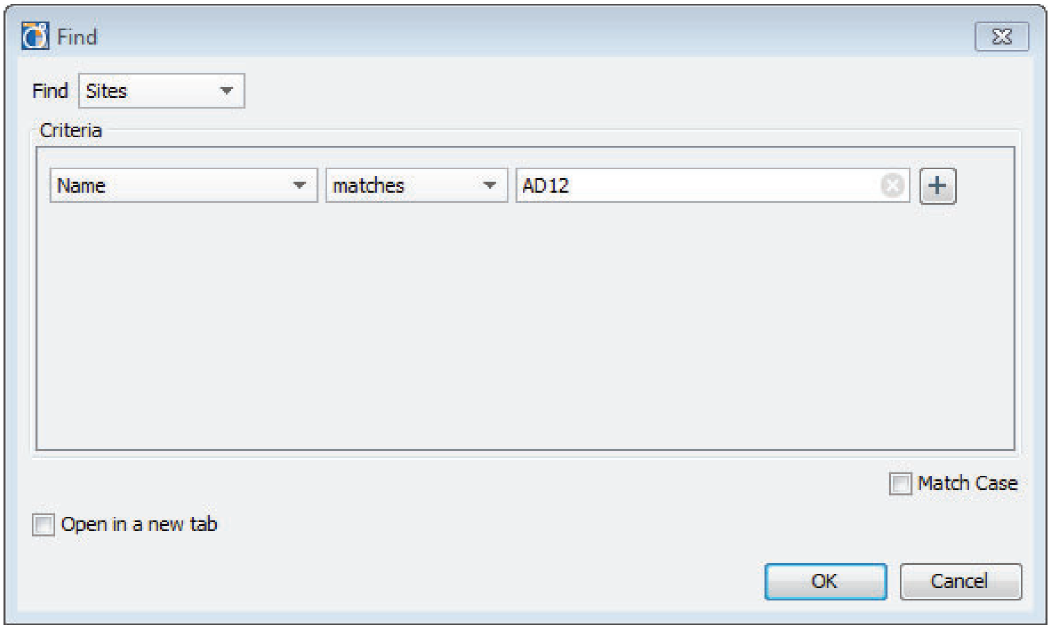
All boxes should be checked, including all of the buffers (BUFGCTRL, BUFHCE, BUFR, BUFIO, and BUFMRCE). The BUFMRCE needs to be included in the design if the BUFIO or the BUFR is used so that the clock can route between the two even if the BUFMRCE is not actually used.

Note: The CMT column is directly adjacent to the I/O column, and the routing resources to the I/O are not dense. There are enough routing resources to jump over an unused CMT column to gain access to the I/O routing without causing any routing issues.

The following procedure is for placing the individual I/O pins using the PlanAhead tool:

Note: For more experienced users of the PlanAhead tools, it might be easier to add I/O ports from the I/O ports tab in the I/O Planning view. Add the port by opening the Scalar ports folder and entering the desired pin location under the site column for the desired port. Otherwise, use the following procedure.

1. Go to the Device tab and open the synthesized design if it is not already opened. Select **Synthesis > Open Synthesized Design** in the Flow Navigator window.
2. To specify the I/O pin for clk_p, select **Edit > Find** (shortcut: **Ctrl-F**). A Find window appears, as shown in Figure 3-1.



Find: Sites

Criteria: Name | matches | AD12

□ Match Case

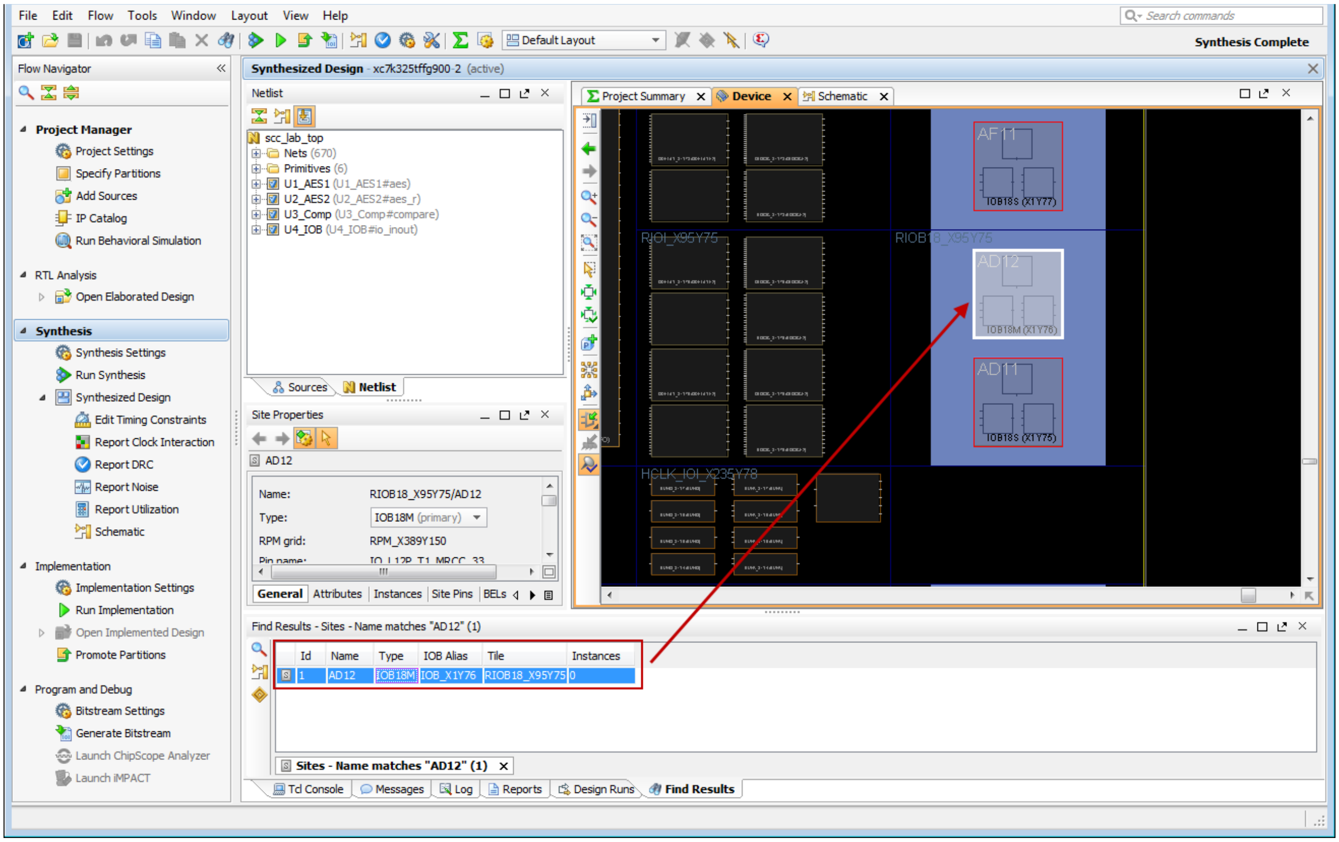
□ Open in a new tab

X1085_c3_01_112912

Figure 3-1: **Find Site**

3. Select **Sites** from the Find list box.
4. Select **Name** and **matches** from the two list boxes under Criteria.
5. Type **AD12** in the remaining field box and click **OK**. The search results appear on a tab in the Find Results section at the bottom of the PlanAhead tool window.

6. Select the find result and press **F9** to zoom to and highlight the current selection, as shown in Figure 3-2. Alternatively, select **View > Fit Selection** from the top menu bar to perform the same task.



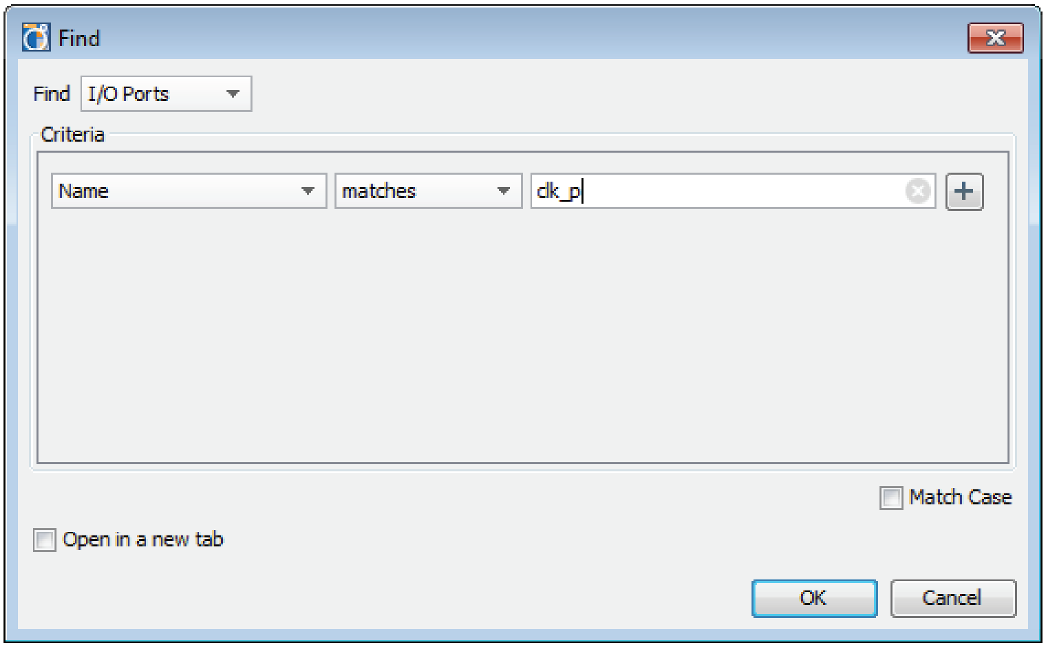
Name: RIOB18_X95Y75/AD12
Type: IOB18M (primary)
RPM grid: RPM_X389Y150
Pin name: IO_L12P_T1_MRCC_33

Id	Name	Type	IOB Alias	Tile	Instances
1	AD12	IOB18M	IOB_X1Y76	RIOB18_X95Y75	0

X1085_c3_02_021513

Figure 3-2: **Find Results**

7. To place clk_p on pin AD12, select **Edit > Find** (shortcut: **Ctrl-F**). A Find window appears, as shown in Figure 3-3.



Find: I/O Ports

Name matches clk_p

□ Match Case

□ Open in a new tab

X1085_c3_03_112912

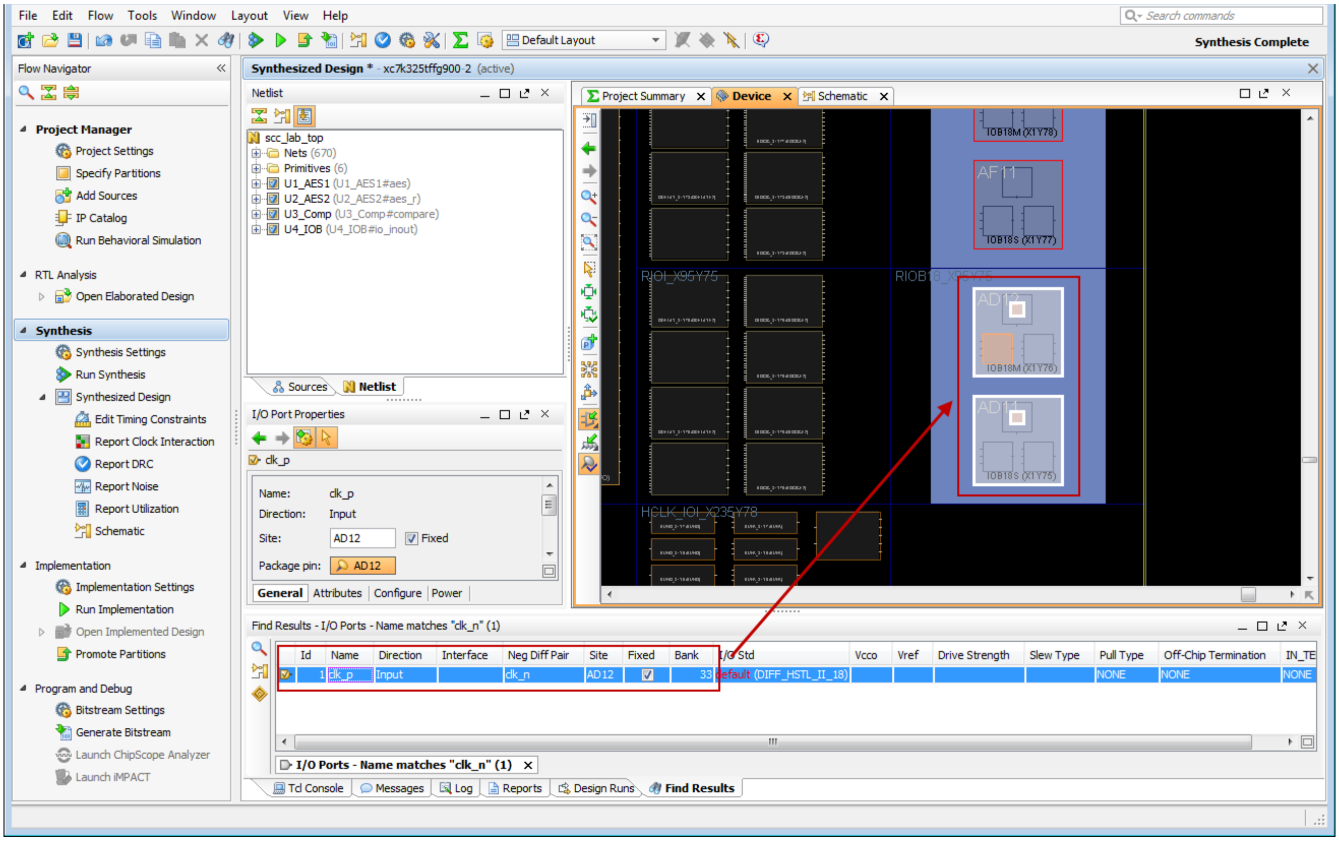
Figure 3-3: **Find I/O Port**

8. Select **I/O Ports** from the Find: list box.
9. Select **Name** and **matches** from the two list boxes under Criteria.
10. Type **clk_p** 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.

    Note: Alternatively, it might be quicker to select **Windows > I/O Ports** to search for and select each I/O port individually as it is being placed.

11. Click and drag the find result **clk_p** to the site identified (AD12) in step 2 through step 6, as shown in Figure 3-4.

    Note: Because this clock input is a differential input, both I/O pins get placed as a pair: AD12 for clk_p and AD11 for clk_n. The positive signal clk_p must be manually placed on the IOB18M pin, which causes the negative signal clk_n to be automatically placed on the IOB18S pin. It might be necessary to drag the pointer slightly above the desired IOB pair, but the AD12/AD11 IOB pair is highlighted in orange when the pointer is on the correct IOB pair. After the placement, the find results reflect that clk_p/clk_n are anchored to the AD12/AD11 IOB pair, as shown in Figure 3-4.



scc_lab_top
- Nets (670)
- Primitives (6)
- U1_AES1 (U1_AES1#aes)
- U2_AES2 (U2_AES2#aes_r)
- U3_Comp (U3_Comp#compare)
- U4_IOB (U4_IOB#io_inout)

clk_p

Name: clk_p
Direction: Input
Site: AD12 ☑ Fixed
Package pin: AD12

General | Attributes | Configure | Power

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

	Id	Name	Direction	Interface	Neg Diff Pair	Site	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip Termination	IN_TE
☑	1	clk_p	Input		clk_n	AD12	☑	33	default (DIFF_HSTL_II_18)					NONE	NONE	NONE

I/O Ports - Name matches "clk_n" (1)

X1085_c3_04_021513

Figure 3-4: **Drag clk_p to the Specified Site**

12. Repeat step 2 to step 11 to place reset at AK4.
13. Repeat step 2 to step 11 to place push_button at P27.
14. Repeat step 2 to step 11 to place led at AB9.
15. Repeat step 2 to step 11 to place mode at Y25.
16. Repeat step 2 to step 11 to place key_sel at Y28.
17. Repeat step 2 to step 11 to place key_addr[0] at AA28.
18. Repeat step 2 to step 11 to place key_addr[1] at W29.
19. Repeat step 2 to step 11 to place key_addr[2] at Y29.

The user can choose to drag and drop the I/O pins for a bus individually if the placement location of each pin is critical. However, to avoid manually placing many pins in a bus, the PlanAhead tool has an option to drag and drop the complete bus to a starting I/O pin location and auto-fill the pins within a specified I/O bank.

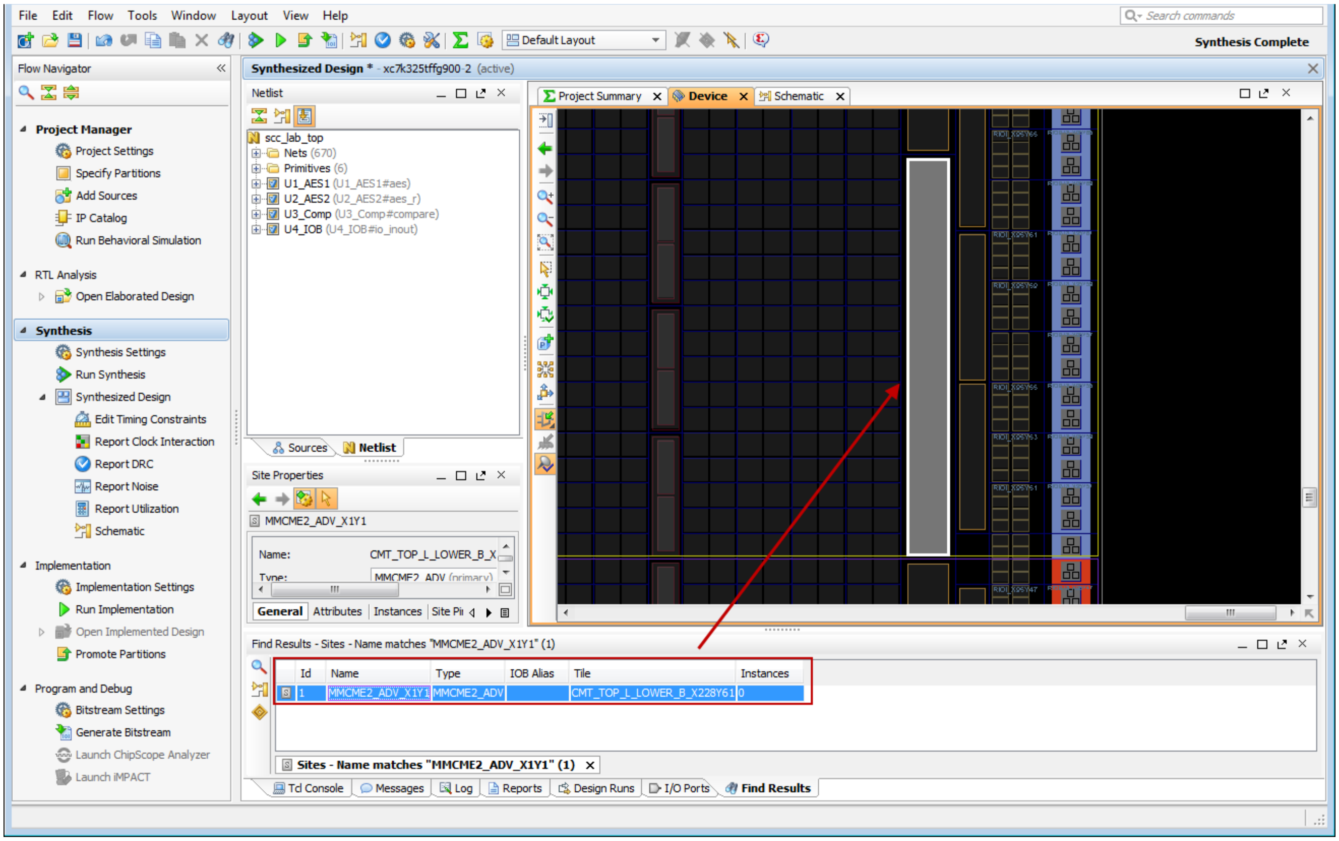
To place the I/O pins for a bus, all the signals of the bus must be selected at one time and placed at the top or bottom I/O pin of the specified I/O bank. The PlanAhead tool auto-fills the pins into the selected I/O bank. If there are more signals in the bus than can fit into one I/O bank, the PlanAhead tool must place the remaining group of signals already selected and waiting for pin assignment into the next chosen I/O bank. In the Device Planner window, scroll to the next desired I/O pin location in the desired I/O bank and drop the remaining group of the bus on that pin. The rest of the I/O pins for the bus are placed automatically.

20. The data_pins bus, data_pins[63:0], is selected to start I/O placement at the top of I/O bank 15 at pin location M19. Select **Edit > Find** (shortcut: **Ctrl-F**). A Find window appears.
21. Select **Sites** from the Find pull-down menu.
22. Select **Name** and **matches** from the two pull-down menus under Criteria.
23. Type **M19** in the remaining field box and click **OK**.
24. Select the find result and press **F9** to zoom to the current selection.
25. To place data_pins[63:0] starting on pin M19, select **Edit** > **Find** (shortcut: **Ctrl-F**).
26. Select **I/O Ports** from the Find: pull-down menu.
27. Select **Name** and **matches** from the two pull-down menus under Criteria.
28. To select the entire bus, type **data_pins*** in the field box and click **OK**.
29. Select the entire bus in the Find Results window and drag the result, **data_pins[63:0]**, to the site (M19) identified in step 20 through step 24.
30. All 64 bus pins do not fit in one I/O bank (50 I/Os per bank). Using the arrow keys, scroll down in the Device Planner window to expose pin R19 (top I/O pin of the I/O bank directly below). Click pin **R19** to drop the remaining bus signals to be placed in I/O bank 14.
31. The key_pins[63:0] bus is selected to start I/O placement at the bottom of I/O bank 13 at pin location AE26. Select **Edit > Find** (shortcut: **Ctrl-F**). A Find window appears.
32. Select **Sites** from the Find pull-down menu.
33. Select **Name** and **matches** from the two pull-down menus under Criteria.
34. Type **AE26** in the remaining field box and click **OK**.
35. Select the find result and press **F9** to zoom to the current selection.
36. To place key_pins[63:0] starting on pin AE26, select **Edit** > **Find** (shortcut: **Ctrl-F**).
37. Select **I/O Ports** from the Find: pull-down menu.
38. Select **Name** and **matches** from the two pull-down menus under Criteria.
39. To select the entire bus, type **key_pins*** in the field box and click **OK**.
40. Select the entire bus in the Find Results window and drag the result, **key_pins[63:0]**, to the site (AE26) identified in step 31 through step 35.
41. All 64 bus pins do not fit in one I/O bank (50 I/Os per bank). Using the arrow keys, scroll up in the Device Planner window to expose pin W19 (bottom I/O pin of the I/O bank directly above). Click pin **W19** to drop the remaining bus signals to be placed in I/O bank 14.

Placing Clocking Resources

This procedure describes how to place the clocking resources using the PlanAhead tool.

1. Select **Edit > Find** (shortcut: **Ctrl-F**).
2. Select **Sites** from the Find: pull-down menu.
3. Select **Name** and **matches** from the two pull-down menus under Criteria.
4. Type **MMCME2_ADV_X1Y1** in the field box, and click **OK**. The search results appear on a tab in the Find Results section at the bottom left of the PlanAhead tool window. Select the result and press **F9** to zoom to the current selection as shown in Figure 3-5.



Synthesized Design * - xc7k325tffg900-2 (active)

Site Properties: MMCME2_ADV_X1Y1

Name: CMT_TOP_L_LOWER_B_X

Type: MMCME2_ADV (primary)

Find Results - Sites - Name matches "MMCME2_ADV_X1Y1" (1)

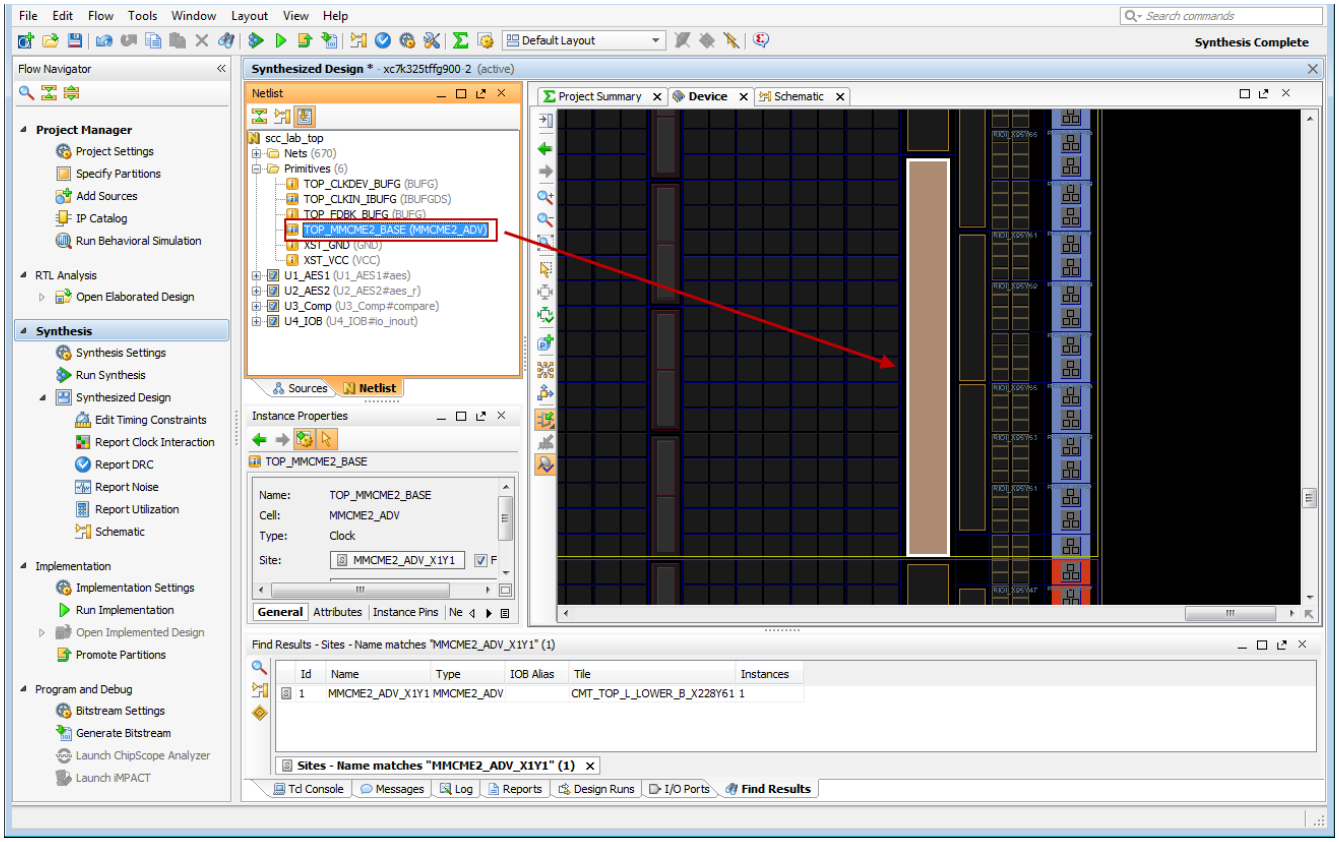
	Id	Name	Type	IOB Alias	Tile	Instances
S	1	MMCME2_ADV_X1Y1	MMCME2_ADV		CMT_TOP_L_LOWER_B_X228Y61	0

Sites - Name matches "MMCME2_ADV_X1Y1" (1)

X1085_c3_05_021513

Figure 3-5: **Search Results for MMCME2_ADV_X1Y1**

5. In the Netlist window, expand Primitives, then drag **TOP_MMCME2_BASE** to the MMCME2_ADV_X1Y1 box, as shown in Figure 3-6.



scc_lab_top
Nets (670)
Primitives (6)
TOP_CLKDEV_BUFG (BUFG)
TOP_CLKIN_IBUFG (IBUFGDS)
TOP_FDBK_BUFG (BUFG)
TOP_MMCME2_BASE (MMCME2_ADV)
XST_GND (GND)
XST_VCC (VCC)
U1_AES1 (U1_AES1#aes)
U2_AES2 (U2_AES2#aes_r)
U3_Comp (U3_Comp#compare)
U4_IOB (U4_IOB#io_inout)

TOP_MMCME2_BASE

Name: TOP_MMCME2_BASE
Cell: MMCME2_ADV
Type: Clock
Site: MMCME2_ADV_X1Y1 ☑ F

Id	Name	Type	IOB Alias	Tile	Instances
1	MMCME2_ADV_X1Y1	MMCME2_ADV		CMT_TOP_L_LOWER_B_X228Y61	1

X1085_c3_06_021513

Figure 3-6: **Placing MMCME2_ADV_X1Y1**

6. Repeat step 1 to step 5 to place **TOP_CLKDEV_BUFG** at BUFGCTRL_X0Y7.
7. Repeat step 1 to step 5 to place **TOP_FDBK_BUFG** at BUFGCTRL_X0Y6.

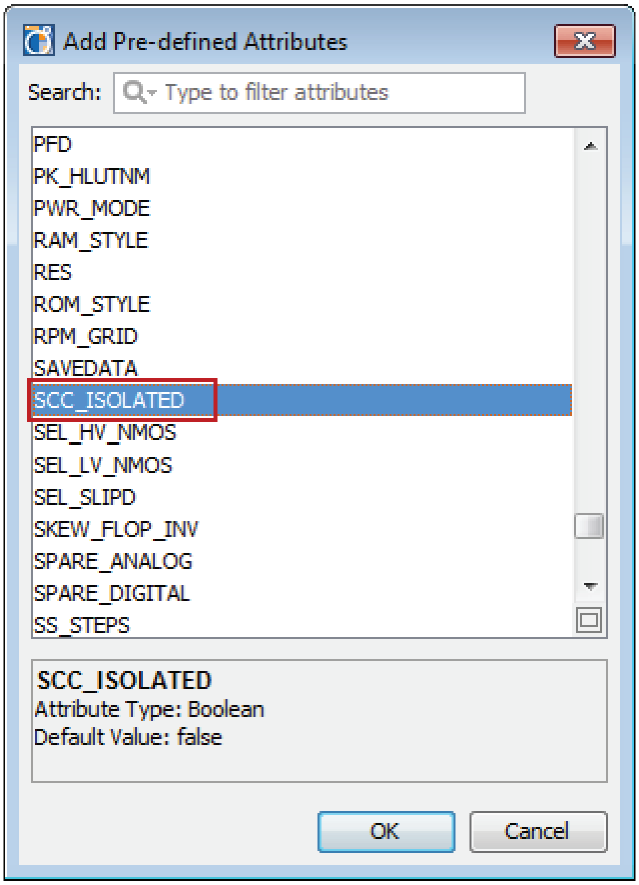
Defining Attributes for ISO Partition

IDF rules have many additional constraint rules as opposed to a traditional partial reconfiguration (PR) or hierarchical design flow. It is necessary to define several attributes for each partition to invoke these rules.

The SCC_ISOLATED attribute needs to be set for each partition for the back-end implementation tools to use the special Isolated Partition rules, which allow for floorplanning of an isolated design when using the IDF. If the SCC_ISOLATED attribute is not set, it is possible to floorplan only a limited subset of the FPGA components (CLB slices, RAMB, and DSP48E).

1. In the PlanAhead tool, under the Netlist tab, select and right-click the **U1_AES1** (AES) partition. Select **Instance Properties** from the resulting menu.
2. Select the Attributes tab of the Instance Properties window and click the green **+** button to add pre-defined attributes.

3. Select the **SCC_ ISOLATED** general attribute and click **OK** (see Figure 3-7).


Add Pre-defined Attributes

Search: Type to filter attributes

PFD
PK_HLUTNM
PWR_MODE
RAM_STYLE
RES
ROM_STYLE
RPM_GRID
SAVEDATA
SCC_ISOLATED
SEL_HV_NMOS
SEL_LV_NMOS
SEL_SLIPD
SKEW_FLOP_INV
SPARE_ANALOG
SPARE_DIGITAL
SS_STEPS

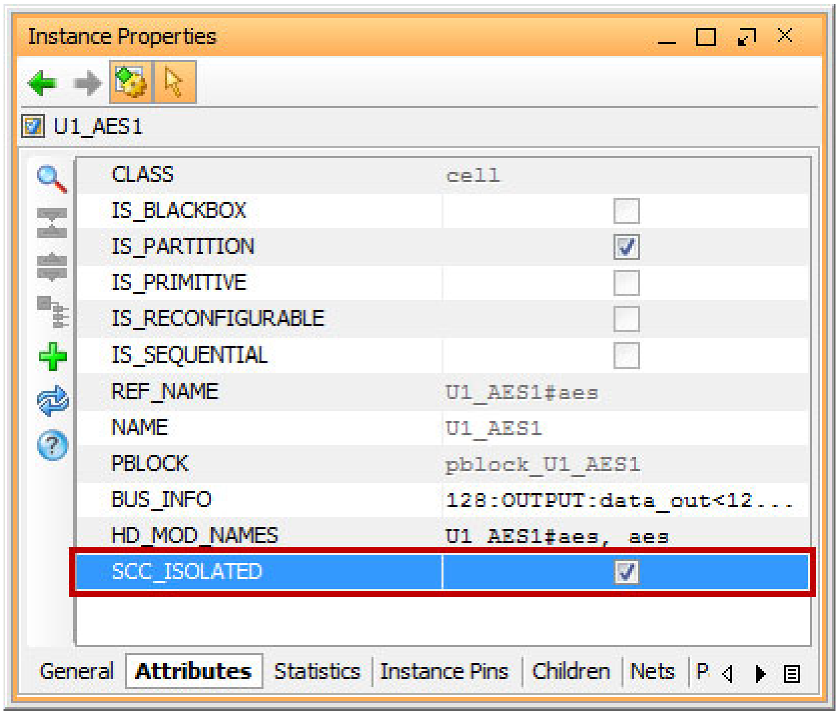
SCC_ISOLATED
Attribute Type: Boolean
Default Value: false

X1085_c3_07_022613

Figure 3-7: **Adding SCC_ISOLATED Attribute**

Note: Setting the SCC_ISOLATED attribute in the PlanAhead tool sets the Isolated=True parameter for that partition in the `xpartition.pxml` file.

4. Select the checkbox for SCC_ ISOLATED in the general attributes list in the Instance Properties window (see Figure 3-8).



CLASS	cell
IS_BLACKBOX	☐
IS_PARTITION	☑
IS_PRIMITIVE	☐
IS_RECONFIGURABLE	☐
IS_SEQUENTIAL	☐
REF_NAME	U1_AES1#aes
NAME	U1_AES1
PBLOCK	pblock_U1_AES1
BUS_INFO	128:OUTPUT:data_out<12...
HD_MOD_NAMES	U1_AES1#aes, aes
SCC_ISOLATED	☑

X1085_c3_08_021513

Figure 3-8: **Select SCC_ISOLATED Attribute**

5. Repeat step 1 to step 4 for the **U2_AES2** (AES_R) partition.
6. Repeat step 1 to step 4 for the **U3_Comp** (COMPARE) partition.
7. Repeat step 1 to step 4 for the **U4_IOB** (IO_INOUT) partition.

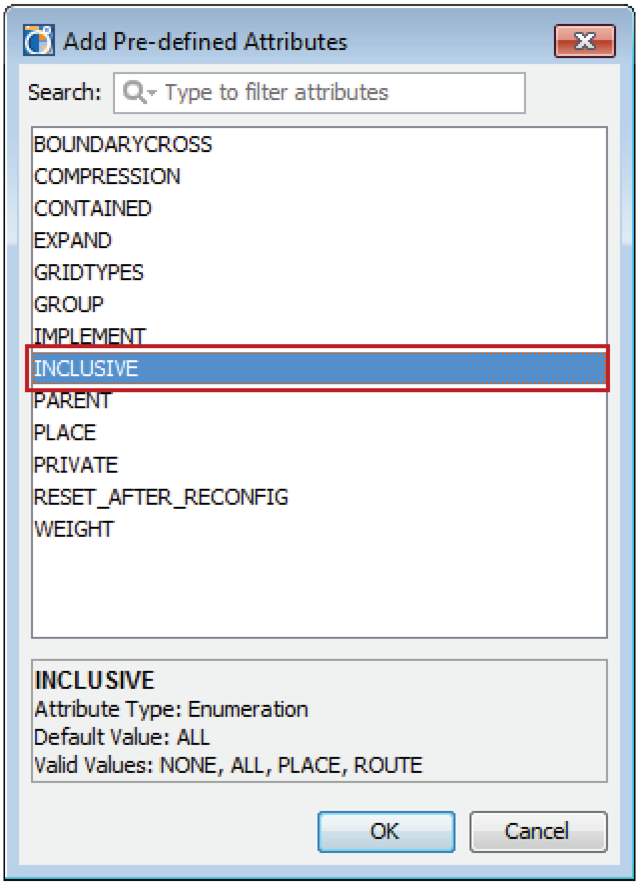
The INCLUSIVE attribute needs to be set to **ALL** for each pblock to allow the back-end implementation tools to include all the resources within the area range.

8. Under the Physical Constraints tab, select and right-click the **pblock_U1_AES** pblock. Select **Pblock Properties** in the resulting menu.

   Note: If the Physical Constraints tab is not viewable, open it by selecting **Window > Physical Constraints**.

9. Click the **Attributes** tab of the Pblock Properties window, and click the green **+** button to add pre-defined attributes.

10. Select **INCLUSIVE** and click **OK** (see Figure 3-9).

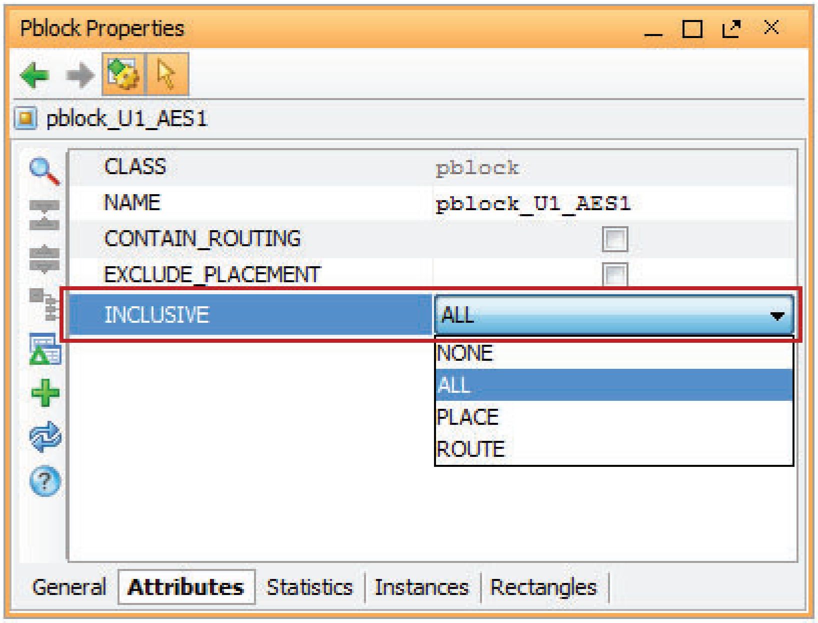


INCLUSIVE
Attribute Type: Enumeration
Default Value: ALL
Valid Values: NONE, ALL, PLACE, ROUTE

X1085_c3_09_022613

Figure 3-9: **Adding INCLUSIVE Attribute**

11. Select **INCLUSIVE** in the general attributes list in the Pblock Properties window and select **ALL** (see Figure 3-10).



pblock_U1_AES1

CLASS	pblock
NAME	pblock_U1_AES1
CONTAIN_ROUTING	□
EXCLUDE_PLACEMENT	□
INCLUSIVE	ALL

X1085_c3_10_022613

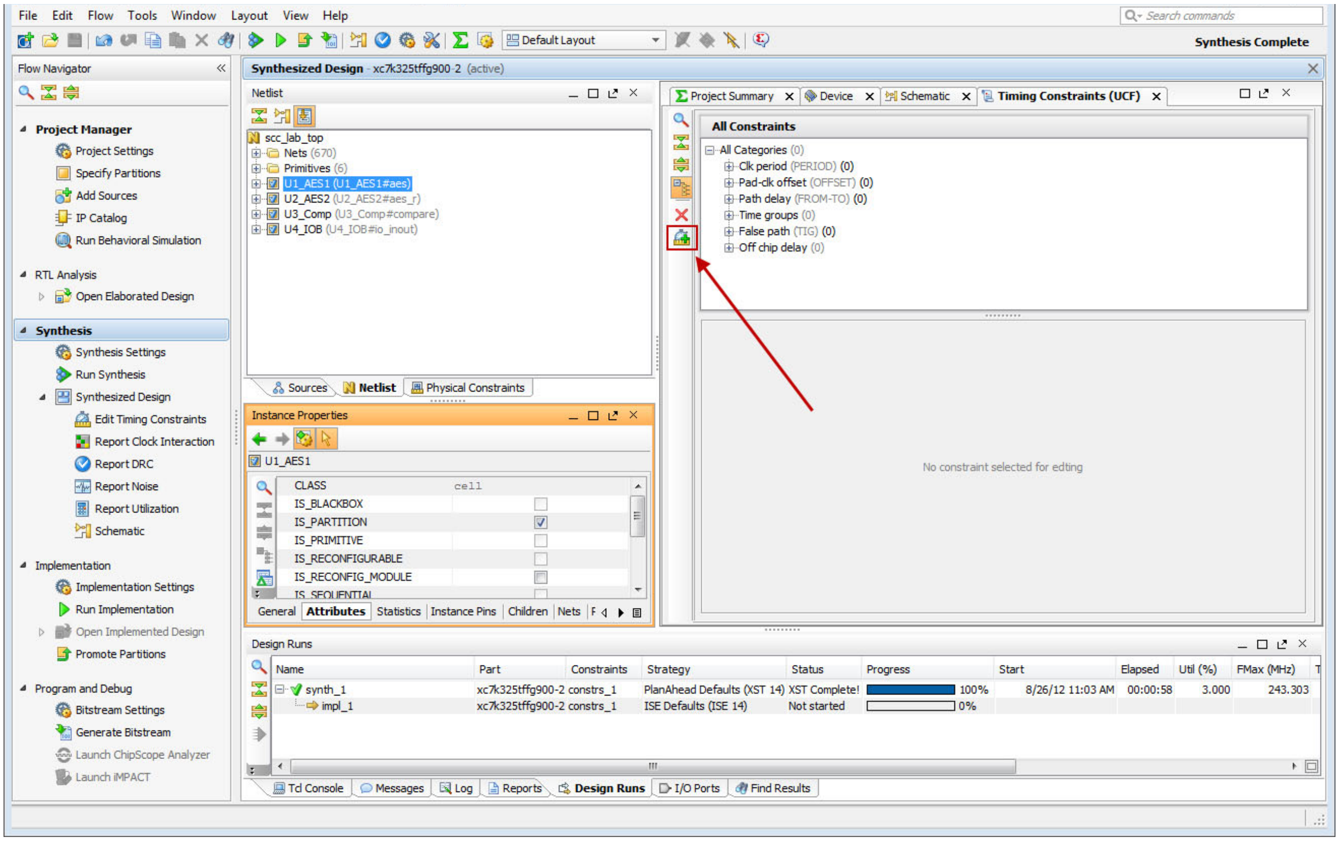
Figure 3-10: **Setting INCLUSIVE Attribute to ALL**

12. Repeat step 8 to step 11 for pblock_U2_AES2 and set INCLUSIVE to **ALL**.
13. Repeat step 8 to step 11 for pblock_U3_Comp and set INCLUSIVE to **ALL**.
14. Repeat step 8 to step 11 for pblock_U4_IOB and set INCLUSIVE to **ALL**.
15. Save the project to update the target constraints file with the added constraints.

Setting Up Timing Constraints

This section describes how to apply timing constraints to each module and the top-level design using the PlanAhead tool.

1. Select **Timing Constraints** under the Window option in the PlanAhead tool to display the Timing Constraints window.
2. In the Timing Constraints tab, click the **Create Timing Constraint** button located at the left side of the Constraints window (see Figure 3-11).



CLASS	cell
IS_BLACKBOX	☐
IS_PARTITION	☑
IS_PRIMITIVE	☐
IS_RECONFIGURABLE	☐
IS_RECONFIG_MODULE	☐
IS_SEQUENTIAL	☐

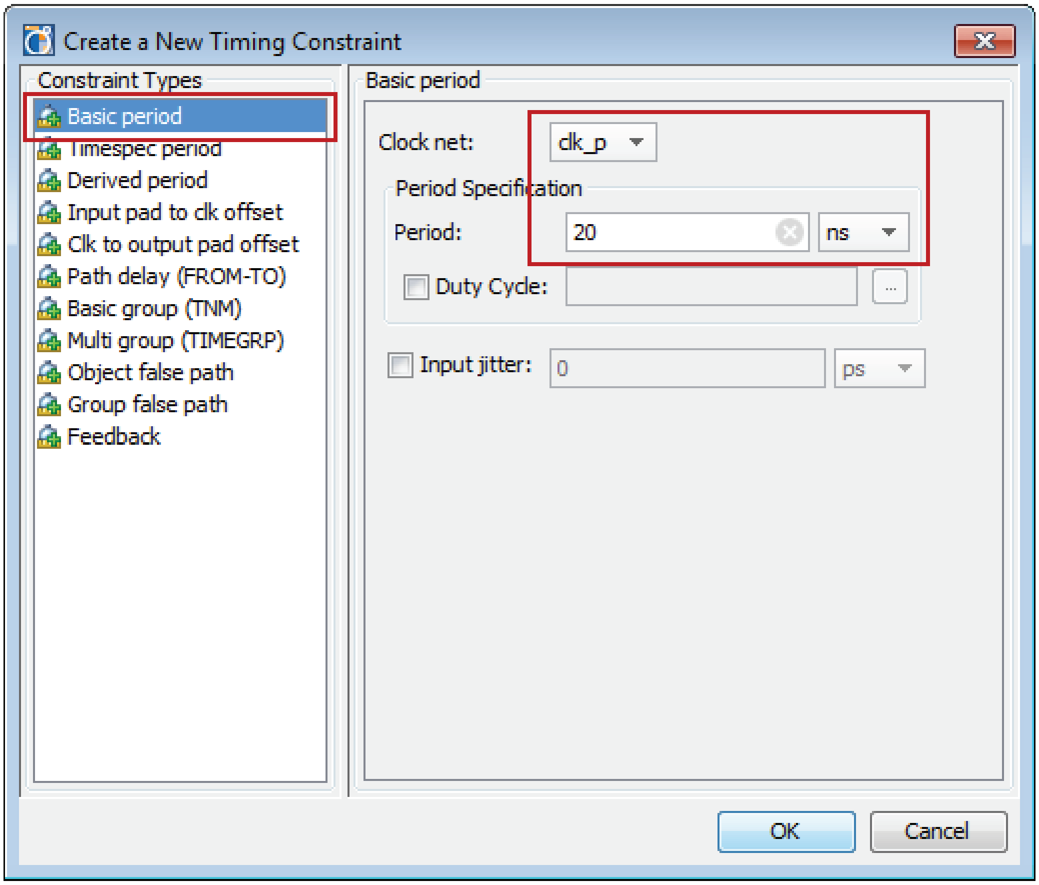
Name	Part	Constraints	Strategy	Status	Progress	Start	Elapsed	Util (%)	FMax (MHz)
synth_1	xc7k325tffg900-2	constrs_1	PlanAhead Defaults (XST 14)	XST Complete!	100%	8/26/12 11:03 AM	00:00:58	3.000	243.303
impl_1	xc7k325tffg900-2	constrs_1	ISE Defaults (ISE 14)	Not started	0%				

X1085_c3_11_021513

Figure 3-11: **Create New Timing Constraint**

3. Add a Basic Period constraint of 20 ns to the clk_p clock input by entering the value as shown in Figure 3-12.

   Note: Optionally, the *Xilinx Timing Constraints User Guide* recommends using the Timespec Period constraint instead of the Basic Period to have the constraints properly propagate through the mixed-mode clock manager (MMCM) and create the appropriate period constraints on the output clocks of the MMCM.



Create a New Timing Constraint

Constraint Types: Basic period (selected)

Basic period

Clock net: clk_p

Period Specification

Period: 20 ns

□ Duty Cycle:

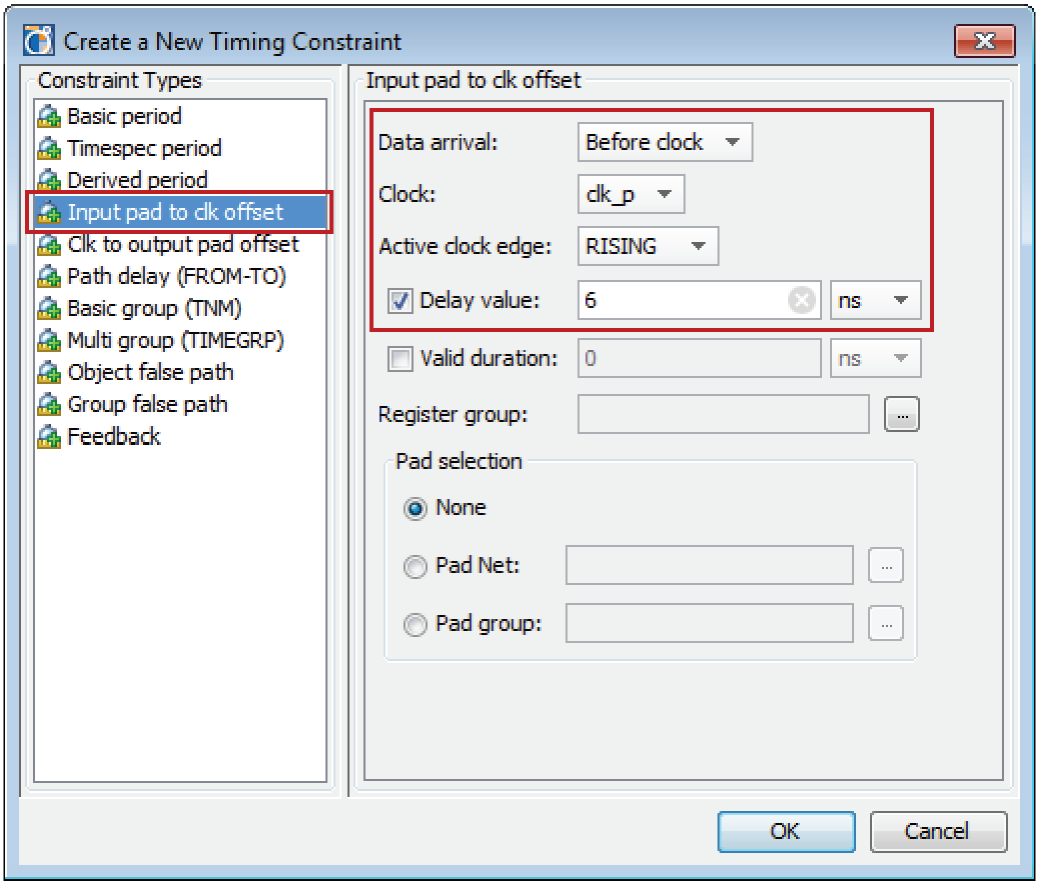
□ Input jitter: 0 ps

X1085_c3_12_022613

Figure 3-12: **New Basic Period Constraint**

4. Click **OK**.

5. Create a new timing constraint by selecting **Input pad to clk offset**. Set the Data arrival option to **Before clock**. Set Clock to **clk_p**. Set Active clock edge to **RISING**. Check the **Delay value** box and add a global input constraint of **6 ns** to the clock input by entering the value as shown in Figure 3-13.



Input pad to clk offset

Data arrival: Before clock

Clock: clk_p

Active clock edge: RISING

☑ Delay value: 6 ns

□ Valid duration: 0 ns

Register group:

Pad selection

● None

○ Pad Net:

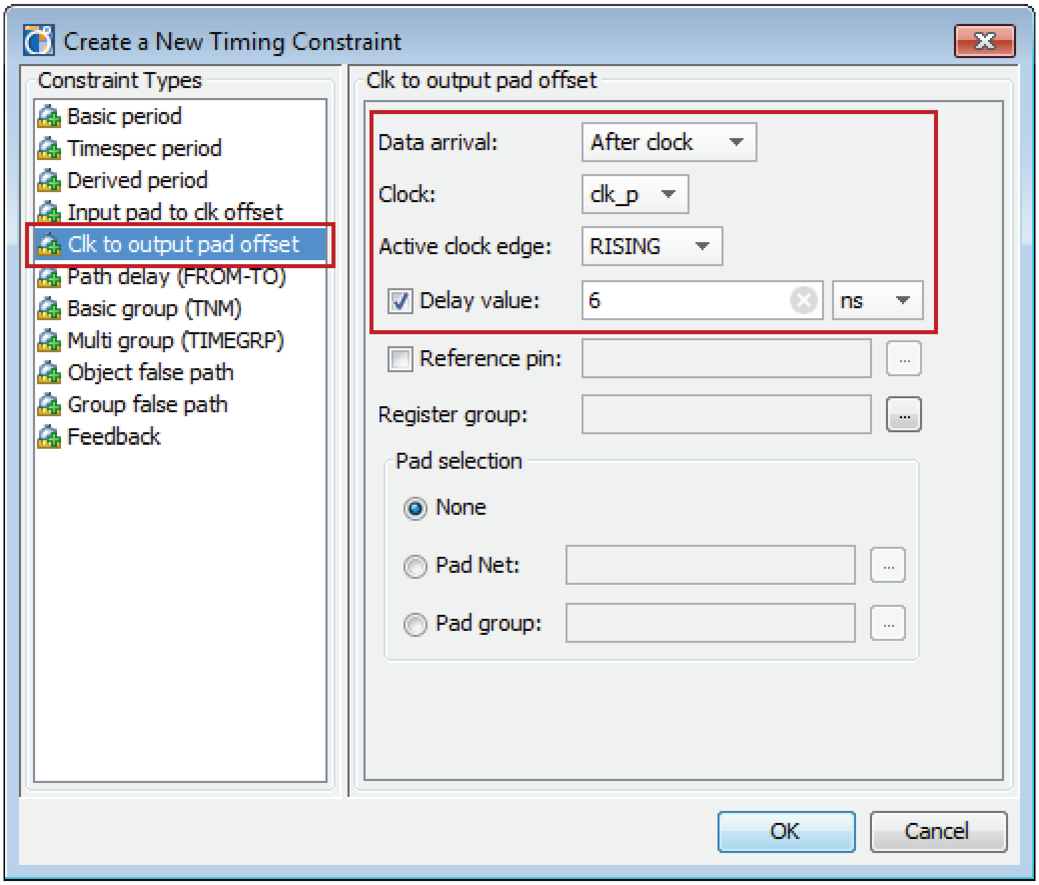
○ Pad group:

X1085_c3_13_022613

Figure 3-13: **New Timing Constraint – Input to Clk**

6. Click **OK**.

7. Create a new timing constraint by selecting **Clk to output pad offset**. Set the Data arrival option to **After clock**. Set Clock to **clk_p**. Set Active clock edge to **RISING**. Check the **Delay value** box and add a global output constraint of **6 ns** to the clock input by entering the value as shown in Figure 3-14.



Create a New Timing Constraint

Clk to output pad offset

Data arrival: After clock

Clock: clk_p

Active clock edge: RISING

☑ Delay value: 6 ns

□ Reference pin:

Register group:

Pad selection

● None

○ Pad Net:

○ Pad group:

Figure 3-14: **New Timing Constraint – Clk to Output**

8. Click **OK**.

Setting Up Area Groups for ISO Regions

Area groups constrain FPGA resources. Their size can be changed using the PlanAhead tool. For this exercise, the specific dimensions of the area groups are presented. However, a general method for configuring an area group for an arbitrary ISO region is demonstrated in this section:

Note: It is recommended to save the project often during the manual floorplanning steps, especially between creating new area groups.

1. Under the Physical Constraints tab, select and right-click the **pblock_U1_AES1** (AES) block. Select **Set Pblock Size** from the menu.

2. For reference, Figure 3-15 shows where on the device the first isolated region, pblock_U1_AES1 (AES), is drawn. The upper-left corner of the pblock is started near the intersection of clock regions X0Y2, X1Y2, X0Y1, and X1Y1 of the Design Planner window. It is helpful to zoom out while searching for the area referenced in Figure 3-15 to locate the first pblock.

   Note: In PlanAhead 14.4, if the coordinates of the rectangle are known, the user can enter the X/Y grid locations of the rectangle directly into the properties window, which might be easier than trying to size the area group graphically.

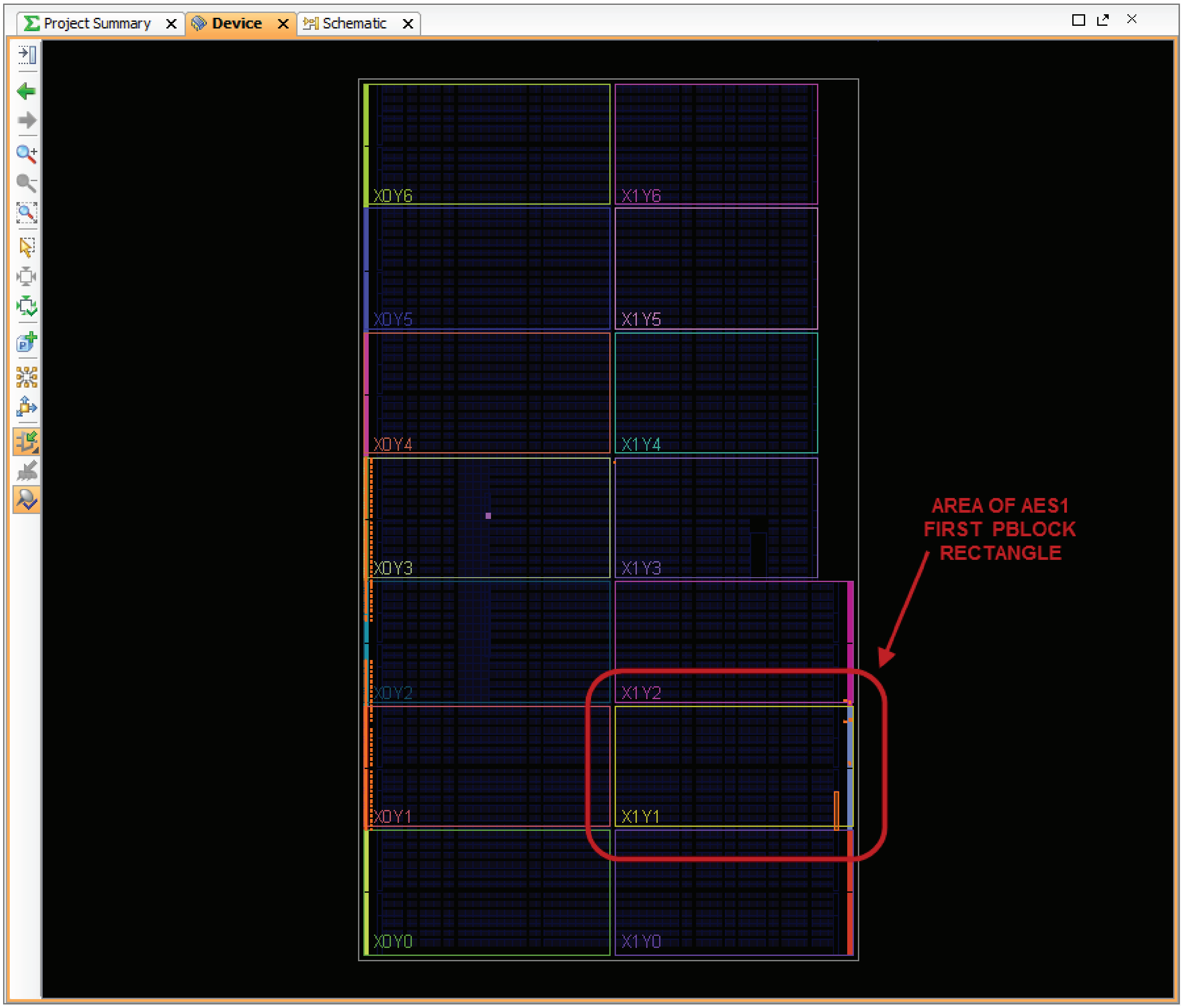
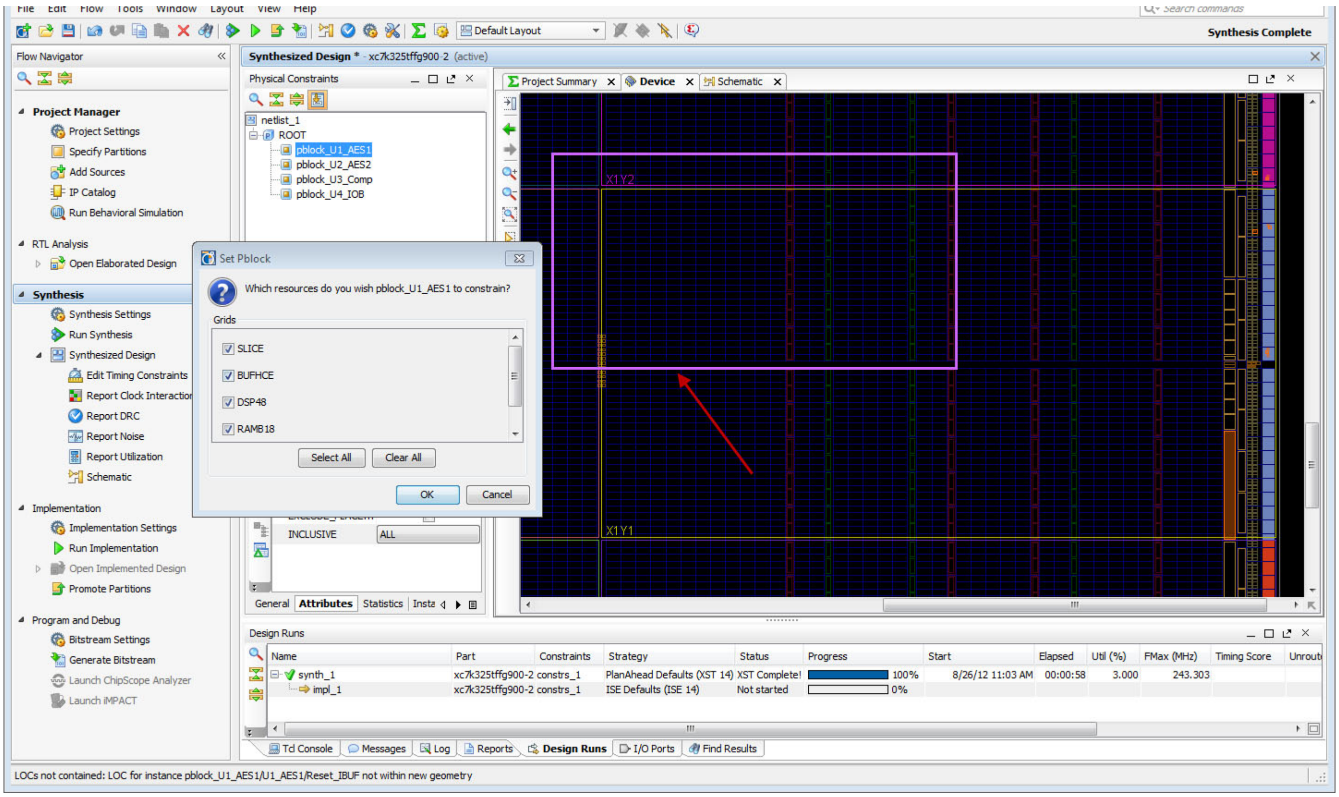


Figure 3-15: **XC7K325T-2FFG900 Whole View Reference for First Pblock**

3. Draw a rectangle, as shown in Figure 3-16 (accuracy is not important at this stage.)



Set Pblock

Which resources do you wish pblock_U1_AES1 to constrain?

Grids

☑ SLICE

☑ BUFHCE

☑ DSP48

☑ RAMB18

Select All | Clear All

OK | Cancel

Design Runs

Name	Part	Constraints	Strategy	Status	Progress	Start	Elapsed	Util (%)	FMax (MHz)	Timing Score	Unrout
synth_1	xc7k325tffg900-2	constrs_1	PlanAhead Defaults (XST 14)	XST Complete!	100%	8/26/12 11:03 AM	00:00:58	3.000	243.303		
impl_1	xc7k325tffg900-2	constrs_1	ISE Defaults (ISE 14)	Not started	0%						

LOCs not contained: LOC for instance pblock_U1_AES1/U1_AES1/Reset_IBUF not within new geometry

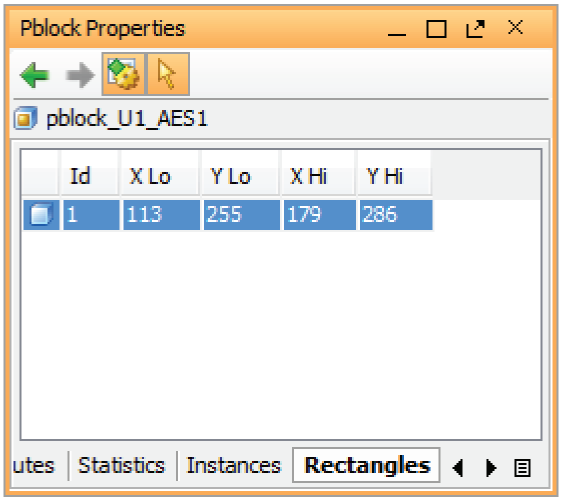
X1085_c3_16_021513

Figure 3-16: **Set Pblock – pblock_U1_AES1**

4. A dialog box (shown in Figure 3-16) appears with the available resources and associated checkboxes. All boxes should be checked including all of the buffers (BUFGCTRL, BUFHCE, BUFR, BUFIO, and BUFMRCE). Even though most of these blocks might not be used in the design, it is important to select them to take advantage of their routing resources. The only exception to this is when not specifically using the CMT clocking components (MMCME2_ADV, PLLE2_ADV, PHASER_IN_PHY, PHASER_OUT_PHY, PHY_CONTROL, IN_FIFO, and OUT_FIFO) in an area group, they should not be included.
5. For this pblock rectangle, no CMT clocking components appear in the enclosed area, so none are displayed or checked in the dialog box.

   Note: IDF requires that all clock buffers and 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.
6. Click **OK**.
7. In the Choose LOC Mode dialog box, select **Leave all location constraints in their current position**.
8. Click **OK**.
9. Ensure that **pblock_U1_AES1** (AES) is selected in the Physical Constraints view.

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



	Id	X Lo	Y Lo	X Hi	Y Hi
	1	113	255	179	286

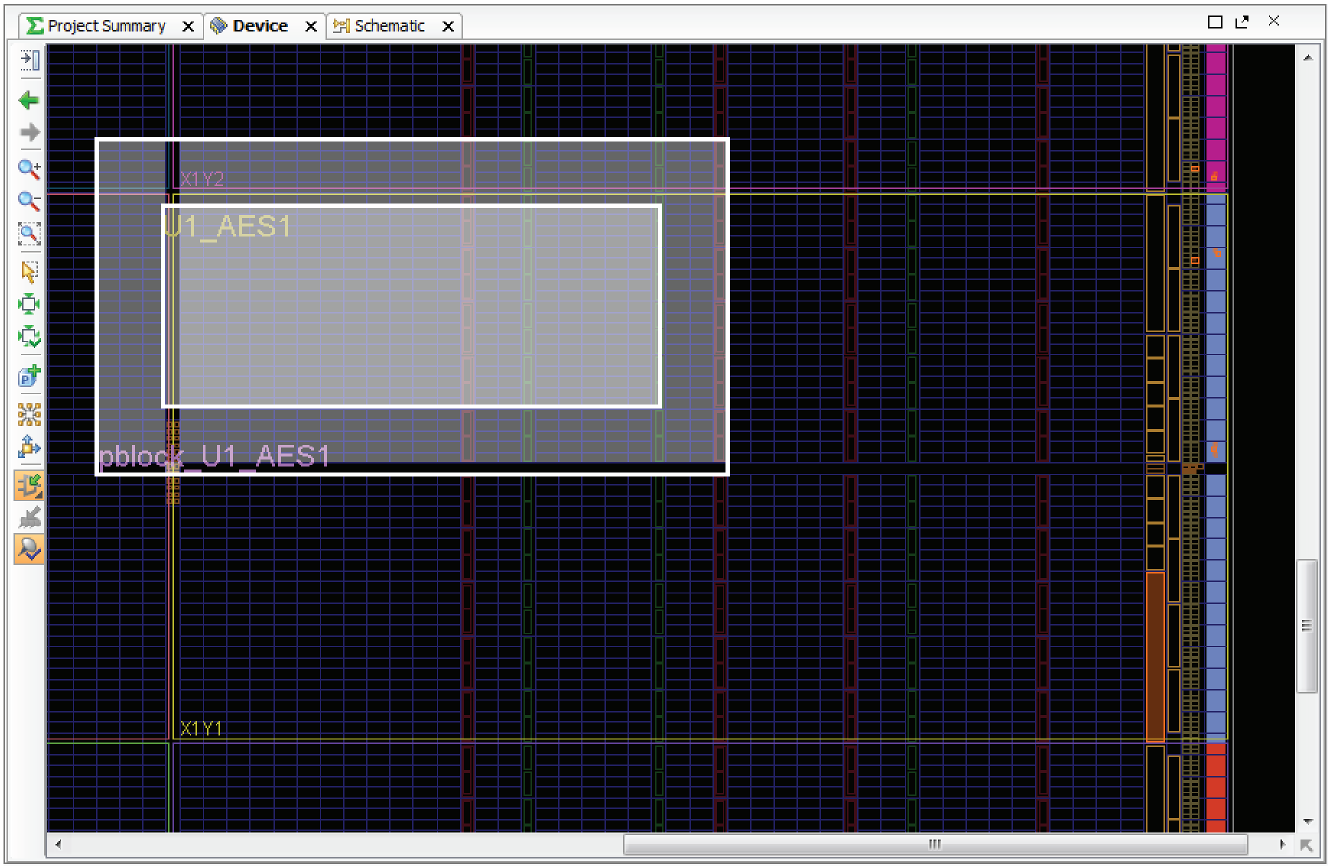
X1085_c3_17_022613

Figure 3-17: **Pblock Properties**

11. After the preliminary U1_AES1 (AES) block is drawn, adjust the rectangle graphically as necessary to match these coordinates (the final pblock rectangle should look like that shown in Figure 3-18):

    - X Lo = 113
    - Y Lo = 255
    - X Hi = 179
    - Y Hi = 286

    Note: Pblock coordinates also can be adjusted by typing the correct coordinates directly into the Pblock Properties Rectangles tab.

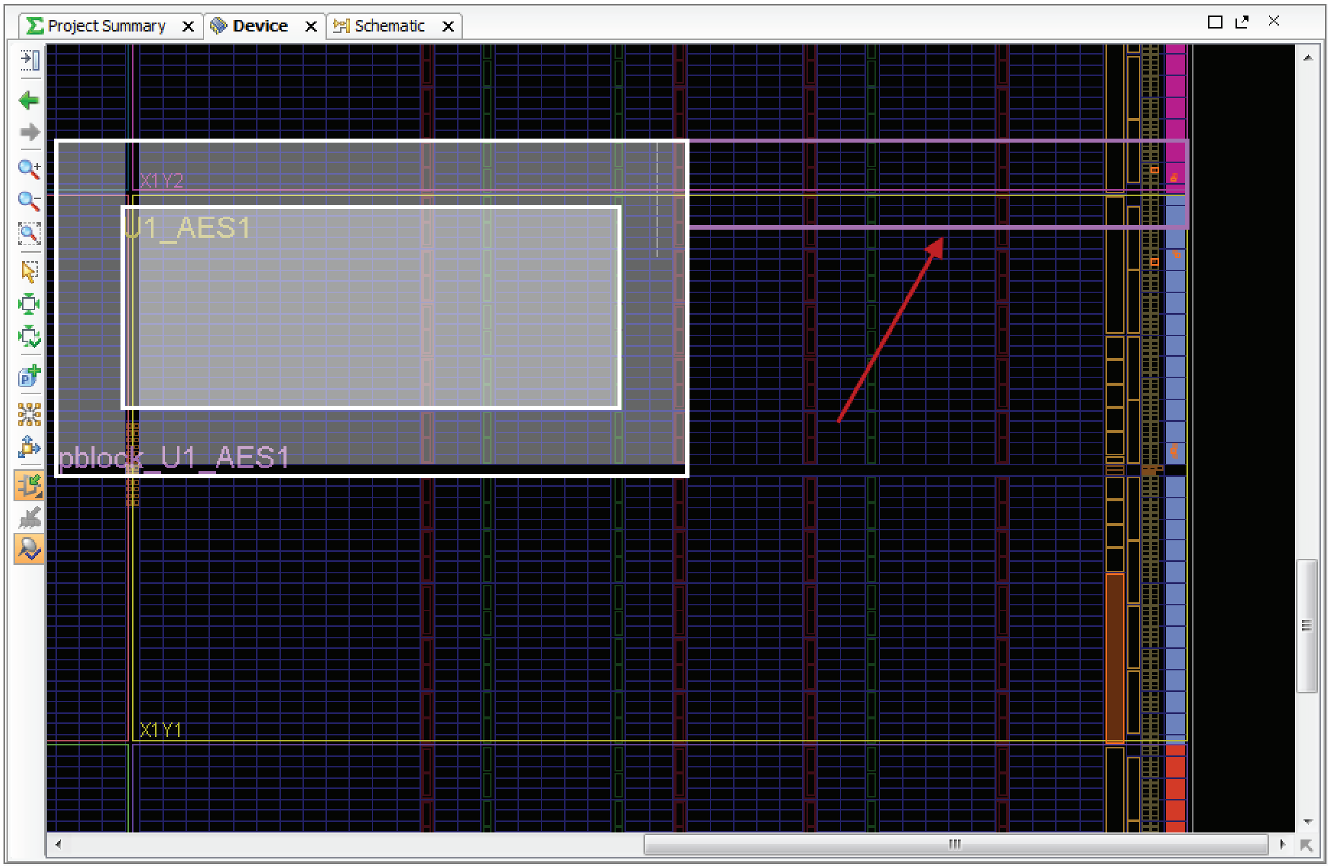


X1085_c3_18_022613

Figure 3-18: **First pblock_U1_AES1 Pblock Floorplan**

12. The second rectangle is drawn for pblock_U1_AES1. Under the Physical Constraints tab, select **pblock_U1_AES1**.
13. Right-click **pblock_U1_AES1** and select **Add Pblock Rectangle** from the pull-down menu.

14. Draw a new rectangle to the right side of the first rectangle to make a resultant L-shaped area as shown in Figure 3-19 (accuracy is not important at this stage.)



X1085_c3_19_022613

Figure 3-19: **Add Pblock Rectangle – pblock_U1_AES1**

15. A dialog box appears with available ranges and associated checkboxes. All boxes should be checked including all of the buffers (BUFGCTRL, BUFHCE, BUFR, BUFIO, and BUFMRCE). Even though most of these blocks might not be used in the design, it is important to select them to take advantage of their routing resources.

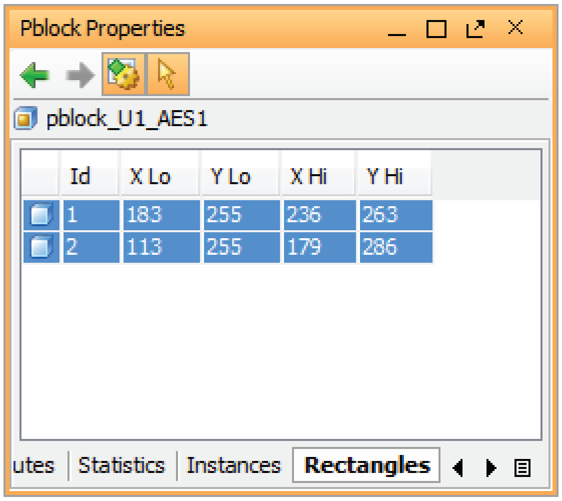
16. For this pblock rectangle, no CMT clocking components appear in the enclosed area so none are displayed or checked in the dialog box.

    Note: IDF requires that all clock buffers and 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.

17. Click **OK**.

18. Ensure that **pblock_U1_AES1** is selected in the Physical Constraints view.

19. Select the **Rectangles** tab in the Pblock Properties window to view all of the rectangles created for this pblock (Figure 3-20). Verify that the coordinates match what is displayed in Figure 3-20 and adjust accordingly.



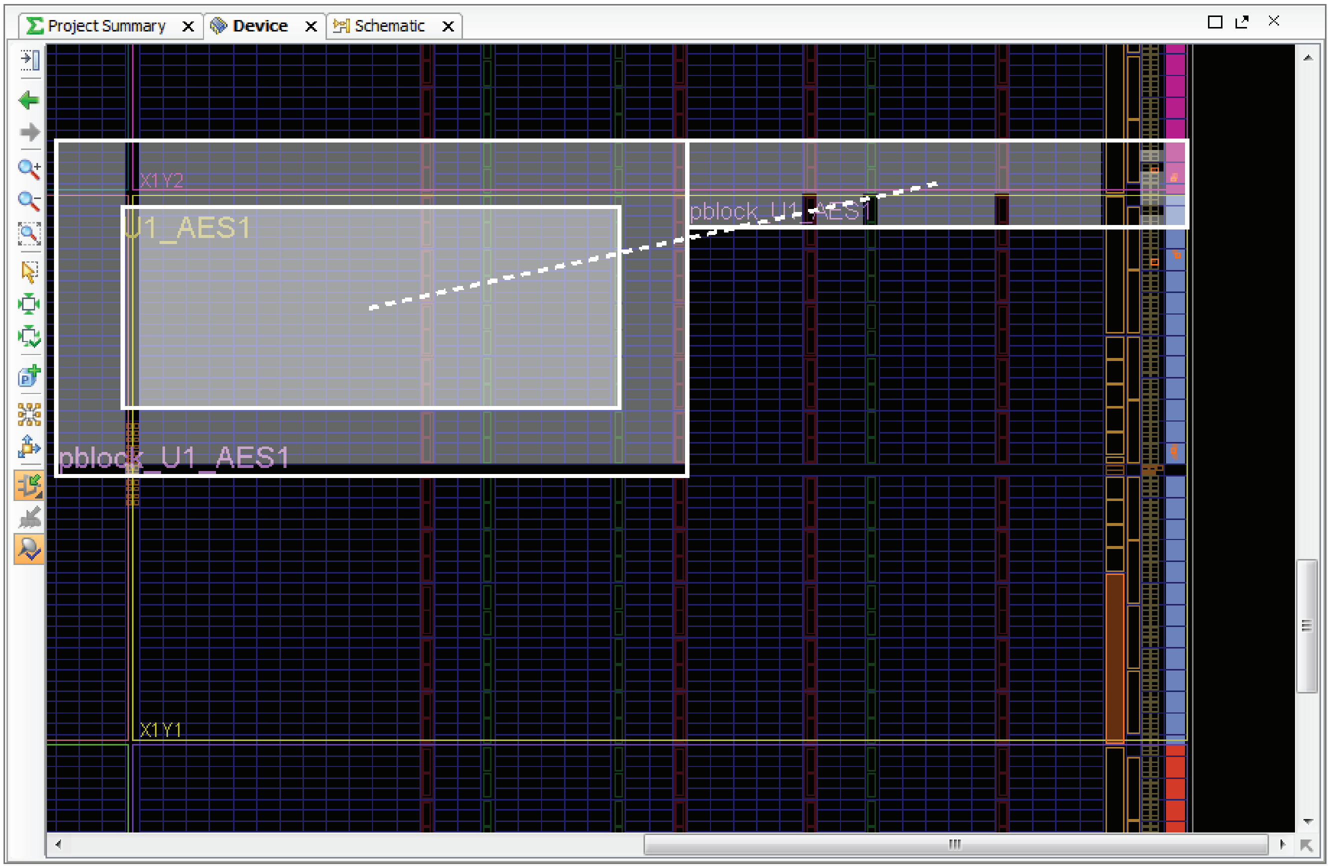
Id	X Lo	Y Lo	X Hi	Y Hi
1	183	255	236	263
2	113	255	179	286

X1085_c3_20_022613

Figure 3-20: **Pblock Properties**

Note: The PlanAhead tool might resize the rectangle to many different combinations of smaller rectangles. Make sure that all of the needed resources inside the area group are fully enclosed (such as block RAMs, DSPs, and MMCMs).

The first two pblocks for pblock_U1_AES1 are shown in Figure 3-21.



X1085_c3_21_022613

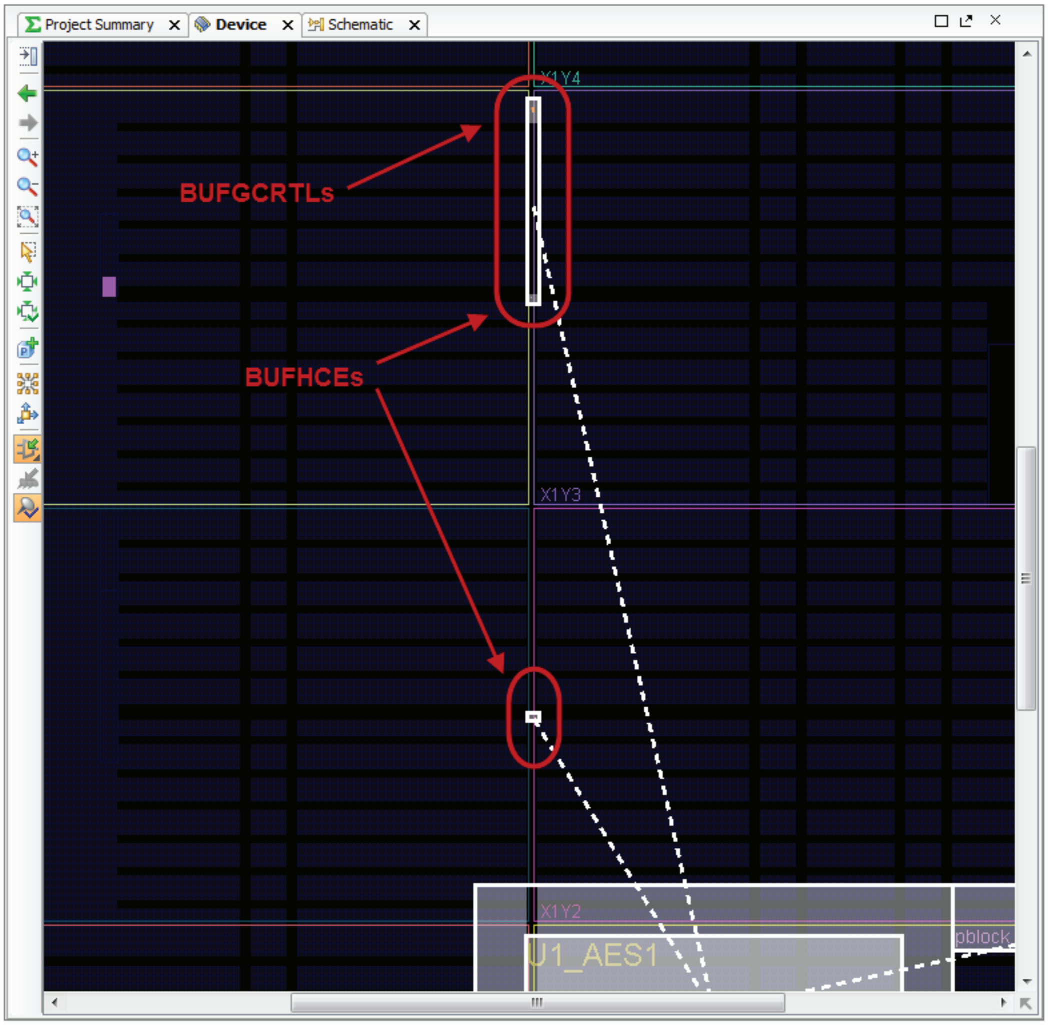
Figure 3-21: **First and Second pblock_U1_AES1 Pblocks Floorplan**

As stated in Placing I/Os, page 27, all clock resources used must be physically owned by one of the isolated regions. The two BUFGCTRL clock buffers used in this design must be included in pblock_U1_AES1. There are also BUFHCE buffers in the center column of the device that must be included in pblock_U1_AES1. All of these buffers are enclosed in two separate rectangles as described here:

1. The third and fourth rectangles are drawn for pblock_U1_AES1. Under the Physical Constraints tab, select **pblock_U1_AES1**.
2. For each of the two rectangles, right-click **pblock_U1_AES1** and select **Add Pblock Rectangle** from the pull-down menu.

3. Draw a new rectangle above the existing pblock_U1_AES1 rectangle to enclose the BUFGCTRL and BUFHCE buffers as shown in Figure 3-22.

   Note: It is better to make the rectangles bigger than needed because the PlanAhead tool resizes the rectangles to include the required BUFGCTRL and BUFHCE buffers.



X1085_c3_22_022613

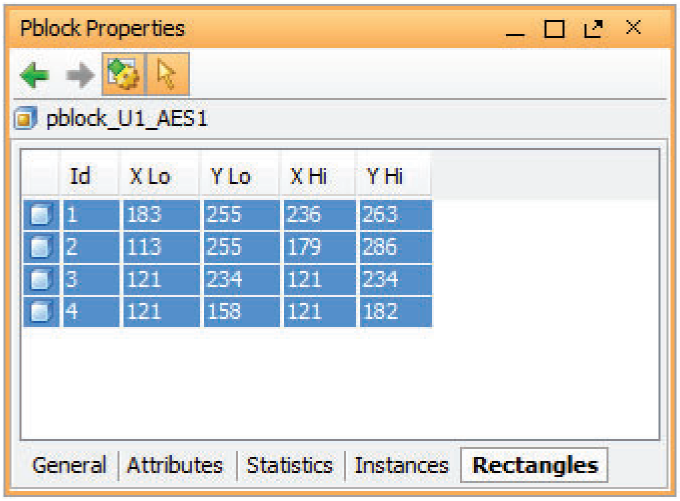
Figure 3-22: **Add Pblock Rectangle – Additional Pblocks for pblock_U1_AES1**

4. A dialog box appears with available resources and associated checkboxes for each rectangle drawn. Both boxes must be checked. The only resources that should appear in this pblock are BUFGCTRL and BUFHCE.

   Note: IDF requires that all clock buffers and 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.

5. Click **OK** for each rectangle.
6. Ensure that **pblock_U1_AES1** is selected in the Physical Constraints view.

7. Select the **Rectangles** tab in the Pblock Properties window to view all of the rectangles created for this pblock (Figure 3-23). Verify that the coordinates match what is displayed in Figure 3-23 and adjust accordingly.



Pblock Properties

pblock_U1_AES1

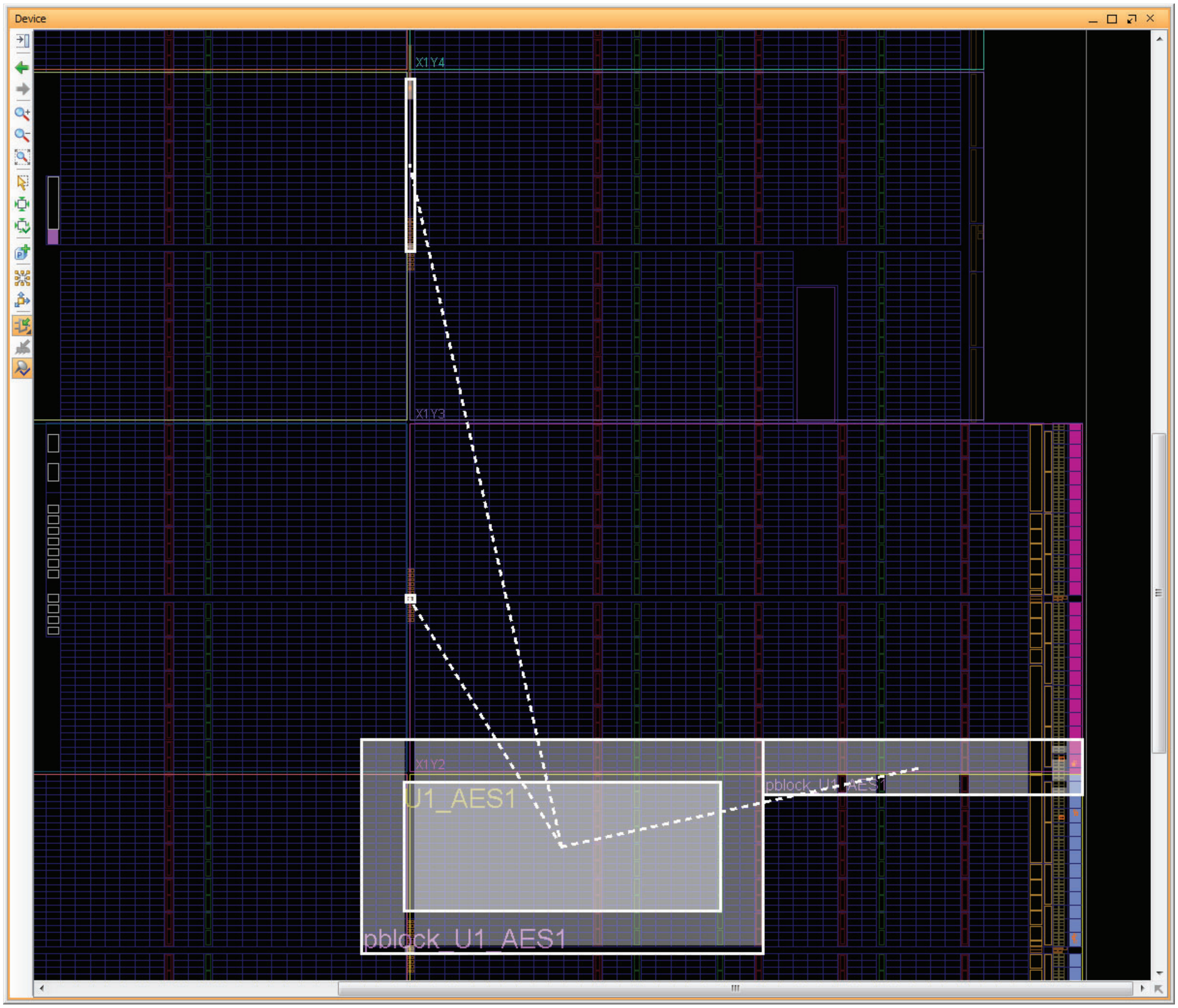
Id	X Lo	Y Lo	X Hi	Y Hi
1	183	255	236	263
2	113	255	179	286
3	121	234	121	234
4	121	158	121	182

X1085_c3_23_022613

Figure 3-23: **Pblock Properties**

Note: The PlanAhead tool might resize the rectangle to many different combinations of smaller rectangles. Make sure that all of the needed resources inside the area group that are required are fully enclosed (such as block RAMs, DSPs, and MMCMs).

The completed final pblock including the rectangles for the BUFGCTRL and BUFHCE buffers for pblock_U1_AES1 is shown in Figure 3-24.



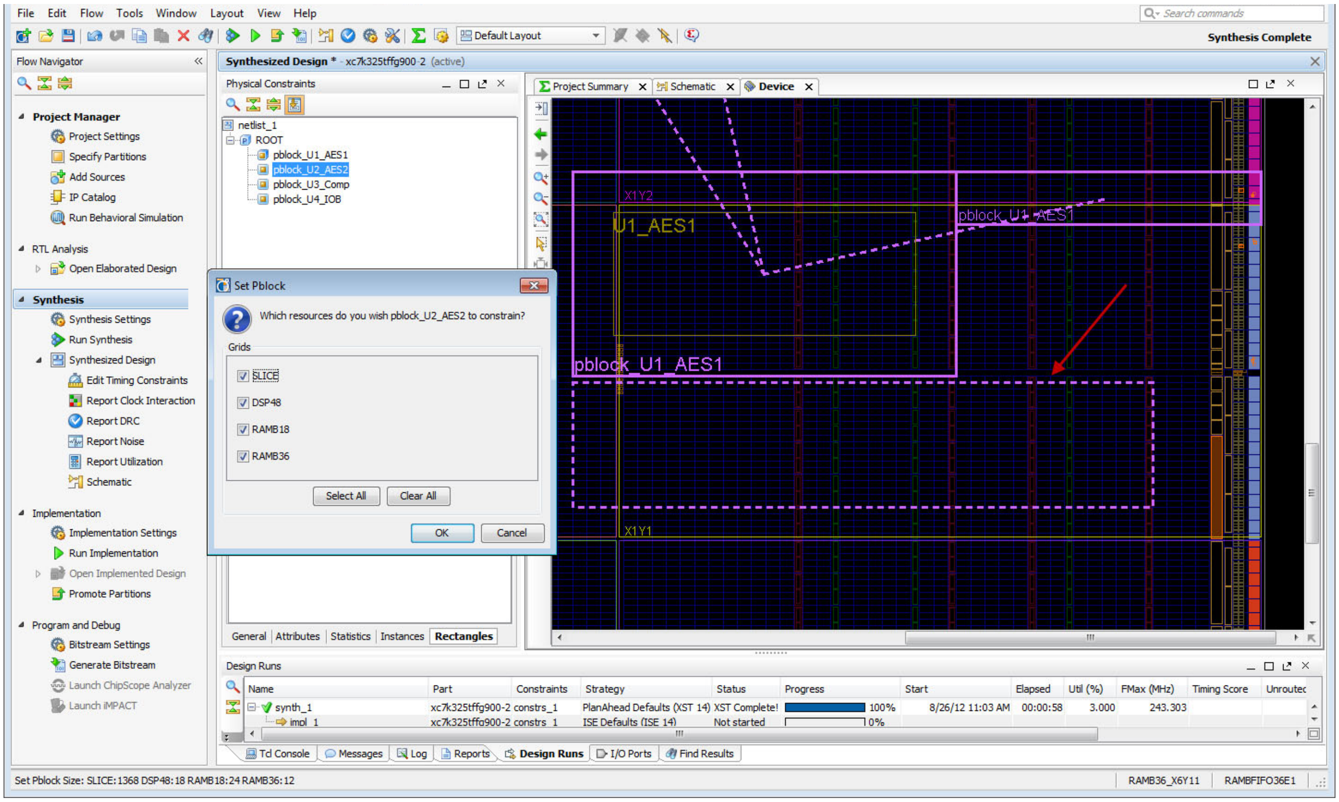
X1085_c3_24_022613

Figure 3-24: **Completed Final Pblock Floorplan for pblock_U1_AES1**

8. Under the Physical Constraints tab, select and right-click the block **pblock_U2_AES2**. Select **Set Pblock Size** from the menu.

9. Draw a rectangle (as shown in Figure 3-25) below the pblock area of AES1 in the Design Planner window (accuracy is not important at this stage).

   Note: To maintain the required isolation between isolated regions, leave one fence tile. For example, leave a configurable logic block (CLB) of isolation between the AES1 and AES2 partitions.



Set Pblock

Which resources do you wish pblock_U2_AES2 to constrain?

Grids

☑ SLICE

☑ DSP48

☑ RAMB18

☑ RAMB36

Select All | Clear All

OK | Cancel

Design Runs

Name	Part	Constraints	Strategy	Status	Progress	Start	Elapsed	Util (%)	FMax (MHz)	Timing Score	Unroutec
synth_1	xc7k325tffg900-2	constrs_1	PlanAhead Defaults (XST 14)	XST Complete!	100%	8/26/12 11:03 AM	00:00:58	3.000	243.303		
impl_1	xc7k325tffg900-2	constrs_1	ISE Defaults (ISE 14)	Not started	0%						

Set Pblock Size: SLICE:1368 DSP48:18 RAMB18:24 RAMB36:12 | RAMB36_X6Y11 | RAMBFIFO36E1

X1085_c3_25_021513

Figure 3-25: **Set Pblock – pblock_U2_AES2**

10. A dialog box (Figure 3-25) appears with available resources and associated checkboxes. All boxes should be checked, including all of the buffers (BUFGCTRL, BUFHCE, BUFR, BUFIO, and BUFMRCE). Even though most of these blocks might not be used in the design, it is important to select them to take advantage of their routing resources. The only exception to this is when not specifically using the CMT clocking components (MMCME2_ADV, PLLE2_ADV, PHASER_IN_PHY, PHASER_OUT_PHY, PHY_CONTROL, IN_FIFO, and OUT_FIFO) in an area group, they should not be included.

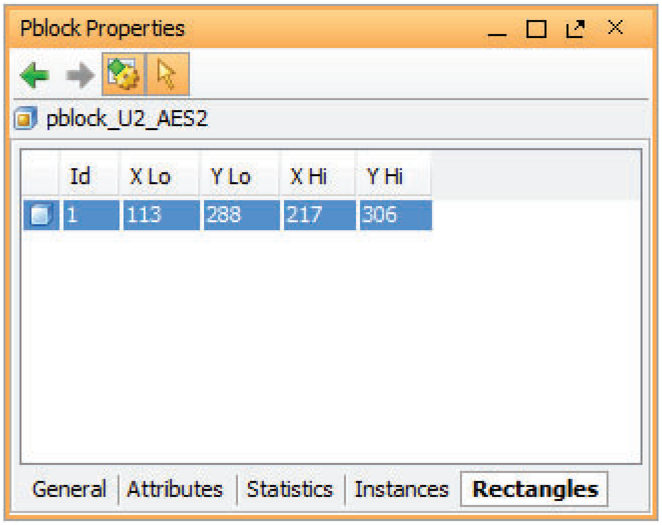
11. For this pblock rectangle, no CMT clocking components appear in the enclosed area so none are displayed or checked in the dialog box.

    Note: IDF requires that all clock buffers and 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.

12. Click **OK**.

13. Ensure that **pblock_U2_AES2** is selected in the Physical Constraints view.

14. Select the **Rectangles** tab in the Pblock Properties window (Figure 3-26).



Id	X Lo	Y Lo	X Hi	Y Hi
1	113	288	217	306

X1085_c3_26_022613

Figure 3-26: **Pblock Properties**

15. After the preliminary U2_AES2 block is drawn, adjust the rectangle graphically as necessary to match these coordinates (the pblock should look like that shown in Figure 3-27):
    - X Lo = 113
    - Y Lo = 288
    - X Hi = 217
    - Y Hi = 306

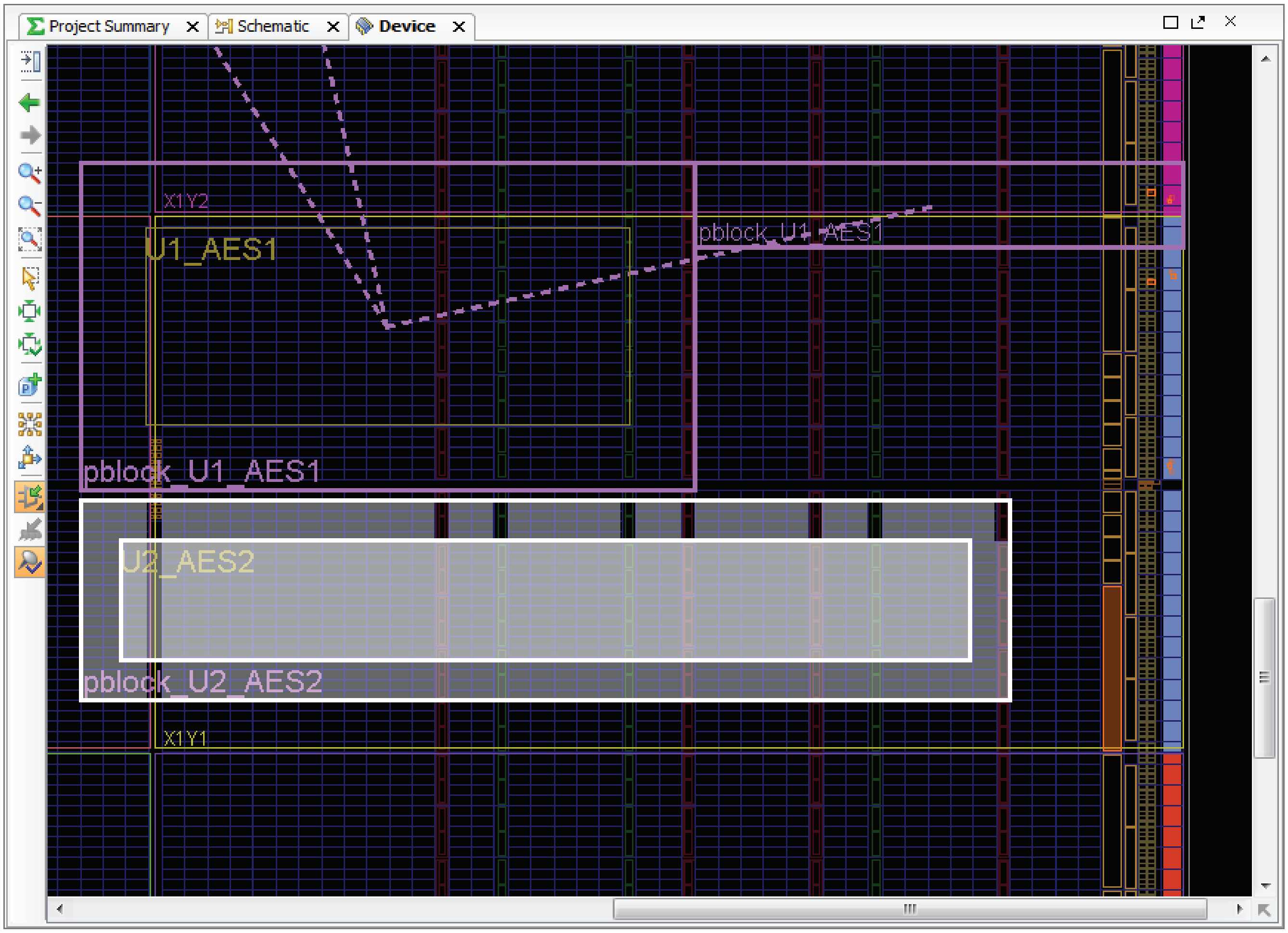
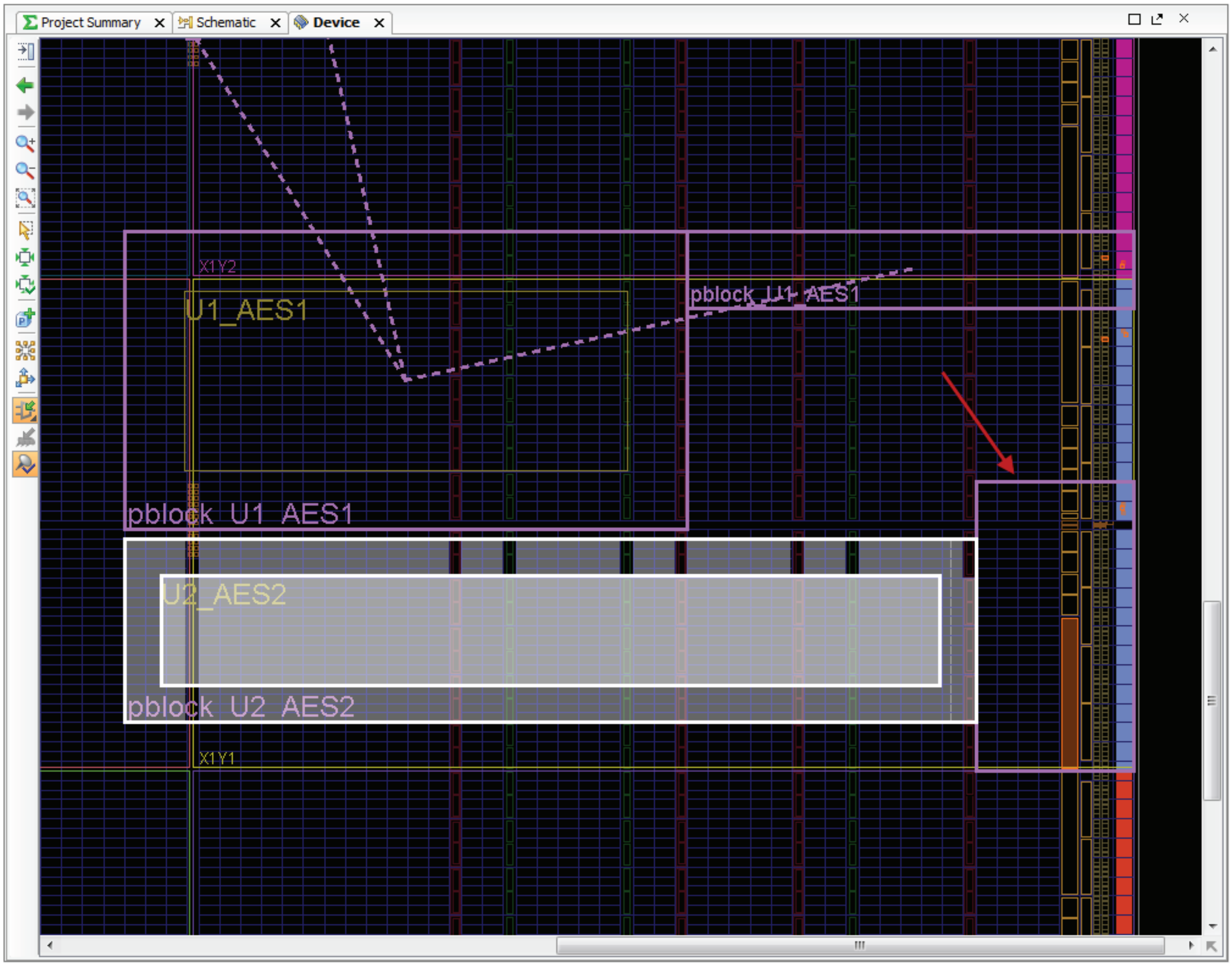


Figure 3-27: **First pblock_U2_AES2 Pblock Floorplan**

16. The second rectangle is drawn to complete pblock_U2_AES2. Under the Physical Constraints tab, select **pblock_U2_AES2**.
17. Right-click **pblock_U2_AES2** and select **Add Pblock Rectangle** from the pull-down menu.

18. Draw a new rectangle to the right side of the first rectangle to make a resultant T-shaped area as shown in Figure 3-28 (accuracy is not important at this stage).



X1085_c3_28_022613

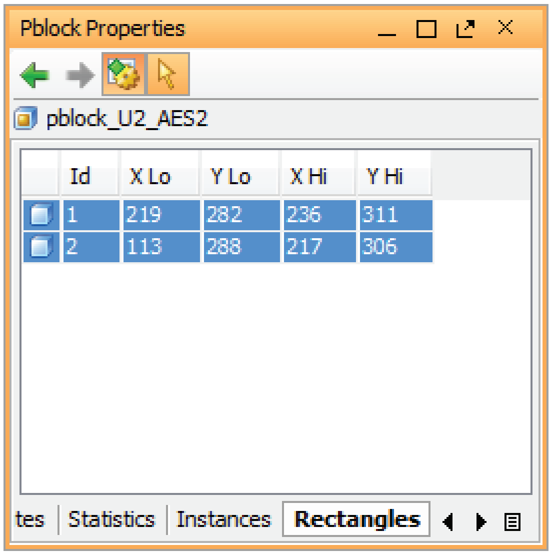
Figure 3-28: **Add Pblock Rectangle – Second pblock_U2_AES2**

19. A dialog box appears with available ranges and associated checkboxes. All boxes should be checked, including all of the buffers (BUFGCTRL, BUFHCE, BUFR, BUFIO, and BUFMRCE). Even though most of these blocks might not be used in the design, it is important to select them to take advantage of their routing resources.

20. For the CMT clocking components in this Pblock rectangle, include (check) **MMCME2_ADV**, **IN_FIFO**, and **OUT_FIFO**. Do not include (uncheck) **PHASER_IN_PHY**, **PHASER_OUT_PHY**, **PHASER_REF**, and **PHY_CONTROL**. The MMCM is checked because it is the only CMT clocking component being used in this partition. The adjacent IN_FIFOs and OUT_FIFOs for the MMCM must also be included in the area group along with the MMCM. See (XAPP1086), *Developing Secure and Reliable Single FPGA Designs with Xilinx 7 Series FPGAs Using the Isolation Design Flow* for information on using the MMCM and the adjacent IO_FIFO.

Note: IDF requires that all clock buffers and 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.

21. Click **OK**.
22. Ensure that **pblock_U2_AES2** is selected in the Physical Constraints view.
23. Select the **Rectangles** tab in the Pblock Properties window to view all of the rectangles created for this pblock (Figure 3-29). Verify that the coordinates match what is displayed in Figure 3-29 and adjust accordingly.



Pblock Properties

pblock_U2_AES2

Id	X Lo	Y Lo	X Hi	Y Hi
1	219	282	236	311
2	113	288	217	306

X1085_c3_29_022613

Figure 3-29: **Pblock Properties**

Note: The PlanAhead tool might resize the rectangle to many different combinations of smaller rectangles. Make sure that all of the needed resources inside the area group are fully enclosed (such as block RAMs, DSPs, and MMCMs). Ensure that there is one fence tile, for example, a CLB, of separation between the pblocks AES1 and AES2.

The completed pblock for pblock_U2_AES2 (AES_R) is shown in Figure 3-30.

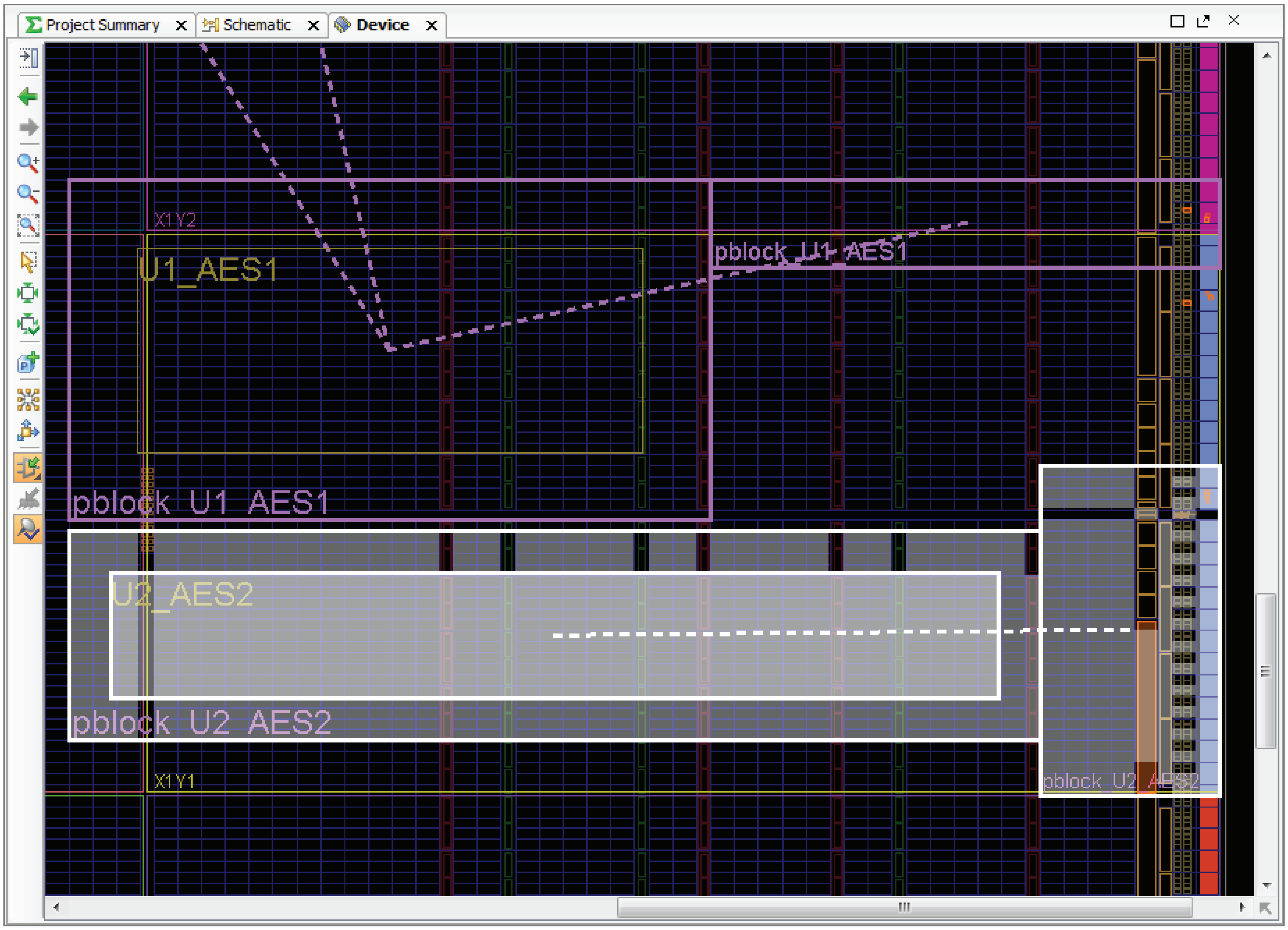
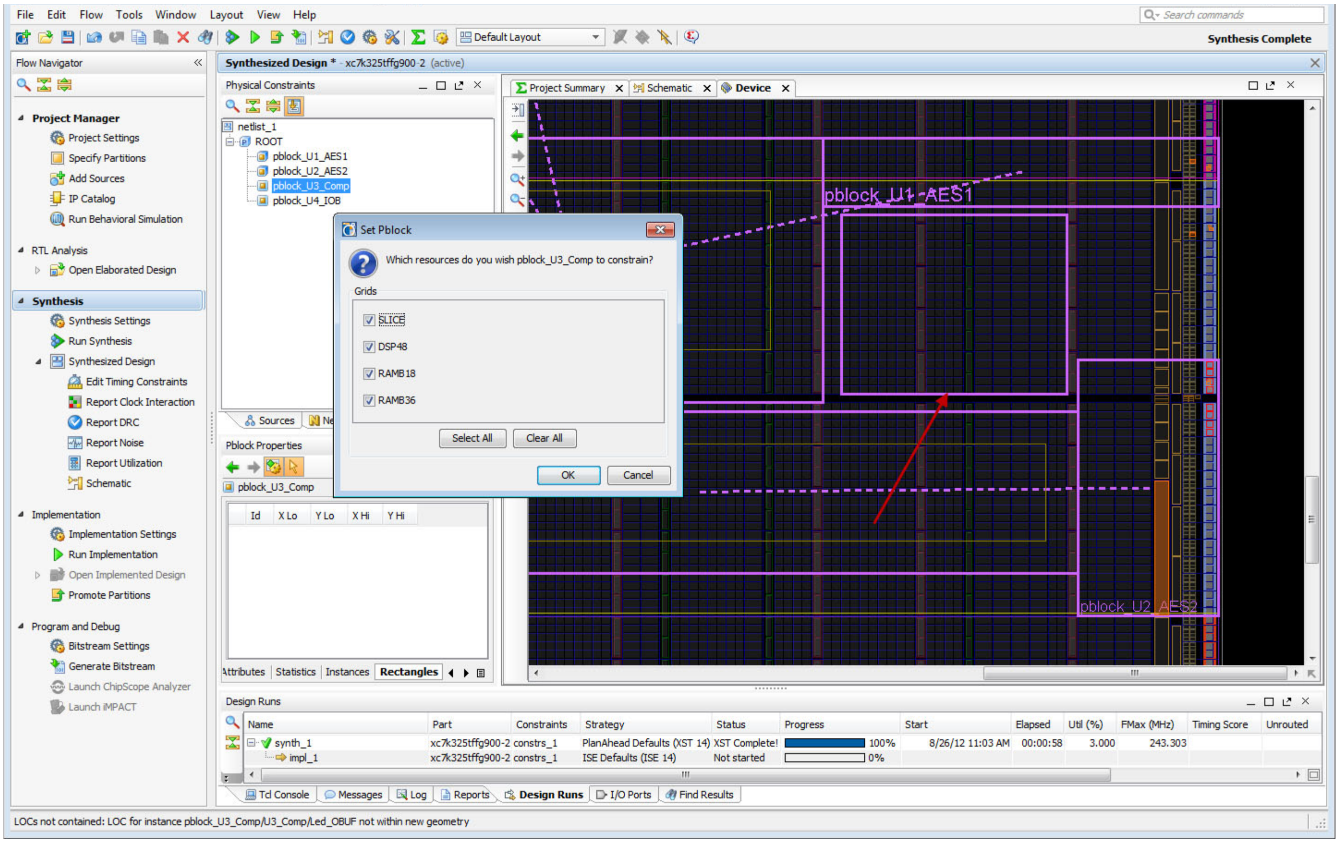


Figure 3-30: **Completed pblock_U2_AES2 Pblock Floorplan**

24. Under the Physical Constraints tab, select and right-click the block **pblock_U3_Comp**. Select **Set Pblock Size** from the menu.

25. Draw a rectangle as shown in Figure 3-31 between the AES1 and AES2 blocks (accuracy is not important at this stage). The Comp block has signals that route to both the AES1 and AES2 blocks so coincident borders are required between these blocks to provide the needed routing channels. Remember to leave a single fence tile for isolation around the perimeter of the Comp pblock.



Set Pblock

Which resources do you wish pblock_U3_Comp to constrain?

Grids

☑ SLICE
☑ DSP48
☑ RAMB18
☑ RAMB36

Select All | Clear All

OK | Cancel

Design Runs

Name	Part	Constraints	Strategy	Status	Progress	Start	Elapsed	Util (%)	FMax (MHz)	Timing Score	Unrouted
synth_1	xc7k325tffg900-2	constrs_1	PlanAhead Defaults (XST 14)	XST Complete!	100%	8/26/12 11:03 AM	00:00:58	3.000	243.303		
impl_1	xc7k325tffg900-2	constrs_1	ISE Defaults (ISE 14)	Not started	0%						

LOCs not contained: LOC for instance pblock_U3_Comp/U3_Comp/Led_OBUF not within new geometry

X1085_c3_31_021513

Figure 3-31: **Set Pblock – pblock_U3_Comp Pblock Floorplan**

26. A dialog box (shown in Figure 3-31) appears with available resources and associated checkboxes. All boxes should be checked, including all of the buffers (BUFGCTRL, BUFHCE, BUFR, BUFIO, and BUFMRCE). Even though most of these blocks might not be used in the design, it is important to select them to take advantage of their routing resources. The only exception to this is when not specifically using the CMT clocking components (MMCME2_ADV, PLLE2_ADV, PHASER_IN_PHY, PHASER_OUT_PHY, PHY_CONTROL, IN_FIFO, and OUT_FIFO) in an area group, they should not be included.

27. For this pblock rectangle, no CMT clocking components appear in the enclosed area so none are displayed or checked in the dialog box.

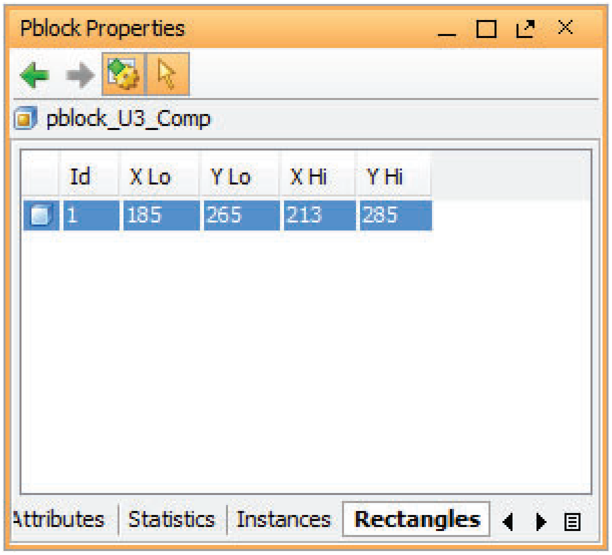
Note: IDF requires that all clock buffers and clocking components be assigned to the AREA GROUP that physically contains the component, even though the component is not logically instantiated in the HDL of that module.

28. Click **OK**.

29. In the Choose LOC mode dialog box, select **Leave all location constraints in their current position**.

30. Click **OK**.

31. Ensure that **pblock_U3_Comp** is selected in the Physical Constraints view.
32. Select the **Rectangles** tab in the Pblock Properties window (Figure 3-32).

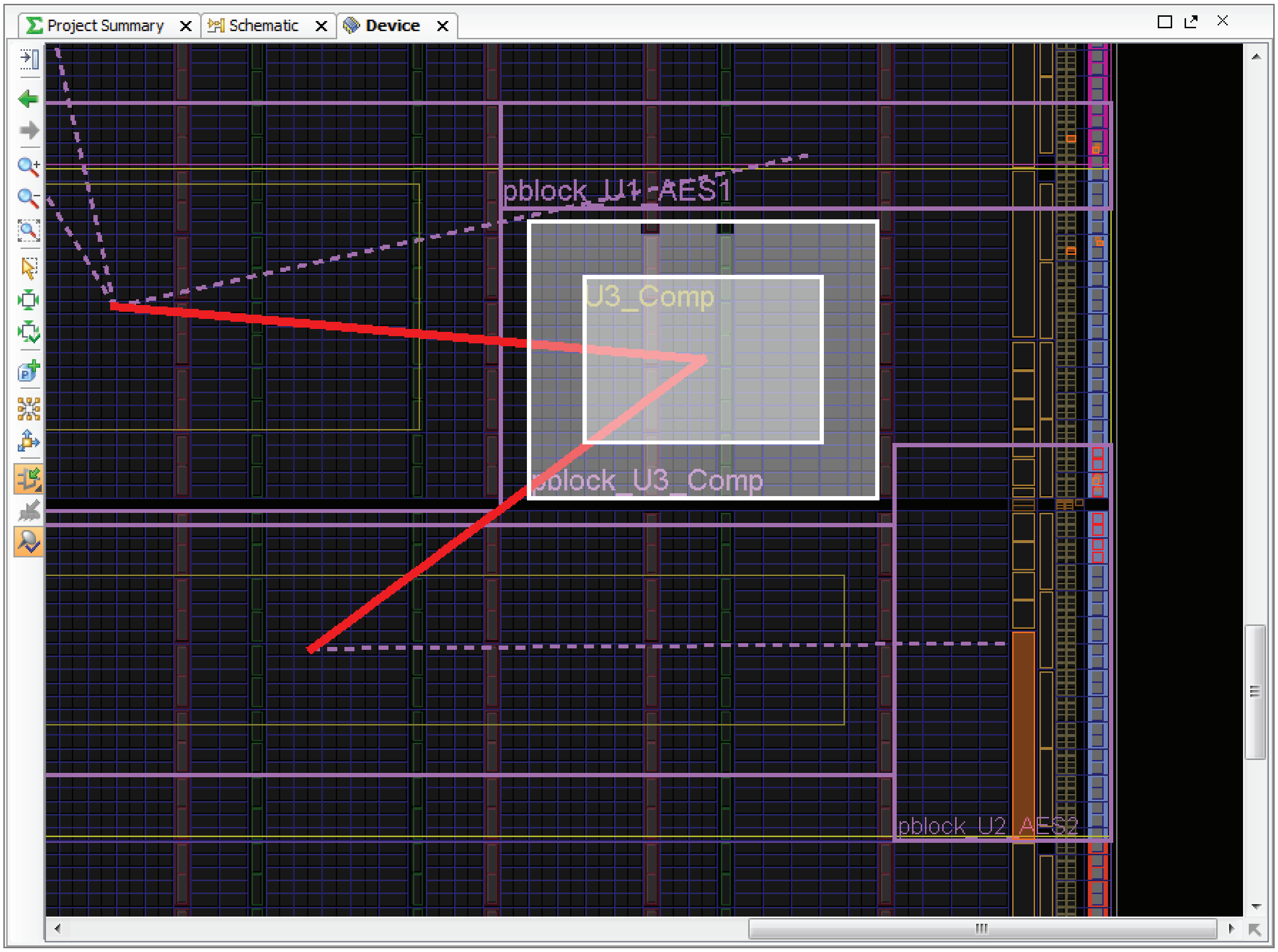


Id	X Lo	Y Lo	X Hi	Y Hi
1	185	265	213	285

X1085_c3_32_022613

Figure 3-32: **Pblock Properties**

33. After the preliminary U3_Comp (COMPARE) block is drawn, adjust the rectangle graphically as necessary to match these coordinates (the pblock should look like that shown in Figure 3-33):
    - X Lo = 185
    - Y Lo = 265
    - X Hi = 213
    - Y Hi = 285



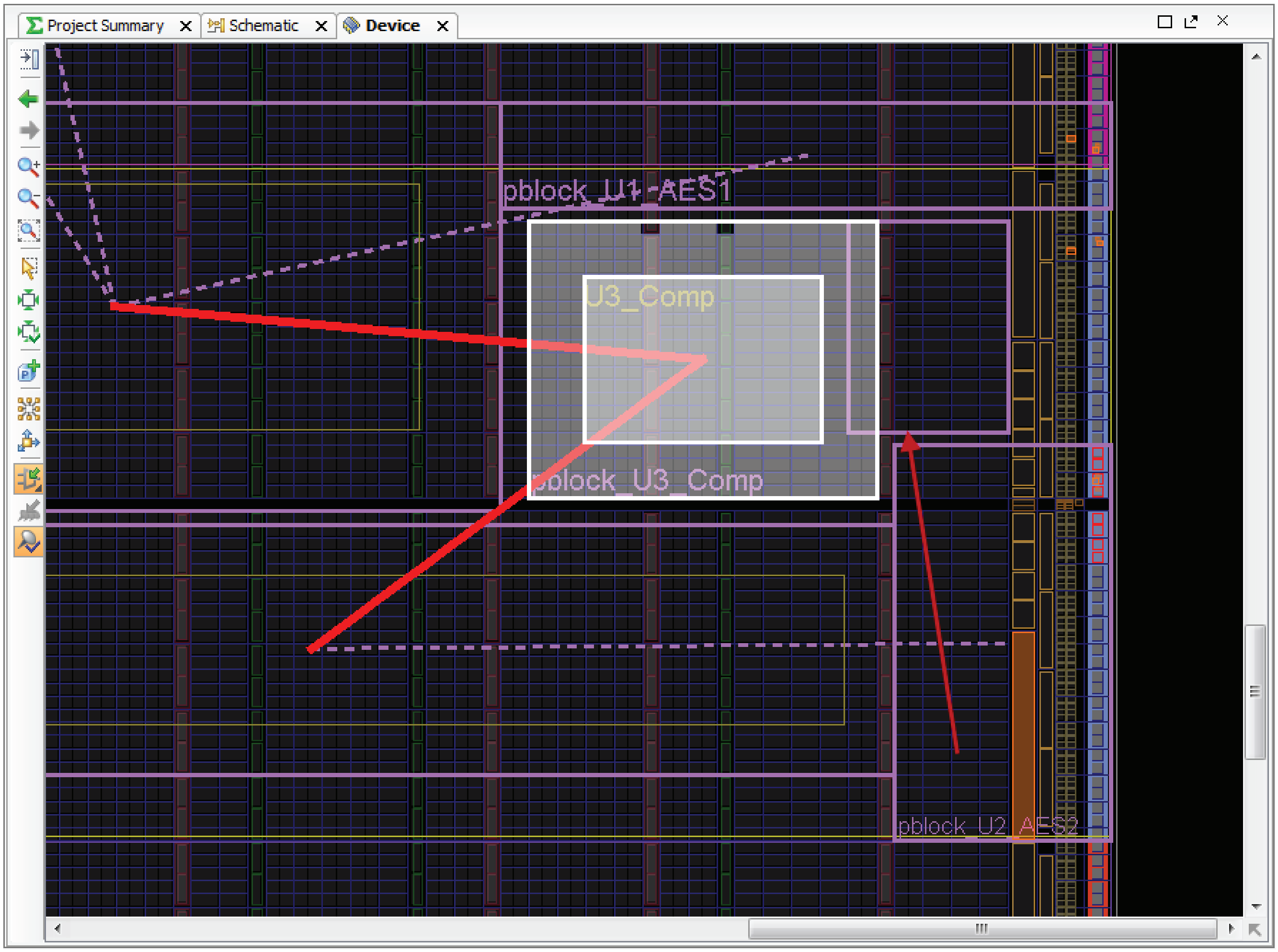
X1085_c3_33_022613

Figure 3-33: **First pblock_U3_Comp Pblock Floorplan**

34. The second rectangle is drawn for pblock_U3_Comp. Under the Physical Constraints tab, select **pblock_U3_Comp**.

35. Right-click **pblock_U3_Comp** and select **Add Pblock Rectangle** from the pull-down menu.

36. Draw a new rectangle to the right side of the first rectangle not including any of the CMT components to create the area shown in Figure 3-34 (accuracy is not important at this stage).



X1085_c3_34_022613

Figure 3-34: **Add Pblock Rectangle – pblock_U3_Comp Pblock Floorplan**

37. A dialog box appears with available ranges and associated checkboxes. All boxes should be checked, including all of the buffers (BUFGCTRL, BUFHCE, BUFR, BUFIO, and BUFMRCE). Even though most of these blocks might not be used in the design, it is important to select them to take advantage of their routing resources.

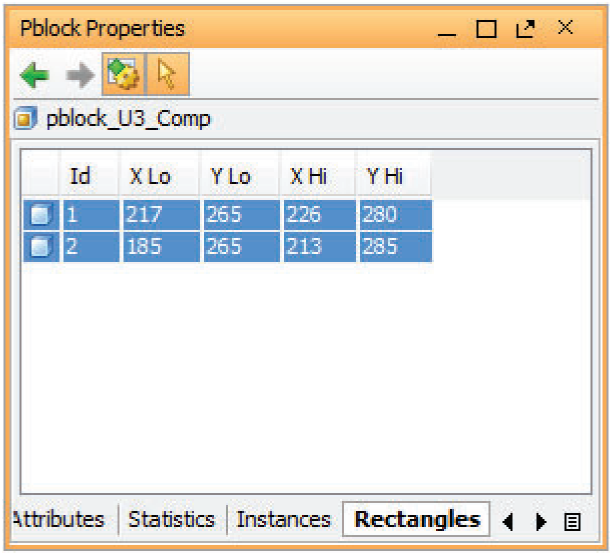
38. For this pblock, the CMT clocking components are not needed so they are excluded completely for this enclosed area. None are displayed or checked in the dialog box.

    Note: IDF requires that all clock buffers and 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.

39. Click **OK**.

40. Ensure that **pblock_U3_Comp** is selected in the Physical Constraints view.

41. Select the **Rectangles** tab in the Pblock Properties window to view the rectangles created for this Pblock (Figure 3-35). Verify that the coordinates match what is displayed in Figure 3-35 and adjust accordingly.



Id	X Lo	Y Lo	X Hi	Y Hi
1	217	265	226	280
2	185	265	213	285

X1085_c3_35_022613

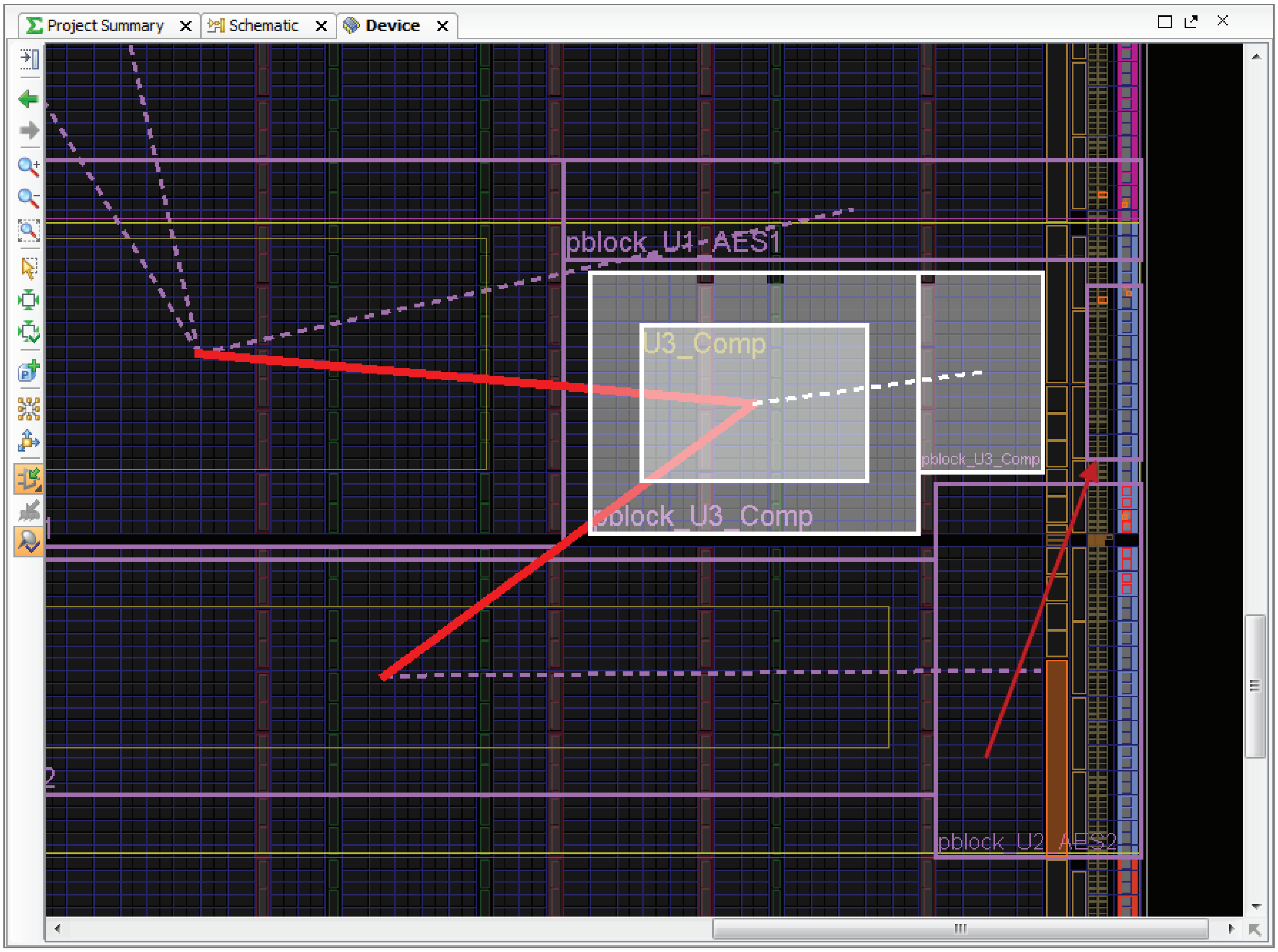
Figure 3-35: **Pblock Properties**

Note: The PlanAhead tool might resize the rectangle to many different combinations of smaller rectangles. Make sure that all of the needed resources inside the area group are fully enclosed (such as block RAMs, DSPs, and MMCMs). Ensure that there is one fence tile, for example, a CLB, of separation between the pblocks AES1, AES2, and Comp.

42. The third rectangle is drawn to complete pblock_U3_Comp. Under the Physical Constraints tab, select **pblock_U3_Comp**.

43. Right-click **pblock_U3_Comp** and select **Add Pblock Rectangle** from the pull-down menu.

44. Draw a new rectangle to the right side of the second rectangle enclosing only I/O components and buffers while excluding all the CMT components (leaving a gap for the CMT components) to create the area shown in Figure 3-36 (accuracy is not important at this stage).



X1085_c3_36_022613

Figure 3-36: **pblock_U3_Comp Pblock Floorplan**

45. A dialog box appears with available ranges and associated checkboxes. All boxes should be checked, including all of the buffers (BUFGCTRL, BUFHCE, BUFR, BUFIO, and BUFMRCE). Even though most of these blocks might not be used in the design, it is important to select them to take advantage of their routing resources.

46. For this pblock, the CMT clocking components are not needed so they are excluded completely for the enclosed area. None are displayed or checked in the dialog box.

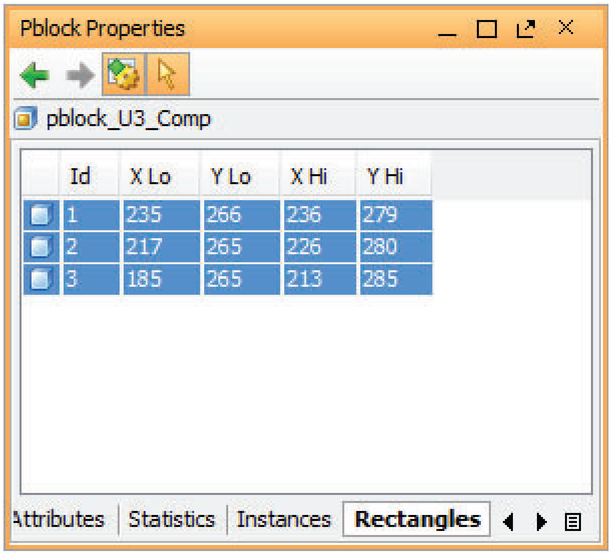
    Note: IDF requires that all clock buffers and clocking components be assigned to the AREA GROUP that physically contains the component, even though the component is not logically instantiated in the HDL of that module.

47. Click **OK**.

48. Ensure that **pblock_U3_Comp** is selected in the Physical Constraints view.

49. Select the **Rectangles** tab in the Pblock Properties window to view all of the rectangles created for this pblock (Figure 3-37). Verify that the coordinates match what is

displayed in Figure 3-37 and adjust accordingly.



Pblock Properties

pblock_U3_Comp

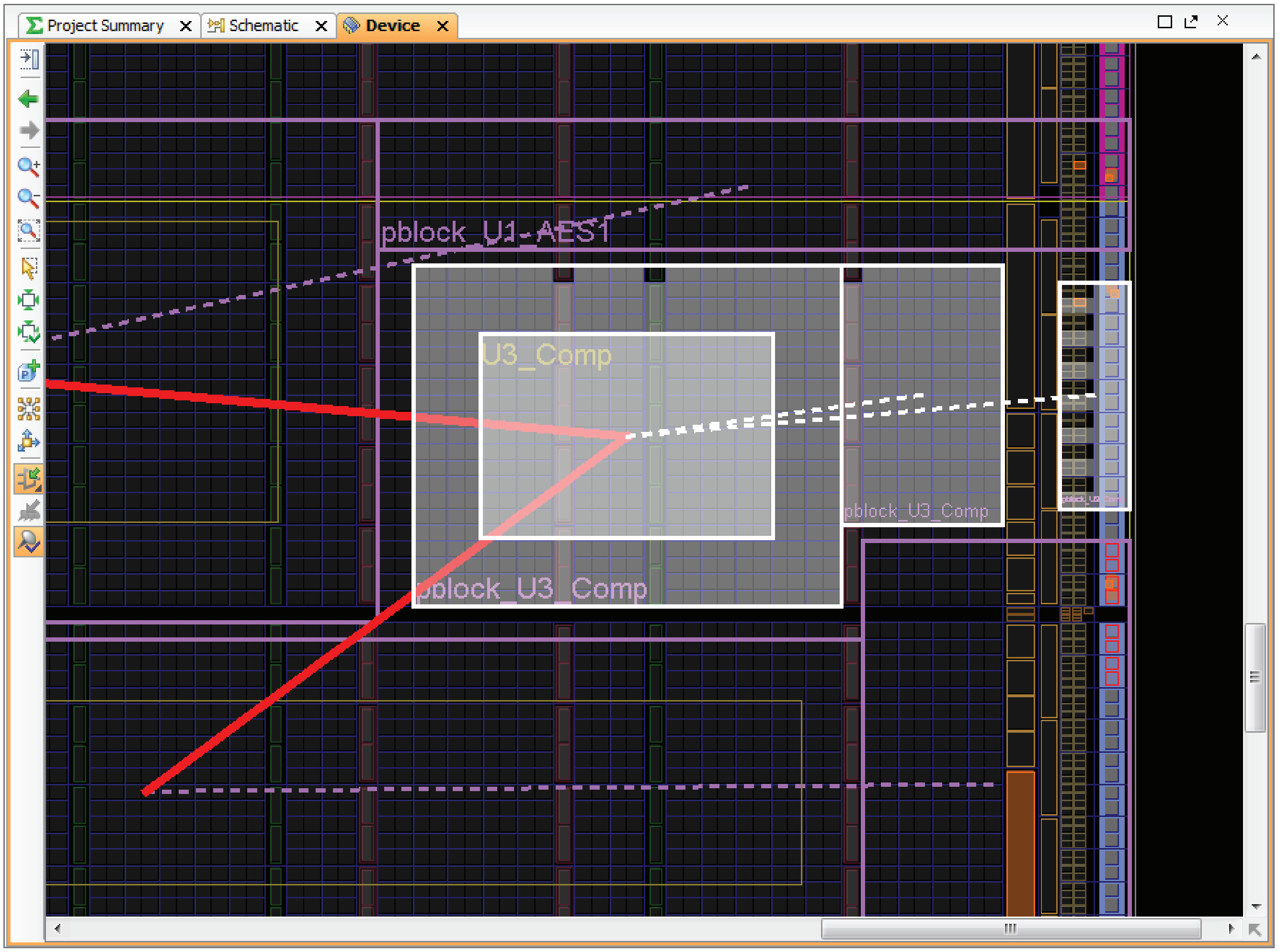
Id	X Lo	Y Lo	X Hi	Y Hi
1	235	266	236	279
2	217	265	226	280
3	185	265	213	285

X1085_c3_37_022613

Figure 3-37: **Pblock Properties**

Note: The PlanAhead tool might resize the rectangle to many different combinations of smaller rectangles. Make sure that all of the needed resources inside the area group are fully enclosed (such as block RAMs, DSPs, and MMCMs). Ensure that there is one fence tile, for example, a CLB, of separation between the pblocks AES1, AES2, and Comp.

The completed pblock for pblock_U3_Comp is shown in Figure 3-38.

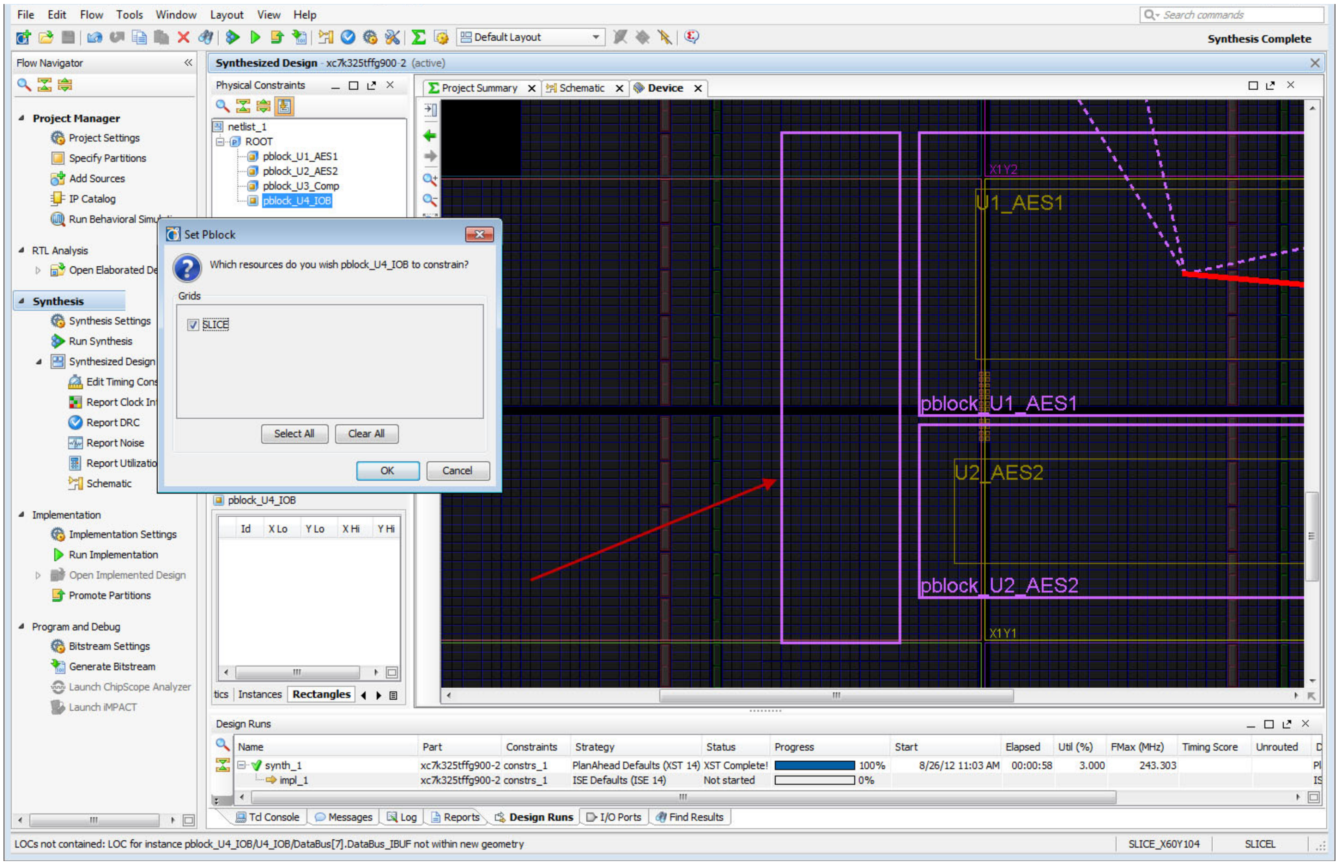


X1085_c3_38_022613

Figure 3-38: **Completed pblock_U3_Comp Pblock Floorplan**

50. Under the Physical Constraints tab, select and right-click the block **pblock_U4_IOB** and select **Set Pblock Size** from the menu.

51. Draw a rectangle, as shown in Figure 3-39, to the left of the AES1/AES2 blocks (accuracy is not important at this stage). Remember to leave a single fence tile for isolation between the Comp and the AES1/AES2 pblocks.



Set Pblock

Which resources do you wish pblock_U4_IOB to constrain?

Grids

☑ SLICE

Select All | Clear All

OK | Cancel

Design Runs

Name	Part	Constraints	Strategy	Status	Progress	Start	Elapsed	Util (%)	FMax (MHz)	Timing Score	Unrouted
synth_1	xc7k325tffg900-2	constrs_1	PlanAhead Defaults (XST 14)	XST Complete!	100%	8/26/12 11:03 AM	00:00:58	3.000	243.303		
impl_1	xc7k325tffg900-2	constrs_1	ISE Defaults (ISE 14)	Not started	0%						

LOCs not contained: LOC for instance pblock_U4_IOB/U4_IOB/DataBus[7].DataBus_IBUF not within new geometry | SLICE_X60Y104 | SLICEL

X1085_c3_39_021513

Figure 3-39: **Set Pblock – pblock_U4_IOB Pblock Floorplan**

52. A dialog box (shown in Figure 3-39) appears with available resources and associated checkboxes. All boxes should be checked, including all of the buffers (BUFGCTRL, BUFHCE, BUFR, BUFIO, and BUFMRCE). Even though most of these blocks might not be used in the design, it is important to select them to take advantage of their routing resources. The only exception to this is when not specifically using the CMT clocking components (MMCME2_ADV, PLLE2_ADV, PHASER_IN_PHY, PHASER_OUT_PHY, PHY_CONTROL, IN_FIFO, and OUT_FIFO) in an area group, they should not be included.

53. For this pblock rectangle, no CMT clocking components are in the enclosed area so none are displayed or checked in the dialog box.

Note: IDF requires that all clock buffers and 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.

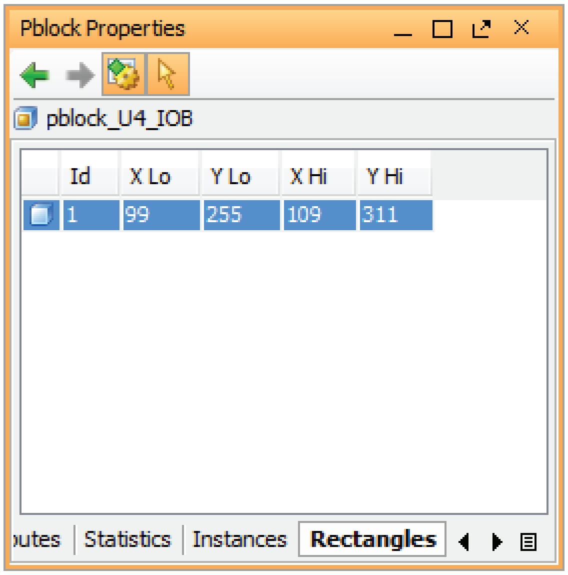
54. Click **OK**.

55. In the Choose LOC mode dialog box, select **Leave all location constraints in their current position**.

56. Click **OK**.

57. Ensure that **pblock_U4_IOB** is selected in the Physical Constraints view.

58. Select the **Rectangles** tab in the Pblock Properties window (Figure 3-40).



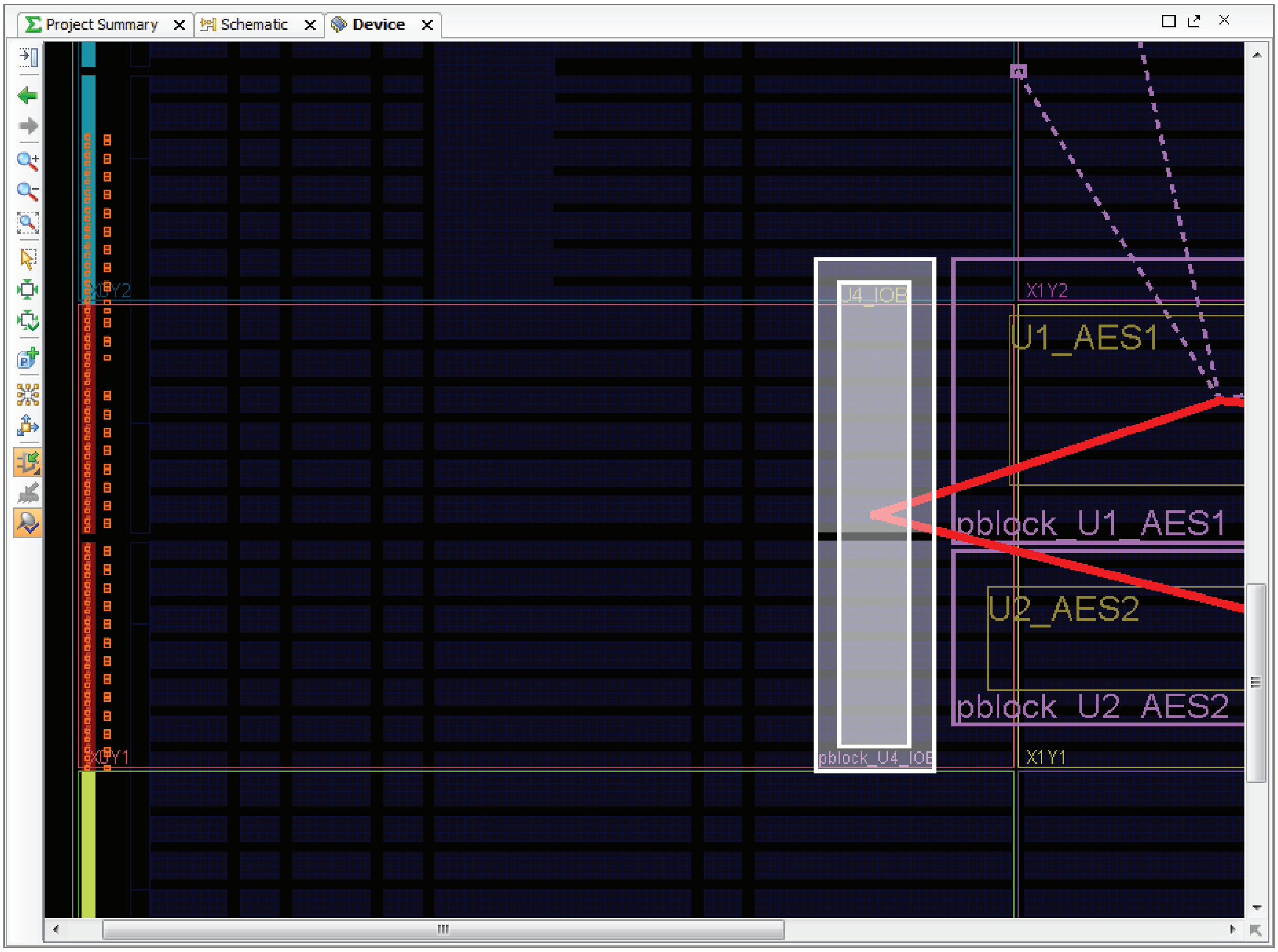
Id	X Lo	Y Lo	X Hi	Y Hi
1	99	255	109	311

X1085_c3_40_022613

Figure 3-40: **Pblock Properties**

59. After the preliminary U4_IOB (INOUT) block is drawn, adjust the rectangle graphically as necessary to match these coordinates (the pblock should look like Figure 3-41):

- X Lo = 99
- Y Lo = 255
- X Hi = 109
- Y Hi = 311

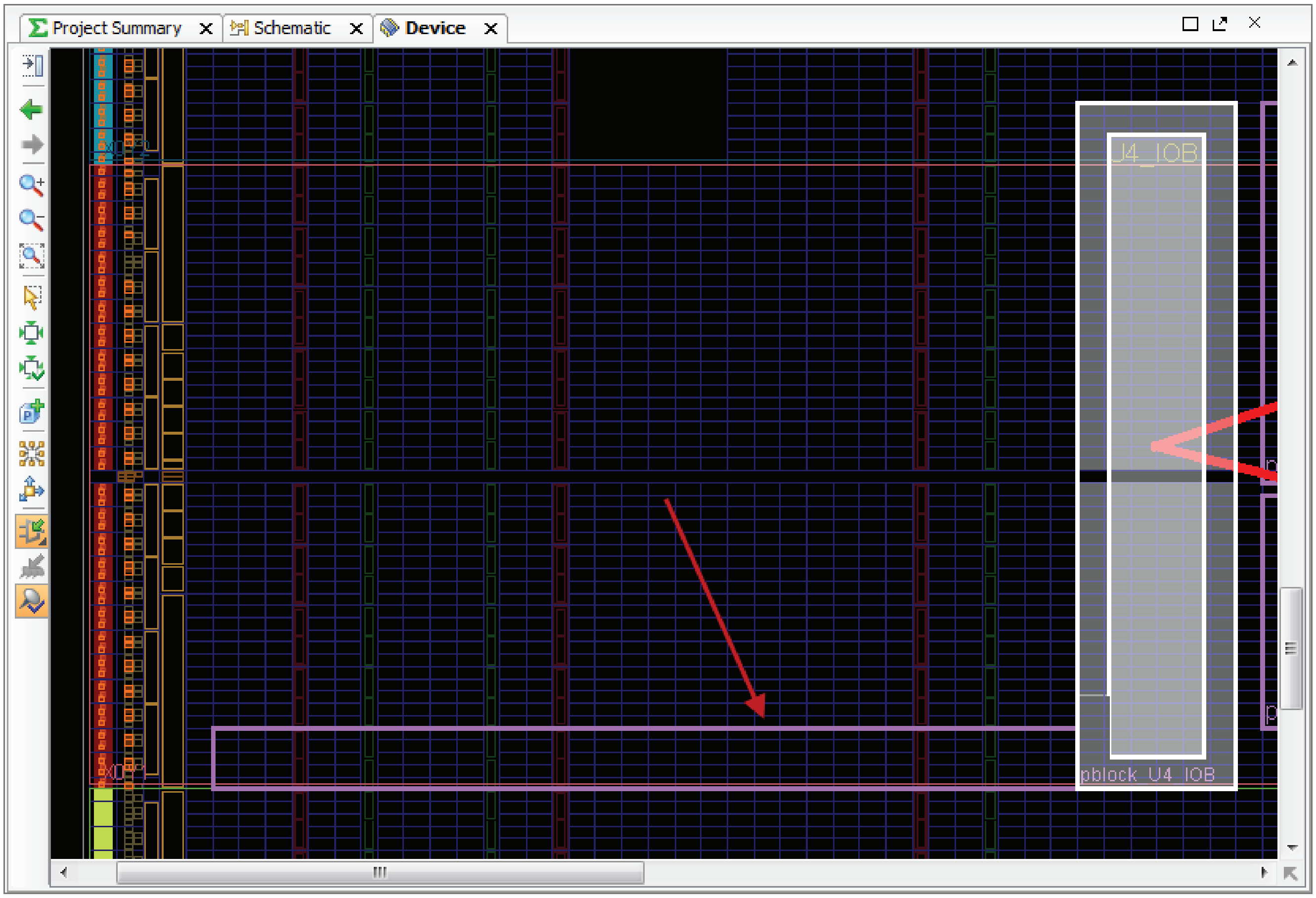


X1085_c3_41_022613

Figure 3-41: **pblock_U4_IOB Pblock Floorplan**

60. The second rectangle is drawn for pblock_U4_IOB. Under the Physical Constraints tab, select **pblock_U4_IOB**.
61. Right-click **pblock_U4_IOB** and select **Add Pblock Rectangle** from the pull-down menu.

62. Draw a new rectangle to the left side of the first rectangle, not including any of the CMT columns, to make the area shown in Figure 3-42 (accuracy is not important at this stage).



X1085_c3_42_022613

Figure 3-42: **pblock_U4_IOB Pblock Floorplan**

63. A dialog box appears with available ranges and associated checkboxes. All boxes should be checked, including all of the buffers (BUFGCTRL, BUFHCE, BUFR, BUFIO, and BUFMRCE). Even though most of these blocks might not be used in the design, it is important to select them to take advantage of their routing resources. Also, all used components must be included.

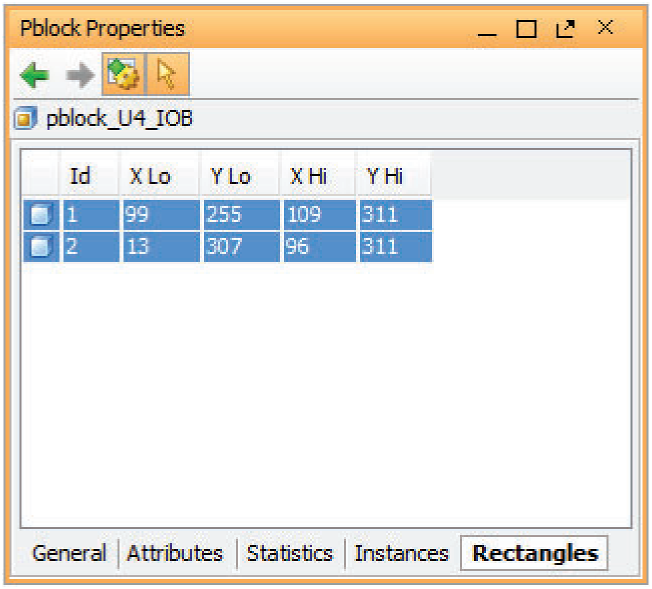
64. For this pblock rectangle, no CMT clocking components are in the enclosed area so none are displayed or checked in the dialog box.

    Note: IDF requires that all clock buffers and 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.

65. Click **OK**.

66. Ensure that **pblock_U4_IOB** is selected in the Physical Constraints view.

67. Select the **Rectangles** tab in the Pblock Properties window to view all of the rectangles created for this pblock (Figure 3-43). Verify that the coordinates match what is displayed in Figure 3-43 and adjust accordingly.



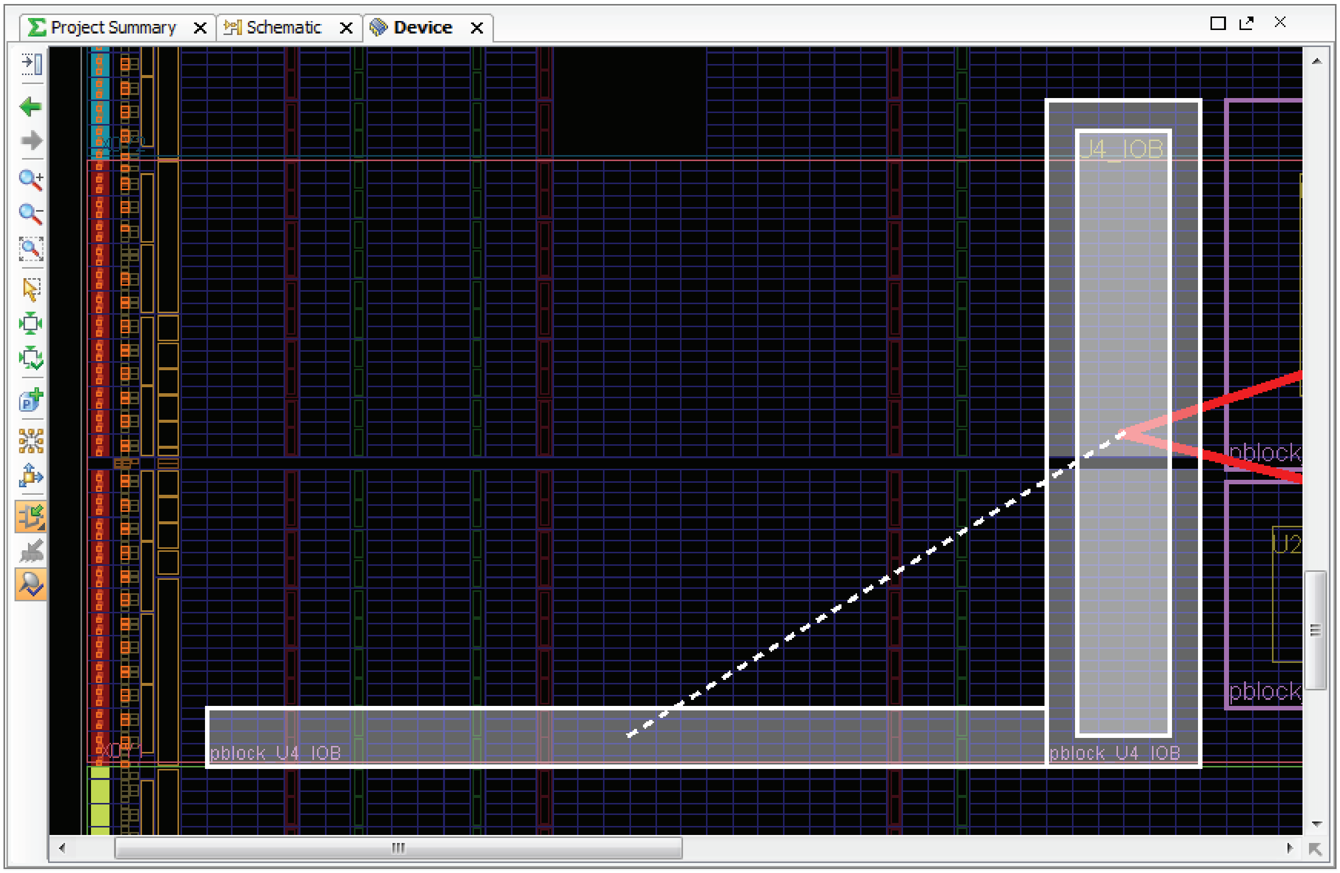
Id	X Lo	Y Lo	X Hi	Y Hi
1	99	255	109	311
2	13	307	96	311

X1085_c3_43_022613

Figure 3-43: **Pblock Properties**

Note: The PlanAhead tool might resize the rectangle to many different combinations of smaller rectangles. Make sure that all of the needed resources inside the area group are fully enclosed (such as block RAMs, DSPs, and MMCMs). Ensure that there is one fence tile, for example, a CLB, of separation between the pblocks AES1, AES2, Comp, and INOUT.

The first two pblocks for pblock_U4_IOB are shown in Figure 3-44.



X1085_c3_44_022613

Figure 3-44: **First and Second pblock_U4_IOB Pblocks Floorplan**

68. The third rectangle is drawn to complete pblock_U4_IOB. Under the Physical Constraints tab, select **pblock_U4_IOB**.
69. Right-click **pblock_U4_IOB** and select **Add Pblock Rectangle** from the pull-down menu.

70. Draw a new rectangle to the left side of the second rectangle enclosing the I/O components and buffers, the CMT components, and one column of CLBs to make the area shown in Figure 3-45 (accuracy is not important at this stage). The CMT resources are not used in this pblock rectangle so they are excluded, along with their interconnect. Therefore, a single adjacent CLB column is included to the right of the CMT column to supply sufficient interconnects to the I/O column.

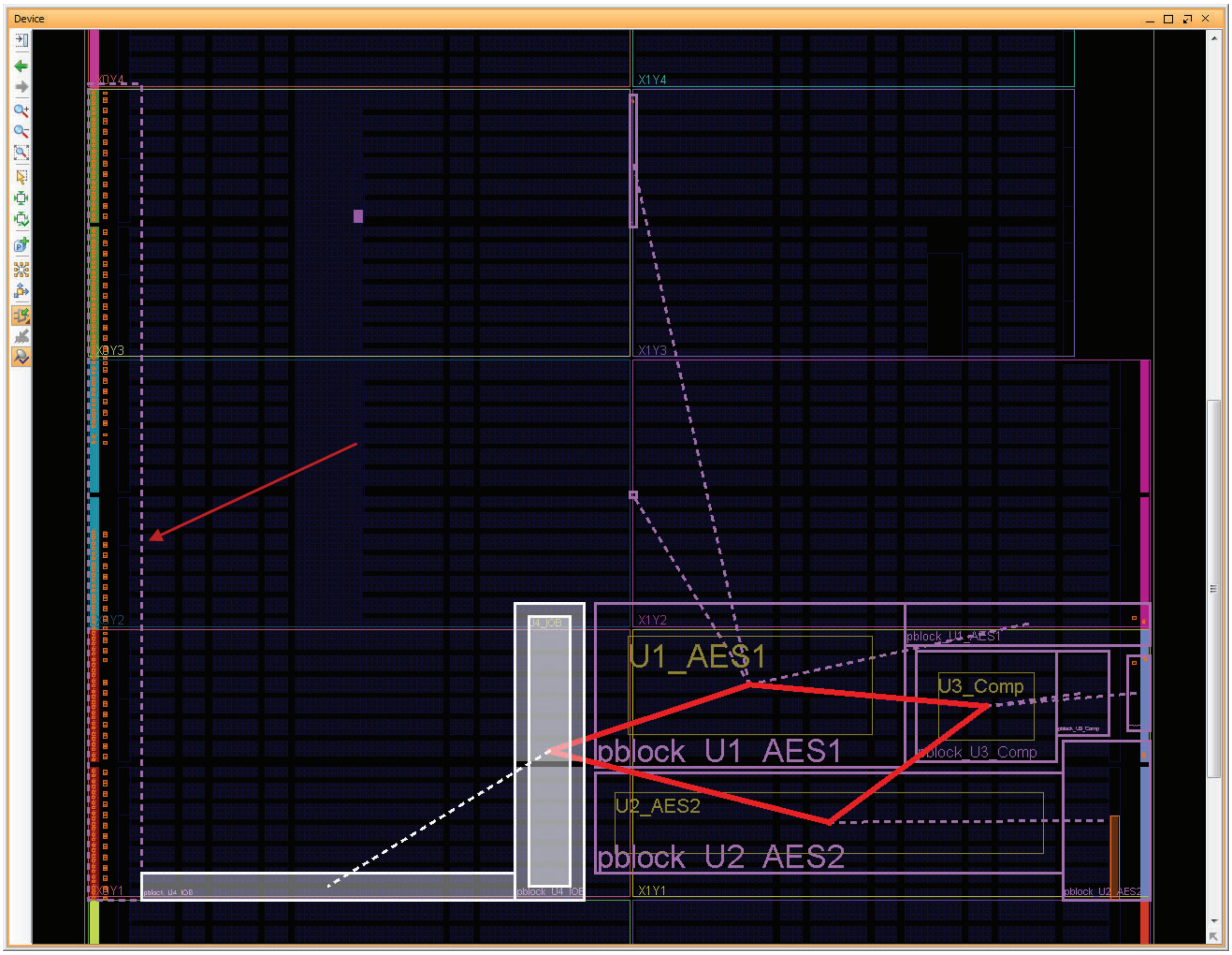


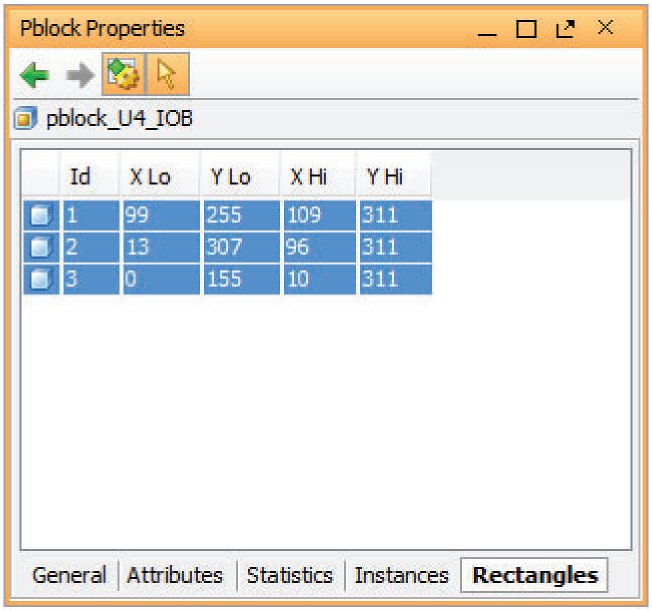
Figure 3-45: **pblock_U3_Comp Pblock Floorplan**

71. A dialog box appears with available ranges and associated checkboxes. All boxes should be checked, including all of the buffers (BUFGCTRL, BUFHCE, BUFR, BUFIO, and BUFMRCE). Even though most of these blocks might not be used in the design, it is important to select them to take advantage of their routing resources.

72. For this pblock rectangle, there are CMT clocking components in the enclosed area but none are used in the design so they are excluded. Uncheck all CMT clocking components (MMCME2_ADV, PLLE2_ADV, PHASER_IN_PHY, PHASER_OUT_PHY, PHY_CONTROL, PHASER_REF, IN_FIFO, and OUT_FIFO) in the dialog box.

Note: IDF requires that all clock buffers and 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.

73. Click **OK**.
74. Ensure that **pblock_U4_IOB** is selected in the Physical Constraints view.
75. Select the **Rectangles** tab in the Pblock Properties window to view all of the rectangles created for this pblock (Figure 3-46). Verify that the coordinates match what is displayed in Figure 3-46 and adjust accordingly.



Id	X Lo	Y Lo	X Hi	Y Hi
1	99	255	109	311
2	13	307	96	311
3	0	155	10	311

X1085_c3_46_022613

Figure 3-46: **Pblock Properties**

Note: The PlanAhead tool might resize the rectangle to many different combinations of smaller rectangles. Make sure that all of the needed resources inside the area group are fully enclosed (such as block RAMs, DSPs, MMCMs, and so on). Ensure that there is one fence tile, for example, a CLB, of separation between the Pblocks AES1, AES2, Comp, and INOUT.

The completed Pblock for pblock_U4_IOB is shown in Figure 3-47.

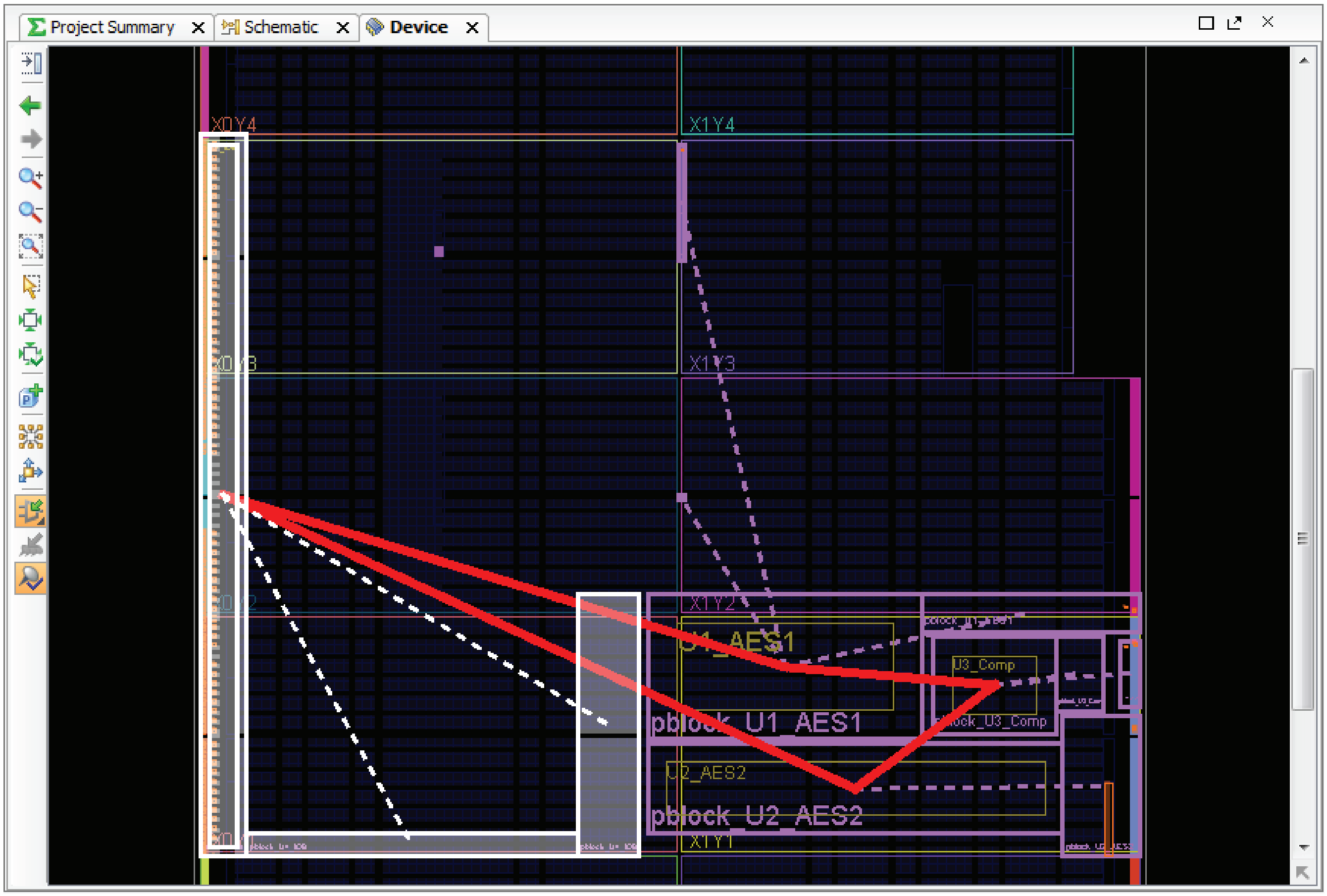
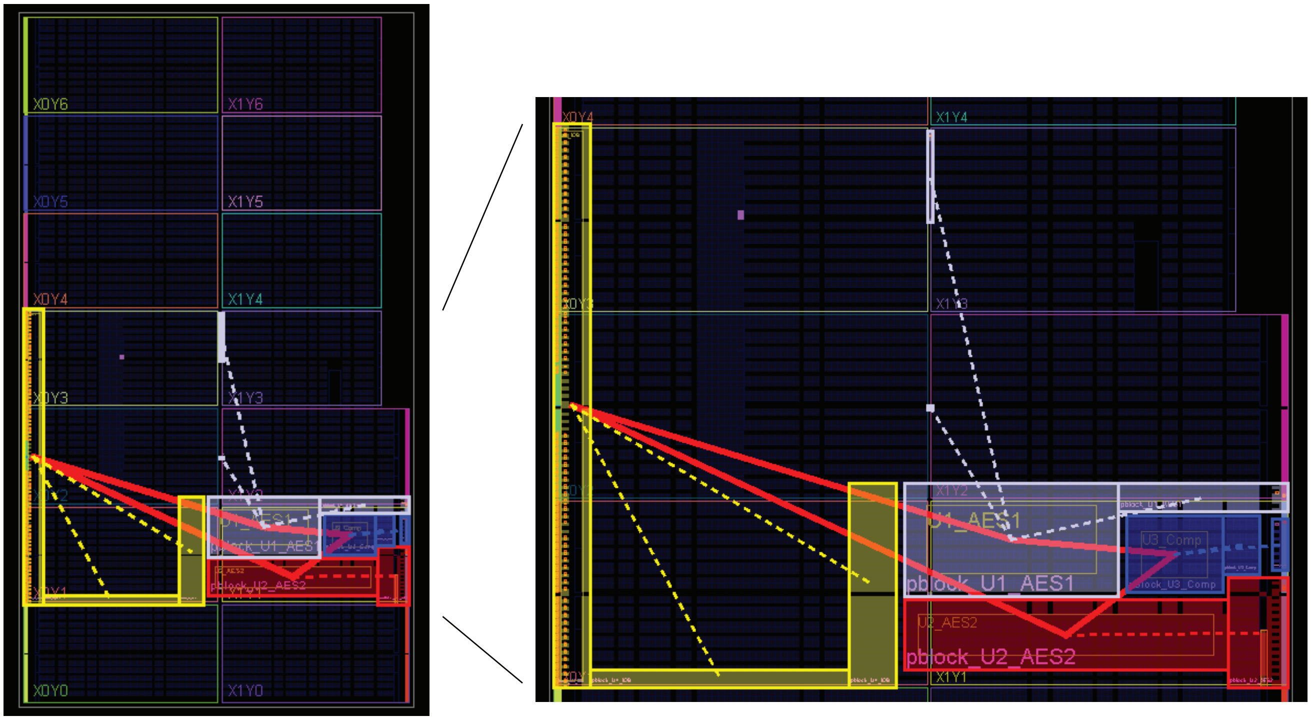


Figure 3-47: **Completed pblock_U4_IOB Pblock Floorplan**

The final floorplan is shown in Figure 3-48. Each block is separated by one user tile (fence) to ensure IDF isolation. A CLB, block RAM, DSP, IOB, or any other site type that contains a global switch (GSM) can be used as a fence tile for this isolation.



X1085_c3_48_022613

Figure 3-48: **Final Floorplan**

Isolation Design Flow Progress – System Floorplanning Block

The System Floorplanning block of the IDF system design flow diagram is complete, as shown in Figure 3-49.

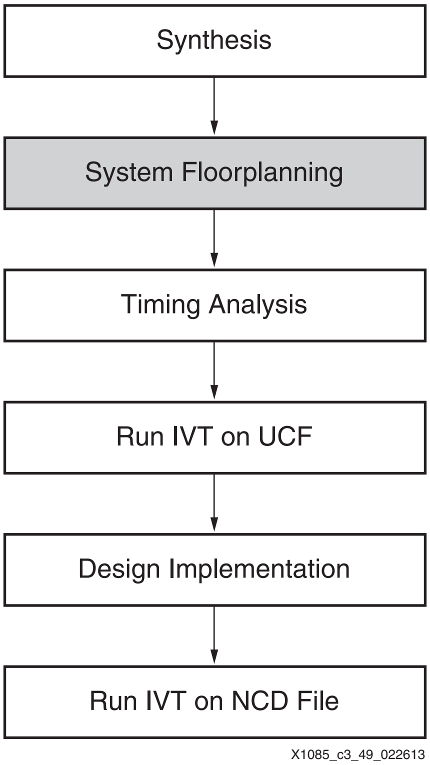
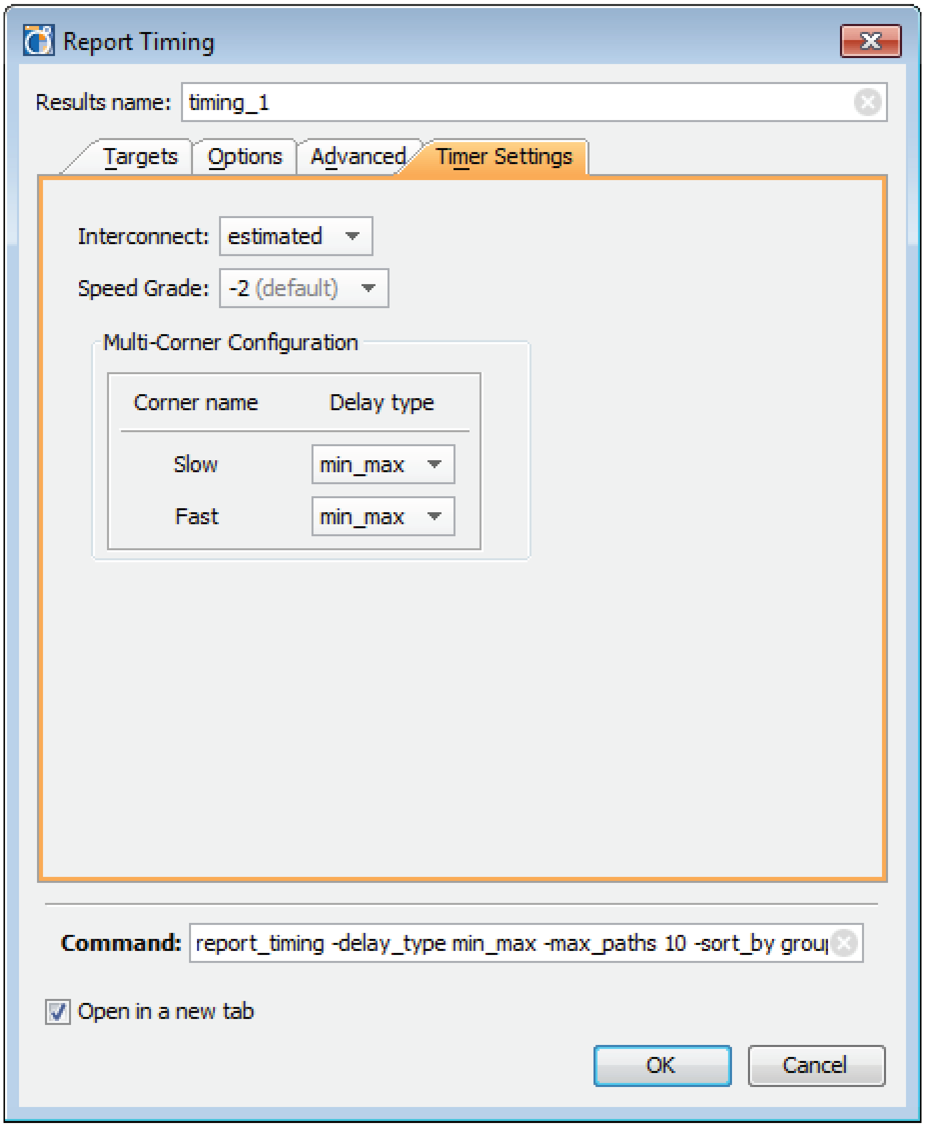


Figure 3-49: **IDF System Design Flow with System Floorplanning Block Completed**

Running Report Timing

This section tests the timing constraints that were set up in Setting Up Timing Constraints, page 39.

1. Select **Tools** > **Timing** > **Report Timing** and click **OK** to run the Report Timing (Figure 3-50).



Results name: timing_1

Interconnect: estimated

Speed Grade: -2 (default)

Multi-Corner Configuration

Corner name	Delay type
Slow	min_max
Fast	min_max

Command: report_timing -delay_type min_max -max_paths 10 -sort_by grou

☑ Open in a new tab

X1085_c3_50_022613

Figure 3-50: **Run Report Timing Dialog**

A new tab, Timing, appears at the bottom with a sub-tab named as specified in the Timing – Report Timing window (in this example, timing_1). As specified when launched, Report Timing reports the 10 paths closest to missing timing.

Isolation Design Flow Progress

The Timing Analysis block of the IDF system design flow diagram is complete, as shown in Figure 3-51.

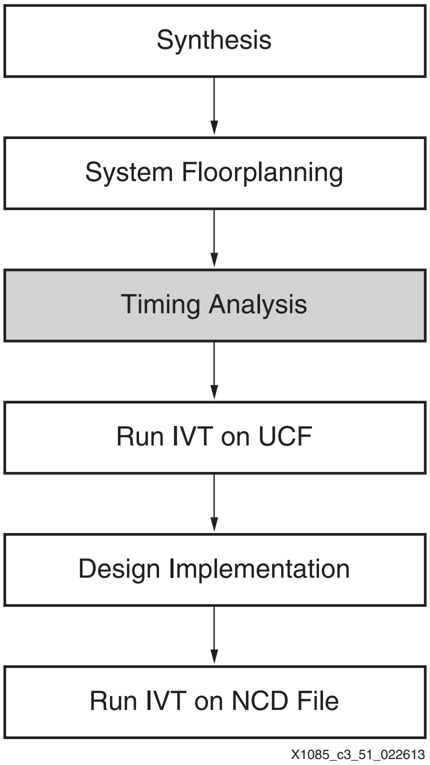


Figure 3-51: **IDF System Design Flow with Timing Analysis Block Completed**

Exporting the Design

At this stage, the design can be exported. Only the UCF must be exported for later checking by the Isolation Verification tool.

Note: At this point, the PlanAhead project should be saved. If an error occurs when saving the project, set the UCF in the Sources window as the target constraints file.

1. To export the constraints file, select **File > Export > Export Constraints** and browse to the desired location, or choose the default location and specify the UCF:

   ..\Xilinx_Design\planahead\FloorPlan_SCC\scc_lab_top.ucf

2. Click **OK**.

Chapter 4

Running Isolation Verification Tool Against 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 IVT features graphical output, there is no graphical user interface.

IVT is run on the UCF to catch pin, I/O bank, and area group isolation faults early in the design when changes are more easily integrated. The steps in this chapter guide the user through the process. After implementation, the IVT NCD test is run against the routed design.

Creating the Pin Isolation Group File

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

1. Open the directory `..\Xilinx_Design\ivt`.
2. Either use the IVT files provided in the `..\Xilinx_Design\ivt` directory, or create a new text file and name the file `scc_lab_top.pig` to match the format of this example:

   Note: *Do not* copy and paste the example PIG file because only some of the `INOUT` isolation group NETs are shown. This example PIG file is for illustrative and format purposes only. If creating a new PIG file, the user should copy each NET statement directly from the UCF.

```
# Place all Global (top level) signals here (each commented out)
# NET "clk_p"           LOC = AD12;
# NET "clk_n"           LOC = AD11;

ISOLATION_GROUP AES BEGIN
NET "reset"         LOC = AK4;
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 = AB9;
END ISOLATION_GROUP

ISOLATION_GROUP INOUT BEGIN
NET "push_button"   LOC = P27;
NET "mode"          LOC = Y25;
NET "key_sel"       LOC = Y28;
NET "key_addr[2]"   LOC = Y29;
  |
  |
  V
NET "data_pins[56]" LOC = R23;
NET "data_pins[58]" LOC = T20;
END ISOLATION_GROUP
```

3. The instructions placed in the IVT PIG file are:
   - The clk_p and clk_n clock pins are commented out because they are not required to be isolated (global nets).
   - The four 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 the section Creating the File Used to Run the IVT UCF Test.

   Note: The IVT PIG file uses UCF syntax for each pin definition. It is useful to copy the pins directly 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.

Creating the File Used to Run the IVT UCF Test

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

5. Open the directory `..\Xilinx_Design\ivt`.
6. Either use the UCF IVT file (`scc_lab_top_ucf.ivt`) provided in the `..\Xilinx_Design\results\ivt` directory, or create a new text file and name the file `scc_lab_top_ucf.ivt` with these contents:

```
-device xc7k325t -package ffg900

# Groups      Isolation Group        Area Group
# -------     ------------------     ----------------
-group        AES                    pblock_U1_AES1
-group        AES_r                  pblock_U2_AES2
-group        COMPARE                pblock_U3_Comp
-group        INOUT                  pblock_U4_IOB

# Pin Isolation Groups
-pig scc_lab_top.pig

# User Constraint File
..\planahead\FloorPlan_SCC\scc_lab_top.ucf

# Output file
-output scc_lab_top_ucf.rpt
```

7. The instructions placed in the IVT command file are:
   - The first line sets the target device and package for IVT to compare against.
   - Four isolation groups are assigned: AES, AES_R, COMPARE, and INOUT.

     Note: These names are arbitrary. If desired, they can be named the same to denote RED and BLACK groups.

- Four area groups are assigned: pblock_U1_AES1, pblock_U2_AES2, pblock_U3_Comp, and pblock_U4_IOB, respectively.

  Note: These names must match the area groups in the UCF.
- IVT is provided with the path to the UCF.
- IVT is provided with the path to the output file and its associated name.

8. Save and close the IVT command file.

Running the IVT UCF Test

These steps describe how to run the IVT UCF test:

9. The `ivt.zip` file posted on 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

   Note: The IVT executable files compiled for the ISE Design Suite 14.4 for both Windows and Linux are also available in the `..\Xilinx_Design\ivt` directory. The IVT executable file chosen to be used is assumed to have been renamed to `ivt.exe` for the examples to run IVT in subsequent sections.
10. Open an ISE command prompt through the Windows Start menu by selecting **Start > All Programs > Xilinx Design Tools > ISE Design Suite 14.4 > Accessories > ISE Design Suite 64 Bit Command Prompt**.

    Note: On a Windows 7 64-bit PC, this provides a command prompt with the Xilinx environment variables and paths configured to run IVT.
11. Navigate to the `..\Xilinx_Design\ivt` directory.
12. Run the UCF test at the command prompt by typing:

    ```
    ivt -verbose -f scc_lab_top_ucf.ivt
    ```

    A successful UCF test run produces the output “`.....SUCCESS!`”.

    Note: It is useful to add this command line to a file with the name `run_ivt_ucf.bat` so that the UCF test can be run by double-clicking the file name in Windows Explorer.

Examining the Output from the IVT UCF Test

IVT creates two types of output files:

- IVT RPT Output File: a standard RPT text report file.
- IVT UCF SVG Output File: a graphical view of the Kintex-7 device with colored tiles denoting the ownership of the tiles by each isolated area.

IVT RPT Output File

This section describes how to read the various sections of the RPT output file from the IVT UCF test.

The IVT UCF test output file is located here:

```
\Xilinx_Design\ivt\scc_lab_top_ucf.rpt
```

The header section of the IVT UCF Report lists the provenance:

```
Isolation Verification Report
```

```
Date: Fri Jan 18 08:50:37 2013
IVT Version: 7.27
ISE Build: P
ISE Development Version: ISE 14.4.N/A
XILINX Directory: C:\Xilinx\14.4\ISE_DS\ISE
ISE Installed Version: 14.4
Command Line: ivt -verbose -device xc7k325t -package
ffg900 -group AES pblock_U1_AES1 -group AES_r
pblock_U2_AES2 -group COMPARE pblock_U3_Comp -group
INOUT pblock_U4_IOB -pig scc_lab_top.pig -output
scc_lab_top_ucf.rpt
../planahead/FloorPlan_SCC/scc_lab_top.ucf
Working Directory: C:\Xilinx_Design\ivt\
Output File: scc_lab_top_ucf.rpt
Part: xc7k325t-ffg900
```

The output report has four key sections:

1. Isolation Groups and Area Groups

```
Section 1 - Area Range Constraints
   Isolation Group         Area Group     Ranges
   ---------------     --------------     -----------------
             AES       pblock_U1_AES1     IOB_X1Y97:IOB...
```

2. Pin Isolation Summary

```
Section 3 - I/O Buffer Isolation Violations
  No I/O buffer isolation violations were found.

Section 4 - Package Pin Isolation Violations
  No package pin isolation violations were found.

Section 5 - I/O Bank Isolation Violations
  No I/O bank violations were found.
```

3. Area Fault Summary

```
Section 6 - Area Group Separation
  No area group range violations were found.
Section 7 - Tiles with Special-Case Adjacency Violations
  No special-case adjacency violations were found.
```

4. Isolation Verification Summary

```
Section 8 - Isolation Verification Summary
  I/O Isolation
    I/O Buffer Isolation Violations: 0
    Package Pin Isolation Violations: 0
    Bank Isolation Violations: 0

  Area Group Isolation
    Area Group Isolation Violations: 0

  Special Fence Rules
    DSP Violations: 0
    Supersite Violations: 0

  UCF Isolation Verification Summary
    Total Isolation Violations: 0

  Isolation analysis completed.
```

```
Elapsed time: 0:00:54
```

IVT UCF SVG Output File

The SVG file `scc_lab_top_ucf.svg` created by IVT for UCF mode gives a graphical view of the Kintex-7 device with colored tiles denoting the ownership of the tiles by each isolated region. The SVG file also visually highlights that a proper fence isolates each of the regions (uncolored tiles). Figure 4-1 shows the SVG output for the Kintex-7 FPGA design used in the SCC lab. The SVG file can be opened using a web browser.

- The YELLOW outline highlights the INOUT isolated partition.
- The GRAY outline highlights the AES isolated partition.
- The RED outline highlights the AES_R isolated partition.
- The BLUE outline highlights the COMP isolated partition.

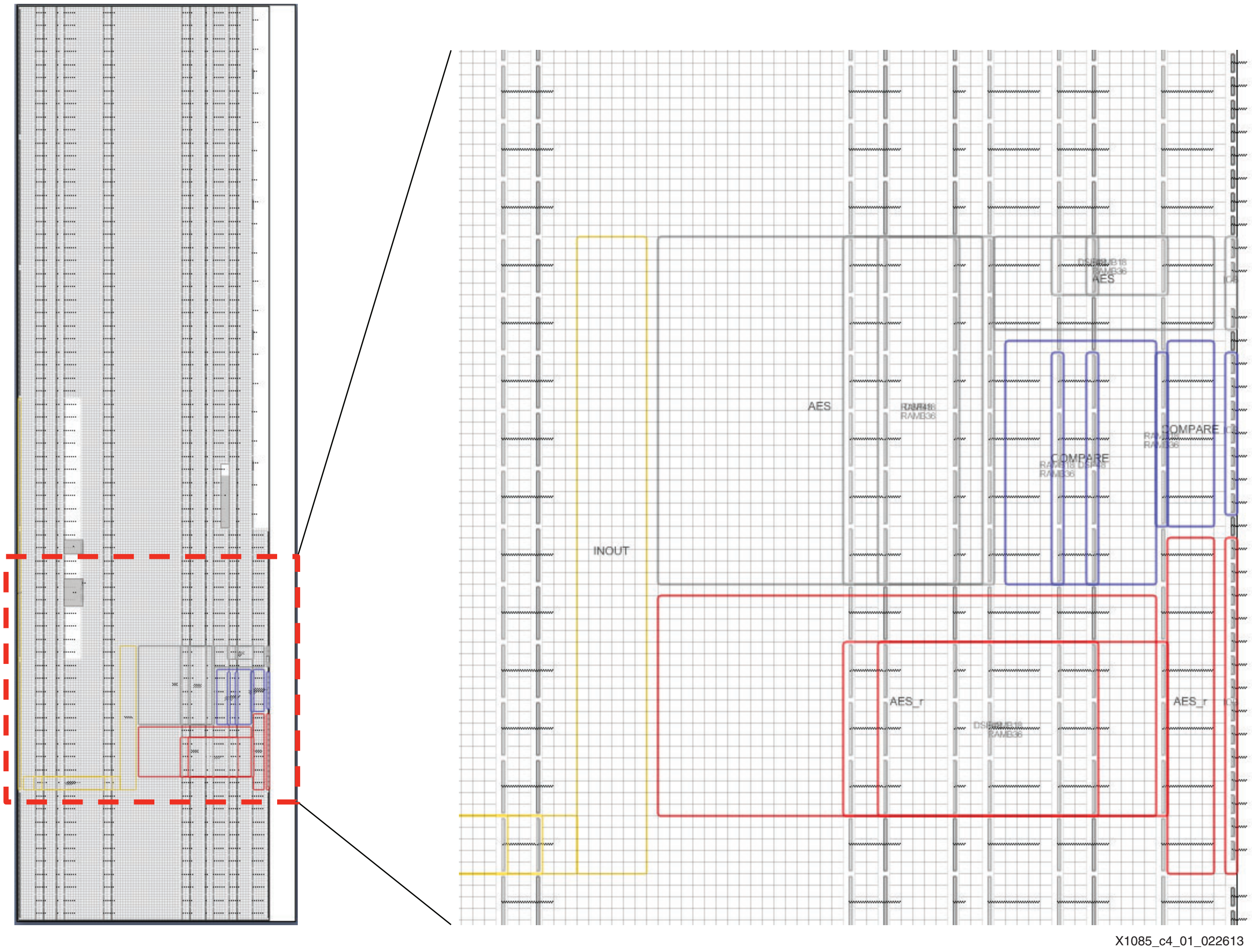


Figure 4-1: **IVT UCF Mode SVG File Graphical Output**

Isolation Design Flow Progress – Run IVT on UCF Block

The Run IVT on UCF block of the IDF system design flow diagram is complete, as shown in Figure 4-2.

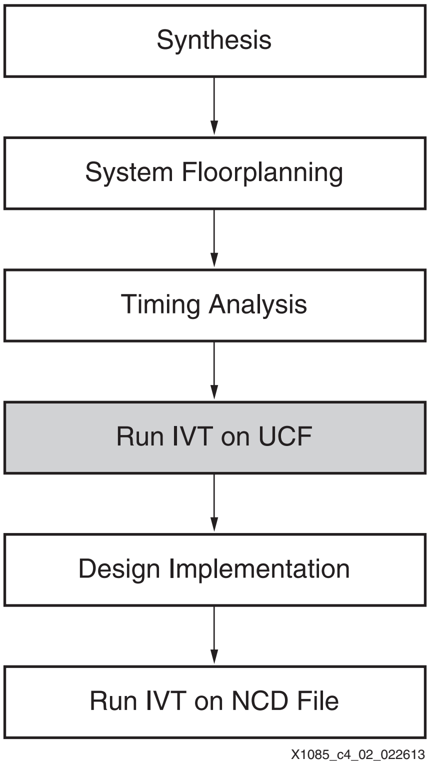


Figure 4-2: **IDF System Design Flow with Run IVT on UCF File Block Completed**

Chapter 5

Implementing Design with PlanAhead Tool

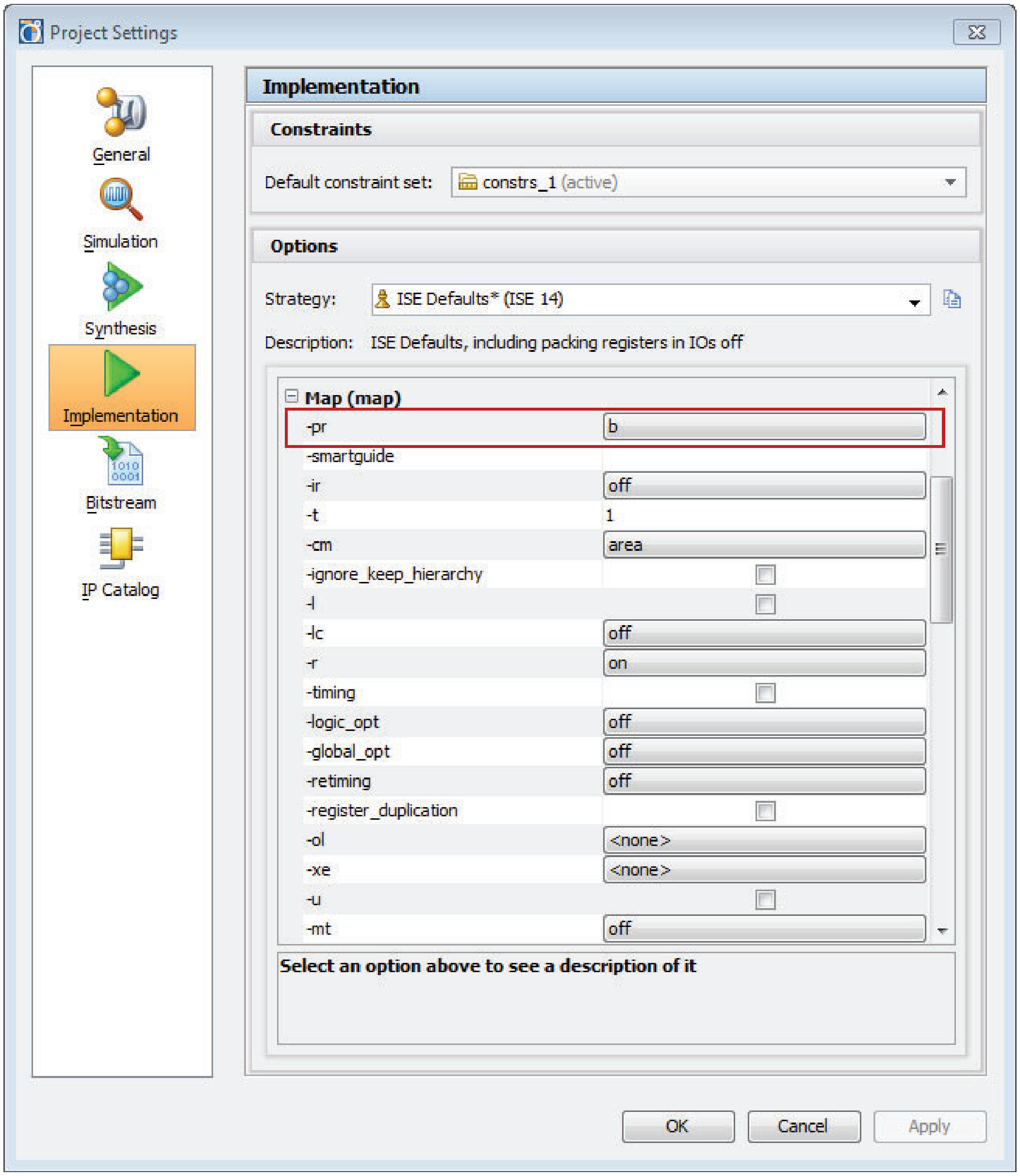
Generating and Running an Implementation

Setting Implementation Options

Before running the implementation, the implementation options must be specified. Implementation options can be set for NGDBuild (translate), Map, PAR (Place and Route), TRCE (Static Timing Report), and BitGen (bit file generation).

Set the implementation options from the Project Settings – Implementation Options window.

1. Select **Implementation > Implementation Settings** in the Flow Navigator window.
2. Use the default setting for Translate.
3. For the Map options, set Pack internal flops/latches into IOBs (-pr) to Both (**b**).
4. Refer to the Project Settings – Implementation Options window shown in Figure 5-1. Use the defaults for the remaining Map settings.
5. Use the default setting for Place and Route (PAR).
6. Click **OK**.



Implementation

Constraints

Default constraint set: constrs_1 (active)

Options

Strategy: ISE Defaults* (ISE 14)

Description: ISE Defaults, including packing registers in IOs off

Map (map)

-pr	b
-smartguide	
-ir	off
-t	1
-cm	area
-ignore_keep_hierarchy	□
-l	□
-lc	off
-r	on
-timing	□
-logic_opt	off
-global_opt	off
-retiming	off
-register_duplication	□
-ol	<none>
-xe	<none>
-u	□
-mt	off

Select an option above to see a description of it

X1085_c5_01_022613

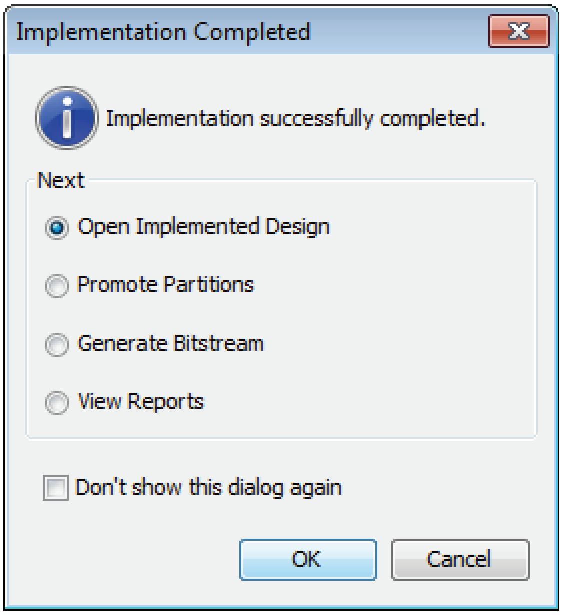
Figure 5-1: **Project Settings - Map (Implementation Options)**

Implementing the Design

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

1. Select **Implement > Run Implementation** (bright green triangle) in the Flow Navigator view to implement the design. If prompted to perform Promote Partitions, click **Launch Run**. If prompted, save the project before launching the implementation.

2. After implementation completes successfully, an Implementation Completed dialog appears, as shown in Figure 5-2.



Implementation Completed

Implementation successfully completed.

Next

● Open Implemented Design

○ Promote Partitions

○ Generate Bitstream

○ View Reports

□ Don't show this dialog again

X1085_c5_02_022613

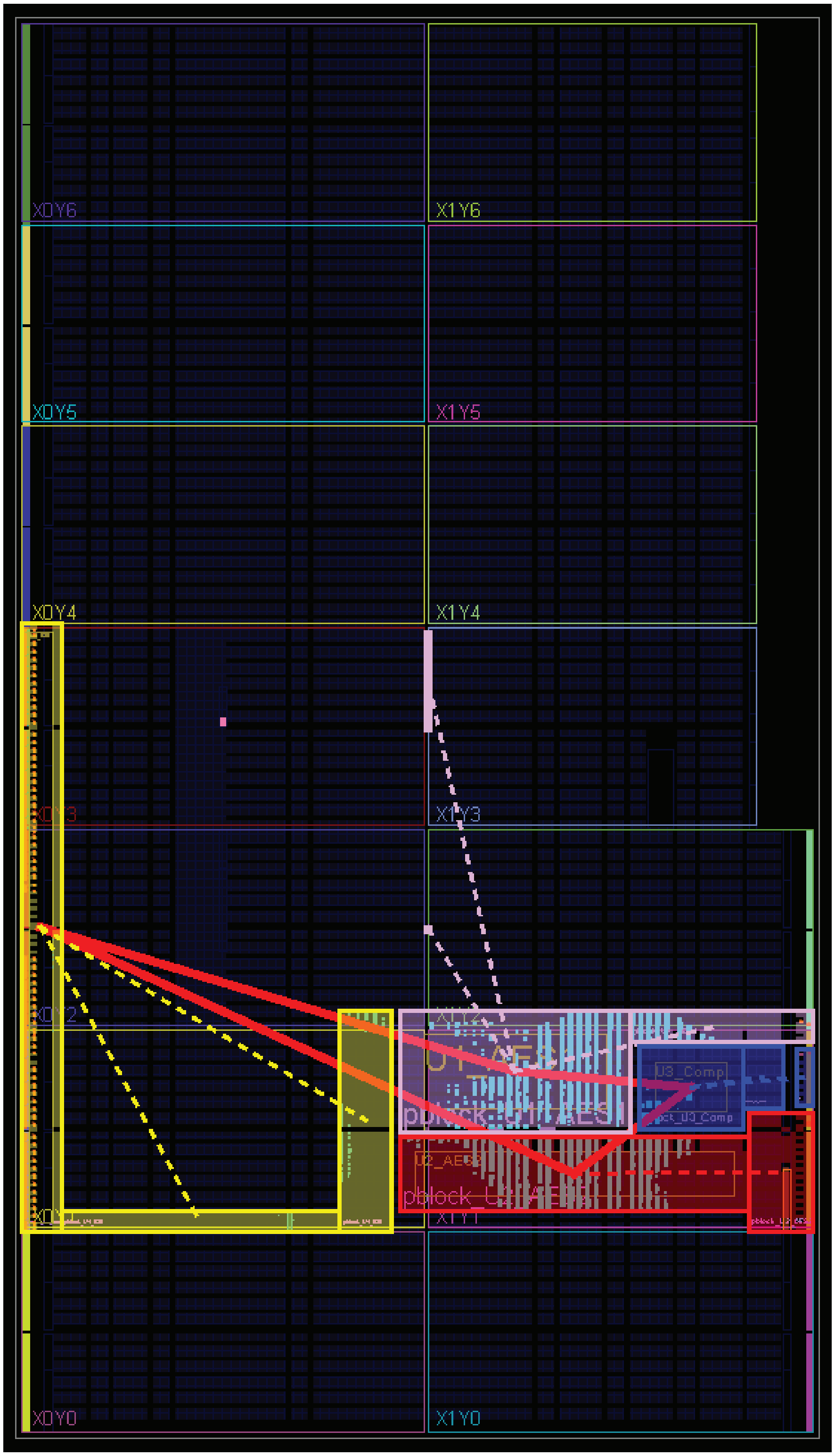
Figure 5-2: **Implementation Completed Dialog**

3. The combined and routed design `scc_lab_top_routed.ncd` is placed in this directory:

```
..\Xilinx_Design\planahead\FloorPlan_SCC\FloorPlan_SCC.runs\impl_1
```

4. Select **Open Implemented Design** in the Implementation Completed dialog and click **OK**.

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

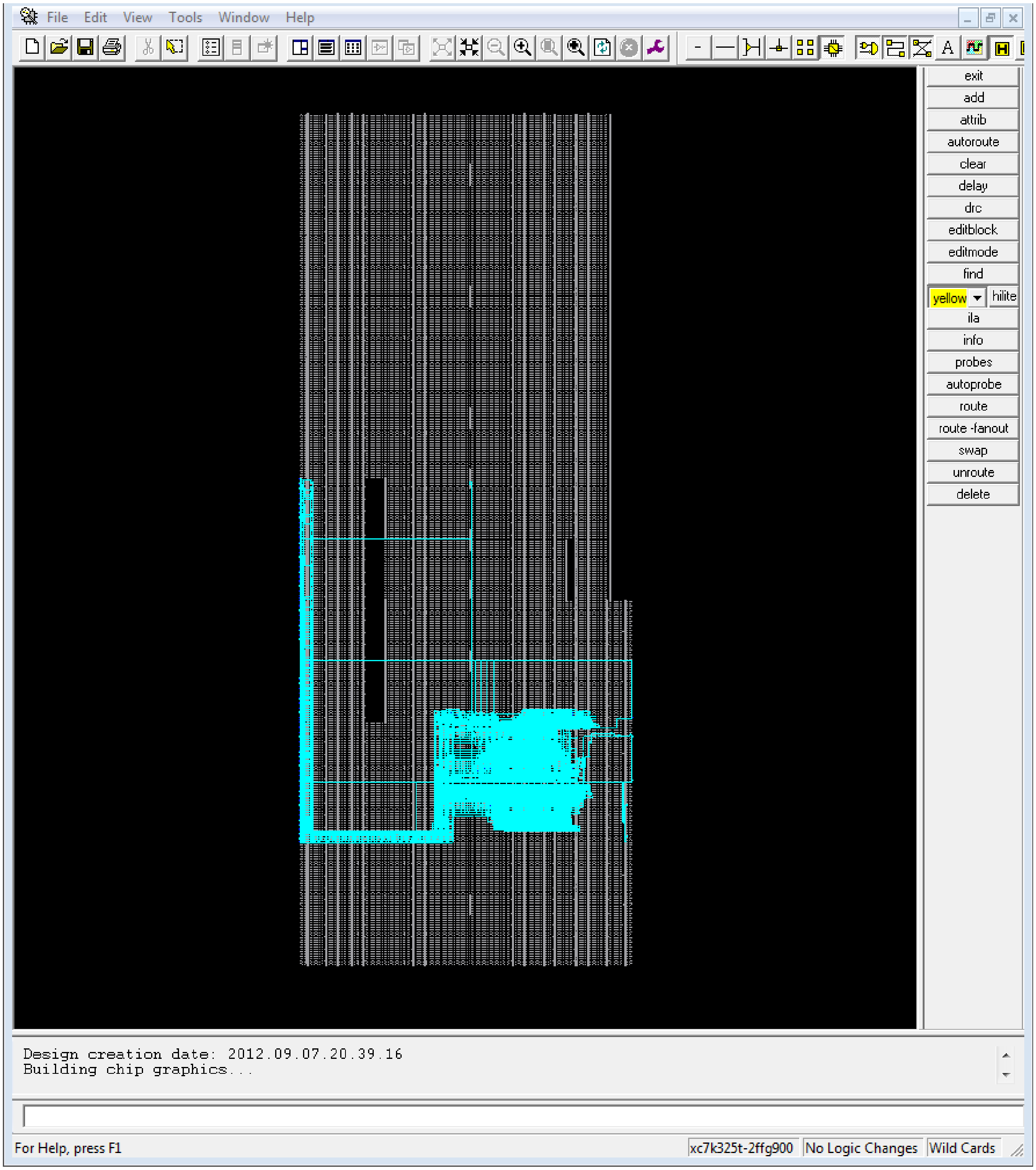


X1085_c5_03_022613

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

The file name for the combined and routed design is `scc_lab_top_routed.ncd` and the design view in the FPGA Editor tool is shown in Figure 5-4.

The FPGA Editor tool can be invoked through the PlanAhead tool by selecting **Implementation > Implemented Design > FPGA Editor** in the Flow Navigator window.



```
Design creation date: 2012.09.07.20.39.16
Building chip graphics...
```

X1085_c5_04_021513

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

Isolation Design Flow Progress – Design Implementation Block

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

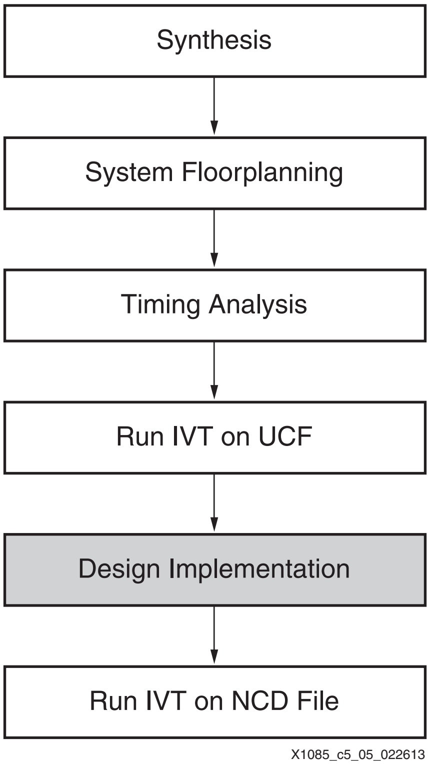


Figure 5-5: **IDF System Design Flow with Design Implementation Block Completed**

Verifying Design with NCD Isolation Verification Tool

Creating File to Run 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. Either use the NCD IVT file (`scc_lab_top_ncd.ivt`) provided in the `\Xilinx_Design\results\ivt` directory, or create a new text file named `scc_lab_top_ncd.ivt` with these contents:

```
# 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
-group           INOUT               U4_IOB

# Combined design
..\planahead\FloorPlan_SCC\FloorPlan_SCC.runs\impl_1\
scc_lab_top_routed.ncd

# Output Report File
-output scc_lab_top_ncd.rpt
```

The NCD IVT command file sets these options:

- Enables the verbose IVT switch
- Assigns four area groups (isolation groups) to the four instance names in the project NCD file
- Points the IVT tool to the combined NCD file
- Specifies the name and location of the output report file

Note: The isolation group names are arbitrary, but the instance name must match the actual design.

Running IVT NCD Test

These steps describe how to run the IVT NCD test.

1. From the Windows Start menu, open an ISE tools command prompt by selecting **Start > All Programs > Xilinx Design Tools > ISE Design Suite 14.4 > Accessories > ISE Design Suite 64 Bit Command Prompt**.

   Note: On a Windows 7 64-bit PC, this provides a command prompt with the Xilinx environment variables and paths configured to run IVT.

2. Navigate to the `\Xilinx_Design\ivt\` directory.
3. To run the IVT NCD test, type this command at the prompt:

   ```
   ivt -f scc_lab_top_ncd.ivt
   ```

   The output for a successful NCD test run is "`.....SUCCESS!`".

Examining Output from IVT NCD Test

IVT creates two types of output files:

- IVT RPT Output File: a standard RPT text report file.
- IVT NCD SVG Output File: 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_top_ncd.rpt`

   The header section of the IVT NCD report lists the provenance:

   ```
   Isolation Verification Report
      Date: Fri Jan 18 09:08:28 2013
      IVT Version: 7.27
      ISE Build: P
      ISE Development Version: ISE 14.4.N/A
      XILINX Directory: C:\Xilinx\14.4\ISE_DS\ISE
      ISE Installed Version: 14.4
      Command Line: ivt -verbose -group AES U1_AES1
      -group AES_r U2_AES2 -group COMPARE U3_Comp
      -group INOUT U4_IOB -output scc_lab_top_ncd.rpt
      ..\planahead\FloorPlan_SCC\FloorPlan_SCC.runs\
      impl_1\scc_lab_top_routed.ncd
      Working Directory: C:\Xilinx_Design\ivt\
      Output File: scc_lab_top_ncd.rpt
      Part: xc7k325t-2ffg900

      Design: scc_lab_top_routed.ncd
   ```

   All of the groups are listed in the input designs section:

   ```
   Section 1 - Isolated Modules

     Group AES module: U1_AES1
     Group AES_r module: U2_AES2
     Group COMPARE module: U3_Comp
     Group INOUT module: U4_IOB
   ```

2. Ensure that clocks and resets are listed in the Uncategorized User Global Nets section. For IDF trusted routing designs, signals shared between isolated regions are expected and intended and appear here. An example of global signals are these global clock signals, which are expected to be shared outside of an isolated region:

```
Section 2 - Uncategorized User Global Nets
   The nets 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, global clocks,
   trusted inter-region signals, or explicitly permitted
   control signals may be global.
```

3. In the Categorized Nets section, ensure that all remaining clocks are listed in the Nets Driven by Global Clock Sources section:

```
Section 3 - Categorized Nets
   The nets listed below present lesser risk than
   uncategorized nets due to their physical extents or
   signal sources.

   Section 3.1 - Nets Driven By Constants or Unused Blocks

   Section 3.2 - Nets Driven By Ground

   Section 3.3 - Nets Driven By Vcc

   Section 3.4 - Nets Driven by Global Clock Sources
   (BUFG, DCM, PLL, and PMCD)
```

4. In the Identified Networks section, shared networks attached to bus macros are listed. All networks passing through bus macros are considered safe by IVT and are listed in this section:

```
Section 4 - Trusted Bus Macros
   The isolation groups listed for each Trusted Bus
   Macro pin are determined by the blocks the net is
   connected to. Therefore a pin may be associated with
   more than one group if the net connected to it is
   connected to blocks in more than one group.

   No trusted bus macros were found.
```

5. Section 5 lists isolation groups and area groups:

```
Section 5 - Area Range Constraints
   Isolation Group        Area Group    Ranges
   ---------------    --------------   -------------
              AES    pblock_U1_AES1   IOB_X1Y97:IOB...
```

6. Section 6 lists violations for tiles with net content:

```
Section 6 - Tiles with Net Content Violations
   No tiles containing nets from multiple isolation
   groups were found.
```

7. Section 7 lists fence tiles with net content:

```
Section 7 - Tiles in the Fence Containing Nets
   The tiles below are in the nominal fence as defined
   by the region outside all isolation groups. This list
   is intended to aid the user in demonstrating that an
   actual fence has been realized.
```

8. Section 8 lists fence tiles with programming content:

```
Section 8 - Tiles in the Fence Containing Programming
   All the fence tiles are in their default
   configurations.
```

9. Section 9 lists fence tiles with used Programmable Interconnect Points (PIPs):

```
Section 9 - Tiles in the Fence Containing Used PIPs
   No wires with terminals in the fence were found.
```

10. Section 10 lists tiles with net adjacency:

```
Section 10 - Tiles with Net Adjacency Violations
   No tile adjacency violations were found.
```

11. Section 11 lists the special-case adjacency violations:

```
Section 11 - Tiles with Special-Case Adjacency Violations
   No special-case adjacency violations were found.
```

12. Pay special attention to the package pins, I/O buffers, and I/O banks listed in section 12 because the note states the user must verify that pins connected to ignored networks are correct.

```
Section 12 - Package Pins, I/O Buffers, and I/O Banks
   Note: It is incumbent on the user to verify that pins
   connected to ignored nets are correct. For example,
   pins may not be directly connected to trusted bus
   macros, but pins may be connected to clocks, power,
   and global resets.

   Pin(col, row) Bank  I/O Buffer Isolation Group  Net
   ------------- ----  ---------- ---------------  ---
   Y28(  2,  10)   13  IOB_X0Y94            INOUT  U4...
```

13. Section 13 lists violations from I/O buffers:

```
Section 13 - I/O Buffer Isolation Violations
   No I/O buffer isolation violations were found.
```

14. Section 14 lists violations from package pins:

```
Section 14 - Package Pin Isolation Violations
   No package pin isolation violations were found.
```

15. Section 15 lists violations from I/O banks:

```
Section 15 - I/O Bank Isolation Violations
   No I/O bank violations were found.
```

16. Section 16 in the NCD report indicates that there are no faults in the NCD and lists the time elapsed to run the test:

```
Section 16 - Isolation Verification Summary

  Tile Adjacency

    Net Adjacency Violations: 0
    Logic Adjacency Violations: 0

  Tile Content

    Net Content Violations: 0
    Logic Content Violations: 0

  Inter-region Signals
    Inter-region Net PIP Violations: 0
    Inter-region Load Violations: 0

  Special Fence Rules

    DSP Violations: 0
    Supersite Violations: 0

  I/O Isolation

    I/O Buffer Isolation Violations: 0
    Package Pin Isolation Violations: 0
    Bank Isolation Violations: 0

  NCD Isolation Verification Summary

    Total isolation violations: 0
    Unrouted nets: 0


  Isolation analysis completed.

  Elapsed time: 0:01:48
```

IVT NCD SVG Output File

The SVG file `scc_lab_top_ncd.svg` created by IVT for NCD mode gives a graphical view of the Kintex-7 device with colored tiles denoting the ownership of the tiles by each isolated 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 Kintex-7 FPGA design used in the SCC lab.

- YELLOW area highlights the INOUT isolated partition
- GRAY area highlights the AES isolated partition
- RED area highlights the AES_R isolated partition
- BLUE area highlights the COMP isolated partition

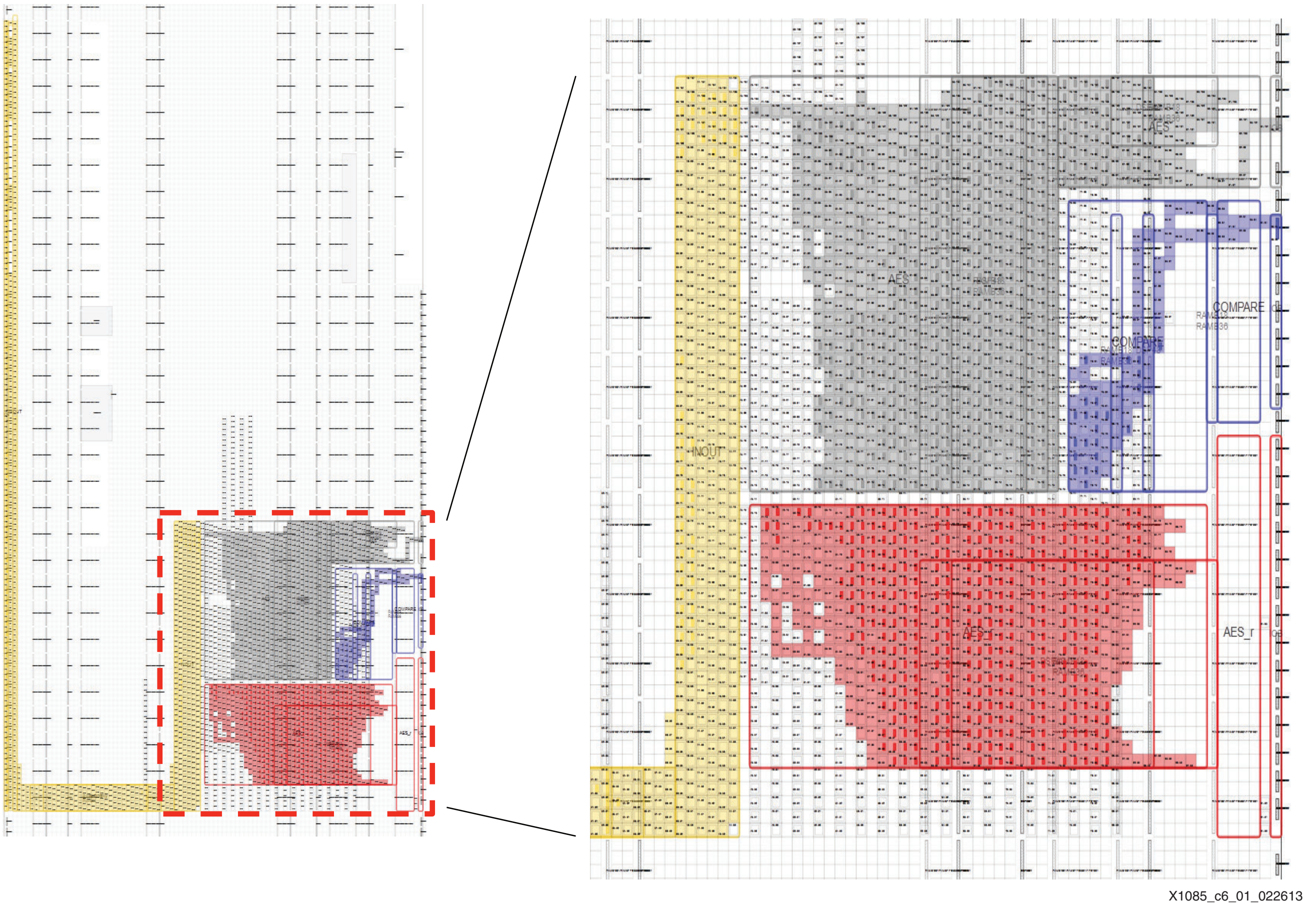


Figure 6-1: **IVT NCD Mode SVG File Graphical Output**

Isolation Design Flow Progress – Run IVT on NCD File Block

The Run IVT on NCD File block of the IDF system design flow diagram is complete, as shown in Figure 6-2.

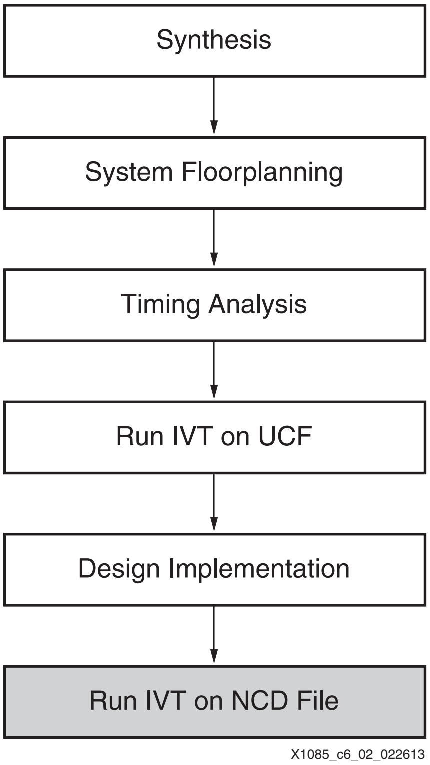


Figure 6-2: **IDF System Design Flow with Run IVT on NCD File Block Completed**

Appendix A

Additional Resources

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see the Xilinx Support website at:

www.xilinx.com/support.

For notification of new or changed documents, subscribe to alerts:

www.xilinx.com/support/alerts

For a glossary of technical terms used in Xilinx documentation, see:

www.xilinx.com/company/terms.htm

Solution Centers

See the Xilinx Solution Centers for support on devices, software tools, and intellectual property at all stages of the design cycle. Topics include design assistance, advisories, and troubleshooting tips.

References

1. Isolation Design Flow page
   www.xilinx.com/applications/isolation-design-flow/index.htm

Further Resources

These documents provide supplemental material useful with this guide:

- *Developing Secure and Reliable Single FPGA Designs with Xilinx 7 Series FPGAs Using the Isolation Design Flow* (XAPP1086)

This application note is also accessible from the Isolation Design Flow page.