PlanAhead Software Tutorial

Leveraging Design Preservation for Predictable Results

UG747 (v 12.3) September 21, 2010

XILINX®

Xilinx is disclosing this Document and Intellectual Property (hereinafter “the Design”) to you for use in the development of designs to operate on, or interface with Xilinx FPGAs. Except as stated herein, none of the
Design may be copied, reproduced, distributed, republished, downloaded, displayed, posted, or transmitted in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Xilinx. Any unauthorized use of the Design may violate copyright laws, trademark laws, the laws of privacy and publicity, and communications regulations and statutes.

Xilinx does not assume any liability arising out of the application or use of the Design; nor does Xilinx convey any license under its patents, copyrights, or any rights of others. You are responsible for obtaining any rights you may require for your use or implementation of the Design. Xilinx reserves the right to make changes, at any time, to the Design as deemed desirable in the sole discretion of Xilinx. Xilinx assumes no obligation to correct any errors contained herein or to advise you of any correction if such be made. Xilinx will not assume any liability for the accuracy or correctness of any engineering or technical support or assistance provided to you in connection with the Design.

THE DESIGN IS PROVIDED “AS IS" WITH ALL FAULTS, AND THE ENTIRE RISK AS TO ITS FUNCTION AND IMPLEMENTATION IS WITH YOU. YOU ACKNOWLEDGE AND AGREE THAT YOU HAVE NOT RELIED ON ANY ORAL OR WRITTEN INFORMATION OR ADVICE, WHETHER GIVEN BY XILINX, OR ITS AGENTS OR EMPLOYEES. XILINX MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE DESIGN, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NONINFRINGEMENT OF THIRD-PARTY RIGHTS.

IN NO EVENT WILL XILINX BE LIABLE FOR ANY CONSEQUENTIAL, INDIRECT, EXEMPLARY, SPECIAL, OR INCIDENTAL DAMAGES, INCLUDING ANY LOST DATA AND LOST PROFITS, ARISING FROM OR RELATING TO YOUR USE OF THE DESIGN, EVEN IF YOU HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THE TOTAL CUMULATIVE LIABILITY OF XILINX IN CONNECTION WITH YOUR USE OF THE DESIGN, WHETHER IN CONTRACT OR TORT OR OTHERWISE, WILL IN NO EVENT EXCEED THE AMOUNT OF FEES PAID BY YOU TO XILINX HEREUNDER FOR USE OF THE DESIGN. YOU ACKNOWLEDGE THAT THE FEES, IF ANY, REFLECT THE ALLOCATION OF RISK SET FORTH IN THIS AGREEMENT AND THAT XILINX WOULD NOT MAKE AVAILABLE THE DESIGN TO YOU WITHOUT THESE LIMITATIONS OF LIABILITY.

The Design is not designed or intended for use in the development of on-line control equipment in hazardous environments requiring fail-safe controls, such as in the operation of nuclear facilities, aircraft navigation or communications systems, air traffic control, life support, or weapons systems (“High-Risk Applications” Xilinx specifically disclaims any express or implied warranties of fitness for such High-Risk Applications. You represent that use of the Design in such High-Risk Applications is fully at your risk.

Demo Design License

The PlanAhead™ software source code includes the source code for the following programs:

Centerpoint XML

- The initial developer of the original code is CenterPoint – Connective Software
- Software Engineering GmbH. portions created by CenterPoint – Connective Software
- Software Engineering GmbH. are Copyright© 1998-2000 CenterPoint - Connective Software Engineering GmbH. All Rights Reserved. Source code for CenterPoint is available at http://www.cpointc.com/XML/

NLView Schematic Engine

- Copyright© Concept Engineering.

Static Timing Engine by Parallax Software Inc.

- Copyright© Parallax Software Inc.

Java Two Standard Edition

- Includes portions of software from RSA Security, Inc. and some portions licensed from IBM are available at http://oss.software.ibm.com/icu4j/
- Powered By JIDE – http://www.jidesoft.com

Free IP Core License

Table of Contents

PlanAhead Software Tutorial

Leveraging Design Preservation for Predictable Results

Introduction

This tutorial provides an overview of the Design Preservation flow, in which you will:

Define Partitions and Implement the design

- Define AREA_GROUP on the partitioned instances and re-implement
- Promote successful implementation results
- Update the top-level partition
- Re-implement the modified top level while importing two unchanged Partitions

Many of the PlanAhead™ software analysis features are covered in more detail in other tutorials, and not every command or command option is covered. The tutorial uses the features contained in the PlanAhead software, which is bundled as a part of the Xilinx® ISE® Design Suite software.

Sample Design Data

This tutorial uses sample design data that is included with the PlanAhead software shipment. The tutorial design data is located in the following directory:

`<`*`ISE_install_dir`*`>/PlanAhead/testcases/PlanAhead_Tutorial.zip`.

The PlanAhead installation process extracts the zip file in the directory automatically. You can extract the zip to a write-accessible location to perform the tutorial. The location of the unzipped `PlanAhead_Tutorial` data is referred to as the `<`*`Extract_Dir`*`>` throughout this document.

Note: If you are using the default `PlanAhead_Tutorial` location at `<`*`ISE_install_dir`*`>/PlanAhead/testcases/PlanAhead_Tutorial`, you can use the **Getting Started > Open an Example Project** links to open the tutorial designs.

Refer to the *Tutorial Description* section of this tutorial for more information about the example design.

Required Xilinx ISE and PlanAhead Software

Access to the ISE Design Suite software is required to perform this tutorial.

Ensure that PlanAhead is operational and the sample design data is installed prior to beginning the tutorial. For installation instructions and information, refer to the *ISE Design Suite 12: Installation, Licensing, and Release Notes* on the Xilinx website:

http://www.xilinx.com/support/documentation/sw_manuals/xilinx12_3/irn.pdf

Required Hardware

Xilinx recommends 2 GB or more of RAM for use with PlanAhead on larger devices. For this tutorial, a smaller design was used. 1 GB of RAM should be sufficient, but it could impact performance.

PlanAhead Documentation and Information

For information about the PlanAhead software, please see the following documents, which are available on the Xilinx website:

- *PlanAhead User Guide* (UG632) - Provides detailed information about the PlanAhead software. http://www.xilinx.com/support/documentation/sw_manuals/xilinx12_3/PlanAhead_UserGuide.pdf
- *Floorplanning Methodology Guide* (UG633) - Provides floorplanning hints. http://www.xilinx.com/support/documentation/sw_manuals/xilinx12_3/Floorplanning_Methodology_Guide.pdf
- *Hierarchical Design Methodology Guide* (UG748) - Provides an overview of the PlanAhead hierarchical design capabilities. http://www.xilinx.com/support/documentation/sw_manuals/xilinx12_3/Hierarchical_Design_Methodology_Guide.pdf

For information about Partial Reconfiguration, see the *Partial Reconfiguration User Guide* (UG702) and tutorials, which are available at the following website: http://www.xilinx.com/tools/partial-reconfiguration

Licenses for Partial Reconfiguration features can be obtained through the Xilinx website at: http://www.xilinx.com/getproduct

For more information about the PlanAhead software, including video demonstrations that cover the benefits of using the software, visit the following website: http://www.xilinx.com/planahead.

Tutorial Description

The sample design used in this tutorial has two sets of synthesis results. The first was created using a standard top-down synthesis flow to be used in a flat implementation. The second set was created using an incremental synthesis flow to be used in the Design Preservation flow, and has separate netlists for module instances that will be partitioned. The design used throughout this tutorial contains:

- A RISC processor
- FFTs
- Gigabit transceivers
- Two USB port modules (to be partitioned)
- An xc6vlx75tff784 device

A small design is used to allow the tutorial to be run with minimal hardware requirements and to enable timely completion, as well as to minimize the data size.

If you have any questions or comments regarding this tutorial, contact Xilinx Technical Support.

Tutorial Objectives

The objectives of this tutorial are to familiarize you with the partitions and the Design Preservation flow using the PlanAhead software.

Tutorial Steps

Step 1 Create a new PlanAhead project

Step 2 Implement and analyze results

Step 3 Re-implement after defining partitions & Pblock ranges

Step 4 Promote successfully implemented partitions

Step 5 Update the top partition

Step 6 Re-implement top while importing USB partitions

Create a New PlanAhead Project Step 1

This tutorial assumes synthesis has already been done, and will use synthesized netlists as the source. For more information on setting up synthesis projects for Design Preservation, refer to the *Hierarchical Design Methodology Guide* (UG748):

http://www.xilinx.com/support/documentation/sw_manuals/xilinx12_3/Hierarchical_Design_Methodology_Guide.pdf

PlanAhead RTL project support for partitioned designs is planned for a future release of PlanAhead.

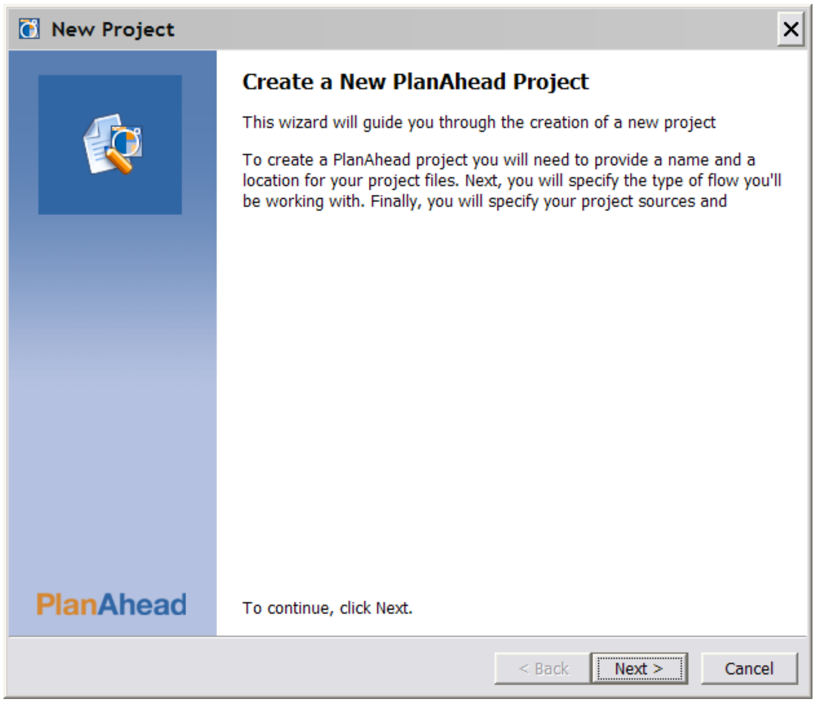
1-1. Create a new netlist PlanAhead project.

1-1-1. Open the PlanAhead software and create the project_pinout I/O Pin Planning project.

- On Windows, double-click the Xilinx PlanAhead 12 Desktop icon, or select **Start > Programs > Xilinx ISE Design Suite 12.3 > PlanAhead > PlanAhead**.
- On Linux, go to `<Extract_Dir>/PlanAhead_Tutorial/Tutorial_Created_Data` directory and type **`planAhead`**.

1-1-2. Open the PlanAhead software. In the Getting Started page, click **Create New Project**.

The Create a New PlanAhead Project page opens (Figure 1).



New Project

Create a New PlanAhead Project

This wizard will guide you through the creation of a new project

To create a PlanAhead project you will need to provide a name and a location for your project files. Next, you will specify the type of flow you'll be working with. Finally, you will specify your project sources and

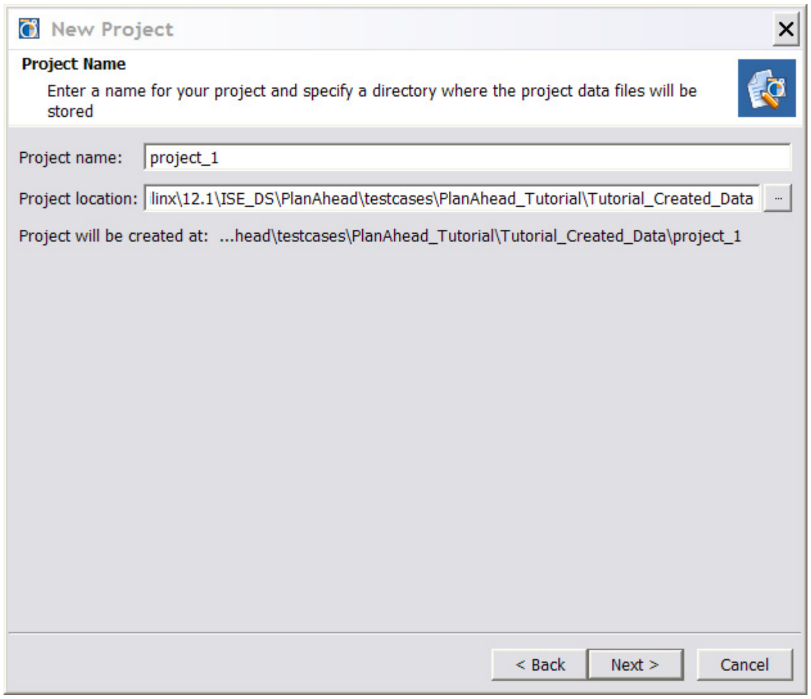
To continue, click Next.

Figure 1: The Create a New PlanAhead Project Page of the New Project Wizard

1-1-3. Click **Next**.

1-1-4. Select a project name and location and click **Next** (Figure 2).

This tutorial uses “project_1” for the project name, and `<Extract_Dir>\PlanAhead_Tutorial\Tutorial_Created_Data` for the project location.



New Project

Project Name

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

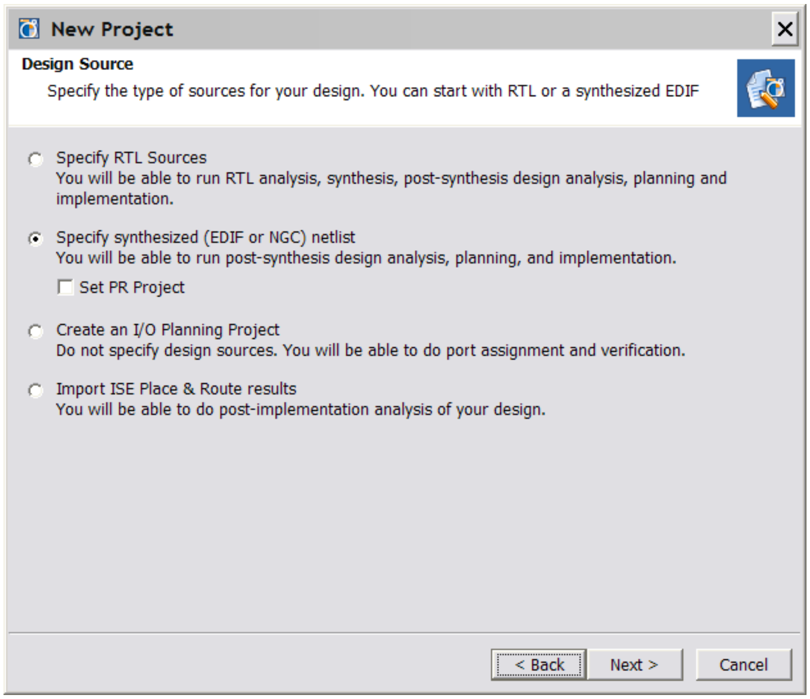
Project name: project_1

Project location: linx\12.1\ISE_DS\PlanAhead\testcases\PlanAhead_Tutorial\Tutorial_Created_Data

Project will be created at: ...head\testcases\PlanAhead_Tutorial\Tutorial_Created_Data\project_1

Figure 2: Project Name Page

1-1-5. In the Design Source page, select **Specify synthesized (EDIF or NGC) netlist** and click **Next** (Figure 3).



New Project

Design Source

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

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

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

□ Set PR Project

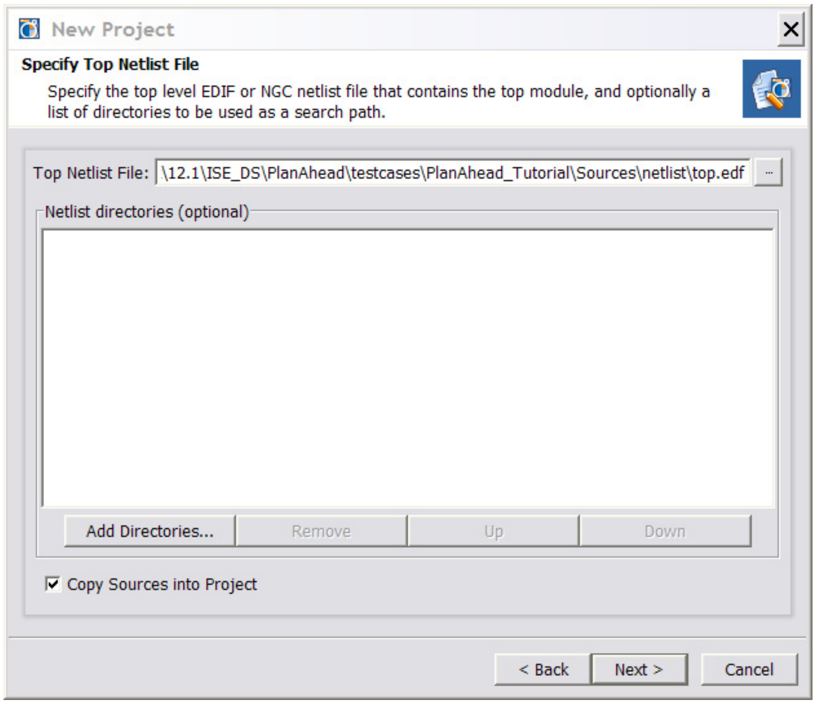
○ Create an I/O Planning Project
Do not specify design sources. You will be able to do port assignment and verification.

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

Figure 3: Design Source Page

1-1-6. In the Specify Top Netlist File page, browse to the netlist located at `<Extract_Dir>\PlanAhead_Tutorial\Sources\netlist\top.edf` and click **Open**.

Because the lower-level netlists are located in this same directory, there is no need to add any optional netlist directories (Figure 4).



New Project

Specify Top Netlist File

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

Top Netlist File: \12.1\ISE_DS\PlanAhead\testcases\PlanAhead_Tutorial\Sources\netlist\top.edf

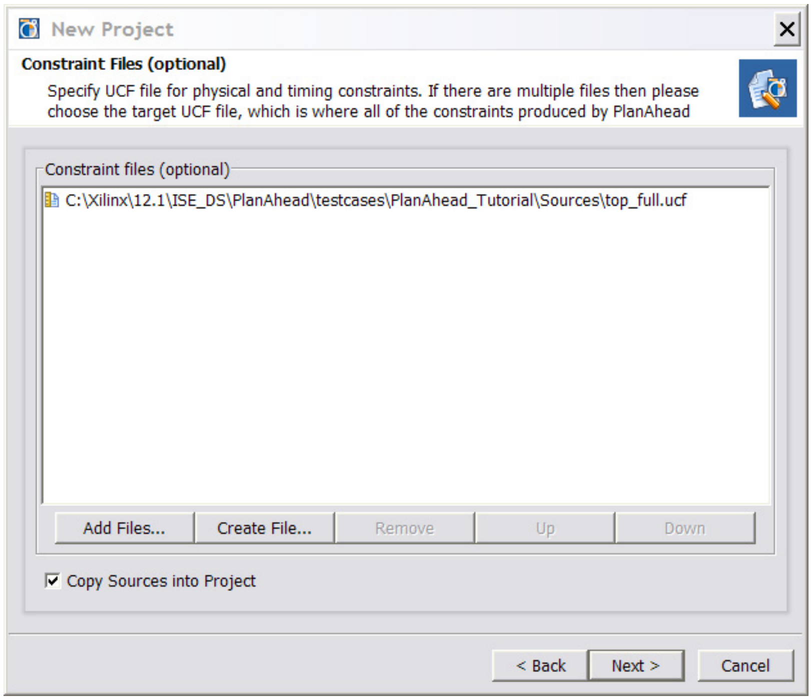
Netlist directories (optional)

☑ Copy Sources into Project

Figure 4: Specifying the Top Netlist File

1-1-7. Click **Next**.

1-1-8. On the Constraints File page, click **Add Files** and specify the UCF located at `<Extract_Dir>\PlanAhead_Tutorial\Sources\top_full.ucf` (Figure 5).



New Project

Constraint Files (optional)

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

Constraint files (optional)

C:\Xilinx\12.1\ISE_DS\PlanAhead\testcases\PlanAhead_Tutorial\Sources\top_full.ucf

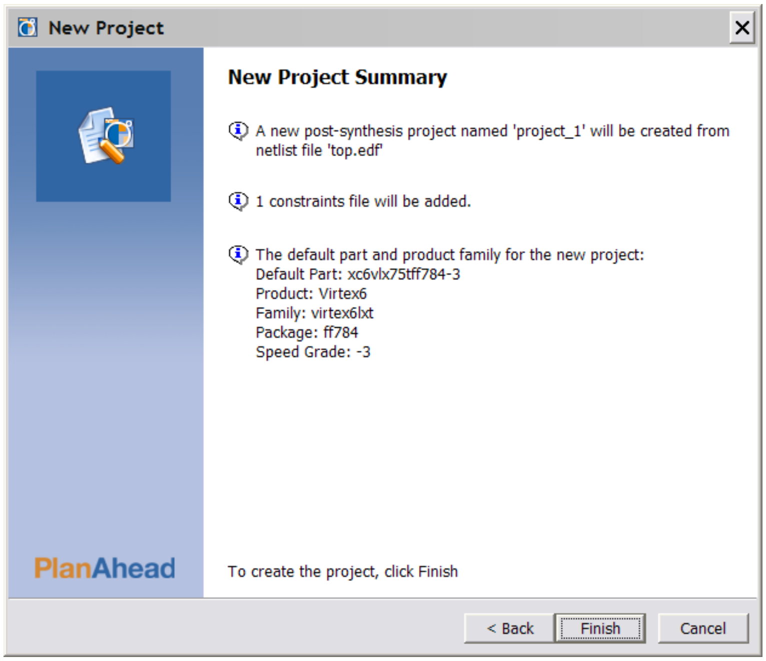
☑ Copy Sources into Project

Figure 5: Specifying the UCF File

1-1-9. Click **Next**.

1-1-10. The Default Part page selects the part from the top-level netlist automatically (xc6vlx75tff784-1). Click **Next.**

1-1-11. Verify the New Project Summary page and click **Finish** to create the project (Figure 6).



New Project

New Project Summary

A new post-synthesis project named 'project_1' will be created from netlist file 'top.edf'

1 constraints file will be added.

The default part and product family for the new project:
Default Part: xc6vlx75tff784-3
Product: Virtex6
Family: virtex6lxt
Package: ff784
Speed Grade: -3

To create the project, click Finish

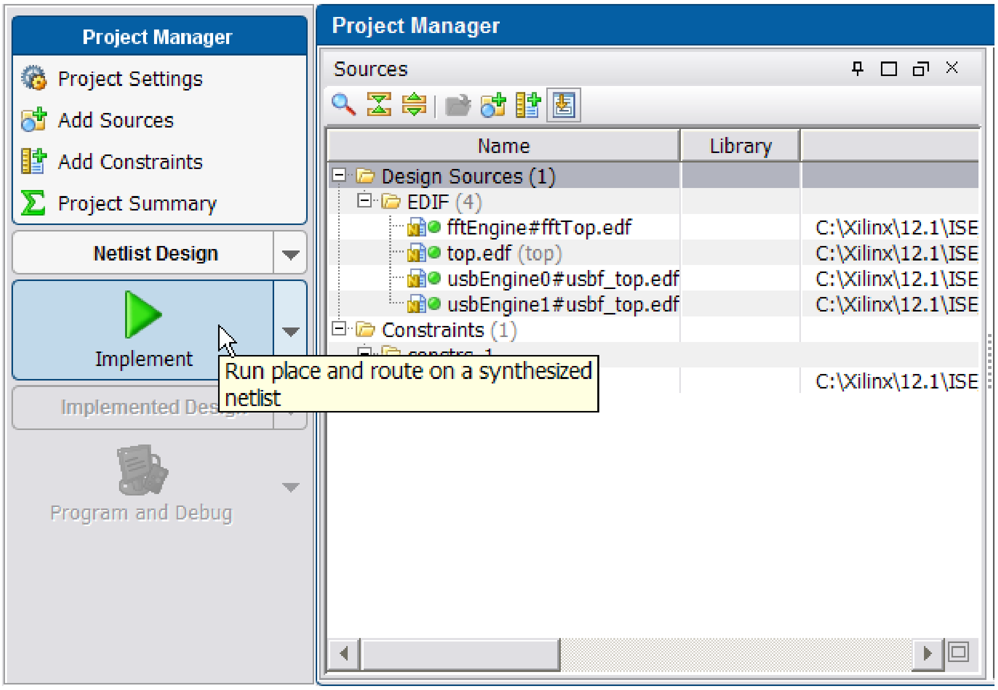
Figure 6: New Project Summary Page

Implement and Analyze Results Step 2

The term "flat flow" refers to a design containing no partitions. When you add partitions, the PlanAhead software invokes the Hierarchical Design (HD) tool set and allows for flows such as the Design Preservation flow. In this tutorial you will implement a flat design, do some analysis on the results, and then add partitions.

1-2. Implement a flat design (no partitions)

1-2-1. From the Flow Navigator, click **Implement** (Figure 7). This launches the ISE implementation tools with a single click. The implementation tools use the current active strategy. The default is the "ISE defaults" strategy.



Run place and route on a synthesized netlist

Figure 7: Implementing the Design

1-2-2. Wait for implementation to complete. If you wish to skip this implementation step, read through step 2-2 to see the analysis of the results, and then continue with the tutorial at step 3.

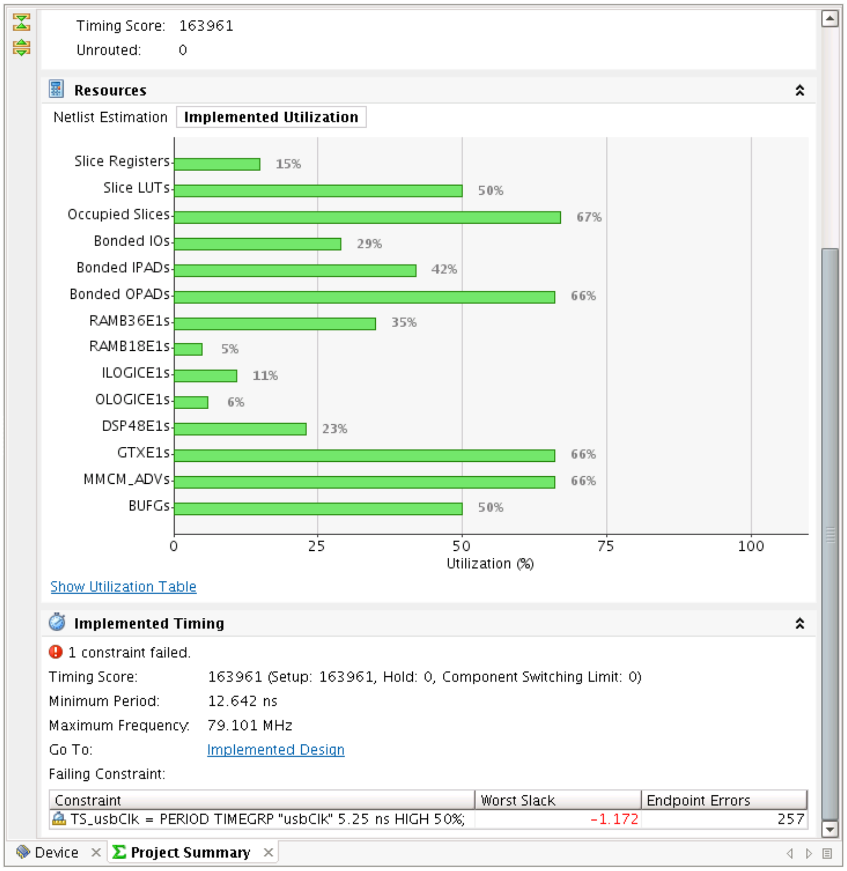
You can monitor the implementation progress several ways:

- From the status bar in the upper right-hand corner of PlanAhead (active run only)
- From the Project Summary view
- From the Design Runs view, which can opened using **Window > Design Runs**
- From the Compilation Log view, which shows more detail

1-3. Analyze the implementation results

1-3-1. When Implementation completes, the Implementation Completed dialog box appears with several choices. Select **Open Implemented Design** and click **OK**. If you do not see this dialog box or choose another option, the results can be loaded by clicking **Implemented Design** in the Flow Navigator.

1-3-2. When the implementation results finish loading, a summary is available from the Project Summary view. Click the **Project Summary** tab and scroll to the bottom to see the Implemented Timing summary (Figure 8). Note the timing score is not "0", indicating that the design has timing failures. The summary lists the failing timing constraint also.



Timing Score: 163961
Unrouted: 0

Resources

Netlist Estimation | Implemented Utilization

Show Utilization Table

Implemented Timing

1 constraint failed.

Timing Score: 163961 (Setup: 163961, Hold: 0, Component Switching Limit: 0)
Minimum Period: 12.642 ns
Maximum Frequency: 79.101 MHz
Go To: Implemented Design
Failing Constraint:

Constraint	Worst Slack	Endpoint Errors
TS_usbClk = PERIOD TIMEGRP "usbClk" 5.25 ns HIGH 50%;	-1.172	257

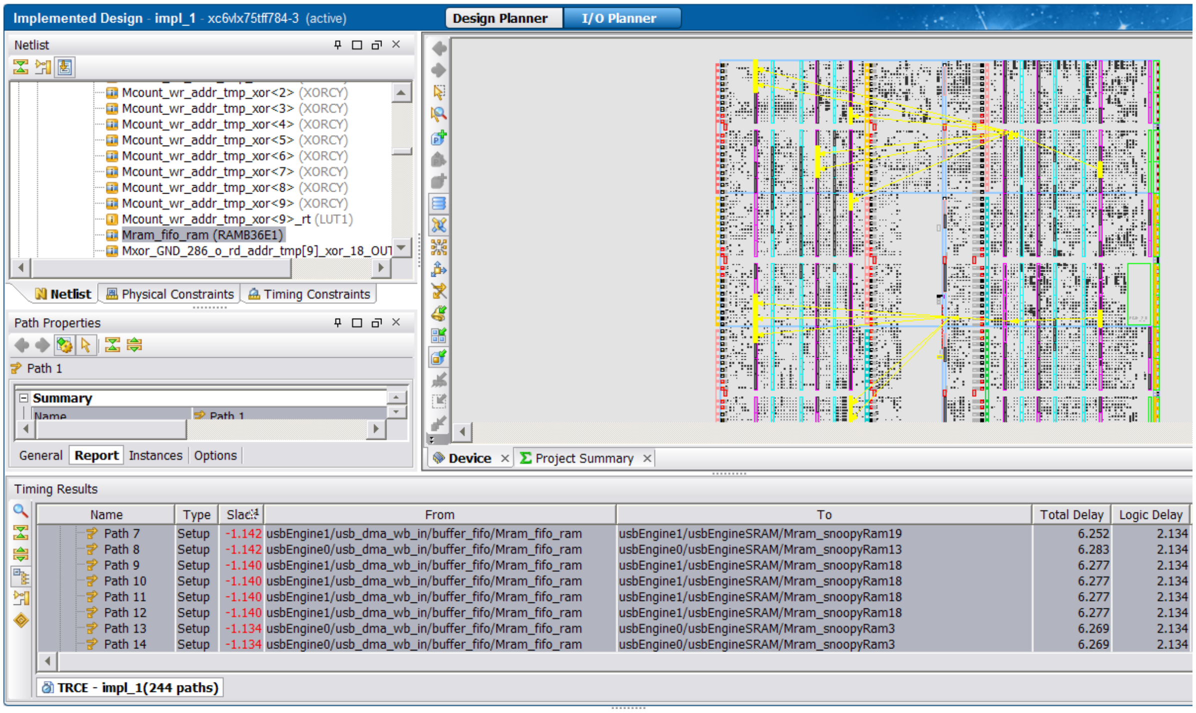
Figure 8: Project Summary View

1-3-3. Now that the implementation results are loaded, the timing report is available from the Timing Results view. You can use this view to see the details of the failing paths.

Note the failing paths in the usbEngine modules. There may also be some failing paths in the fftEngine, but these paths are the result of the router spending minimal time on them after they determine the usbEngine modules will not meet timing.

1-3-4. Click the **Device** tab to switch from the Project Summary view to the Device view.

1-3-5. Highlight the failing paths by selecting them in the Timing Results view (Figure 9).



Mcount_wr_addr_tmp_xor<2> (XORCY)
Mcount_wr_addr_tmp_xor<3> (XORCY)
Mcount_wr_addr_tmp_xor<4> (XORCY)
Mcount_wr_addr_tmp_xor<5> (XORCY)
Mcount_wr_addr_tmp_xor<6> (XORCY)
Mcount_wr_addr_tmp_xor<7> (XORCY)
Mcount_wr_addr_tmp_xor<8> (XORCY)
Mcount_wr_addr_tmp_xor<9> (XORCY)
Mcount_wr_addr_tmp_xor<9>_rt (LUT1)
Mram_fifo_ram (RAMB36E1)
Mxor_GND_286_o_rd_addr_tmp[9]_xor_18_OUT

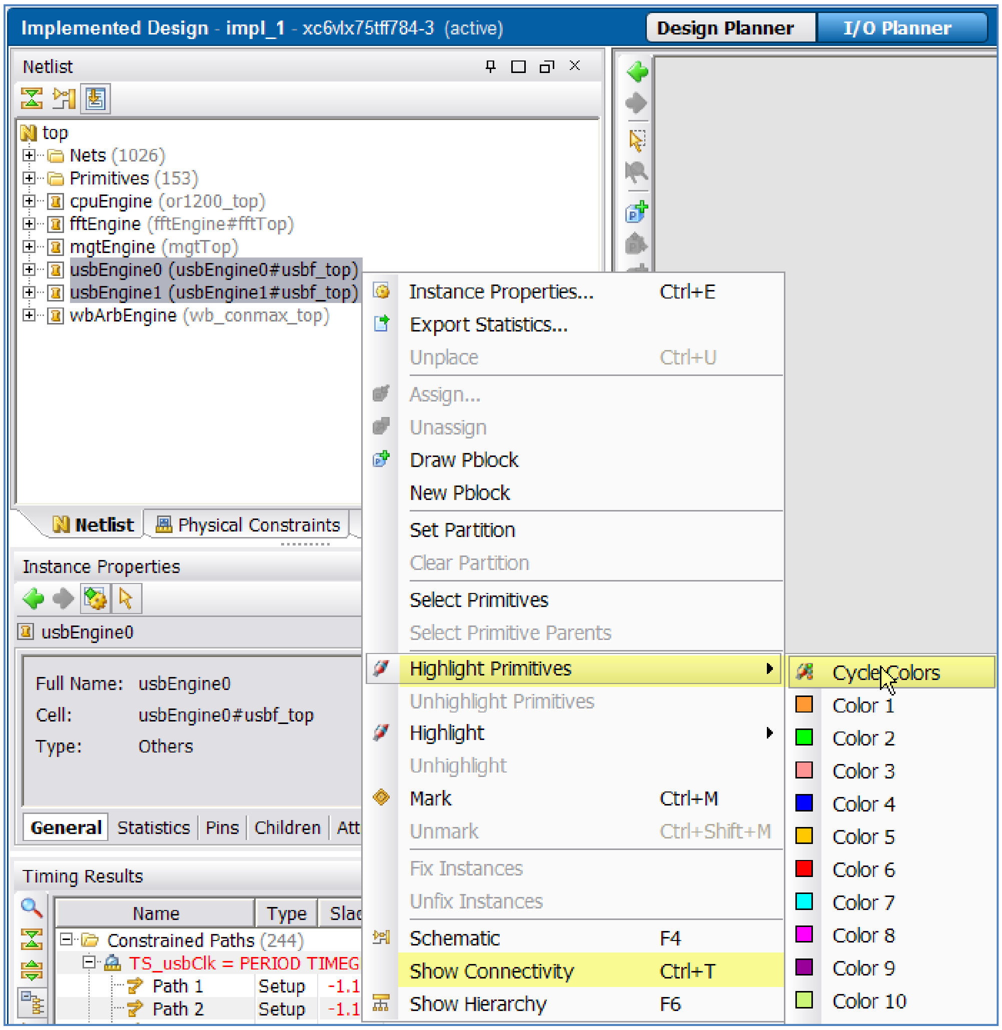
Name	Type	Slack	From	To	Total Delay	Logic Delay
Path 7	Setup	-1.142	usbEngine1/usb_dma_wb_in/buffer_fifo/Mram_fifo_ram	usbEngine1/usbEngineSRAM/Mram_snoopyRam19	6.252	2.134
Path 8	Setup	-1.142	usbEngine0/usb_dma_wb_in/buffer_fifo/Mram_fifo_ram	usbEngine0/usbEngineSRAM/Mram_snoopyRam13	6.283	2.134
Path 9	Setup	-1.140	usbEngine1/usb_dma_wb_in/buffer_fifo/Mram_fifo_ram	usbEngine1/usbEngineSRAM/Mram_snoopyRam18	6.277	2.134
Path 10	Setup	-1.140	usbEngine1/usb_dma_wb_in/buffer_fifo/Mram_fifo_ram	usbEngine1/usbEngineSRAM/Mram_snoopyRam18	6.277	2.134
Path 11	Setup	-1.140	usbEngine1/usb_dma_wb_in/buffer_fifo/Mram_fifo_ram	usbEngine1/usbEngineSRAM/Mram_snoopyRam18	6.277	2.134
Path 12	Setup	-1.140	usbEngine1/usb_dma_wb_in/buffer_fifo/Mram_fifo_ram	usbEngine1/usbEngineSRAM/Mram_snoopyRam18	6.277	2.134
Path 13	Setup	-1.134	usbEngine0/usb_dma_wb_in/buffer_fifo/Mram_fifo_ram	usbEngine0/usbEngineSRAM/Mram_snoopyRam3	6.269	2.134
Path 14	Setup	-1.134	usbEngine0/usb_dma_wb_in/buffer_fifo/Mram_fifo_ram	usbEngine0/usbEngineSRAM/Mram_snoopyRam3	6.269	2.134

Figure 9: Selecting Timing Errors

Looking at the failing paths, you can determine the following:

- o The failing paths appear to be primarily BRAM-related.
- o The BRAMs that connect to each other appear to be very far apart.

1-3-6. To highlight the two usbEngine instance primitives, select them in the netlist view, right-click, and select **Highlight Primitives > Cycle Colors** (Figure 10).



Full Name: usbEngine0
Cell: usbEngine0#usbf_top
Type: Others

Name	Type	Slac
Constrained Paths (244)		
TS_usbClk = PERIOD TIMEG		
Path 1	Setup	-1.1
Path 2	Setup	-1.1

Figure 10: Highlighting usbEngine Primitives

1-3-7. To view the connectivity for the usbEngine cores, select them in the Netlist view, right-click and select **Show Connectivity**.

1-3-8. Look at the results of the previous two steps in the Device view (Figure 11).

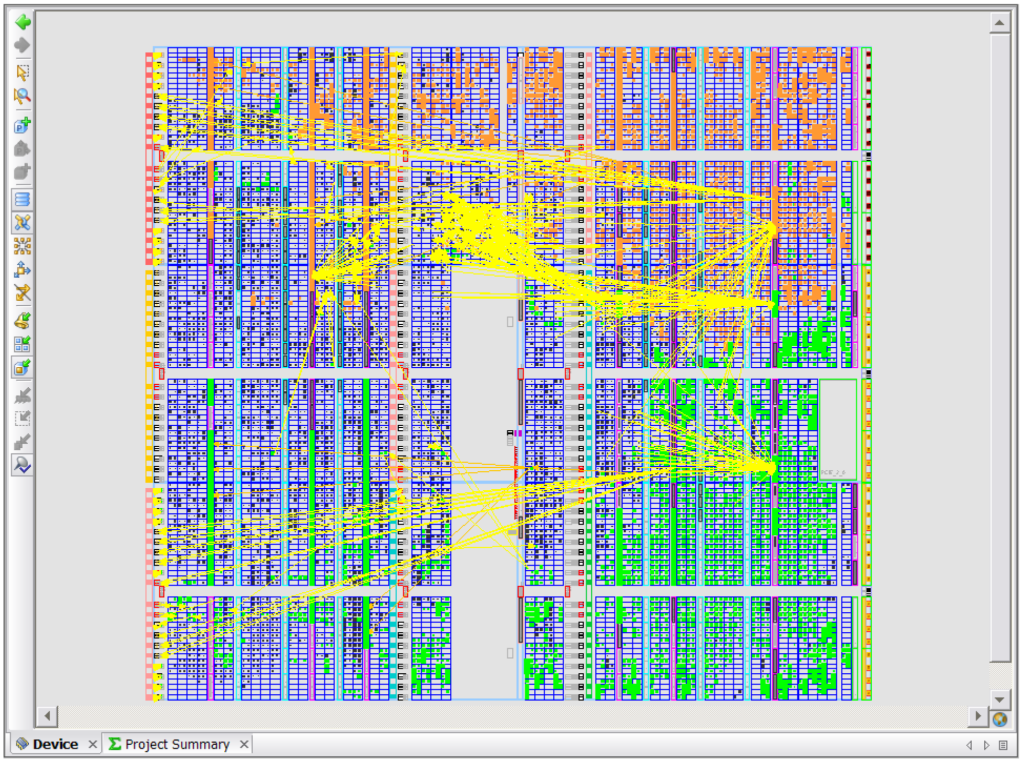


Figure 11: Device View after Selecting Show Connectivity

You can see that the placement of the two usbEngine cores is quite spread out.

- The usbEngine1 instance appears to reside on the top half of the device, while the usbEngine0 instance resides mainly on the lower half of the device.
- You can also see from the connectivity that the two usbEngine cores interface to I/O ports along the left side of the device (usbEngine1 connects to I/O on the top half, while usbEngine0 connects to I/O on the bottom half).

It would be ideal to control the placement of these two cores to force the logic to the left side of the device. This would not only improve timing to the I/O interfaces of these cores, but also would force tighter BRAM placement and free up the rest of the device for other modules.

In addition, it would be helpful to preserve any implementation results where these instances do meet timing. They are the timing critical modules of this design and are completed USB cores with no additional logic changes expected. Consequently, these cores are ideal candidates for partitioning.

You can close the Implemented Design view at this point.

1-3-9. From the Flow Navigator, select **Implemented Design** > **Close**.

Re-implement After Defining Partitions & Pblock Ranges Step 3

Based on the timing results from the previous step, the usbEngine cores have been identified as instances for partitioning because they are timing-critical modules and it would be advantageous to preserve successful results. However, this fact alone does not make these good candidates for partitions. These are good choices for partitions because they are logically isolated from the rest of the design, and have reasonable interface timing. You can use the PlanAhead DRCs to help identify whether or not a module is a good choice for partitions. For more information on how to choose good module instances for partitioning, refer to the *Hierarchical Design Methodology Guide* (UG748):

http://www.xilinx.com/support/documentation/sw_manuals/xilinx12_3/Hierarchical_Design_Methodology_Guide.pdf

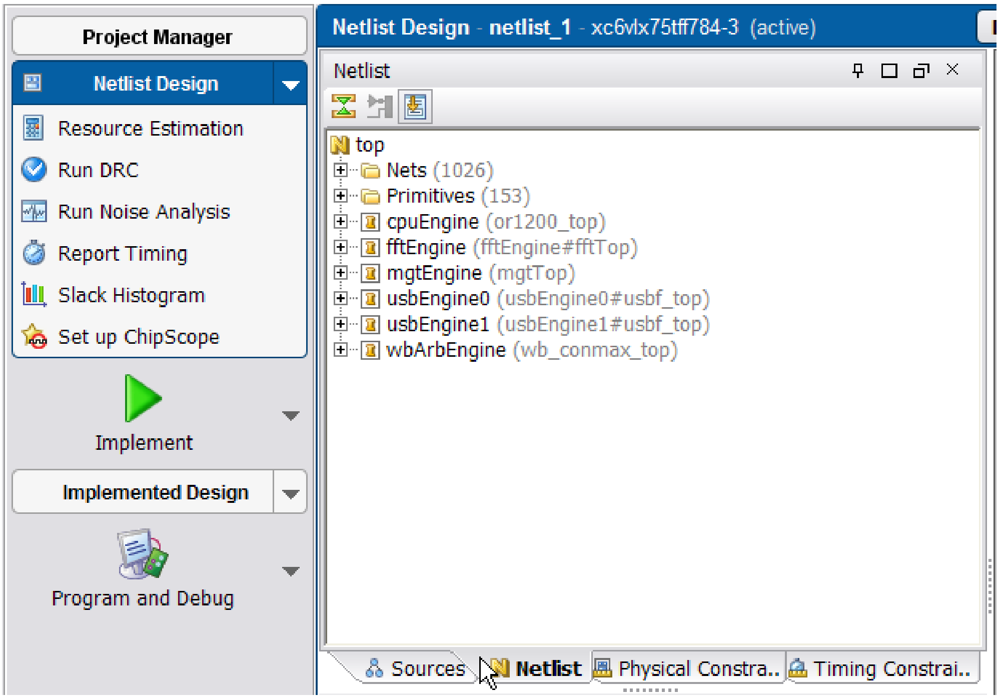
Partitioned instances can be floorplanned just like any other instance, and creating Pblock (AREA_GROUP) constraints can help achieve timing closure and improve runtime. As noted in the previous step, the usbEngine instances interface to I/O along the left side of the part, and reinforces the decision to floorplan these modules to help control placement and routing.

For this tutorial, the AREA_GROUP constraints are provided for the two instances of usbEngine. For more information on creating AREA_GROUP constraints using the PlanAhead software, refer to the *PlanAhead Tutorial: Design Analysis and Floorplanning for Performance* (UG676):

http://www.xilinx.com/support/documentation/dt_planahead_planahead12-3_tutorials.htm

2-1. Create Partitions for the two usbEngine cores.

2-1-1. From the Flow Navigator, open the Netlist Design view and select the Netlist tab (Figure 12).



```
top
  Nets (1026)
  Primitives (153)
  cpuEngine (or1200_top)
  fftEngine (fftEngine#fftTop)
  mgtEngine (mgtTop)
  usbEngine0 (usbEngine0#usbf_top)
  usbEngine1 (usbEngine1#usbf_top)
  wbArbEngine (wb_conmax_top)
```

Figure 12: The PlanAhead RTL Environment

2-1-2. Highlight the two usbEngine instances, right-click and select **Set Partition** (Figure 13). Once the modules have been set as partitions, the icon associated with the instances changes from ▣ to ▢.

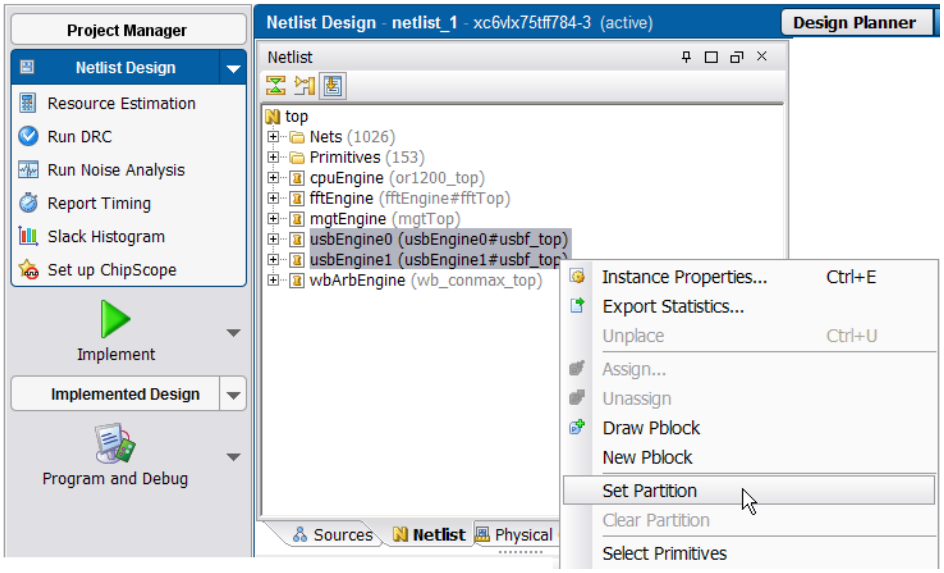


Figure 13: Setting Partitions on usbEngine Cores

2-2. Add AREA_GROUP constraints and launch implementation

2-2-1. From the Netlist view, select and right-click usbEngine1, and select **Draw Pblock**. (**Note**: You can use the Draw Pblock button from the Device view toolbar instead).

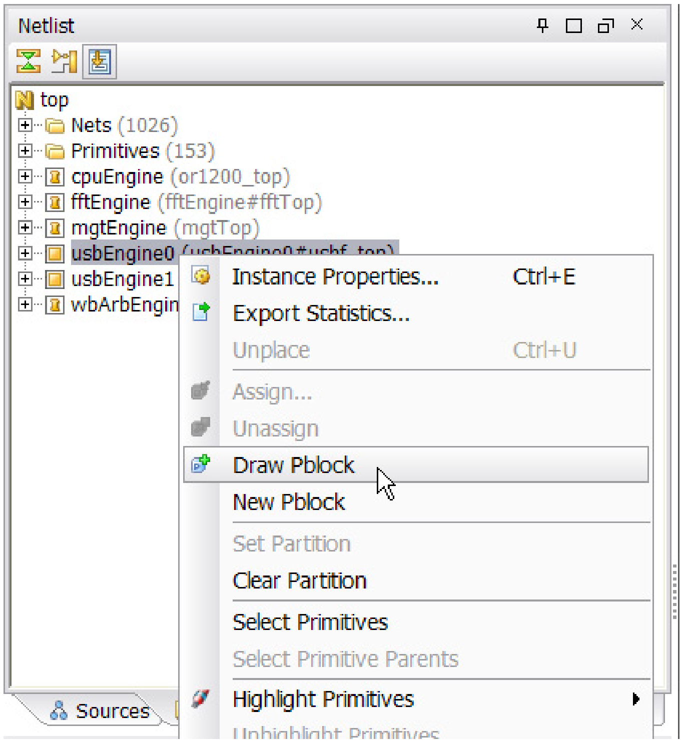


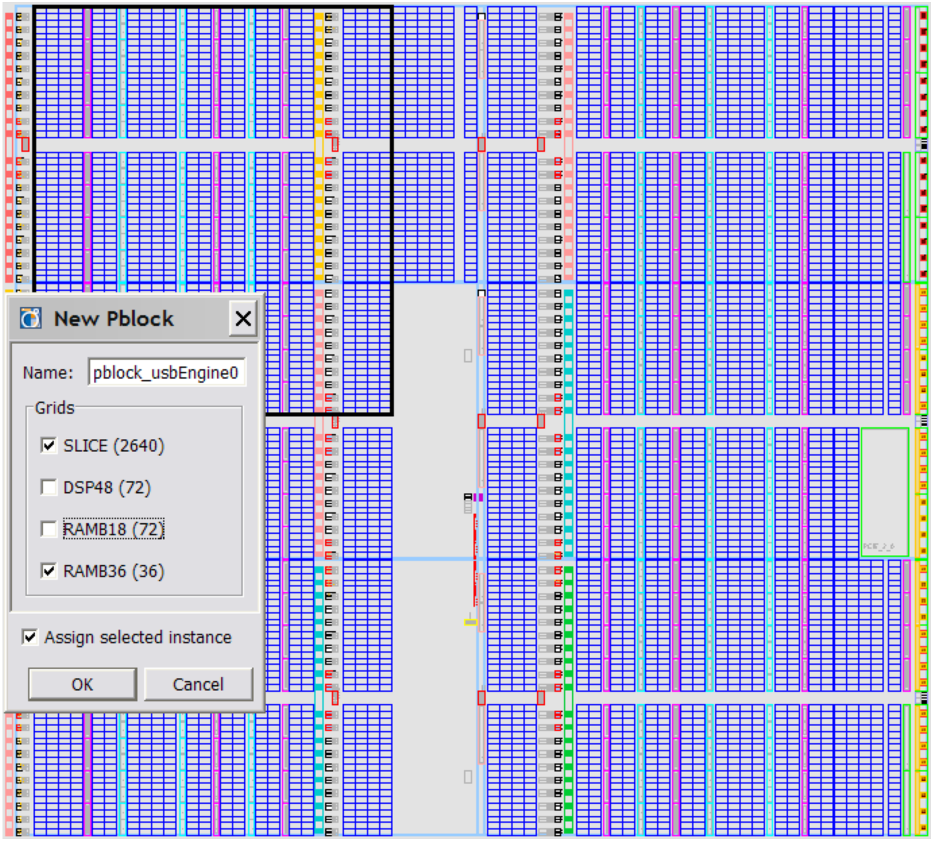
Figure 14: Selecting the Draw Pblock Tool

2-2-2. With the Draw Pblock tool active, move the cursor to the Device view.

2-2-3. Left click on the top-left corner of the device where the CLBs start, and without letting go of the left mouse button, drag to create a rectangle covering most of the top-left quadrant of the device (Figure 15).

2-2-4. In the New Pblock dialog box, verify that the SLICE and RAMB36 grids are selected, and deselect other resources that are not needed (Figure 15).

2-2-5. Click **OK**.



New Pblock

Name: pblock_usbEngine0

Grids

☑ SLICE (2640)

☐ DSP48 (72)

☐ RAMB18 (72)

☑ RAMB36 (36)

☑ Assign selected instance

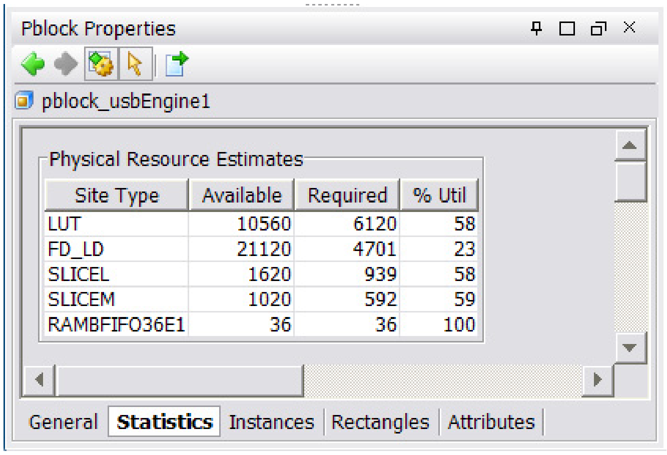
Figure 15: Pblock Rectangle for usbEngine1

The most important thing to note here is the number of available RAMB36 resources. If the Pblock rectangle does not completely cover the RAMB36 resources shown in Figure 15, the number of resources could be less than the required 36. If this is the case, adjust the size of the Pblock rectangle by selecting it and resizing.

2-3. Repeat Step 3-2 for usbEngine0 on bottom-left quadrant.

2-4. Verify that the number of RAMB36 resources is 36.

2-4-1. Select each Pblock and view the Pblock statics in the Pblock Properties view (Figure 16).



Physical Resource Estimates

Site Type	Available	Required	% Util
LUT	10560	6120	58
FD_LD	21120	4701	23
SLICEL	1620	939	58
SLICEM	1020	592	59
RAMBFIFO36E1	36	36	100

Figure 16: Viewing the Pblock Properties

2-4-2. Make any necessary adjustments to the size and location of the two Pblocks to make the floorplan match the one shown in Figure 17.

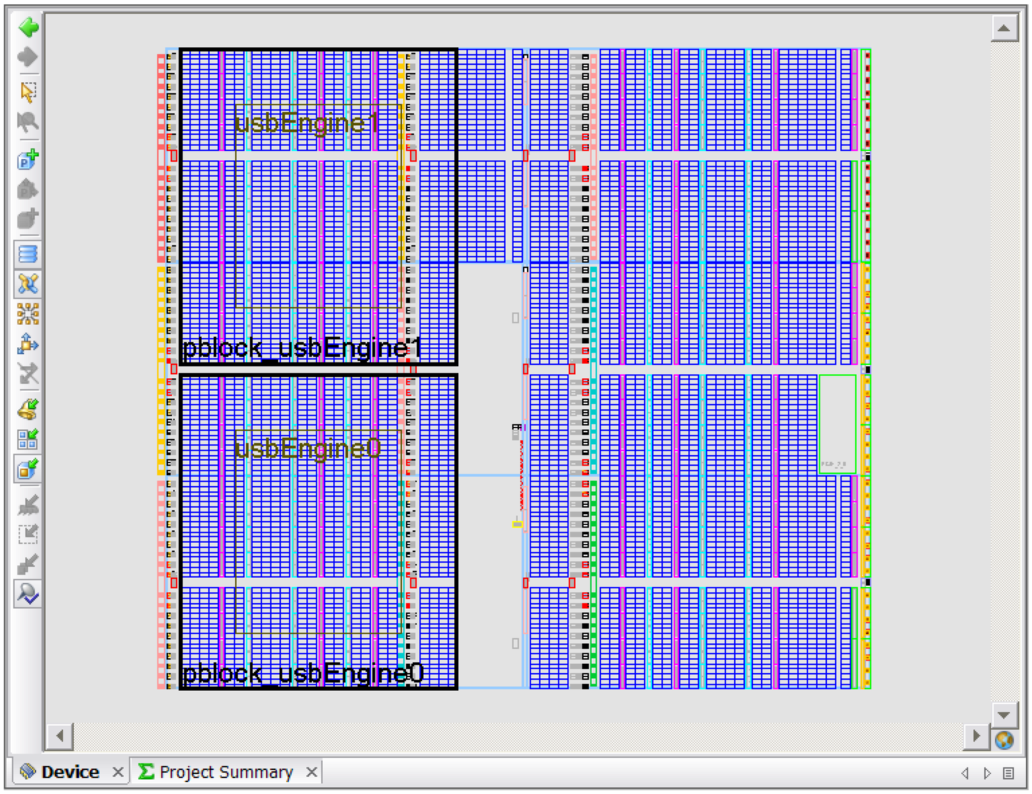


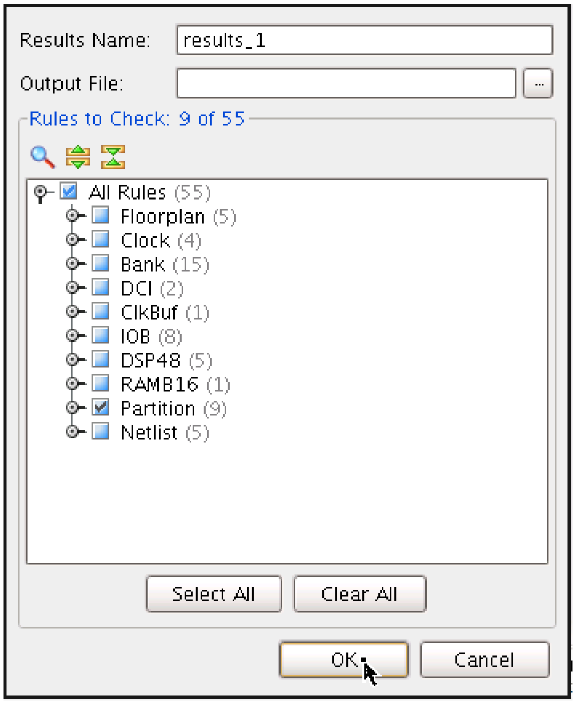
Figure 17: Final Pblock Rectangles

2-5. Run DRC checks on partitions.

Now that partitions have been added to the project, run Design Rule Checks (DRCs) to check for any issues.

2-5-1. Under the Netlist Design in the Flow Navigator, click **Run DRC** (or select **Tools** > **Run DRC**).

2-5-2. From the Run DRC dialog box, unselect all rules except "Partition," and click **OK** (Figure 18).



Results Name: results_1

Output File:

Rules to Check: 9 of 55

- ☑ All Rules (55)
  - □ Floorplan (5)
  - □ Clock (4)
  - □ Bank (15)
  - □ DCI (2)
  - □ ClkBuf (1)
  - □ IOB (8)
  - □ DSP48 (5)
  - □ RAMB16 (1)
  - ☑ Partition (9)
  - □ Netlist (5)

Figure 18: Run Partition DRCs

Note that the DRC returns minor Advisory messages. PlanAhead can report Advisory, Warning, Error, and Fatal messages for the DRC rules. You can ignore the Advisory messages given for this tutorial design. For an actual design, however, all messages reported by the DRCs should be understood and dealt with properly.

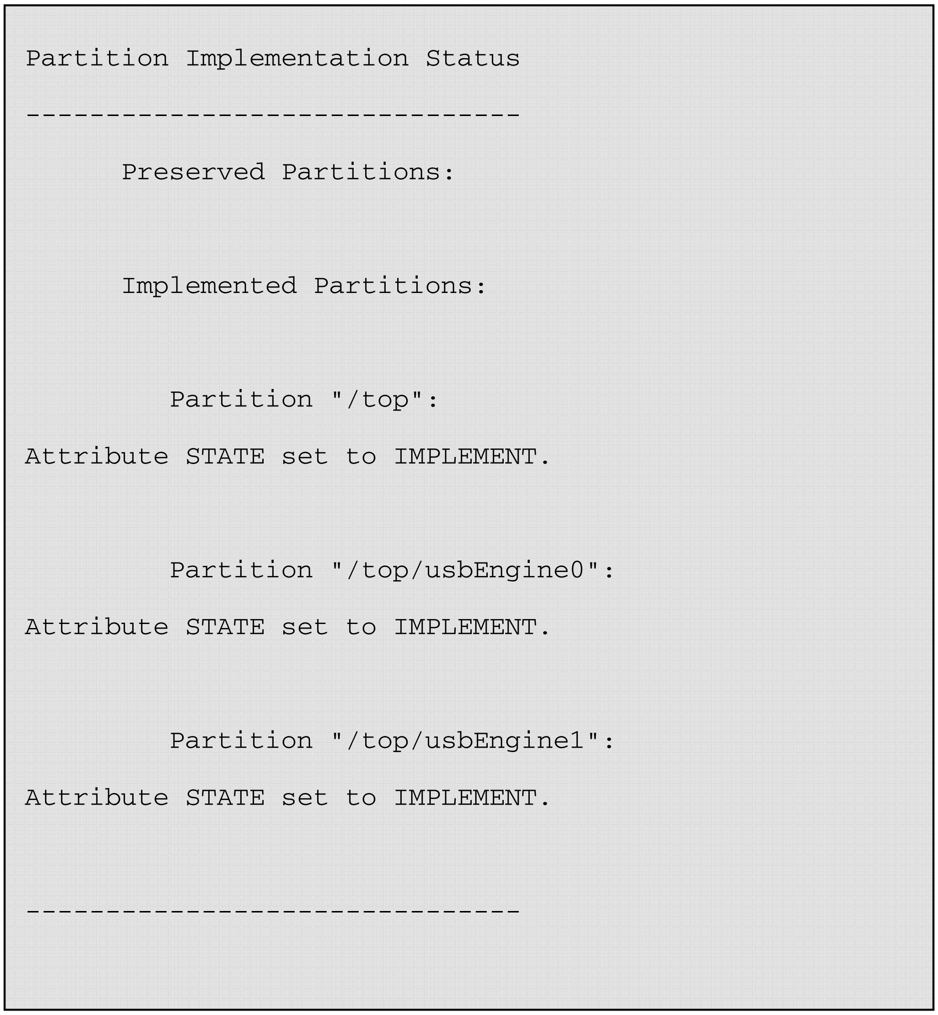
2-5-3. In the Flow Navigator, click **Implement**.

This time there are Partitions and AREA_GROUP constraints on the usbEngine cores, so the results are different.

2-5-4. Once NGDBuild completes, click the Report tab to get a list of all the ISE Implementation report files.

2-5-5. Once processes are finished, report files become available to open. Double-click the NGDBuild Report to open the report file. Scroll to the bottom of the report file and note the partition information (Figure 19).

This information is provided in every report file (NGDBuild, Map, and PAR) and is an easy way to verify the status of all partitions on a given run.



```
Partition Implementation Status
-------------------------------
      Preserved Partitions:


      Implemented Partitions:


         Partition "/top":
Attribute STATE set to IMPLEMENT.


         Partition "/top/usbEngine0":
Attribute STATE set to IMPLEMENT.


         Partition "/top/usbEngine1":
Attribute STATE set to IMPLEMENT.


-------------------------------
```

Figure 19: Partition Implementation Status in Report Files

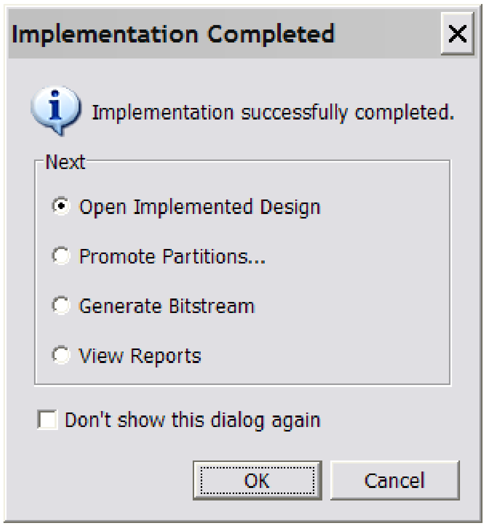
This implementation takes about an hour, depending on the system you are using to run the ISE tools.

Promote Successfully Implemented Partitions Step 4

Once an implementation is successful, the results can be promoted. Promoting results makes a copy of the implementation directory in `<project_name>.promote\X<run_name>` (for example, `project_1.promote\Ximpl_1`). PlanAhead keeps track of the latest promoted run and sets the state of any promoted partitions automatically to import from the most current promote directory. All of this can be managed manually through PlanAhead.

3-1. Promote the successful implementation results

3-1-1. Once the implementation completes, the Implementation Completed dialog box appears, providing several choices (Figure 20).



Implementation Completed

Implementation successfully completed.

Next

● Open Implemented Design

○ Promote Partitions...

○ Generate Bitstream

○ View Reports

□ Don't show this dialog again

Figure 20: Implementation Completed Dialog Box

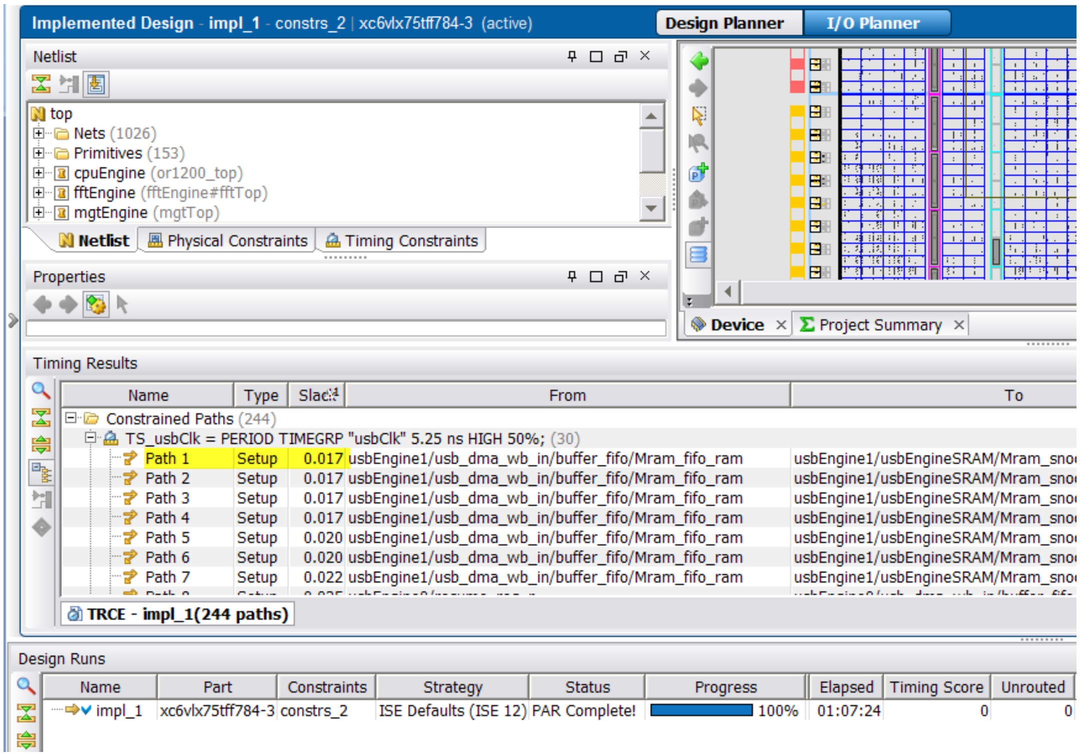
3-1-2. Select **Open Implemented Design** and click **OK**.

Note the other options available. The results could be promoted here using the Promote Partitions option, but this tutorial promotes the results another way.

If you did not see the Implementation Completed dialog box, you can load the results by clicking **Implemented Design** in the Flow Navigator.

3-1-3. To verify the results were successful, look at the final timing score and view the detailed timing report (Figure 21).

Important: This tutorial is being run on the slowest speed grade, and some very small timing errors may still exist at this point. Since the goal of the tutorial is to show the Design Preservation flow and to highlight the exact preservation of placement and routing, these small timing errors can be ignored. If this design were going into hardware, additional time and effort could be spent to close timing completely.



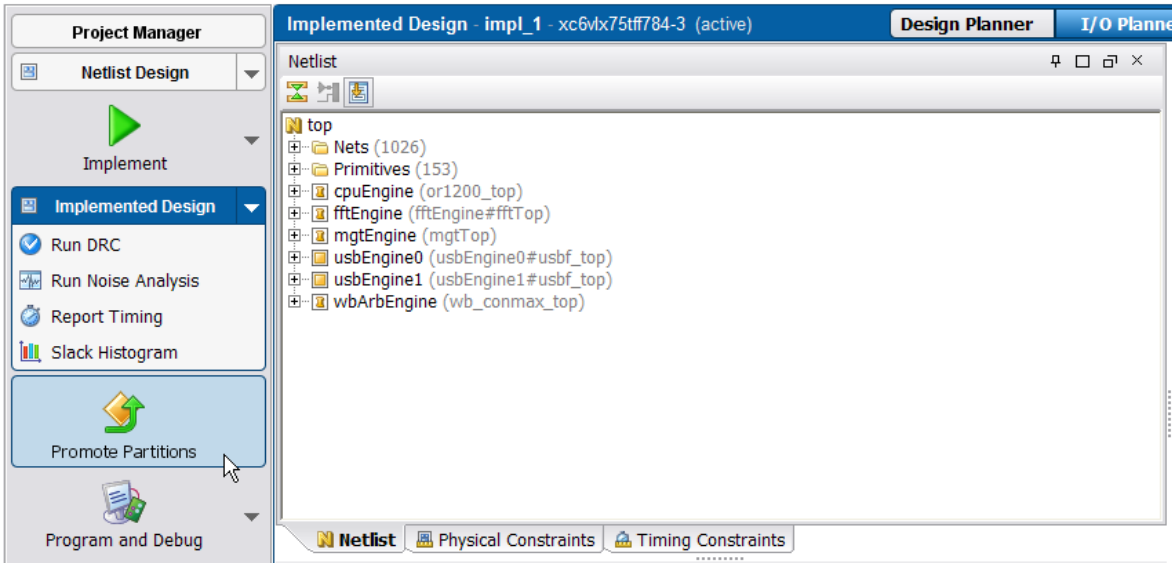
top
- Nets (1026)
- Primitives (153)
- cpuEngine (or1200_top)
- fftEngine (fftEngine#fftTop)
- mgtEngine (mgtTop)

Name	Type	Slack	From	To
Constrained Paths (244)				
TS_usbClk = PERIOD TIMEGRP "usbClk" 5.25 ns HIGH 50%; (30)				
Path 1	Setup	0.017	usbEngine1/usb_dma_wb_in/buffer_fifo/Mram_fifo_ram	usbEngine1/usbEngineSRAM/Mram_sno
Path 2	Setup	0.017	usbEngine1/usb_dma_wb_in/buffer_fifo/Mram_fifo_ram	usbEngine1/usbEngineSRAM/Mram_sno
Path 3	Setup	0.017	usbEngine1/usb_dma_wb_in/buffer_fifo/Mram_fifo_ram	usbEngine1/usbEngineSRAM/Mram_sno
Path 4	Setup	0.017	usbEngine1/usb_dma_wb_in/buffer_fifo/Mram_fifo_ram	usbEngine1/usbEngineSRAM/Mram_sno
Path 5	Setup	0.020	usbEngine1/usb_dma_wb_in/buffer_fifo/Mram_fifo_ram	usbEngine1/usbEngineSRAM/Mram_sno
Path 6	Setup	0.020	usbEngine1/usb_dma_wb_in/buffer_fifo/Mram_fifo_ram	usbEngine1/usbEngineSRAM/Mram_sno
Path 7	Setup	0.022	usbEngine1/usb_dma_wb_in/buffer_fifo/Mram_fifo_ram	usbEngine1/usbEngineSRAM/Mram_sno

Name	Part	Constraints	Strategy	Status	Progress	Elapsed	Timing Score	Unrouted
impl_1	xc6vlx75tff784-3	constrs_2	ISE Defaults (ISE 12)	PAR Complete!	100%	01:07:24	0	0

Figure 21: Verifying Successful Implementation Results

3-1-4. In the Flow Navigator, click **Promote Partitions** to promote the implemented run (Figure 22).



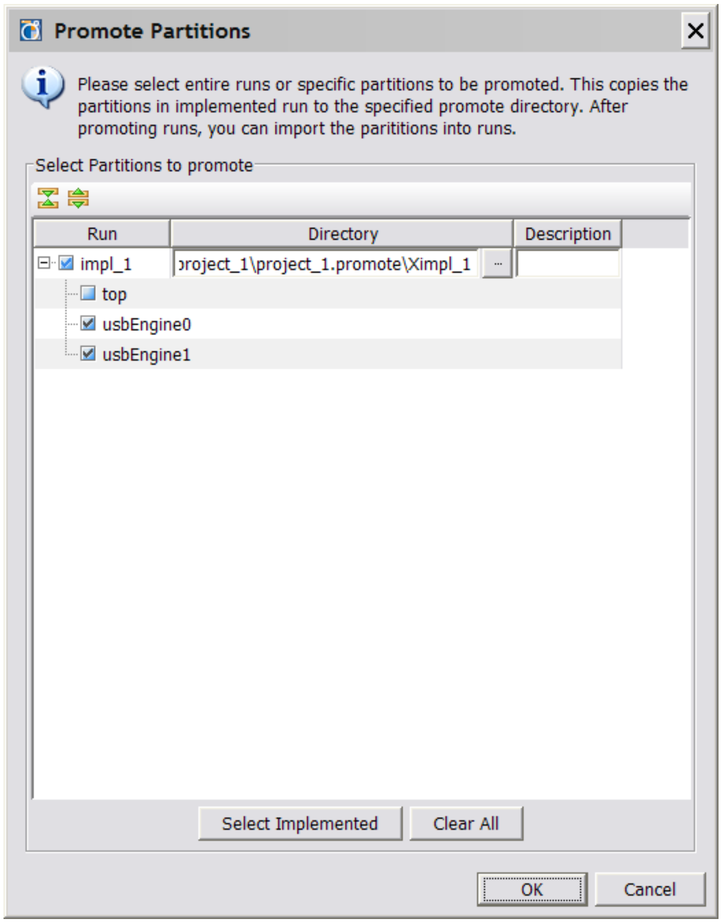
top
- Nets (1026)
- Primitives (153)
- cpuEngine (or1200_top)
- fftEngine (fftEngine#fftTop)
- mgtEngine (mgtTop)
- usbEngine0 (usbEngine0#usbf_top)
- usbEngine1 (usbEngine1#usbf_top)
- wbArbEngine (wb_conmax_top)

Figure 22: Promote Partitions

3-1-5. Verify that the two usbEngine module instances are checked for promoting (Figure 23).

Note that the top-level partition is not selected by default. However, this partition could be promoted like any other partition. For this tutorial, you will update the top-level partition, so there is no need to promote it here.

3-1-6. Click **OK** to promote the two usbEngine partitions. Optionally, you can enter a description about the promoted data.



Promote Partitions

Please select entire runs or specific partitions to be promoted. This copies the partitions in implemented run to the specified promote directory. After promoting runs, you can import the paritions into runs.

Select Partitions to promote

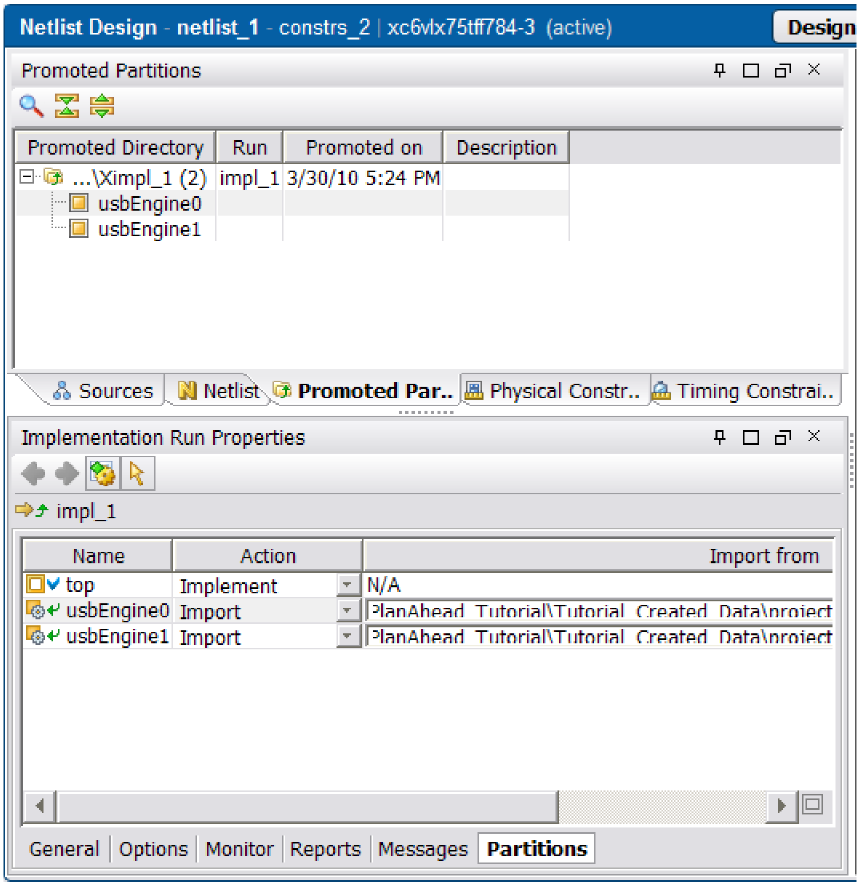
Run	Directory	Description
☑ impl_1	ɔroject_1\project_1.promote\Ximpl_1	
□ top		
☑ usbEngine0		
☑ usbEngine1		

Figure 23: Promote Partitions Dialog Box

3-1-7. Explore the changes caused by promoting partitions (Figure 24):

- In the Netlist Design view, you will now see a Promoted Partitions tab.
- In the Implementation Run Properties window, the Partitions tab now shows the Action on the usbEngine instances as Import.

Note: To access the Implementation Run Properties window, ensure that the Design Runs window is open, and select a run.



Promoted Directory	Run	Promoted on	Description
...\Ximpl_1 (2)	impl_1	3/30/10 5:24 PM	
usbEngine0			
usbEngine1			

Name	Action	Import from
top	Implement	N/A
usbEngine0	Import	PlanAhead_Tutorial\Tutorial_Created_Data\project
usbEngine1	Import	PlanAhead_Tutorial\Tutorial_Created_Data\project

Figure 24: The Promoted Partitions

Update the Top Partition Step 5

PlanAhead monitors remote sources to determine when they have changed. However, if you have copied the sources locally into the PlanAhead project directory, or if a new source exists in another location, you can update a source by removing the old one and adding a new one.

4-1. Update the current top.edf file with a modified version.

4-1-1. Open the Project Manager view by clicking on the Project Manager in the Flow Navigator.

4-1-2. Select `top.edf` in the Sources view.

4-1-3. Right-click `top.edf` and select **Update File** (Figure 25).

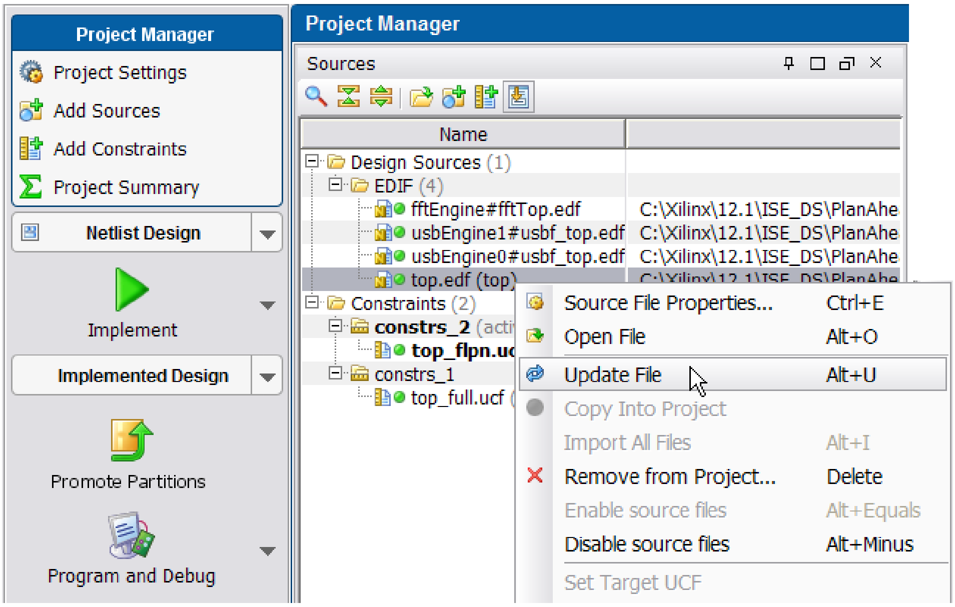
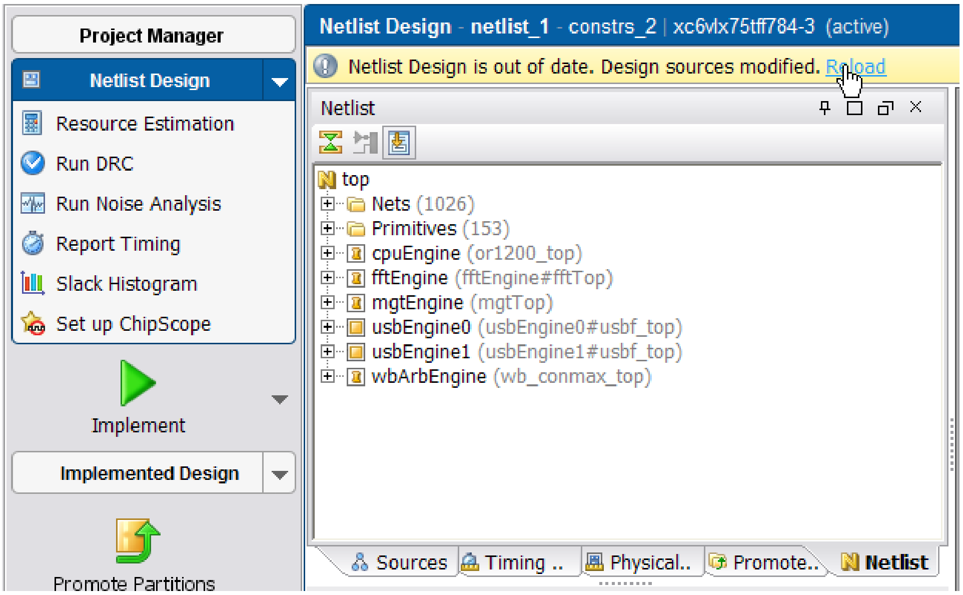


Figure 25: Updating "top.edf"

4-1-4. In the Update File wizard, browse to the modified netlist at `<`*`Extract_Dir`*`>\PlanAhead_Tutorial\Sources\netlist\Iteration_netlist\top.edf`.

4-1-5. Click **Open** to finish updating `top.edf`.

4-1-6. Update the design by going back the Netlist Design view and clicking the **Reload** link in the title bar (Figure 26).



Netlist Design - netlist_1 - constrs_2 | xc6vlx75tff784-3 (active)

Netlist Design is out of date. Design sources modified. Reload

Figure 26: Updating the Design with the Reload Link

Note: If you closed the Netlist Design view when the source file was updated, the new sources are used the next time the view is opened. However, if the Netlist Design view was open when you used the **Update File** command, you must update the design using the **Reload** command.

Re-implement the Top-Level While Importing USB Partitions Step 6

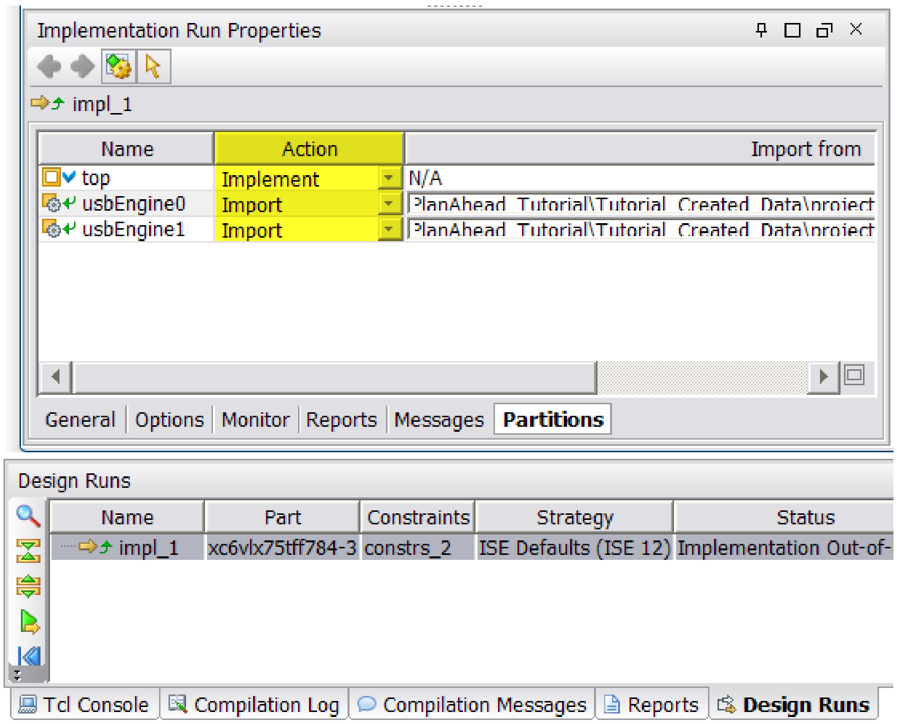
At this point you have implemented the design, promoted the two usbEngine instances, and made a change to the top-level partition. You can now reimplement the modified top-level partition while maintaining an exact copy of the placement and routing results on the two USB cores.

5-1. Verify partition states and re-implement design.

5-1-1. From the Design Runs view (**Window** > **Design Runs**), select **impl_1**.

5-1-2. In the Implementation Run Properties view, select the **Partitions** tab (Figure 27).

5-1-3. Verify that the top-level partition is set to **Implement** and that the two usbEngine partitions are set to **Import**.



impl_1

Name	Action	Import from
top	Implement	N/A
usbEngine0	Import	PlanAhead_Tutorial\Tutorial_Created_Data\project
usbEngine1	Import	PlanAhead_Tutorial\Tutorial_Created_Data\project

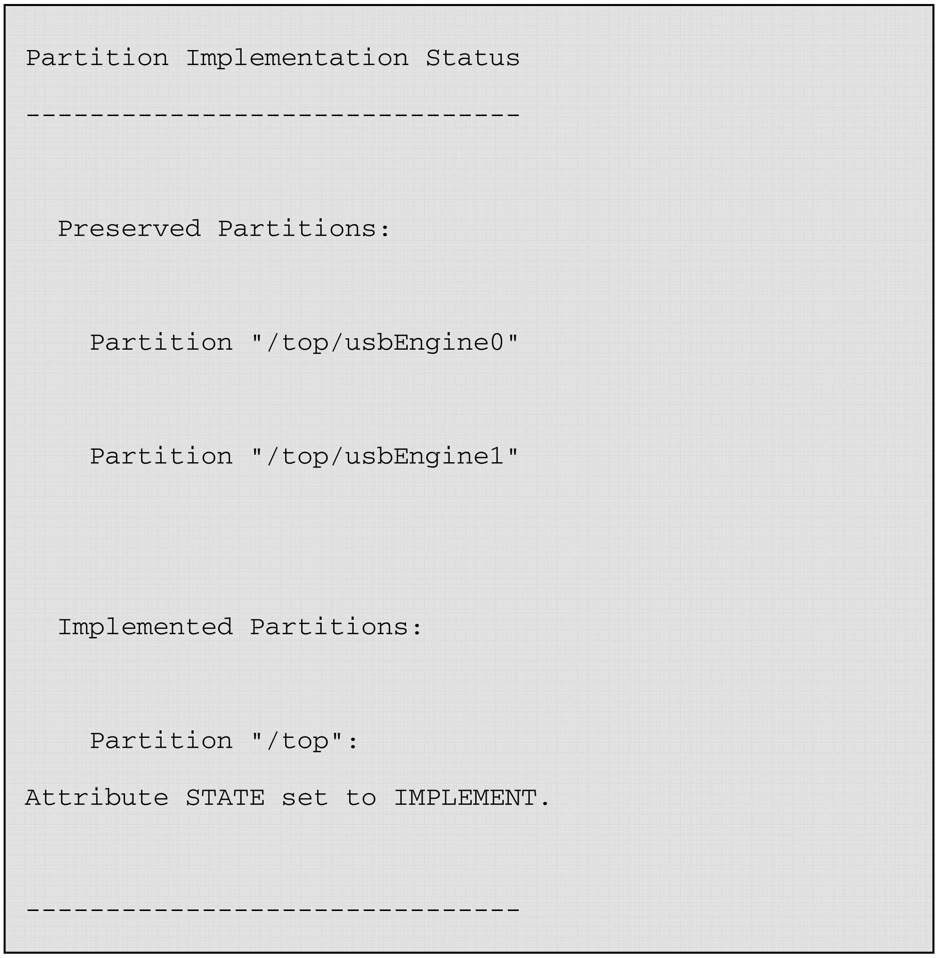
Name	Part	Constraints	Strategy	Status
impl_1	xc6vlx75tff784-3	constrs_2	ISE Defaults (ISE 12)	Implementation Out-of-

Figure 27: Verifying Partition States

5-1-4. In the Flow Navigator, click **Implement** to launch the implementation run. If prompted, click **Yes** to save the project before implementing.

This implementation should go more quickly than the previous implementations because the two large usbEngine cores are being imported. However, it will still take about 30 minutes to complete, depending on the system you are using to run the ISE tools.

5-1-5. Verify that the two usbEngine partitions have been imported and that all timing constraints have been met. Again, if small timing violations existed when the partitions where promoted, then you should expect to see those identical timing violations here. Look at the partition section in the NGDBuild, Map, or PAR reports (Figure 28).



```
Partition Implementation Status
-------------------------------

  Preserved Partitions:

    Partition "/top/usbEngine0"

    Partition "/top/usbEngine1"


  Implemented Partitions:

    Partition "/top":
Attribute STATE set to IMPLEMENT.


-------------------------------
```

Figure 28: Partition Implementation Status in Report Files

Conclusion

In this tutorial, you implemented a flat design with no Area Groups or partitions. You then analyzed the failing paths, added Area Group constraints and partitions to two USB cores, and re-implemented the design. This time the design met timing, so you promoted the successful implementation results to allow for importing in future iterations. The top module was updated requiring implementation to be re-run. However, because the USB cores were not modified, they were imported, and they maintained identical placement and routing results. The output was guaranteed timing results on two large, timing-critical cores for all future iterations of the design (if you made no changes to the usbEngine instances).