Team Design Tutorial

PlanAhead Software

UG839 (v 13.3) October 19, 2011

XILINX®

Xilinx is disclosing this user guide, manual, release note, and/or specification (the “Documentation”) to you solely for use in the development of designs to operate with Xilinx hardware devices. You might not reproduce, distribute, republish, download, display, post, or transmit the Documentation 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. Xilinx expressly disclaims any liability arising out of your use of the Documentation. Xilinx reserves the right, at its sole discretion, to change the Documentation without notice at any time. Xilinx assumes no obligation to correct any errors contained in the Documentation, or to advise you of any corrections or updates. Xilinx expressly disclaims any liability in connection with technical support or assistance that might be provided to you in connection with the Information.

THE DOCUMENTATION IS DISCLOSED TO YOU “AS-IS” WITH NO WARRANTY OF ANY KIND. XILINX MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE DOCUMENTATION, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT OF THIRD-PARTY RIGHTS. IN NO EVENT WILL XILINX BE LIABLE FOR ANY CONSEQUENTIAL, INDIRECT, EXEMPLARY, SPECIAL, OR INCIDENTAL DAMAGES, INCLUDING ANY LOSS OF DATA OR LOST PROFITS, ARISING FROM YOUR USE OF THE DOCUMENTATION.

© Copyright 2011 Xilinx Inc. All Rights Reserved. XILINX, the Xilinx logo, the Brand Window and other designated brands included herein are trademarks of Xilinx, Inc. All other trademarks are the property of their respective owners. The PowerPC name and logo are registered trademarks of IBM Corp., and used under license. All other trademarks are the property of their respective owners.

Table of Contents

Team Design Tutorial

The Team Design flow is a hierarchical design methodology that uses partitions to allow a large, complex design to be broken up into smaller logical blocks.

- These blocks can be implemented independently and simultaneously.
- This allows each block to be individually designed, implemented, and verified in context with the rest of the design.
- When all blocks are complete, the entire design is brought together in an assembly run in which results can be imported and preserved.

A Team Design flow team consists of:

- One team leader
- One or more team members

Tutorial Objectives

This tutorial provides an overview of the Team Design flow. In this tutorial, you will complete a design acting as a team leader and various team members. The objective of this tutorial is to familiarize you with partitions and the Team Design flow using the PlanAhead™ software.

Project Setup

Initially, as team leader you will:

- Define partitions on each team member block.

  Each team member block is a black box.
- Define Pblocks.
- Review existing physical and timing constraints.
- Synthesize an initial version of the design using Xilinx® Synthesis Technology (XST) incremental synthesis.
- Promote synthesis results for the Top partition.
- Run a Design Rules Check (DRC).
- Implement the initial design.
- Create PlanAhead projects for each team member.

Team Member Work

When a PlanAhead project has been created for each team member, you will complete the work of each team member as follows:

- Update the project to include actual design files for the team member block.
- Synthesize the updated team member design while importing the Top partition

  Other team member blocks are black boxes.
- Implement the team member design.

  Other team member blocks are black boxes.
- Promote successful synthesis and implementation results to a location accessible by the team leader.

Design Assembly

The team leader then assembles the design by importing the results from each team member project. You will:

- Assemble the design by importing the results from each team member project.
- Update the team leader project to include NGC files for each team member block.
- Set up implementation to import each team member block while implementing the Top partition.
- Verify design assembles while maintaining placement and routing results from each team member.

Software Requirements

You must have access to the ISE® Design Suite software to perform this tutorial.

Be sure that the PlanAhead software is operational and the sample design data is installed before beginning the tutorial.

The PlanAhead software, Version 13.*x*, supports partitions in RTL projects targeting Spartan®-6, Virtex®-6, and newer families.

For installation instructions and information, see the *ISE Design Suite: Installation and Licensing Guide (UG798)* at http:/ / www.xilinx.com/ support/ documentation/ sw_manuals/ xilinx13_3/ iil.pdf.

Hardware Requirements

Xilinx recommends a minimum of 2 GB of RAM when using the PlanAhead software on larger devices. For this tutorial, a smaller xc6vlx75t design is used, and the number of designs open at one time is limited. Although 1 GB is sufficient, it can impact performance.

Tutorial Design Description

This tutorial uses an incremental synthesis flow to create separate netlist for each partitioned instance. These individual netlists are required for a team design flow.

The design used throughout this tutorial contains:

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

This tutorial uses a small design in order to:

- Allow the tutorial to be run with minimal hardware requirements.
- Enable timely completion.
- Minimize the data size.

Locating Tutorial Design Files

Download the `PlanAhead_Tutorial.zip` file from the Xilinx website:

http://www.xilinx.com/support/documentation/dt_planahead_planahead13-3_tutorials.htm

Extract the zip file contents into any write-accessible location.

The unzipped PlanAhead_Tutorial data directory is referred to in this tutorial as `<Extract_Dir>`.

The tutorial sample design data is modified while performing this tutorial. A new copy of the original PlanAhead_Tutorial data is required each time you run the tutorial.

Step 1: Opening a Project and Elaborating RTL

The PlanAhead software enables you to create several project types depending on where in the design flow the tool is being used. RTL sources can be used to create a project for development and analysis, synthesis, implementation, and bit file creation.

This tutorial uses an existing PlanAhead software project, and focuses on the team design aspects in the software. You will not create a new design using the New Project Wizard.

Opening the PlanAhead RTL Project

To open the PlanAhead RTL project:

1. Open the PlanAhead software.
   - On Windows, select the Xilinx PlanAhead 13 Desktop icon, or select **Start > Programs > Xilinx ISE Design Suite 13 > PlanAhead > PlanAhead**.
   - On Linux, go to the `<Extract_Dir>` directory and type **planAhead**.
2. From the Getting Started page, click **Open Project**.

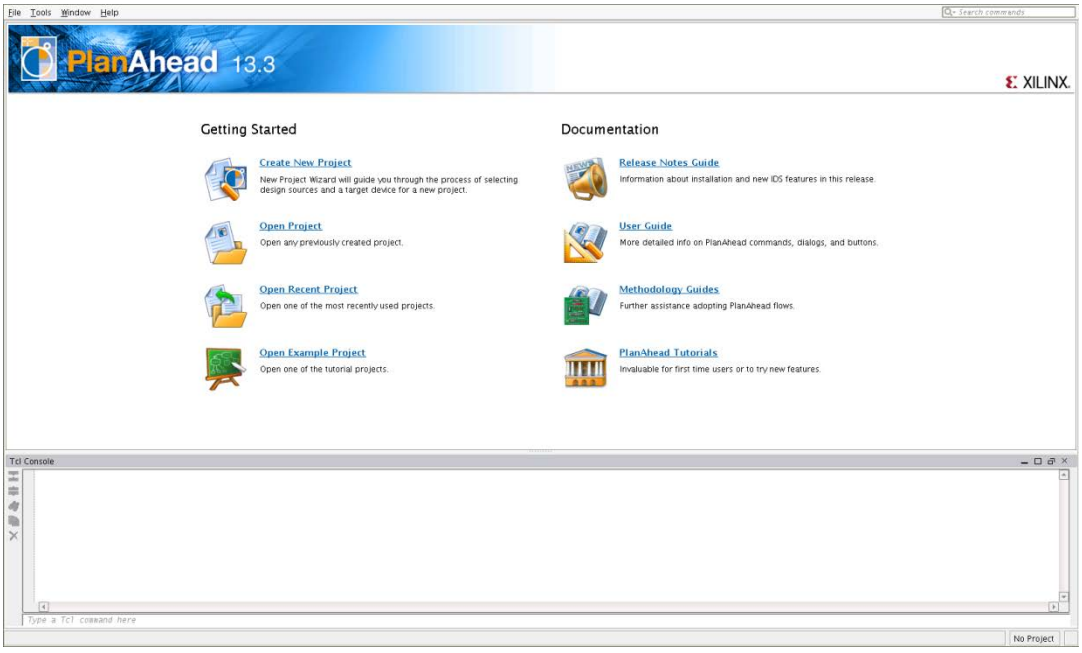

Figure 1: Getting Started Page

3. Browse to `<Extract_Dir>` and open the project file located at:

   `./Projects/project_TL/project_TL.ppr`

The Project Manager view opens. You can view the design source files in the Hierarchy window. These files include:

- VHDL and Verilog files
- A User Constraint File (UCF) named `top_full.ucf`. The UCF contains timing constraints and I/O pin locations.

Elaborating RTL Design

Use the RTL Design view to define partitions in an RTL project. When the RTL Design view is opened:

- The RTL code is elaborated.
- The design hierarchy is displayed.

This is a pre-synthesized view of the design that is used to define partitions and create constraints.

To elaborate the RTL design:

1. In the Flow Navigator, click **RTL Design**.

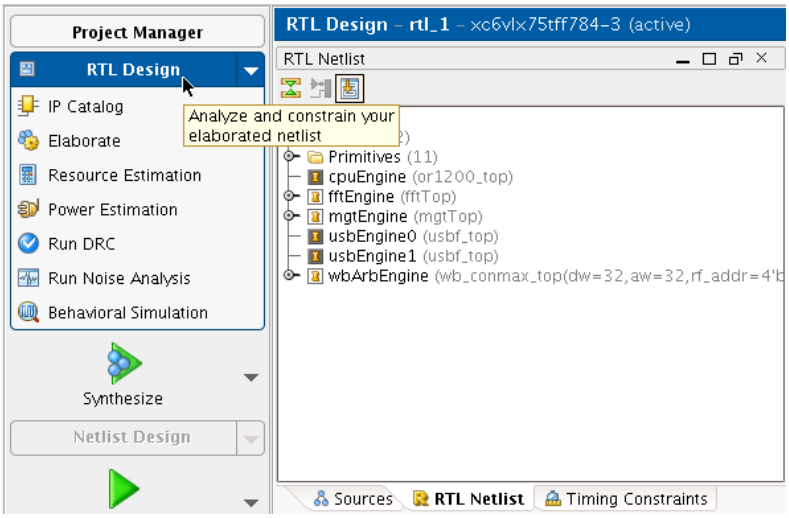

Figure 2: Opening the RTL Design View

   You might see messages warning against undefined instances, and other Critical Warnings. These messages occur because some modules are defined as black boxes. These messages are expected. Ignore them.

2. In the Undefined Modules message windows, click **OK**.

   The Undefined Modules message is expected. Ignore it. This initial version of the design has black box module definitions for two modules. These modules are assigned to each team member later in this tutorial. A definition (`or1200_bb.v` and `usbf_bb.v`) already exists for each module to describe the port direction and port widths.

3. In the Constraint File Critical Warnings message box, click **OK**.

   The Constraint File Critical Warnings are expected. Ignore them. When the RTL Design view opens, it parses the UCF. Because the RTL design view is a pre-synthesized version of the design, there are constraints in the UCF that do not yet exist in the elaborated design, such as I/O buffers.

Step 2: Setting Partitions and Drawing Pblocks

The **usbEngine** instances have already been identified as timing-critical modules. It is advantageous to preserve the successful implementation results of these instances. However, this advantage alone does not make the **usbEngine** instances good candidates for partitions.

The **usbEngine** instances are good candidates for partitions because:

- They are logically isolated from the rest of the design, and
- They have reasonable interface timing by way of registered inputs and outputs.

You can use the PlanAhead design rule checks to help identify modules that are good candidates for partitions.

You can floorplan partitioned instances just like any other instance. Creating Pblock (AREA_GROUP) constraints can help achieve timing closure and improve runtime. The UCF provided with this tutorial constrains the **usbEngine** I/O logic along the left side of the device.

The steps below show you how to set partitions and create appropriate Pblock constraints for the team member instances. Pblocks are required in a Team Design flow for each team member partition.

Setting Partitions on Team Member Instances

To set partitions on team member instances:

1. From the RTL Design view, click the **RTL Netlist** tab.
2. Press **Ctrl** and select:
   - **usbEngine0**
   - **usbEngine1**
   - **cpuEngine**
3. Right-click.
4. Select **Set Partition**.

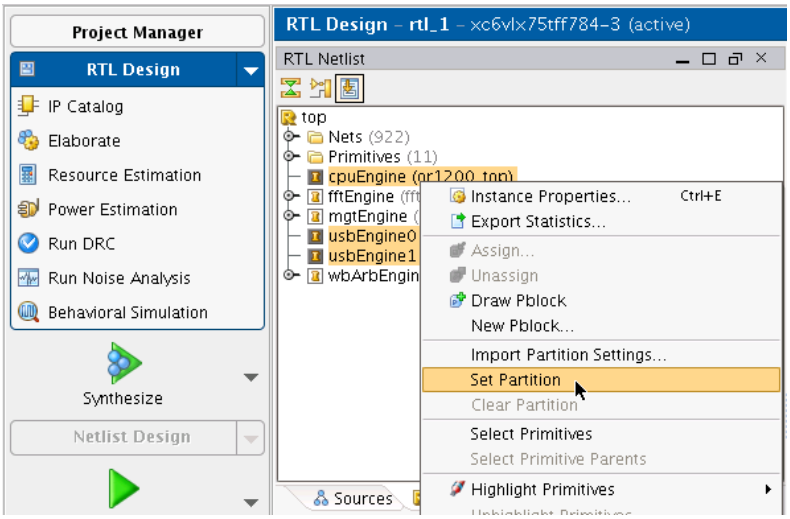

Figure 3: Setting Partitions on usbEngine and cpuEngine Instances

Although three partitions are defined here, there are only two team members in addition to the team leader. Both **usbEngine** instances are managed by a single team member because there is only one set of Hardware Description Language (HDL) code associated with this module.

Drawing Pblocks for the Team Member Partitions

In this step, you:

- Define Pblocks.
- Review existing physical and timing constraints.

This step does not affect synthesis results. You could perform it post-synthesis in the Netlist Design view instead.

To draw Pblocks for the team member partitions:

1. Change the layout to be Floorplanning by selecting **Layout > Floorplanning.**
2. From the RTL Netlist window, select **usbEngine1**.
3. Right-click and select **Draw Pblock**.

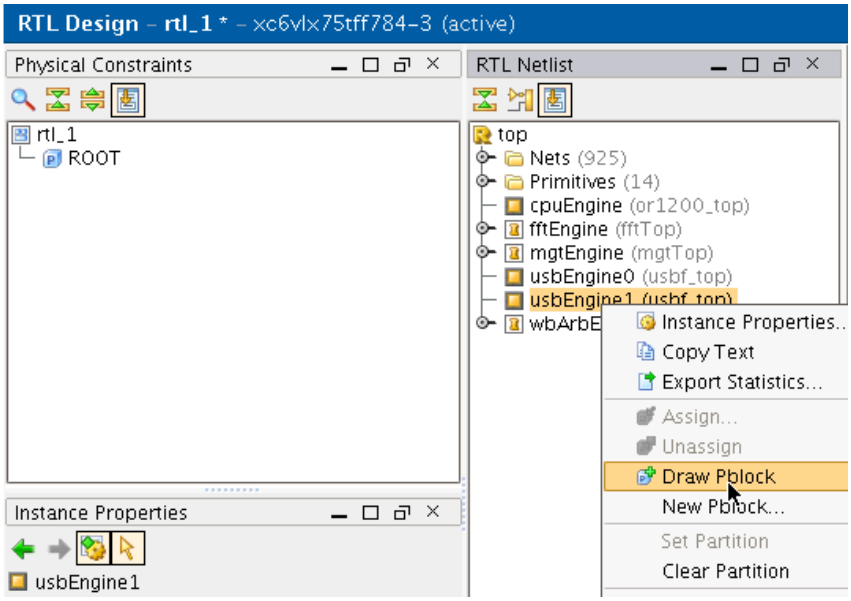

Figure 4: Selecting the Draw Pblock Tool

4. With the Draw Pblock tool active, move the cursor to the Device window.
5. Click the top-left corner of the device where the CLBs start (reference Figure 6 for completed view).

6. Drag down and to the right to create a rectangle covering most of the top-left quadrant of the device.

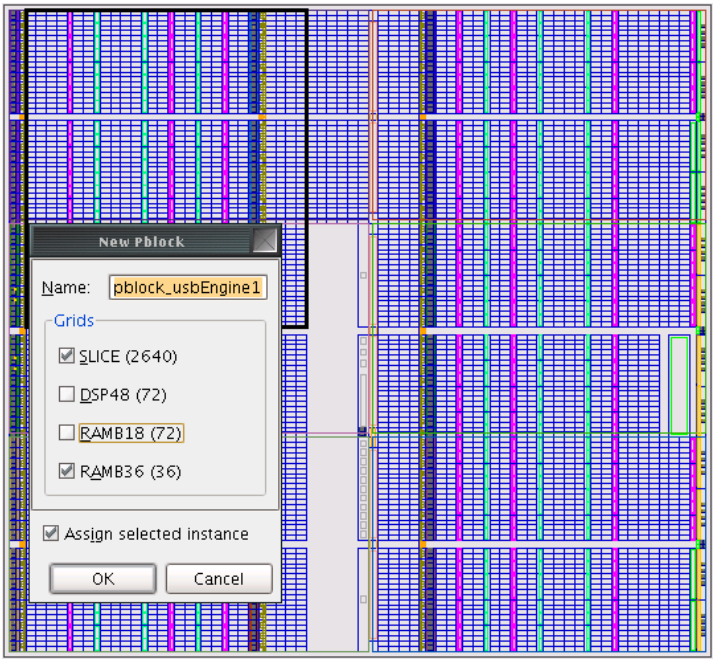

Figure 5: Pblock Rectangle for usbEngine1

7. In the New Pblock dialog box:
   - Verify that the SLICE and RAMB36 grids are selected.
   - Deselect other resources that are not needed.
   - Verify that the number of available RAMB36 is 36.

   If the rectangle does not fully cover the region shown in the figure above (*Pblock Rectangle for usbEngine1*), this number could be less than the required 36, in which case the design fails to place.
8. Click **OK**.
9. If the number of available RAMB36 resources is less than 36, adjust the size of the Pblock rectangle by selecting it and resizing.
10. Repeat the steps above for **usbEngine0** in the bottom-left quadrant.

    Only SLICE and RAMB36 (36 of them) are required for this block.
11. Repeat the steps above for **cpuEngine** in the top-right quadrant.

    For **cpuEngine**, the Pblock contains all four ranges:
    - SLICE
    - DSP48
    - RAMB18
    - RAMB36

The number of resources required for the **cpuEngine** is not as critical. The Pblock must roughly cover the top right quadrant of the die.

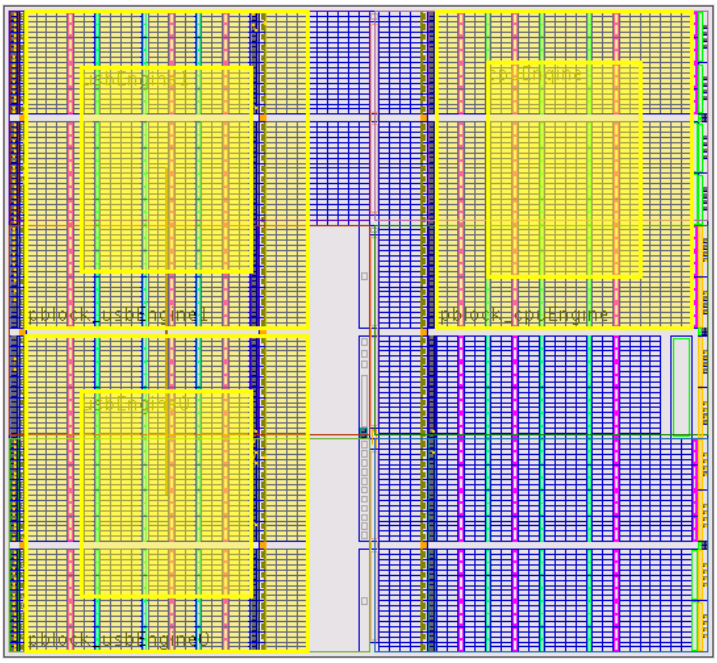

Figure 6: Completed Floorplan for Pblocks usbEngine0, usbEngine1, and cpuEngine

Step 3: Synthesizing and Implementing the Design

In the previous steps, you:

- Defined partitions on the elaborated HDL design.
- Created constraints.

In this step, you will:

- Run synthesis.
- Run DRC.
- Run implementation.

During synthesis, XST recognizes that the design has partitions, and runs the design through an incremental flow.

Normally, XST generates individual NGC files for each partition when using the incremental flow. However, because all team member partitions are black boxes and contain no logic, only the top-level NGC file is created during synthesis.

This tutorial uses an RTL project. For a bottom-up or third-party incremental synthesis flow:

- Synthesis is run outside the PlanAhead software.
- A netlist project is used in the PlanAhead software instead of an RTL project.

Running Synthesis

This design uses default synthesis options.

To run synthesis:

1. In the Flow Navigator, click **Synthesize**.

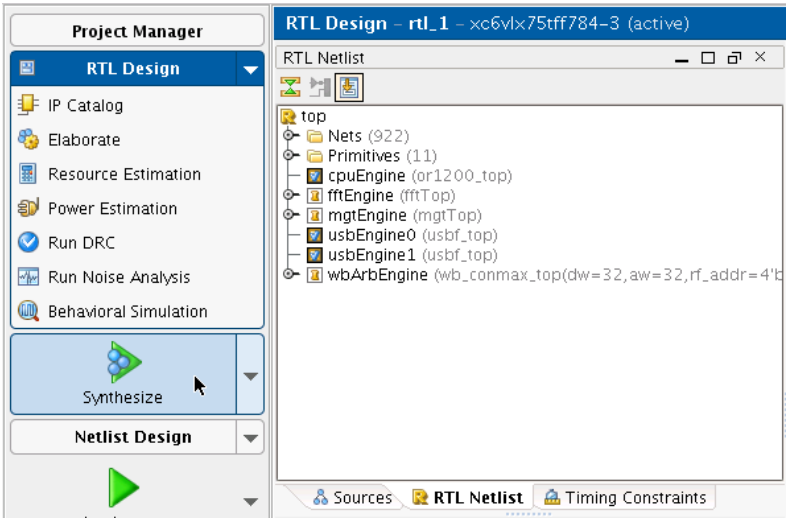

Figure 7: Launch XST Incremental Synthesis

2. Before running synthesis, PlanAhead saves all partition and floorplan changes. If prompted to save your design, click **Save**.
3. After synthesis completes a Synthesis Completed dialog box will appear. You can either choose to Open Netlist Design, or Cancel the dialog and open the Netlist Design in the next step.

Running DRC on Partitions

You can run a Design Rule Check (DRC) on the RTL design, although the checks at this stage are limited. Xilinx recommends that you run a DRC on the post-synthesis design before implementation.

To check DRC, load the Netlist Design view, and run partition-specific DRCs.

1. If not already open, open the Netlist Design view by clicking **Netlist Design** to in the Flow Navigator.

   This loads the synthesis results and allows for additional DRC checking.

2. In the Flow Navigator under Netlist Design, click **Run DRC**.

3. From the Run DRC dialog box, deselect all rules except **Partition** and **Team Design**.

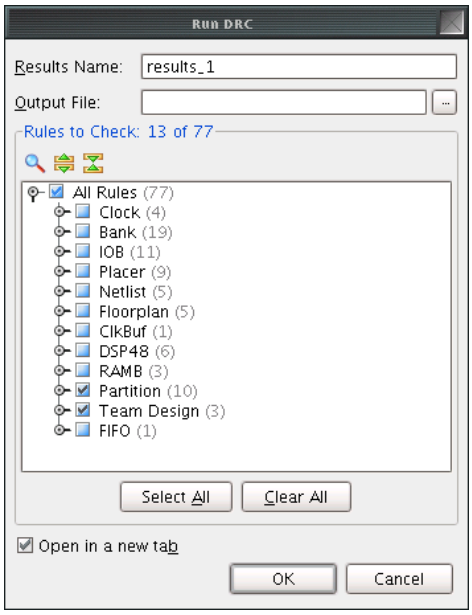

Figure 8: Run Partition Design Rule Checks

4. Click **OK**.

DRC Messages

The PlanAhead software can report the following message types for the DRC rules:

- Advisory
- Warning
- Error
- Fatal

In this case, the DRC returns several warning messages, along with other minor messages. Ignore them. The DRC messages are caused by the black boxes in the design.

In an actual design, investigate all DRC messages, and correct any serious issues.

Running Implementation

Before transmitting the design to each team member, the team leader now runs implementation to check that all timing and floorplan constraints are valid. Because the two **usbEngine** instances and the **cpuEngine** instance are black boxes, no real useful implementation results are obtained.

To run implementation:

1. In the Flow Navigator, click **Implement**.
2. Click the **Report** tab to see a list of all implementation reports.
3. As each process finishes, the related reports are available for review.
4. After NGDBuild finishes, double-click the **NGDBuild Report** to open the report.

5. Scroll to the bottom of the report file to review the partition information.

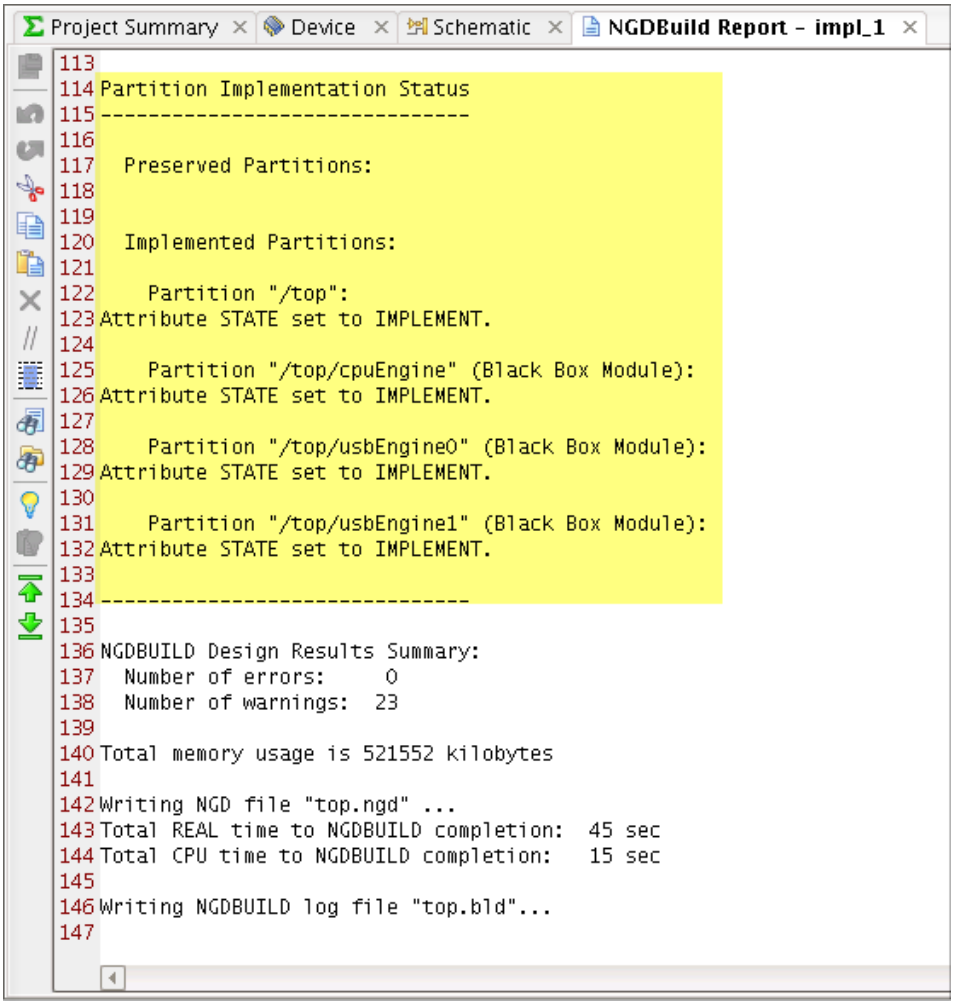

Figure 9: Partition Implementation Status in Report Files

The three team member partitions are all listed as Black Box Module. Partition information is provided in every report file (NGDBuild, Map, and PAR). This allows you to easily verify the status of all partitions on a given run.

Step 4: Promoting Synthesis and Creating Team Member Projects

The design is now ready for the team members to start their work.

- Before generating PlanAhead projects for each team member, promote the initial *synthesis* results.
- Because the *implementation* results of the previous step are not used by team members, the *implementation* results need not be promoted.

Promoting Successful Synthesis Results

To promote the successful synthesis results after synthesis (or implementation) has completed:

1. Click **Promote Partition i**n the Flow Navigator.
2. In the Promote Partitions dialog box, select only the Top partition to be promoted in the **synth_1** run.

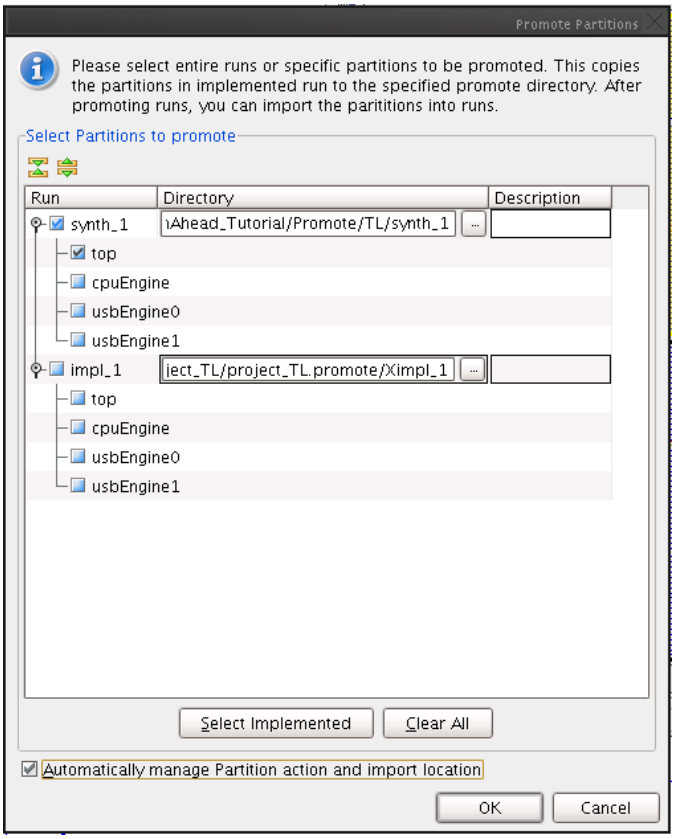

Figure 10: Promote Partitions Dialog Box

Do not to promote other partitions at this stage. No useful data exists for the other partitions. Promoting them will require additional steps by each team member to change the promoted partition state values back to implement for their runs.

3. Set the directory for **synth_1** to:

   `<Extract_Dir>/Promote/TL/synth_1`

4. Enter an optional description about the promoted data.
5. Verify that **Automatically Manage Partition Action and Import Location** is checked.

   Checking **Automatically Manage Partition Action and Import Location** enables the PlanAhead software to update the partition state and import location for the next synthesis run, whether by the team leader or by a team member. If this box is not checked, the team leader must manage these attributes.

6. Click **OK** to promote the Top partition synthesis results.
7. View the changes caused by promoting partitions. See the figure below (*Promoted Partitions Window*).

8. A Promoted Partitions tab now appears in the RTL Design view.

   If you do not see this tab, select **Window > Promoted Partitions**.

The Specify Partitions dialog box now shows that the Action on the Top partition is Import.

To open the Specify Partitions dialog box, click the **Synthesize** button pull-down menu in the Flow Navigator.

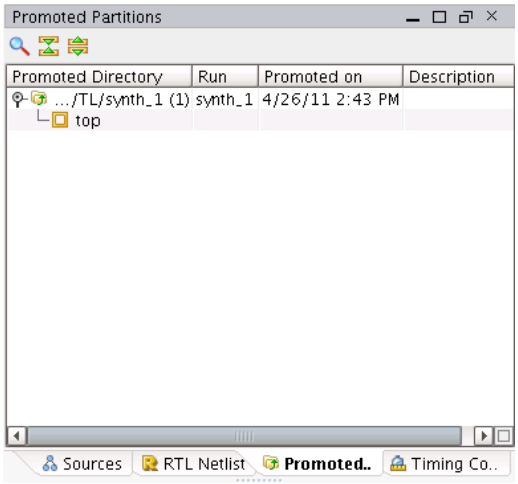

Figure 11: Promoted Partitions

Creating Team Member Projects for usbEngine and cpuEngine

To create a consistent starting point for all team members, create multiple copies of the initial project. For this tutorial, you will create:

- One project for the **cpuEngine** instance
- One project for the two **usbEngine** instances.

  These instances are the same module and are managed by a single team member.

To create team member projects for **usbEngine** and **cpuEngine:**

1. Select **File > Save Project As**.
2. Enter **project_TM_usbEngine** as the Project Name
3. Specify the following as the Project location:

   `<Extract_Dir>/Tutorial_Created_Data`
4. Click **OK**.
5. Repeat the above steps with the following parameters:
   - Project Name

     project_TM_cpuEngine
   - Project location

     `<Extract_Dir>/Tutorial_Created_Data`

Step 5: Working on cpuEngine as Team Member

Acting as the team leader in the previous steps, you have:

- Set up the design.
- Created a workspace for each team member.

You will now switch roles and implement each block as a team member.

While the work of the team members is usually done in parallel, for this tutorial each team member block is handled sequentially.

For each team member project, you will:

- Update the project sources to include the completed logic for the team member block.
- Synthesize the project.
- Implement the project.
- Promote the results.

Adding Sources to cpuEngine

Begin by adding sources to the **cpuEngine** project since it is already open from the last step. Make sure that the current PlanAhead project is **project_TM_cpuEngine**. If it is not, open this project now.

To add sources to **cpuEngine**:

1. Switch to the Project Manager view.
2. In the Sources window, use the Hierarchy tab to expand top and select cpuEngine (`or1200_bb.v`).

   This is the black box module for **cpuEngine** that was used for the initial project. You will now replace this black box module with logic.
3. Right-click, and select **Remove from Project**.
4. Click **OK**.

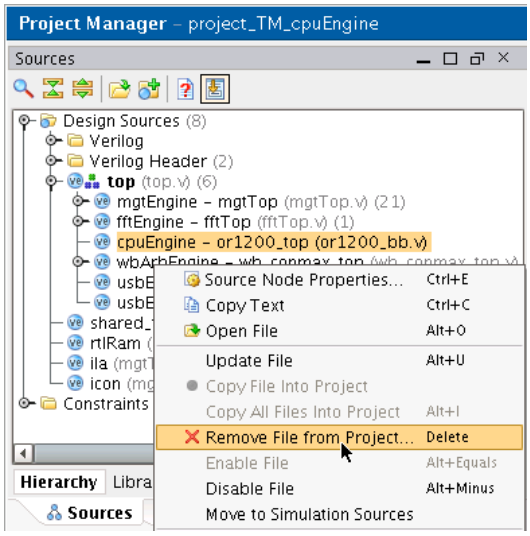

Figure 12: Remove or1200_bb.v Black Box Module

5. At the top of the Sources window, click the **Add Sources** icon ( ).
6. From the Add Sources wizard, select **Add or Create Design Sources**.
7. Click **Next**.
8. Click **Add Directories** to add the completed sources for the **cpuEngine** instance.
9. Browse to the following location:

   `./Sources/hdl/`.
10. Select the `or1200` directory.
11. Click **Select**.

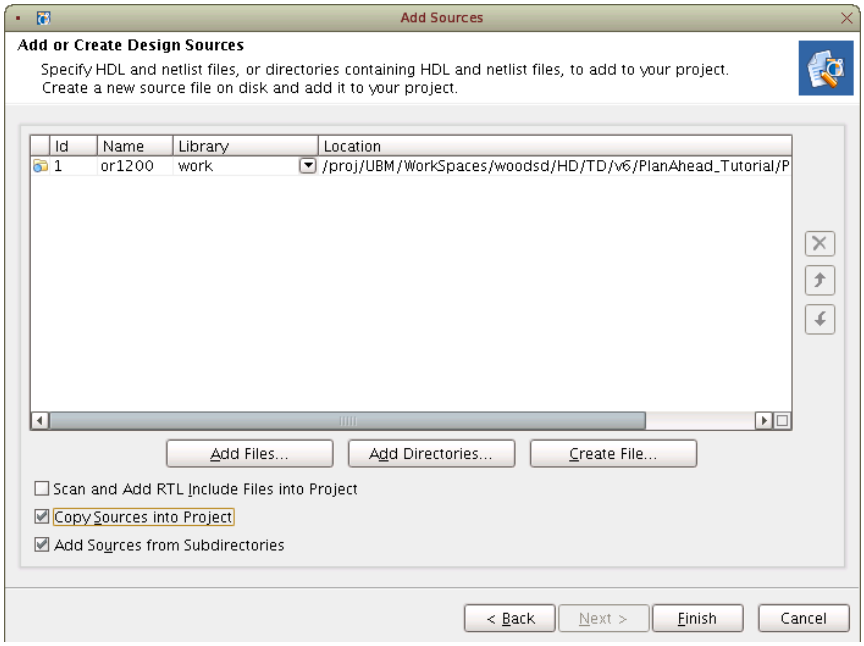

Figure 13: Add Source Directory or1200

12. Verify the directory to be added.
13. Click **Finish**.
14. You should now see the cpuEngine module can be expanded in the Hierarchy sources view.

Synthesizing and Implementing cpuEngine

The RTL Design and Netlist Design views are now out of date. To update them, you will:

- Reload (or close and reopen) these views.
- Verify the partition settings for synthesis and implementation.
- Launch the runs.
- Promote the results.

To reload the design:

1 If the RTL Design view was closed, click **RTL Design** in the Flow Navigator to load the latest version of the design.

2. If the RTL Design view is open, the banner across the top states that the view is out of date. Click **Reload** to reload the view.

Figure 14: Banner with RTL Design Status

3. Verify that **cpuEngine** is no longer a black box.

   When the RTL Design view is reloaded or reopened, a message declares *Undefined Modules Found*. The message now refers only to the **usbEngine** module (**usbf_top**). This module remains a black box.

4. Click **OK**.

5. In the Flow Navigator, click the **Synthesis** or **Implement** button pull-down menu.

6. Select **Specify Partitions**.

   In the Specify Partitions dialog box, verify the partition settings for synthesis and implementation.

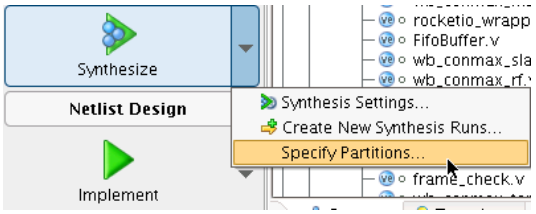

Figure 15: Specify Partitions

7. From the Synthesis tab, verify that:

   a) Top is set to **Import**.

   b) All other partitions are set to **Implement**.

   There is no other data to import besides the top-level synthesis results. Top is the only partition that has been promoted.

   If Top is not set to *import* for synthesis, either of the following might have occurred:

   - Top was not promoted correctly.
   - Automatically Manage Partition Action and Import Location was not checked in the Promote Partition dialog box.

If Top was not correctly promoted, it cannot be imported now.

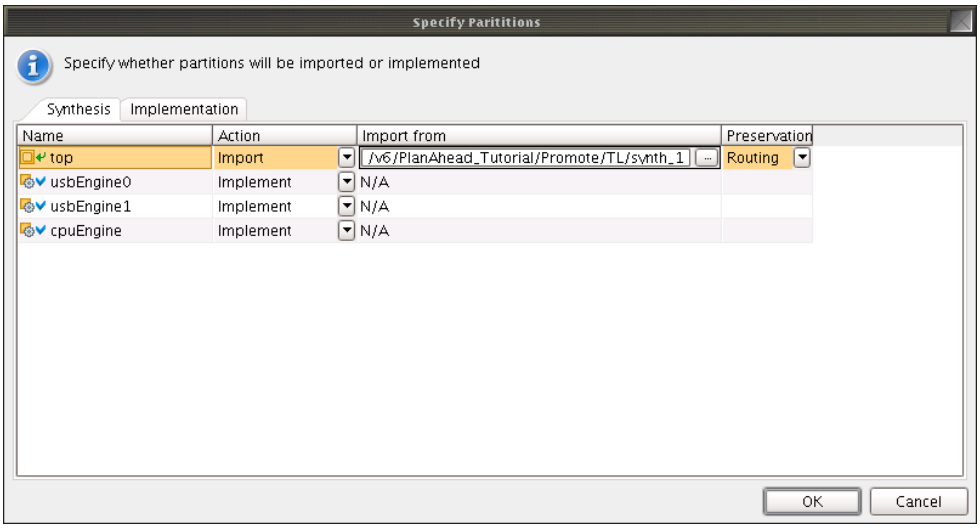

Figure 16: Specify Partitions Dialog Box

8. Click the **Implementation** tab to verify that all partitions are set to **Implement**.
9. Click **OK**.

Optional Steps

You can skip the next two steps to save time. Instead of running synthesis and implementation, open the completed project at:

```
<Extract_Dir>/Projects/project_TM_cpuEngine_completed
```

1. In the Flow Navigator, click **Implement** to synthesize and implement the design.

   A message reports that synthesis is out-of-date.

2. Click **Yes** to launch synthesis prior to implementation.

Verifying and Promoting Results of cpuEngine

If you ran synthesis and implementation, the Implementation Completed dialog box displays when implementation completes.

1. In the Implemented Completed dialog box, select **Open Implemented Design**.
2. Click **OK**.

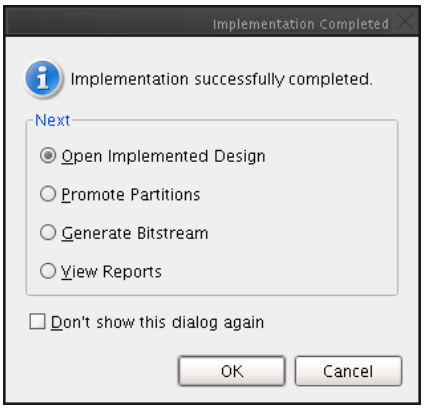

Figure 17: Specify Partitions Dialog Box

If you skipped running implementation and opened the **project_TM_cpuEngine_completed** design, or if the Implemented Completed dialog box did not open:

3. In the Flow Navigator, click **Implemented Design**.

   - The Implemented Design view shows that:
     - The Pblock for **cpuEngine** is relatively full.
     - The Pblocks for the two **usbEngine** instances are relatively empty.
   - Because the **usbEngine** instances were black boxes, no logic exists for these two blocks in the implemented design.
   - The logic belonging to the Top partition is highlighted in yellow in the following figure (*Implemented Design View of the cpuEngine Project*).
     - This logic appears in the **usbEngine** Pblocks because it was not restricted from being placed inside these Pblocks.

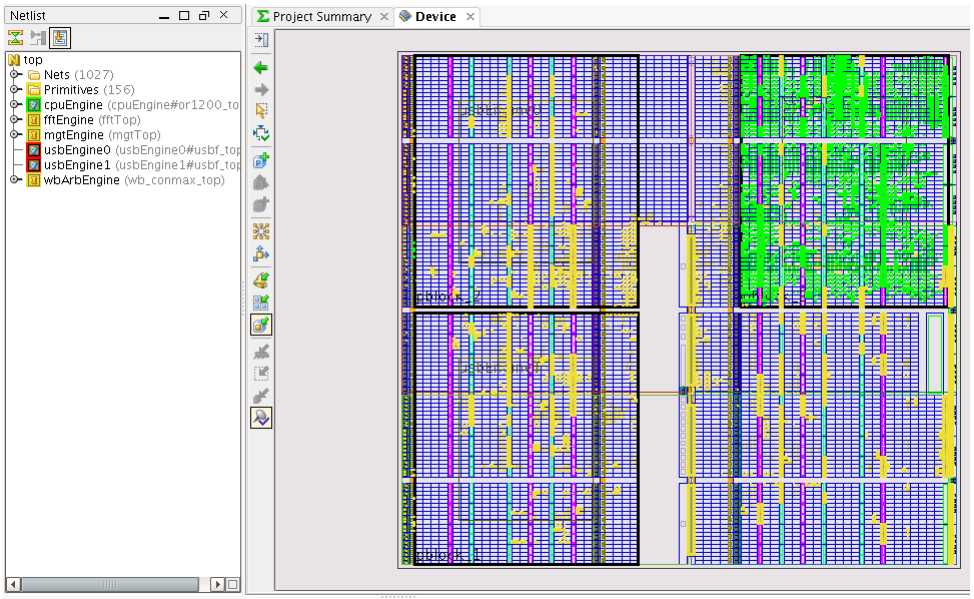

Figure 18: Implemented Design View of the cpuEngine Project

The Highlight Primitives tool can be used to create a highlighted view as shown in the figure above (*Implemented Design View of the cpuEngine Project*).

4. To verify that timing has been met, review any of the following:
   - The Timing Score in the Design Runs view
   - The details of the Timing Results window
   - The Timing Score in the Project Summary window
5. In the Flow Navigator, click **Promote Partition** to promote the successful synthesis and implementation results.

   If the RTL Design view is not opened, you are prompted to open it. Click **OK**.
6. Select only the **cpuEngine** instance in both the synthesis and implementation runs.
7. Set the `Promote` directories.
   - `synth_1: <Extract_Dir>/Promote/cpuEngine/synth_1`
   - `impl_1: <Extract_Dir>/Promote/cpuEngine/impl_1`

   The `Promote` directory used in team design should be a network location or repository that is accessible by all team members. This tutorial changes the default promote location to a folder outside the project directory structure to emulate such a location.
8. Uncheck **Automatically Manage Partition Action and Import Location**.

   While disabling **Automatically Manage Partition Action and Import Location** is not critical to this tutorial, Xilinx recommends removing it for team members in a team design flow.

   The **cpuEngine** team member might continue working on this block. Removing this option prevents the software from automatically importing **cpuEngine** on the next run.

   These promoted results are imported by the team leader in a later step, but are not imported by the team member who is promoting it.

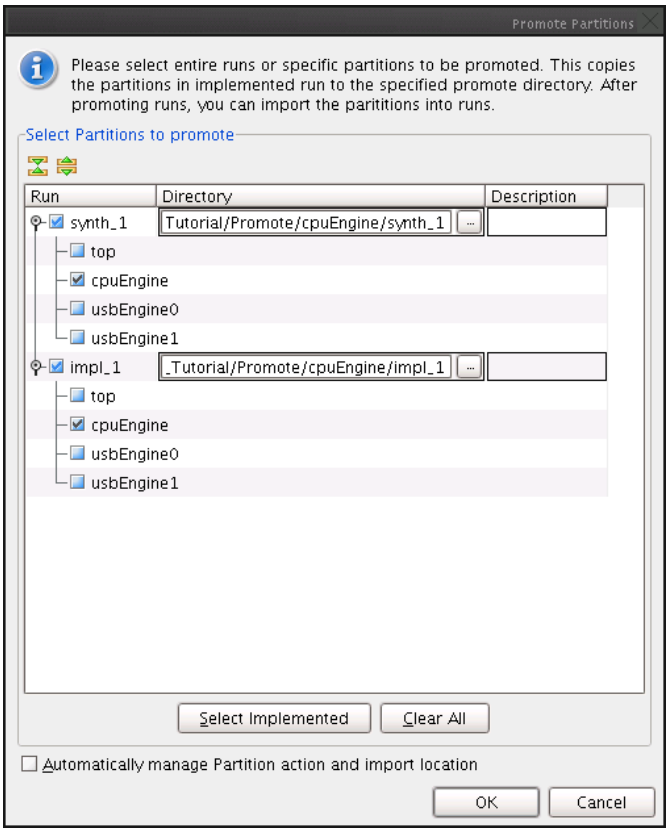

Figure 19: Promote Settings for cpuEngine

9. Click **OK**.

Step 6: Working on usbEngine as Team Member

In this Step, you will repeat the procedures of Step 5 as a team member working on **usbEngine**.

If multiple processors are available, you can perform this step in parallel while the **cpuEngine** project is implementing.

Adding Sources to usbEngine

To add sources to **usbEngine:**

1. Close the current open project unless:
   - You are doing this step in parallel, and
   - The **cpuEngine** has not yet finished implementation, or has not been promoted.
2. Open the PlanAhead project:

   ```
   <Extract_Dir>/Tutorial_Created_Data/project_TM_usbEngine
   ```
3. Click the **Project Manager** view.
4. In the Sources window, use the Hierarchy tab to expand top and select one of the two usbEngine instances (`usbf_bb.v`).

This black box module definition is used for the two **usbEngine** instances in the initial project. It is now replaced with logic.

5. Right-click.
6. Select **Remove from Project**.
7. Click **OK**.

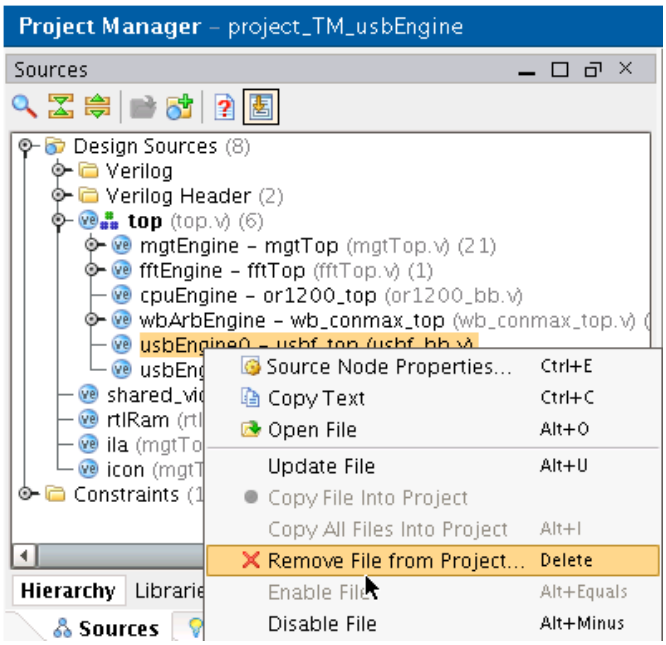

Figure 20: Remove Black Box Module usbf_bb.v

8. At the top of the Sources window, click the **Add Sources** icon ( ).
9. From the Add Sources wizard, select **Add or Create Design Sources**.
10. Click **Next**.
11. Click **Add Directories**.
12. Browse to:

    `/Sources/hdl/`
13. Select the `usbf` directory to add the completed sources for the **usbEngine** instance.
14. Click **Select**.
15. Verify the directory to be added.
16. Click **Finish**.

Synthesizing and Implementing usbEngine

To synthesize and implement **usbEngine:**

1. In the Flow Navigator, click **RTL Design** to open the RTL Design view.
2. Verify that the two **usbEngine** instances are no longer black boxes.

   When the RTL Design view is reloaded or reopened, a message declares *Undefined Modules Found*. The message now refers only to the **cpuEngine** module (**or1200_top**). This module remains a black box.
3. Click **OK**.
4. In the Flow Navigator, click the **Synthesis** or **Implement** button pull-down menu.
5. Select **Specify Partitions**.
6. From the Synthesis tab, verify that:
   - Top is set to **Import**.
   - All other partitions are set to **Implement**.

   There is no other data to import besides the top-level synthesis results. Top is the only partition that has been promoted.

   If Top is not set to *import* for synthesis, either of the following might have occurred:
   - Top was not promoted correctly.
   - **Automatically Manage Partition Action and Import Location** was not checked in the Promote Partition dialog box.

   If Top was not correctly promoted, it cannot be imported now.
7. Click the **Implementation** tab to verify that all partitions are set to **Implement**.
8. Click **OK** to exit the Specify Partitions dialog box.

Optional Steps

You can skip the next two steps to save time. Instead of running synthesis and implementation, open the completed project at:

```
<Extract_Dir>/Projects/project_TM_usbEngine_completed
```

1. In the Flow Navigator, click **Implement** to synthesize and implement the design.

   A message reports that synthesis is out-of-date.
2. Click **Yes** to launch synthesis prior to implementation.

Verifying and Promoting Results of usbEngine

To verify and promote results of **usbEngine**:

1. Open the Implemented Design view.

   If you ran synthesis and implementation, the Implementation Completed dialog box displays when implementation completes.
2. In the Implemented Completed dialog box, select **Open Implemented Design**.

3. Click **OK**.

   If you skipped running implementation and opened the **project_TM_usbEngine_completed** design, or if the Implemented Completed dialog box did not open in the Flow Navigator, click **Implemented Design**.

4. The Implemented Design view shows that:

   - The Pblocks for **usbEngine0** and **usbEngine1** are relatively full.
   - The Pblock for the **cpuEngine** instance is relatively empty.

   Because the **cpuEngine** instance was a black box, it is expected to be empty. However, some logic can be seen in this region. The logic from the top level partition was not restricted from being placed inside the **cpuEngine** Pblock.

   You can use the Highlight Primitives tool to create a highlighted view.

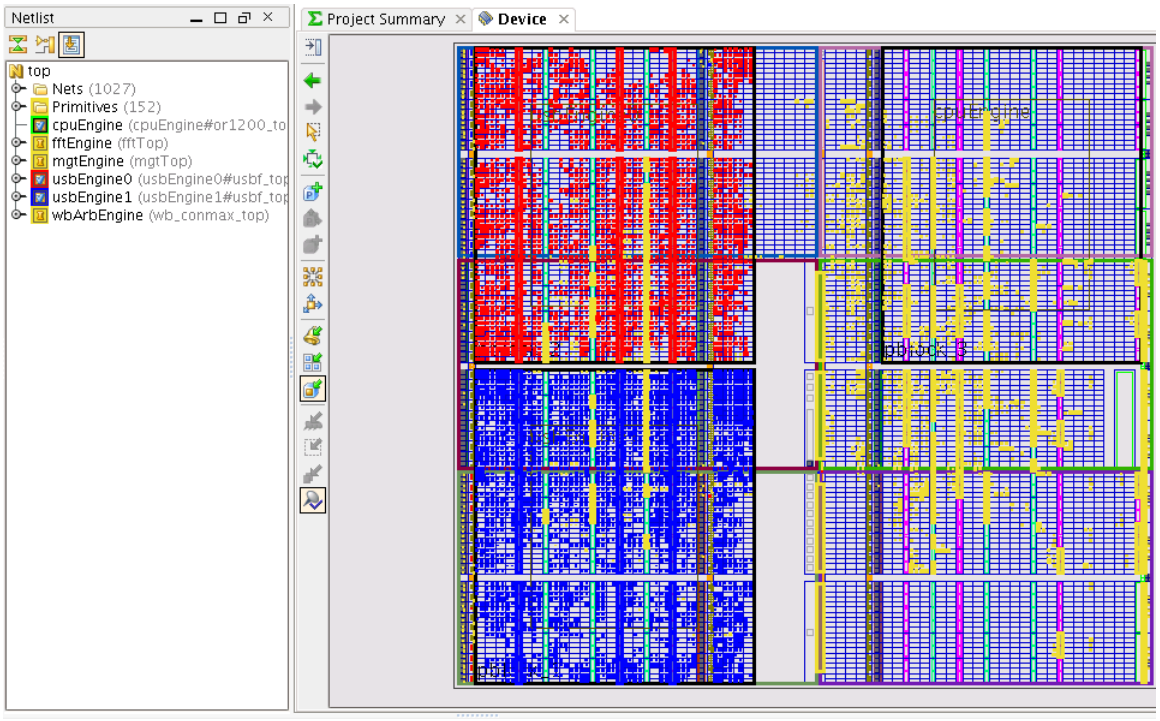

Figure 21: Implemented Design View of the usbEngine Project

5. To verify that timing has been met, review any of the following:

   - The Timing Score in the Design Runs view
   - The details of the Timing Results window
   - The Timing Score in the Project Summary window

6. In the Flow Navigator, click **Promote Partition** to promote the successful synthesis and implementation results.

7. If the RTL Design view is not opened, you are prompted to open it. Click **OK**.

8. Select only the **usbEngine0** and **usbEngine1** instances in both the synthesis and implementation runs.

9. Set the Promote directories.
   - `synth_1: <Extract_Dir>/Promote/usbEngine/synth_1`
   - `impl_1: <Extract_Dir>/Promote/usbEngine/impl_1`
10. Uncheck **Automatically Manage Partition Action and Import Location**.

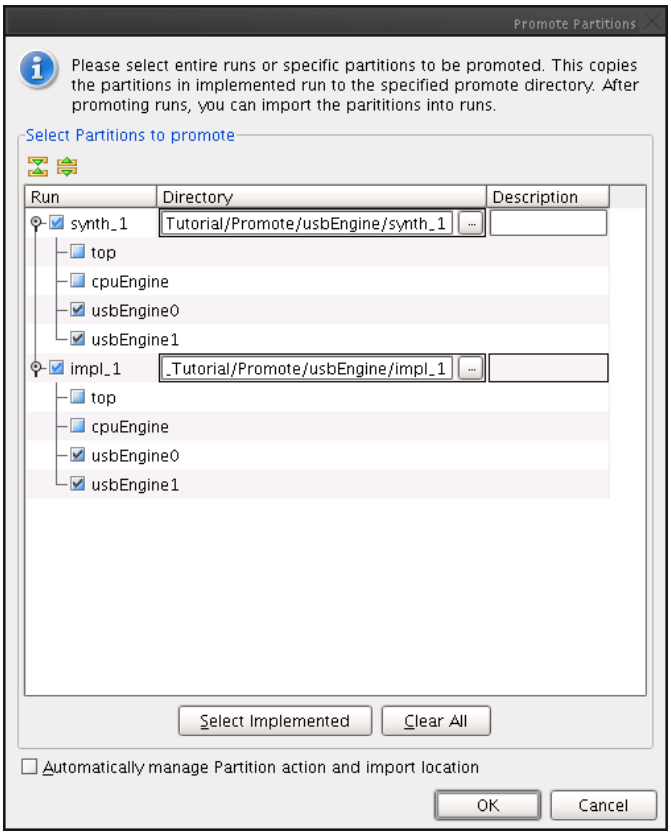

Figure 22: Promote Settings for usbEngine0 and usbEngine1

11. Click **OK** to promote the **usbEngine0** and **usbEngine1** partitions.

Step 7: Running the Assembly as Team Leader

In the previous steps, each team member synthesized, implemented, and promoted a version of their block. The team leader can now assemble the design. The team leader:

- Reopens the team leader project.
- Sets up the partitions for import.

Setting Up the Team Leader Project To Do an Assembly Run

To set up the team leader project to do an assembly run:

1. Reopen the following project:

   `<Extract_Dir>/Projects/project_TL`

   This is the original project. It already contains:

   - The defined team member partitions
   - The successful synthesis results of Top
   - The Pblocks for the team member blocks
2. Update the project to define the team member blocks as logic instead of black boxes. To do this, add the successful synthesis results (NGC) from each team member's promoted synthesis run.
3. From the Project Manager, click **Add Sources**.
4. Select **Add or Create Design Sources**.
5. Click **Next**.
6. Select **Add Files**.
7. Add the following files:
   - `<Extract_Dir>/Promote/usbEngine/synth_1/usbEngine0#usbf_top.ngc`
   - `<Extract_Dir>/Promote/usbEngine/synth_1/usbEngine1#usbf_top.ngc`
   - `<Extract_Dir>/Promote/cpuEngine/synth_1/cpuEngine#or1200_top.ngc`
8. Click **Finish**.

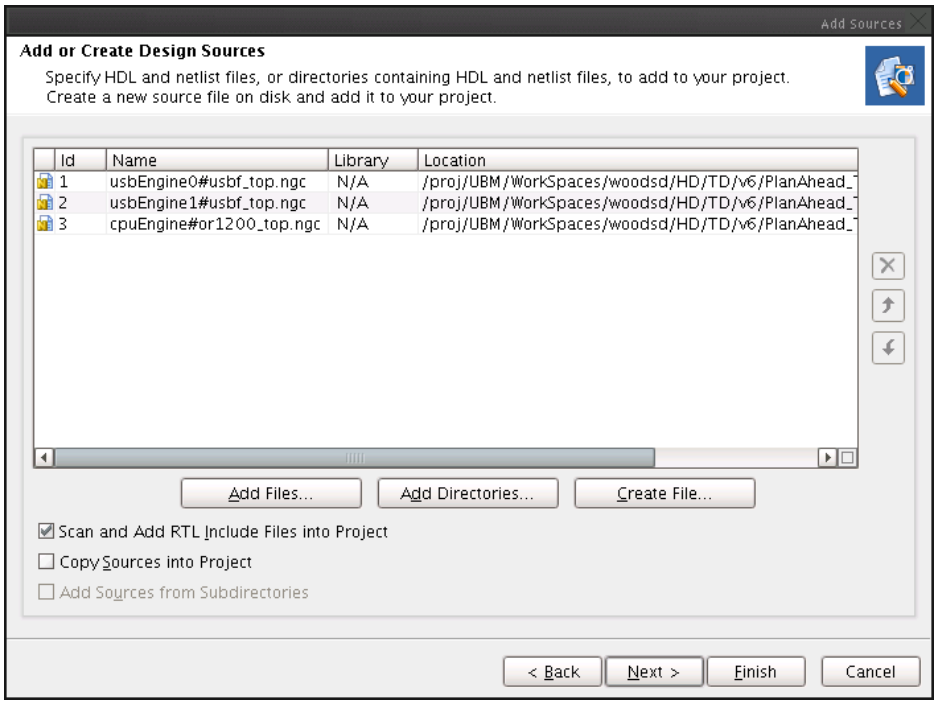

Figure 23: Add Post Synthesis Results for Team Member Partitions

9. In the Flow Navigator, click **Netlist Design** to open the Netlist Design view.

   All partitions are now defined for this project. It is not necessary to rerun synthesis. The latest synthesis results of all team members (including the team leader) are used for this assembly run.

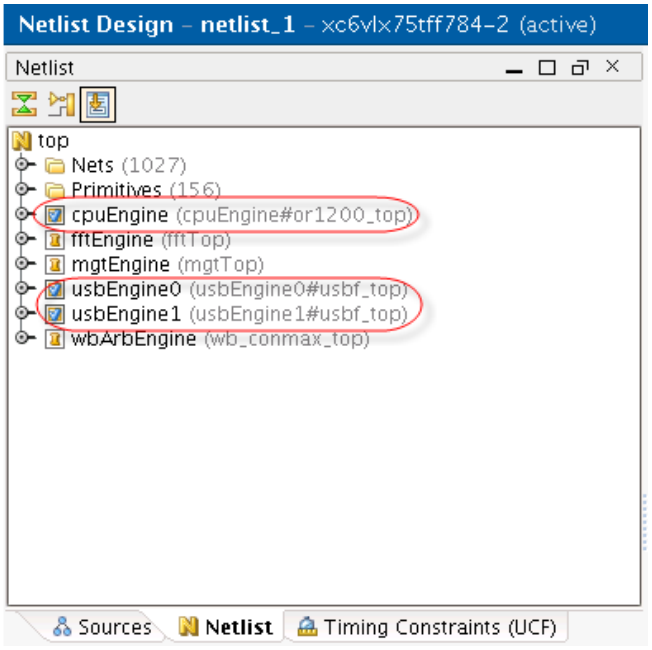

Figure 24: Netlist Design With All Partitions Defined, and No Black Boxes

Updating Partition Settings and Running the Assembly

To update partition settings and run the assembly:

1. In the Flow Navigator, click the **Implement** pull-down menu.
2. Select **Specify Partitions** to access the Implementation Partition settings.
3. Leave the Top partition action as **Implement**.
4. Change the following remaining partition actions to **Import**:
   - `cpuEngine`
   - `usbEngine0`
   - `usbEngine1`
5. Change the **Import from** field to the location to which the team members promoted the implementation results.
   - `cpuEngine: <Extract_Dir>/Promote/cpuEngine/impl_1`
   - `usbEngine0: <Extract_Dir>/Promote/usbEngine/impl_1`
   - `usbEngine1: <Extract_Dir>/Promote/usbEngine/impl_1`

   Because both **usbEngine** instances were implemented together and promoted together, the **Import From** location is also the same.

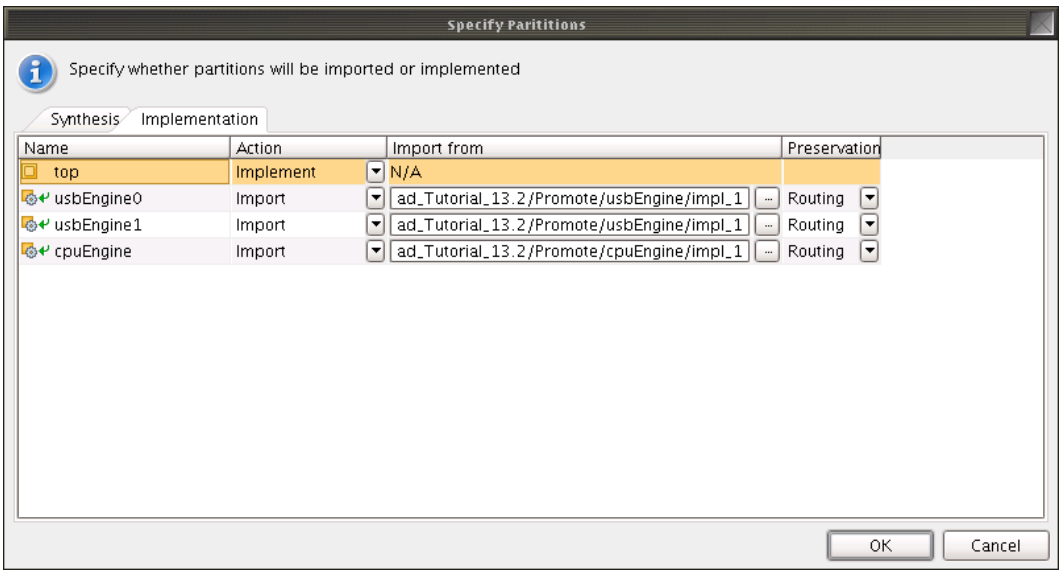

Figure 25: Partition Settings for Assembly Run

6. Click **OK**.
7. Click **OK** on the Reset out-of-date run dialog to reset **impl_1**.

   You do not need to run the next step in which you run implementation. To save time, open the completed project at:

   `<Extract_Dir>/Projects/project_TL_assembled`

7. Optional. Click **Implement** in the Flow Navigator to launch implementation and assemble the full design.

Verifying and Promoting Assembly Results

In this design:

- Each team member block is logically isolated from other team member blocks.
- All the boundaries are registered.
- The placement of each team member block was sufficient the first time through.

For more complex designs, this assembly run may be the first of many. The process described in this tutorial may be repeated as necessary.

Each team member project can now start to import other team member blocks from this promoted assembly run. This helps interface timing to converge early in the design cycle, instead of leaving other team member blocks as black boxes and discovering interface timing issues at the end of the design.

To verify and promote the assembly results:

1. Open the Implemented Design view.
2. In the Implemented Completed dialog box, select **Open Implemented Design**.
3. Click **OK**.

If you skipped running implementation and opened the `project_TL_assembled` design, or if the Implemented Completed dialog box did not open, in the Flow Navigator, click **Implemented Design**.

When the Implemented Design view opens, you can see the entire design. The figure below (*Implemented Design View of the Assembled Design*) shows an example of this placement. The figure uses the highlight colors shown in the following table.

Item	Highlight Colors
usbEngine	red blue
logic belonging to the Top partition	yellow
cpuEngine	green

You can use the Highlight Primitives tool to create this highlighted view.

Figure 26: Implemented Design View of the Assembled Design

Verifying That Timing Has Been Met

To verify that timing has been met, review any of the following:

- The Timing Score in the Design Runs view
- The details of the Timing Results window
- The Timing Score in the Project Summary window

Verifying That All Team Member Partitions Were Imported

To verify that all team member partitions were imported, review the Implemented Partition section of the Project Summary view.

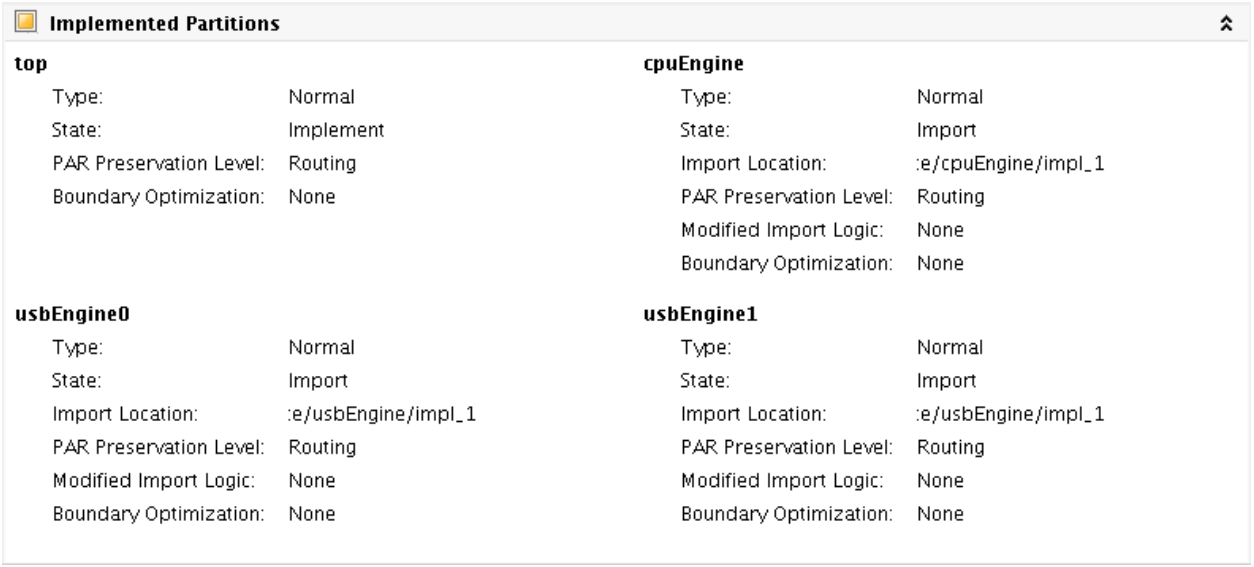

Figure 27: Partition Summary Section of Project Summary

Promoting Successful Synthesis and Implementation Results

Modifications to the design may require some or all of this flow to be repeated. Using these successful assembly results in future iterations may be beneficial. To promote the successful synthesis and implementation results:

1. In the Flow Navigator, click **Promote Partition**.
2. If the RTL Design view is not opened, you are prompted to open it. Click **OK**.
3. Select all partitions in both the synthesis and implementation to be promoted.

   Because the placement of the Top partition meets all interface timing, each team member can now begin to import Top, as well as other team member partitions to maintain this interface timing in all future iterations.
4. Set the `Promote` directories:
   - `synth_1: <Extract_Dir>/Promote/TL/synth_1`
   - `impl_1: <Extract_Dir>/Promote/TL/impl_1`

   For this tutorial, the synthesis results for the team leader have not changed from the last time they were promoted. However, the team leader might develop additional logic in the top level design in parallel with the other team members. In this case, promoting synthesis and implementation results at this stage ensures that all team members are working on the latest code.
5. Uncheck **Automatically Manage Partition Action and Import Location**.

   Disabling **Automatically Manage Partition Action and Import Location** prevents the software from automatically changing the import locations of the team member partitions to the location to which they were just promoted (`Promote/TL`).

Instead, the software maintains the locations defined in the previous steps. This simplifies running the assembly step in future iterations.

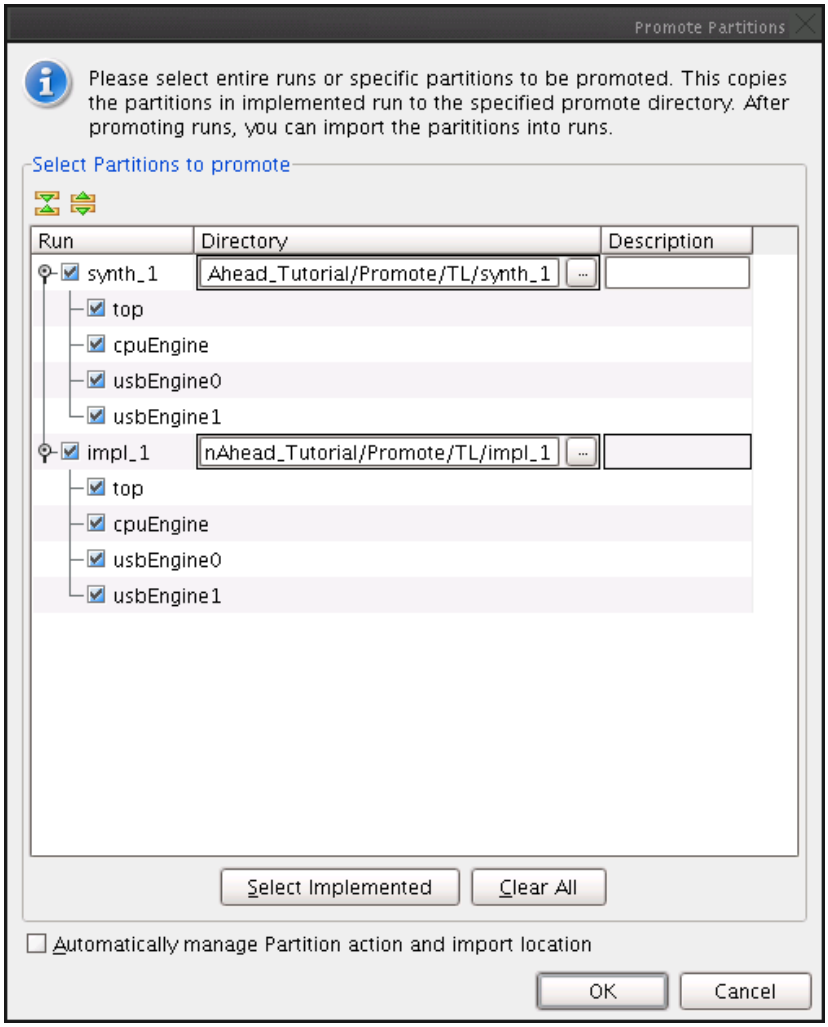

Figure 28: Promote Settings for Assembled Design

6. Click **OK**.

Conclusion

In this tutorial, you did the following:

- Used partitions and the Team Design methodology to synthesize and implement a design in multiple parts.
- Assembled the full design.
- Synthesized and implemented blocks as black boxes.
- Promoted successfully synthesized and implemented partitions to a central repository.
- Created PlanAhead projects for each team member.