PlanAhead Software Tutorial

RTL Design and IP Generation with CORE Generator

UG 675 (v 12.3) September 21, 2010

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

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Free IP Core License

This is the Entire License for all of our Free IP Cores.

Redistribution and use in source, netlist, binary and silicon forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of ASICS World Services, the Authors and/or the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Table of Contents

PlanAhead Software Tutorial

RTL Design and IP Generation with CORE Generator

Introduction

This tutorial provides an overview of the RTL development and analysis environment, in which you will:

- Import RTL sources using the Text Editor
- Run elaboration to compile the RTL
- Use a variety of RTL analysis features to explore your compiled RTL design. These include:
  - Analyzing the RTL logic hierarchy using the RTL schematic
  - Estimating RTL resources using power consumption
  - Running RTL DRCs.
- Go through the steps to browse the Xilinx® IP Catalog and to customize and implement an IP core in the design.

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 product, which is bundled as a part of the Xilinx ISE® Design Suite.

Sample Design Data

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

`<ISE_install_Dir>/PlanAhead/testcases/PlanAhead_Tutorial.zip`

Extract the zip file into any write-accessible location. The location of the unzipped `PlanAhead_Tutorial` data is referred to as the `<Extract_Dir>` throughout this document.

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

Xilinx ISE and PlanAhead Software

By default, the PlanAhead software is installed with the ISE Design Suite. Ensure that the PlanAhead software is operational and the sample design data is installed before beginning the tutorial. For installation instructions and information, see 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 software on larger devices. For this tutorial a smaller design was used and a limited number of designs open at any one time. 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 with your software:

- *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 additional information about PlanAhead, including video demonstrations, go to http://www.xilinx.com/planahead

Tutorial Description

The small sample design used in this tutorial has a set of RTL design sources consisting of Verilog and VHDL. The VHDL sources are from multiple VHDL libraries. The design used throughout this tutorial contains:

- A RISC processor
- A pseudo FFT
- Gigabit transceivers
- Two USB port modules
- An xc6vlx75tff784-1 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 with regards to this tutorial, contact Xilinx Technical Support.

Tutorial Objectives

The objectives of this tutorial are to familiarize you with the RTL development and analysis process using the PlanAhead software.

Tutorial Steps

Step 1 Creating a New RTL Project

Step 2 Using the Sources View and the Text Editor

Step 3 Elaborating and Analyzing the RTL Design

Step 4 Estimating Resource Utilization and Power

Step 5 Run RTL Design Rule Checks (DRCs)

Step 6 Selecting IP from the Xilinx IP Catalog

Step 7 Customizing and Instantiating IP

Step 8 Generating IP

Step 1: Creating a New RTL Project Step 1

PlanAhead software enables several project types to be created 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.

1-1. Open the software.

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

The PlanAhead Getting Started Help page opens.

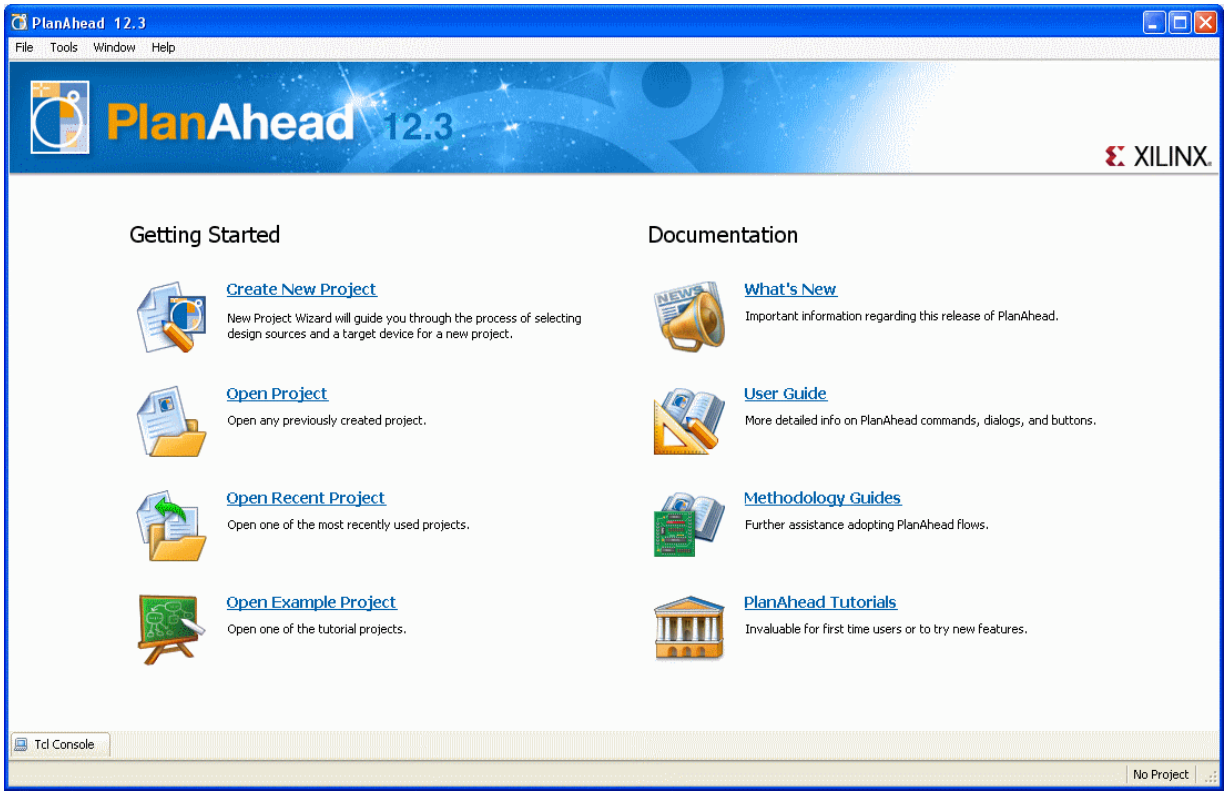

Figure 1: The PlanAhead Getting Started Page

Notice that the *PlanAhead Getting Started* page contains links to open or create projects and view the documentation.

1-2. Create a new RTL project called *project_rtl* using the RTL source files in the *<Extract_Dir>*`\PlanAhead_Tutorial\Sources\hdl` directory.

1-2-1. Select the **Create New Project** link on the Getting Started page.

1-2-2. In the Create a New PlanAhead Project confirmation dialog box, click **Next**.

The Project Name page of the New Project wizard opens.

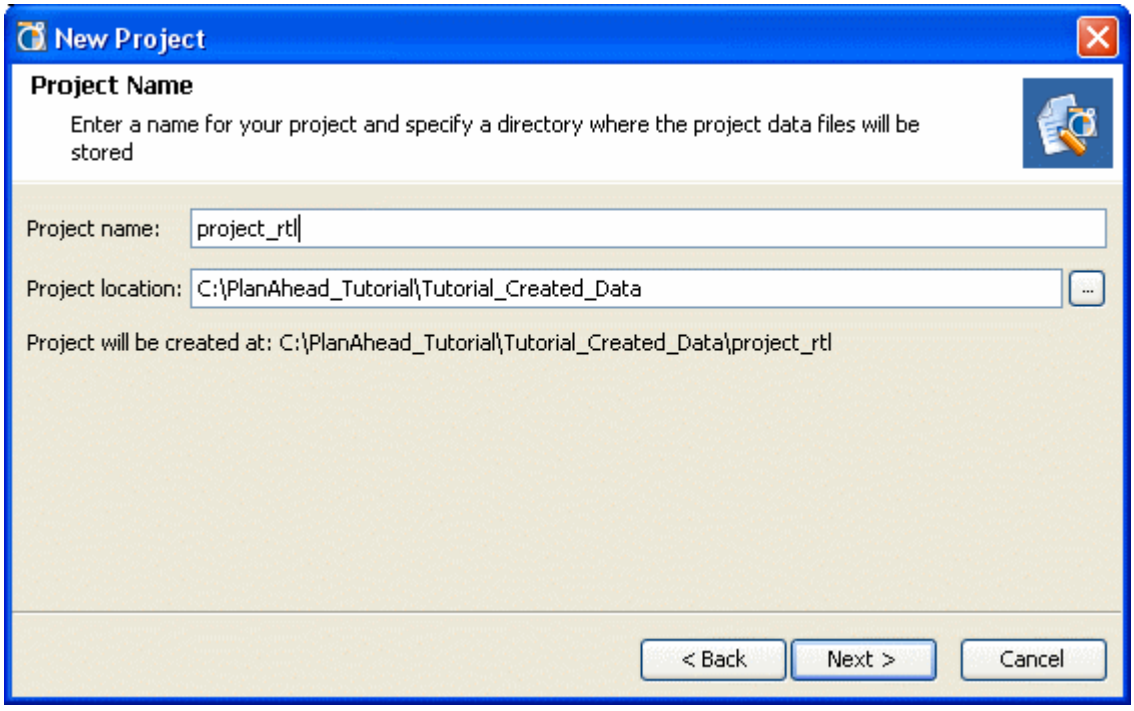

Figure 2: Project Name Page

1-2-3. Browse to and select the following folder: <*Extract_Dir*>**PlanAhead_Tutorial\Tutorial_Created_Data**.

1-2-4. Use the default Project name: **project_rtl**, then click **Next**.

The Design Source page opens.

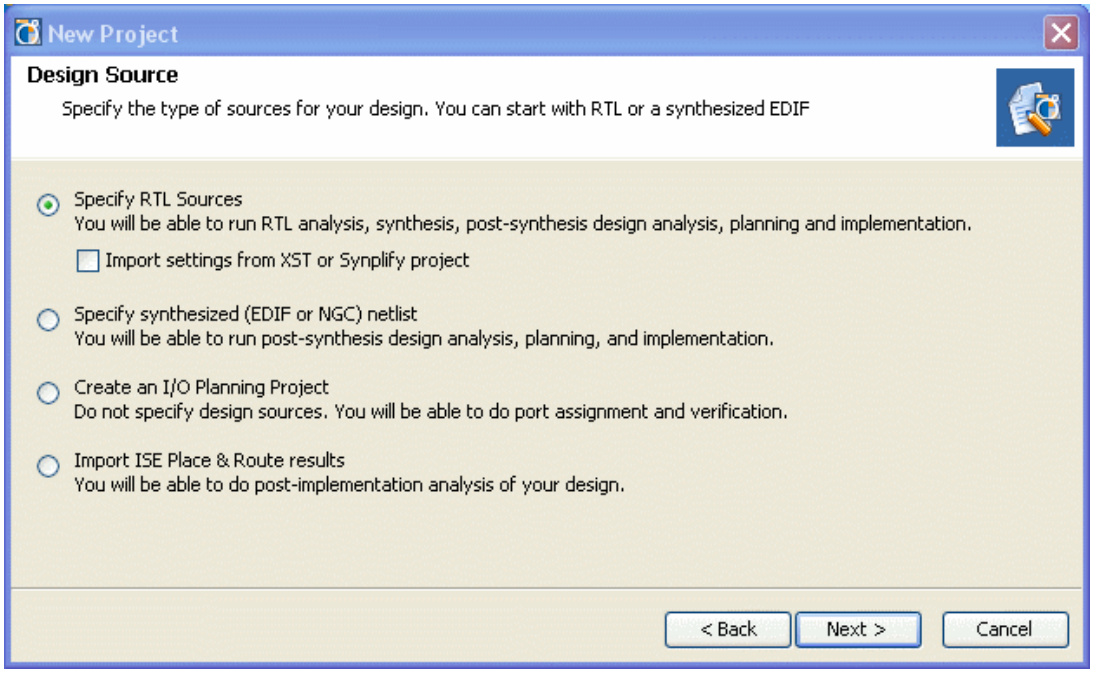

Figure 3: Electing to Import RTL Sources

1-2-5. Click the **Specify RTL Sources** option, and click **Next**.

The Add/Create Sources page opens.

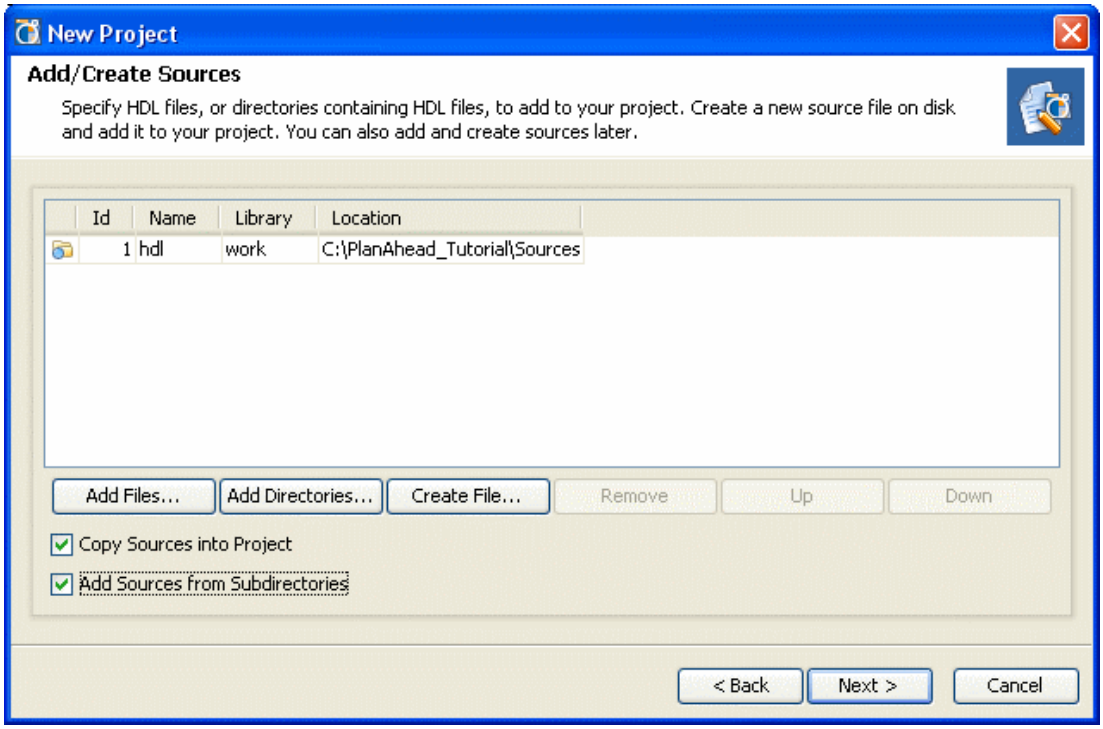

Figure 4: Selecting Sources to Add to the Project

1-3. Add directories and files.

1-3-1. Select the **Add Directories** button and browse to select the following directory: *<Extract_Dir>*/**PlanAhead_Tutorial/Sources/hdl**.

1-3-2. Verify that the Copy Sources into Project and Add Sources from Subdirectories options are set to **on**.

1-3-3. Adjust the page so that it is identical to Figure 4, and click **Next**.

The Add/Create Constraints Files page opens.

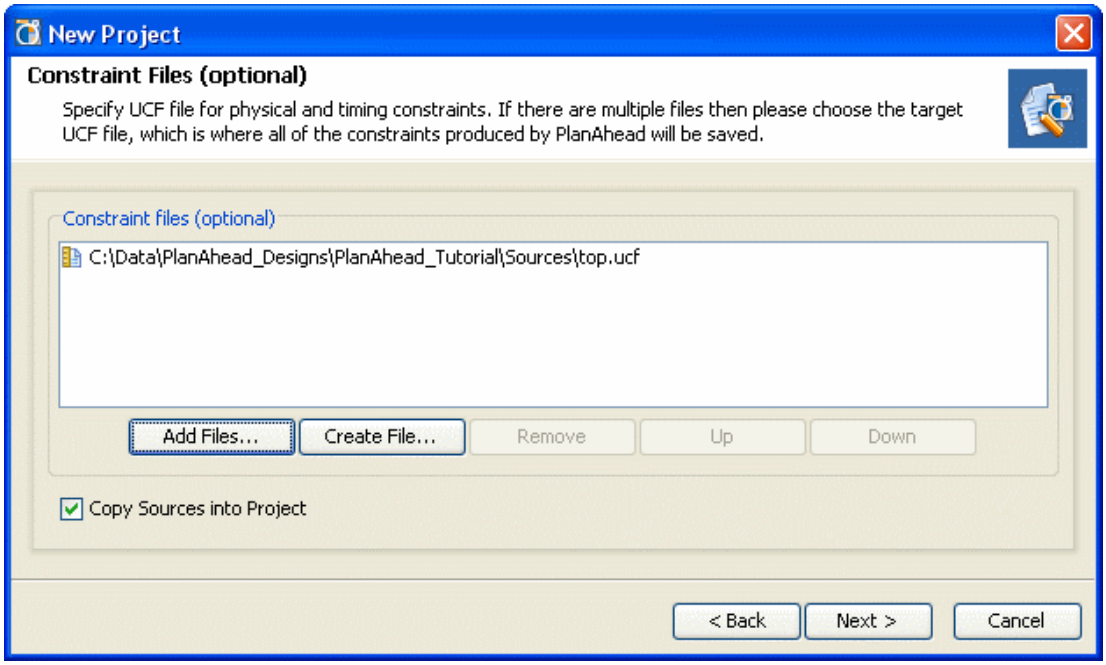

Figure 5: Selecting Constraint Files to Add to the Project

1-4. Add a constraints file.

1-4-1. Select the **Add Files** button and browse to select the following file: `<Extract_Dir>/PlanAhead_Tutorial/Sources/top.ucf`.

1-4-2. Click **Open**.

1-4-3. Click **Next**.

The Default Part page opens.

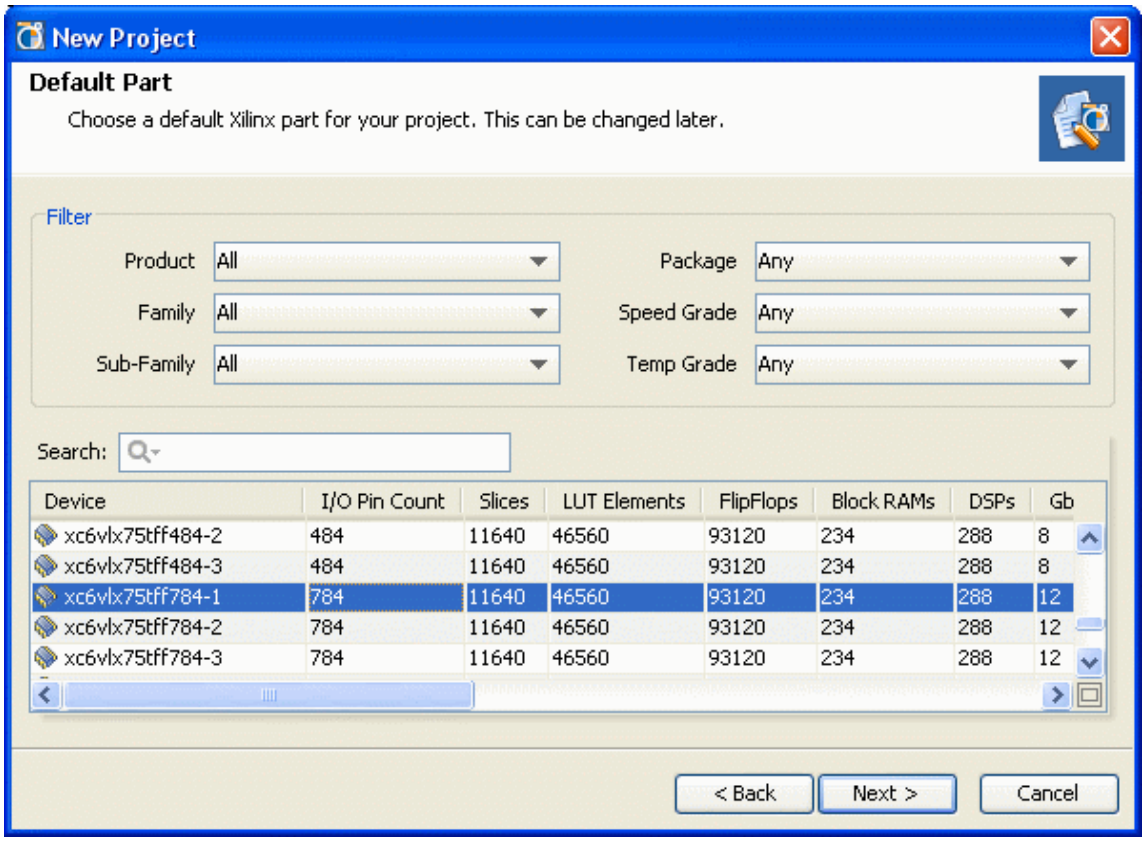

Figure 6: Selecting a Family and Default Part

1-5. Select a default part.

1-5-1. Select the **xc6vlx75tff784-1** device, and click **Next**.

1-5-2. Review the New Project Summary page, and click **Finish**.

The PlanAhead Environment opens.

Step 2: Using the Sources View and the Text Editor Step 2

The PlanAhead software allows different file types to be added as design sources, including Verilog, VHDL, and NGC format cores. The files display by category in the Sources view. A Text Editor is supplied to create or modify RTL sources.

2-1. Explore the Sources view and Project Summary.

2-1-1. Examine the information in the Project Summary. More information will be displayed as the design progresses.

2-1-2. Examine the Sources view.

2-1-3. Collapse the Verilog Folder by clicking on the minus sign (-) next to it (Figure 7).

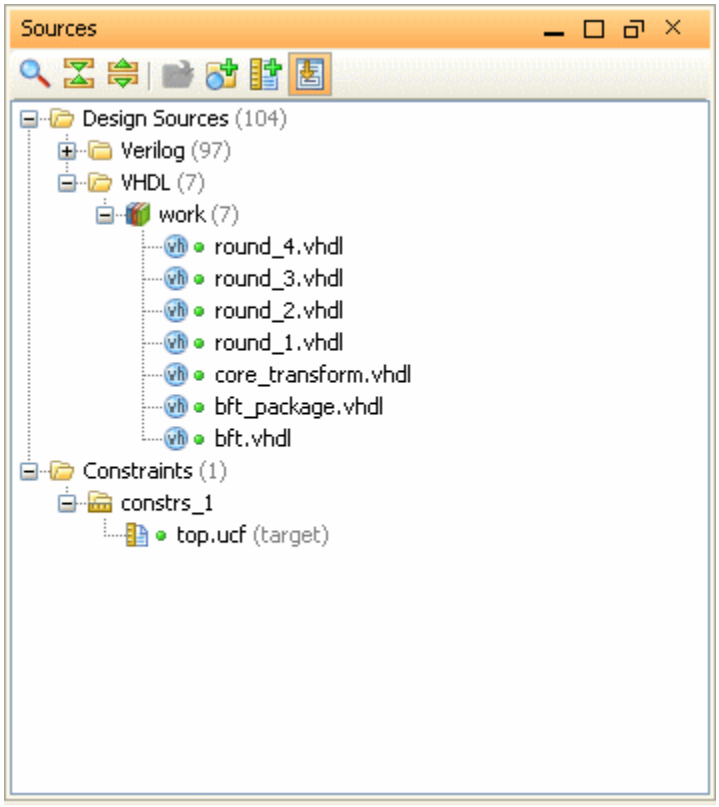

Figure 7: Viewing Sources Grouped by Type

Notice the Design Sources are grouped by file type.

2-1-4. Right-click on one of the VHDL source files and select **Source File Properties** from the popup menu.

2-1-5. View the source file information in the Source File Properties view (Figure 8).

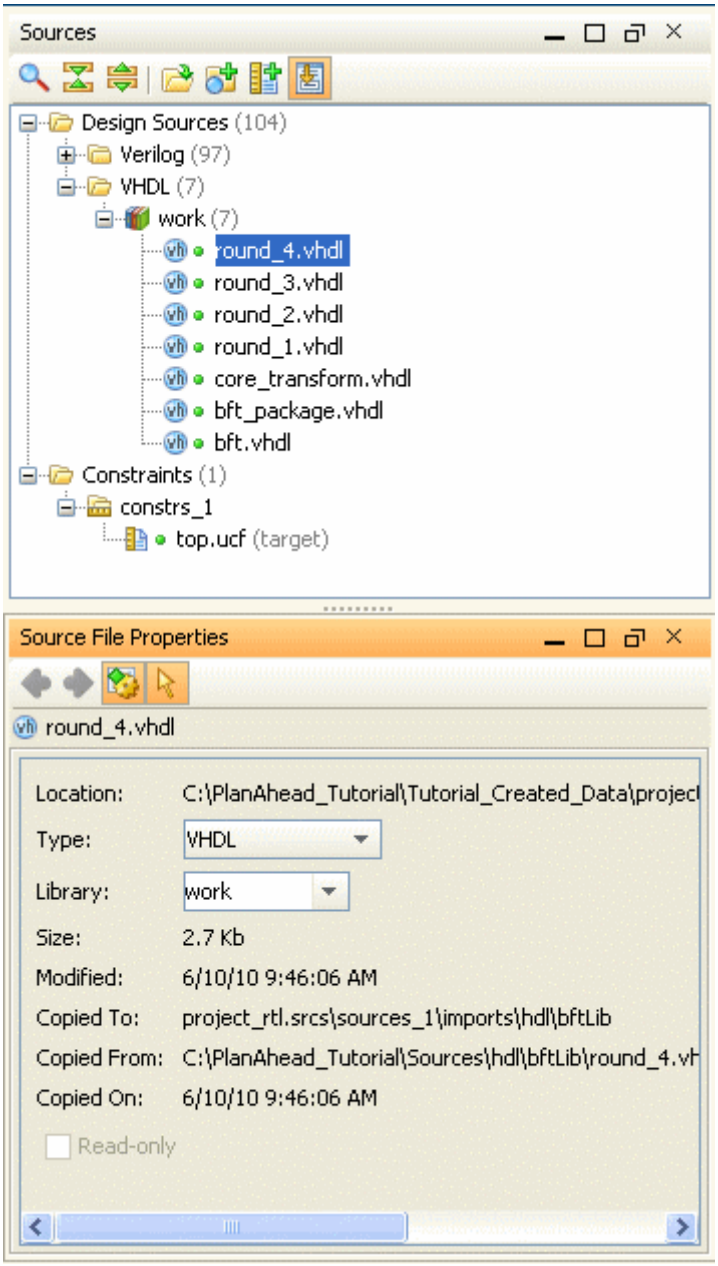

Figure 8: Viewing the Source Files Properties

2-1-6. Close the Source file Properties view by clicking on the X button in the view banner.

2-2. Set the VHDL Library to bftLib for selected VHDL Sources.

2-2-1. In the Sources view, use the **Shift** key to select all VHDL sources *except* the `bft.vhdl` file.

2-2-2. Right-click the selected items and select **Set Library**.

2-2-3. In the Specify Library dialog box, type **`bftLib`**, and click **OK**.

Notice the selected files are now shown under the `bftLib` VHDL library folder (**Error! Reference source not found.**).

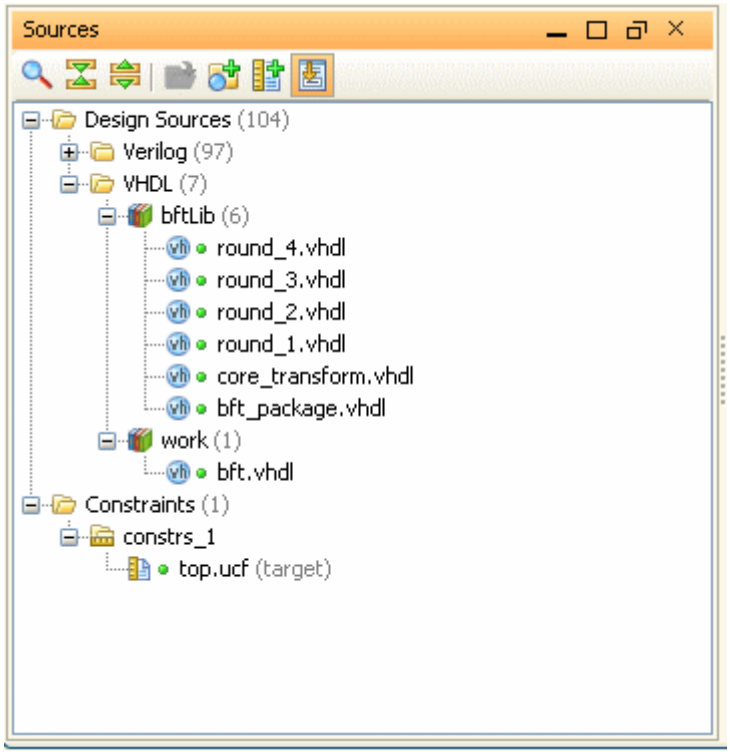

Figure 9: Setting the VHDL Library

2-3. Explore the Sources view commands.

2-3-1. Select one of the VHDL sources in the Sources view.

2-3-2. Right-click to review the available commands in the Sources view pop-up menu. To dismiss it, press the **Esc** key.

2-4. Use the Text Editor to view Source file content.

2-4-1. In the Sources view, double-click on one of the VHDL source files to open it in the Text Editor.

2-4-2. Right-click in the Text Editor to view the available popup commands.

2-4-3. Select the **Find in Files** popup command to invoke the Find in Files dialog box.

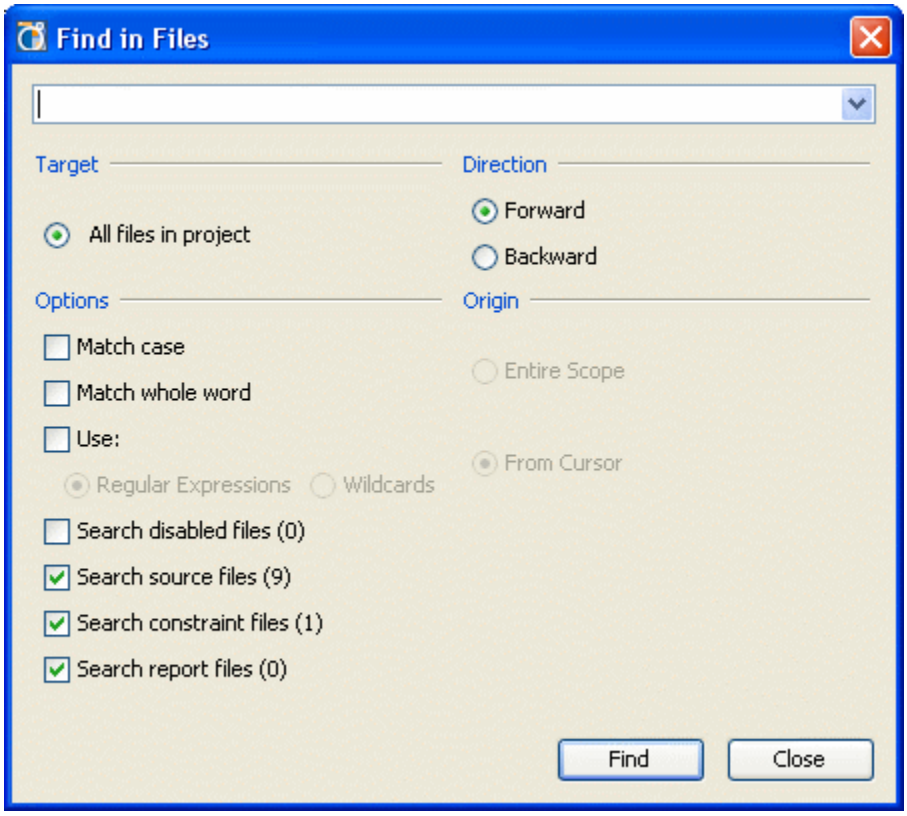

Figure 10: Using the Find in Files Command

2-4-4. Type **`clk`** and click **Find**.

The Find in Files view displays in the messaging area at the bottom of the PlanAhead environment.

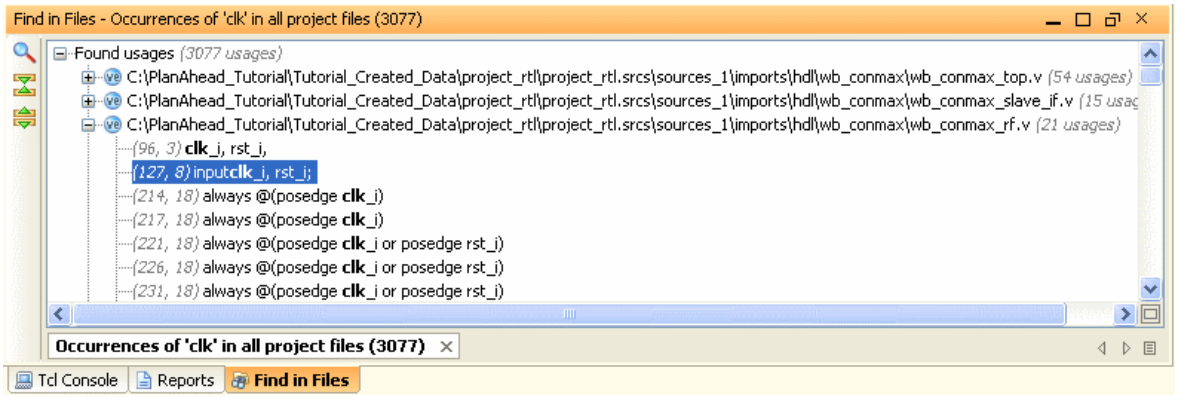

Figure 11: Viewing the Find in Files Results

2-4-5. In the Find in Files view, expand and select one of the Occurrences of `clk` and notice that the Text Editor now displays the file and occurrence.

2-4-6. In the Find in Files view banner.

2-4-7. In each of the open RTL file tabs in the Text Editor.

2-5. Create a new RTL source file and import a template.

PlanAhead enables new Verilog or VHDL source files to be created. Standard Xilinx templates can be used a starting point for a variety of logic and code constructs.

2-5-1. In the Flow Navigator under Project Manager, select **Add/Create Sources**

2-5-2. Select the **Create File** button in the Add/Create Sources dialog.

The Create Source File dialog box opens (Figure 12).

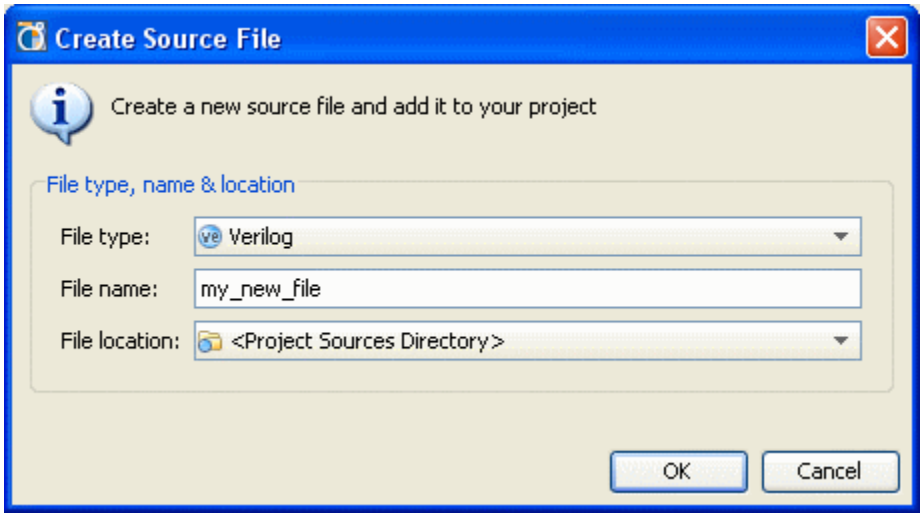

Figure 12: Create Source File Dialog Box

2-5-3. In the File name field, type **`my_new_file`**.

2-5-4. Click **OK**.

2-5-5. Click **OK** in the Add/Create Sources dialog.

Notice that the new empty file is now listed in the Verilog folder in the Sources view.

2-5-6. In the Sources view, double-click *my_new_file.v* to open it in the Text Editor.

2-5-7. Right-click and select the **Insert Template** popup command in the Text Editor to open the Insert Template dialog box (Figure 13).

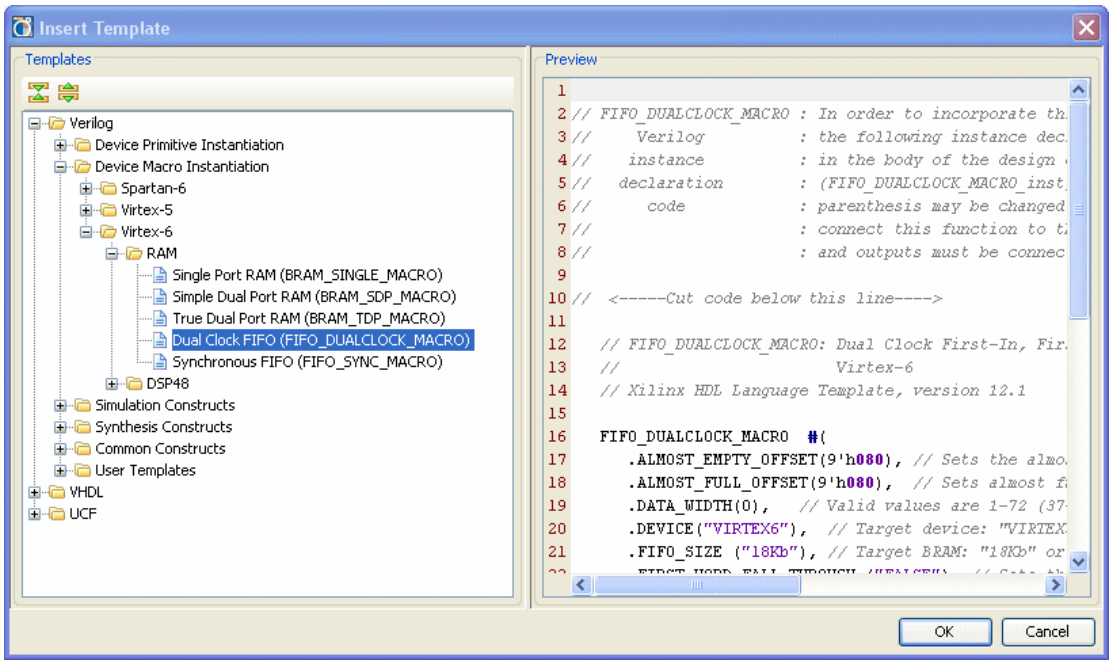

Figure 13: Insert Template Screen

2-5-8. Expand the Verilog folder to examine the types of templates available, select one, and click **OK**. Notice the template text is now inserted in the new source file.

2-5-9. Close the Text Editor by clicking the X button in the view tab.

2-5-10. In the Save Text Editor Changes dialog box, click **No**.

Step 3: Elaborate and Analyze the RTL Design Step 3

PlanAhead provides RTL elaboration capability to compile RTL source files in the project. Displayed compilation errors and warnings are cross-selectable to the lines that are in error in the RTL code. The RTL logic hierarchy is expanded and available for analysis. Once elaborated, all of the RTL views enable cross-selection of logic objects. Opening the RTL Design from the Flow Navigator will automatically elaborate the RTL design and present the Design Planner and I/O Planner view layouts.

- The RTL Netlist and Hierarchy views display the logic hierarchy of the design.
- The RTL Schematic enables interactive logic exploration.
- The Find command enables searching of RTL logic objects.
- The Instance Properties view displays information about the selected logic instantiation including resource estimation.
- The RTL DRCs highlight potential areas of the design to improve power or performance.

3-1. Elaborate and Open the RTL Design using `top` as the top-level module.

3-1-1. In the Flow Navigator, select the RTL Design button.

3-1-2. In the Top Module dialog box, type **`top`** in the Top Module Name field and click **OK** to start the elaboration (Figure 14).

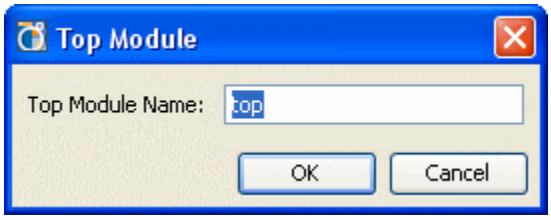

Figure 14: Top Module Dialog Box

The RTL design elaborates and the Elaboration Messages view opens (Figure 15).

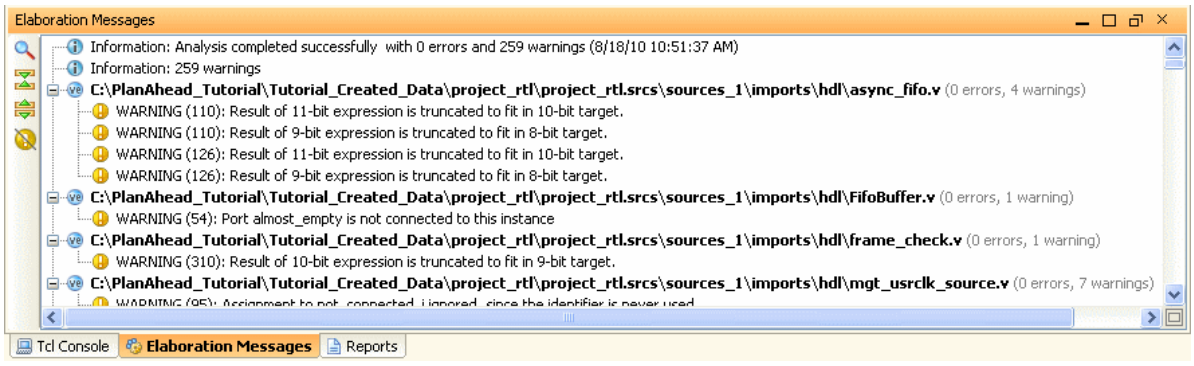

Figure 15: Viewing Elaboration Messages

3-2. Review the various Elaboration Warnings.

3-2-1. In the Elaboration view, click the Hide Warning Messages button .

Notice that there are no Errors in the design. If there were errors messages, they would display in the filtered list.

3-2-2. In the Elaboration Messages view, click the Hide Warning Messages button again to display the Warnings.

3-2-3. In the Elaboration Messages view, click one of the Warning messages. The offending line in the RTL file displays in the Text Editor. The Source file is opened, if needed.

3-2-4. Close the Elaboration Messages view by clicking the X button in top right corner of the view banner.

3-2-5. Close the Text Editor by clicking the X button of all open RTL files.

3-3. Examine the RTL logic hierarchy.

3-3-1. In the RTL Netlist view, expand the **`usbEngine0`** instance by clicking the plus sign (+) next to it.

3-3-2. Select the **`usbEngine0/u0`** instance.

3-3-3. Right-click and select the **Show Definition** popup command.

In the Text Editor notice that the RTL file containing the `usbg_utmi_if` module instantiation is opened (Figure 16).

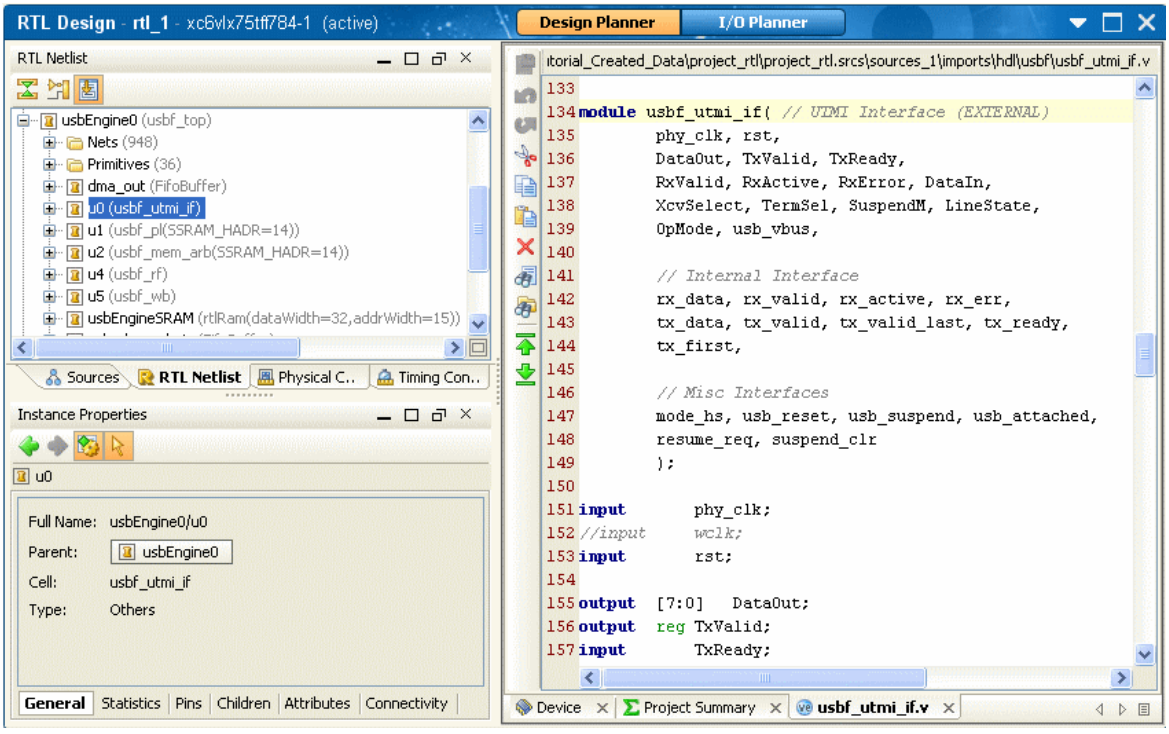

Figure 16: Viewing the RTL Logic Hierarchy

3-3-4. In the RTL netlist view, right-click and select the **Show Source** popup command, and see that the RTL line containing the `usbf_utmi_if` code opens in the Text Editor.

3-3-5. In the RTL view, right-click and select the **Show Hierarchy** popup command.

The RTL Hierarchy view opens with the selected module highlighted. The modules display with rectangles sized relative to the amount of logic contained in them, making it easy to locate large modules (Figure 17).

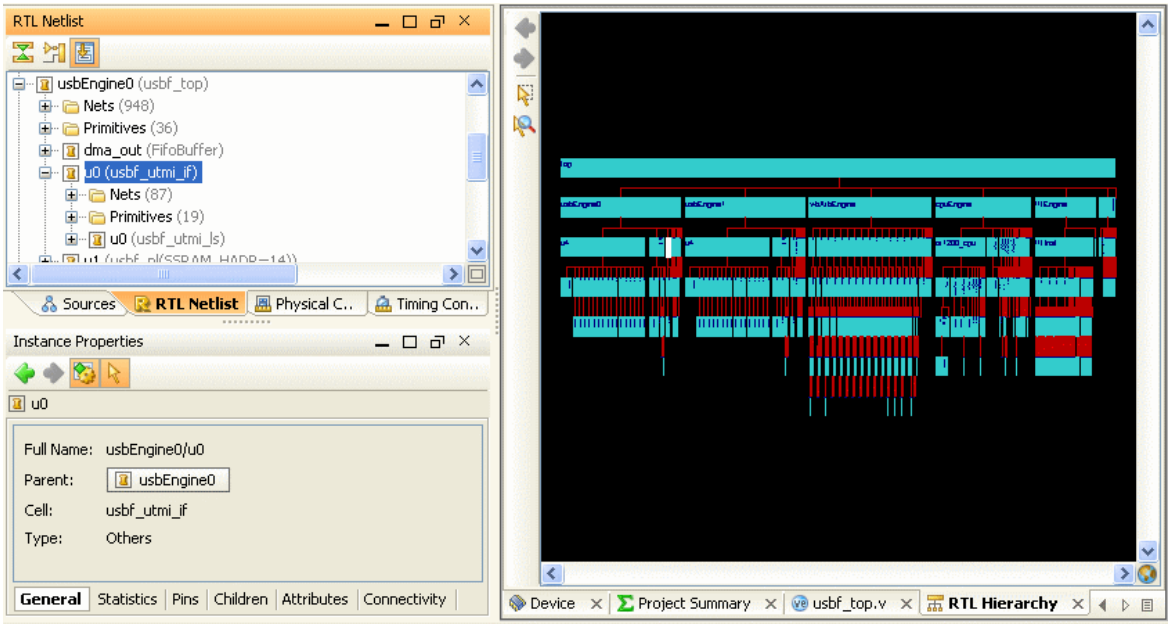

Figure 17: Displaying Modules in the RTL Hierarchy View

3-3-6. In RTL Hierarchy tab, click the X button.

3-3-7. In the Text Editor, click the X button for all open RTL files.

3-4. Examine the RTL Schematic.

3-4-1. In the RTL Netlist view, expand and select the **`usbEngine0/u0/u0`** instance (the level below the previous selection).

3-4-2. Select the Schematic button in the RTL Netlist view, or right-click and select the **Schematic** popup command.

3-4-3. Inside the `u0` module, double-click the **`LineState`** pin to expand the logic.

3-4-4. Zoom Fit the RTL Schematic view.

Hint: This can be done using a cursor stroke: click and drag the left mouse button in the RTL Schematic view from the lower right to the upper left (Figure 18).

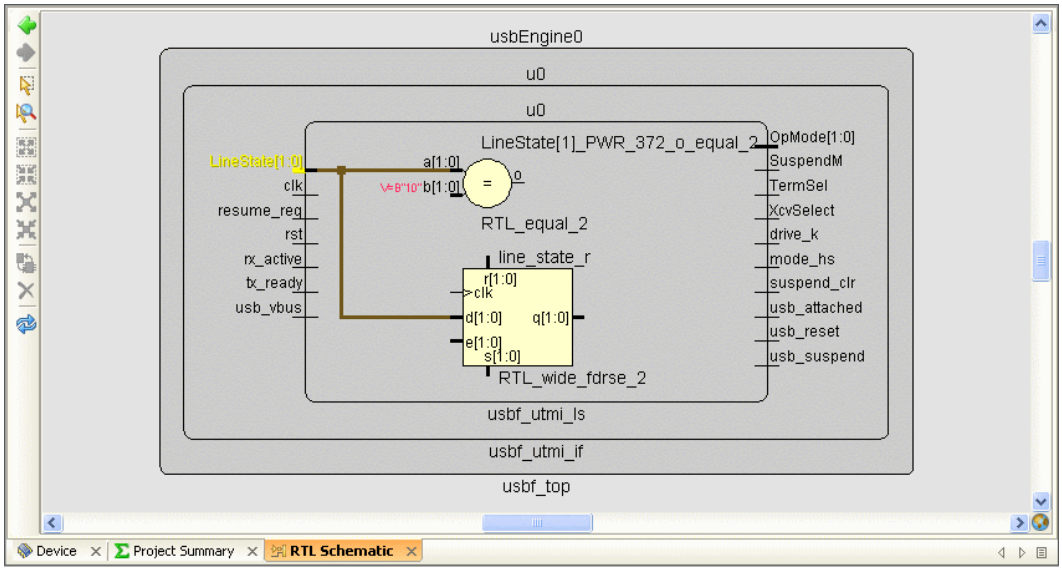

Figure 18: Viewing the RTL Schematic

3-4-5. Outside the `u0` module double-click on the **`LineState`** pin to expand the logic (Figure 19).

3-4-6. Zoom Fit the RTL Schematic view.

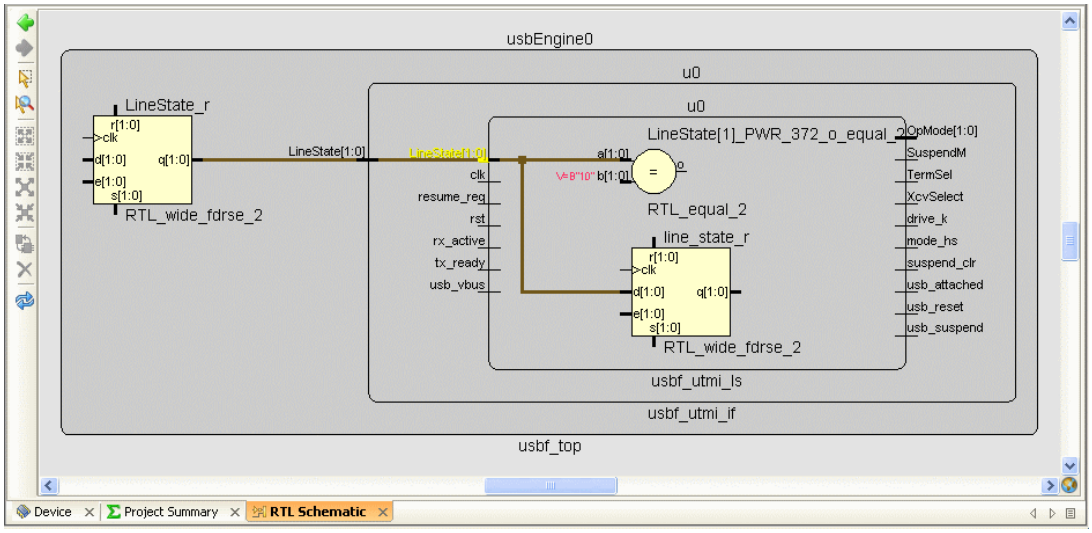

Figure 19: Expanding Logic in the RTL Schematic View

Further schematic exploration capabilities are covered in the *PlanAhead Tutorial: Design Analysis and Floorplanning for Performance* (UG676).

3-4-7. On the left side of the RTL Schematic view, select the RTL_wide_fdrse_2 instance.

3-4-8. In the RTL Schematic view, right-click and select the **Show Source** popup command to see that the RTL file with the logic definition displays.

3-4-9. Close the Text Editor and the RTL Schematic.

3-4-10. In the RTL Netlist view, click the Collapse All button. 

3-5. Use the Find command to locate RTL Block RAM logic.

3-5-1. Click the Find button in the main toolbar  or select **Edit > Find** to open the Find dialog box (Figure 20).

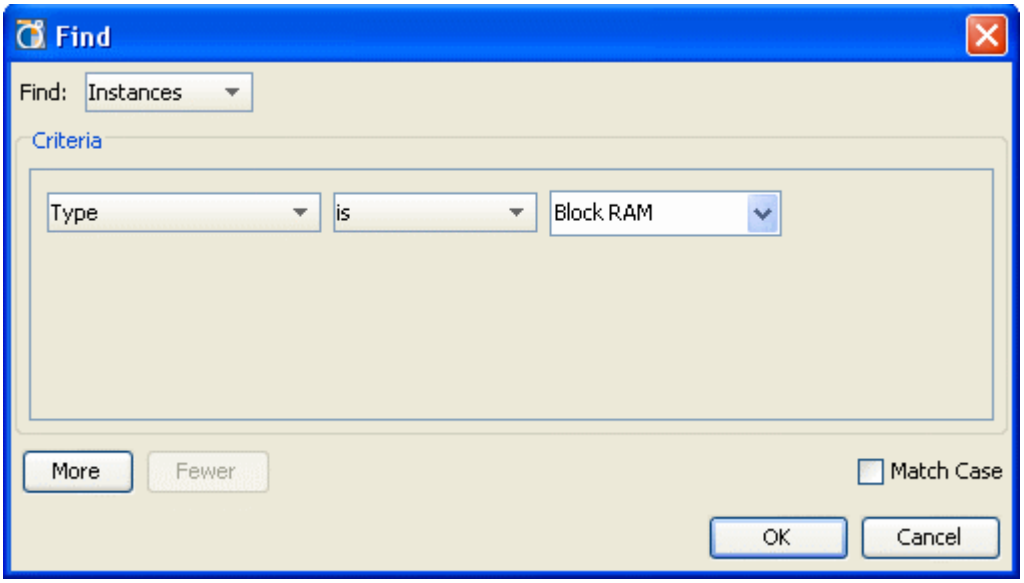

Figure 20: Searching for RTL Logic using the Find Dialog Box

3-5-2. Examine the Find filter options.

3-5-3. Set the **Criteria** to `Type is Block RAM`, and click **OK.**

The Find Results view displays (Figure 21).

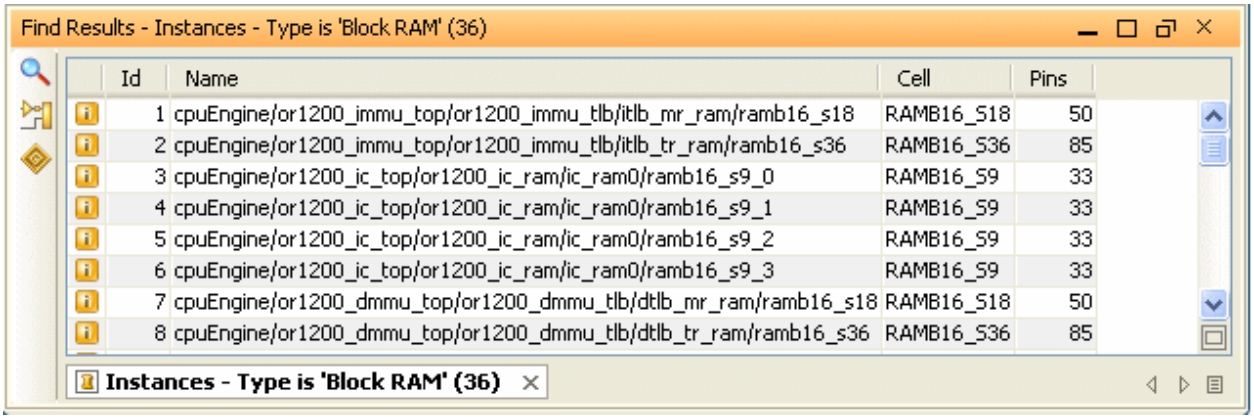

Figure 21: Find Results for RTL Block RAM Search

Notice that the Find Results view displays the results of the search and selects one of the Block RAMs in the list.

3-5-4. Right-click and select **Show Source**.

Notice the instance is selected in the RTL Netlist view and displayed in the Text Editor.

3-5-5. Close the Find Results view and the file in the Text Editor.

Step 4: Estimating Resource Utilization and Power Step 4

4-1. Examine the Resource Estimation options.

4-1-1. Select the **Resource Estimation** command in the Flow Navigator.

The Resource Estimation view is displayed (Figure 22).

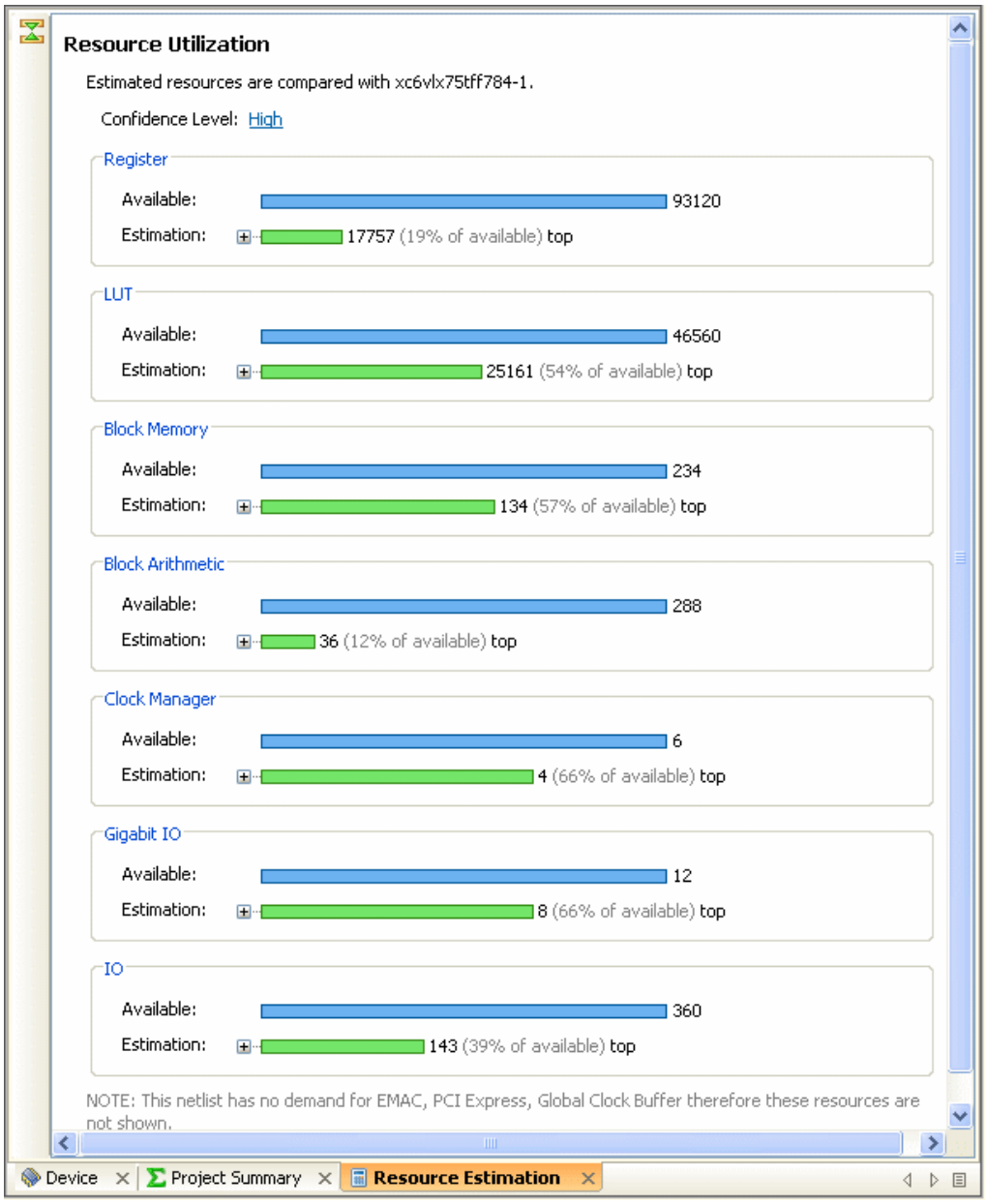

Figure 22: Viewing the RTL Resource Estimation

4-1-2. Expand the Block Memory Estimation tree and others to explore the hierarchical chart report.

4-1-3. Close the Resource Estimation view by clicking on the X icon in the Resource Estimation tab.

4-2. Examine resource estimates for RTL instances.

4-2-1. In the RTL Netlist view, select **top** and see that the RTL Macro Resources displays in the Netlist Properties view (Figure 23).

4-2-2. If the Netlist Properties view is not displayed, right-click and select the **Netlist Properties** popup command.

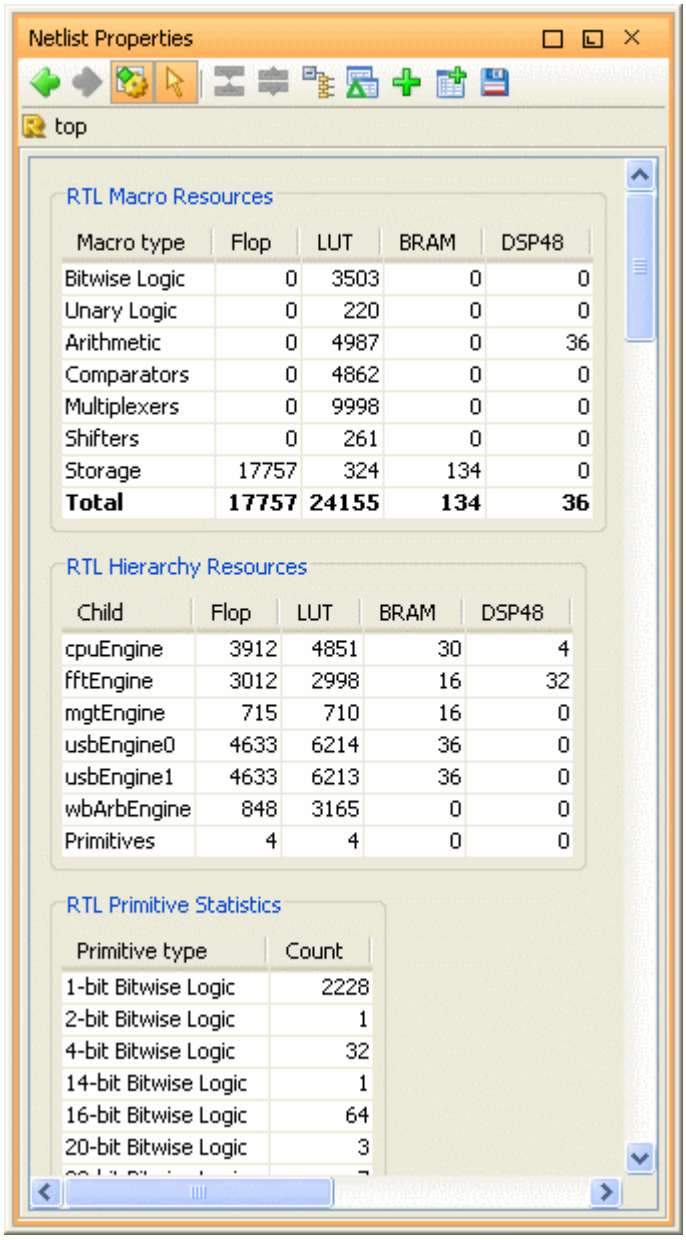

Figure 23: Viewing RTL Resource Estimates

4-2-3. Scroll down the Netlist Properties to examine the information including RTL Memory Resources, RTL Hierarchy Resources, RTL Primitive Statistics, Net Boundary Statistics, and Clock Report.

4-2-4. In the RTL Netlist view, select any of the other modules and examine the same estimates for the selected module. You may need to click the **Statistics** tab in the Instance Properties view.

4-3. Estimate Power consumption for the RTL design.

4-3-1. Click **Power Estimation** in the Flow Navigator.

The Power Estimation dialog box opens (Figure 24).

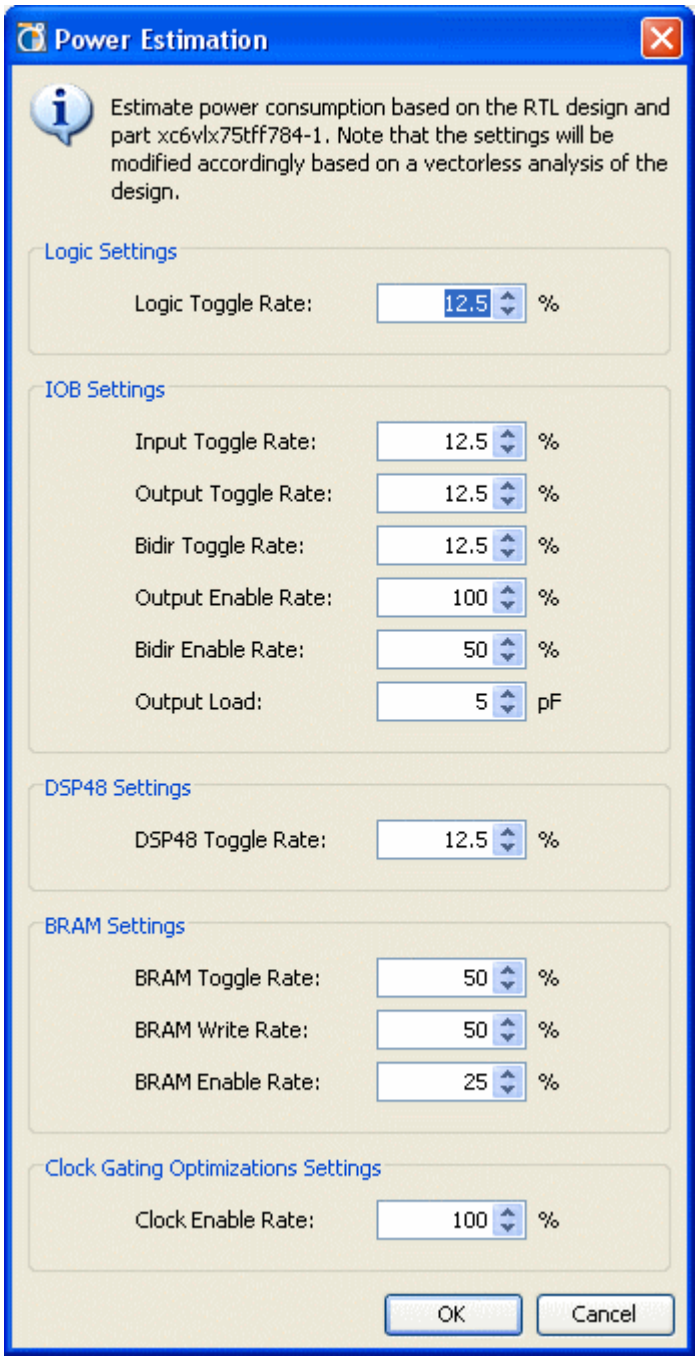

Figure 24: Power Estimation Settings

4-3-2. Accept the default Toggle and Enable Rate values and click **OK.**

4-3-3. The Power Estimation view opens (Figure 25).

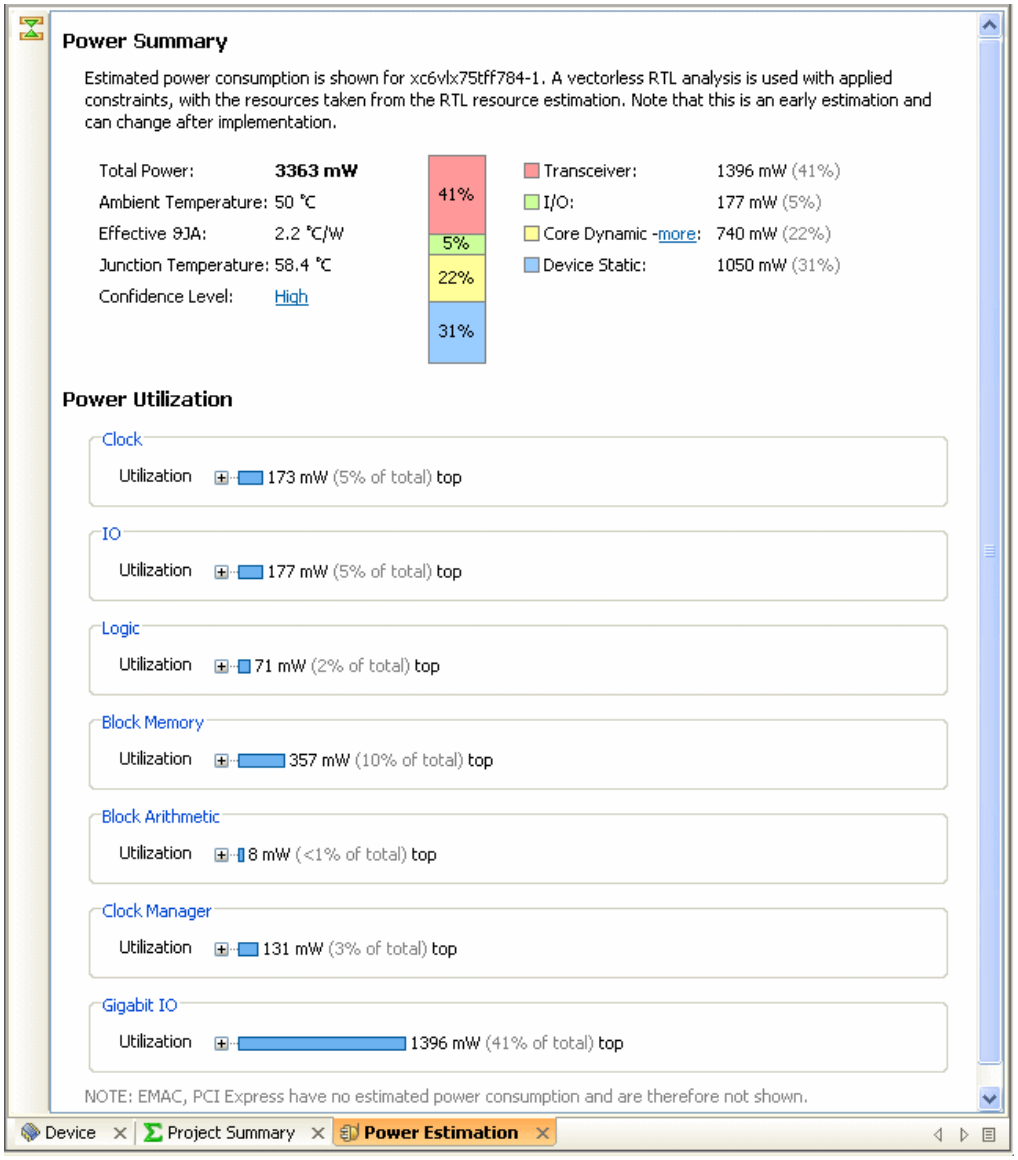

Figure 25: Power Estimation for the RTL Design

4-3-4. Scroll down and expand the Utilization trees for the various types of resources.

4-3-5. Close the Power Estimation view.

Step 5: Running RTL Design Rule Checks Step 5

PlanAhead provides Design Rule Checks (DRC) that can be run on the RTL Design. These include LINT-style RTL checks for power or performance improving suggestions.There are also basic I/O bank and voltage rules for the RTL Design. Once the design is synthesized, a more comprehensive set of logic design, I/O, and clock DRCs is available for the Netlist Design.

5-1. Run DRCs.

5-1-1. From the Flow Navigator or the Tools menu, select **Run DRC**.

5-1-2. In the Run DRC dialog box, expand and examine the RTL rules (Figure 26), and click **OK**.

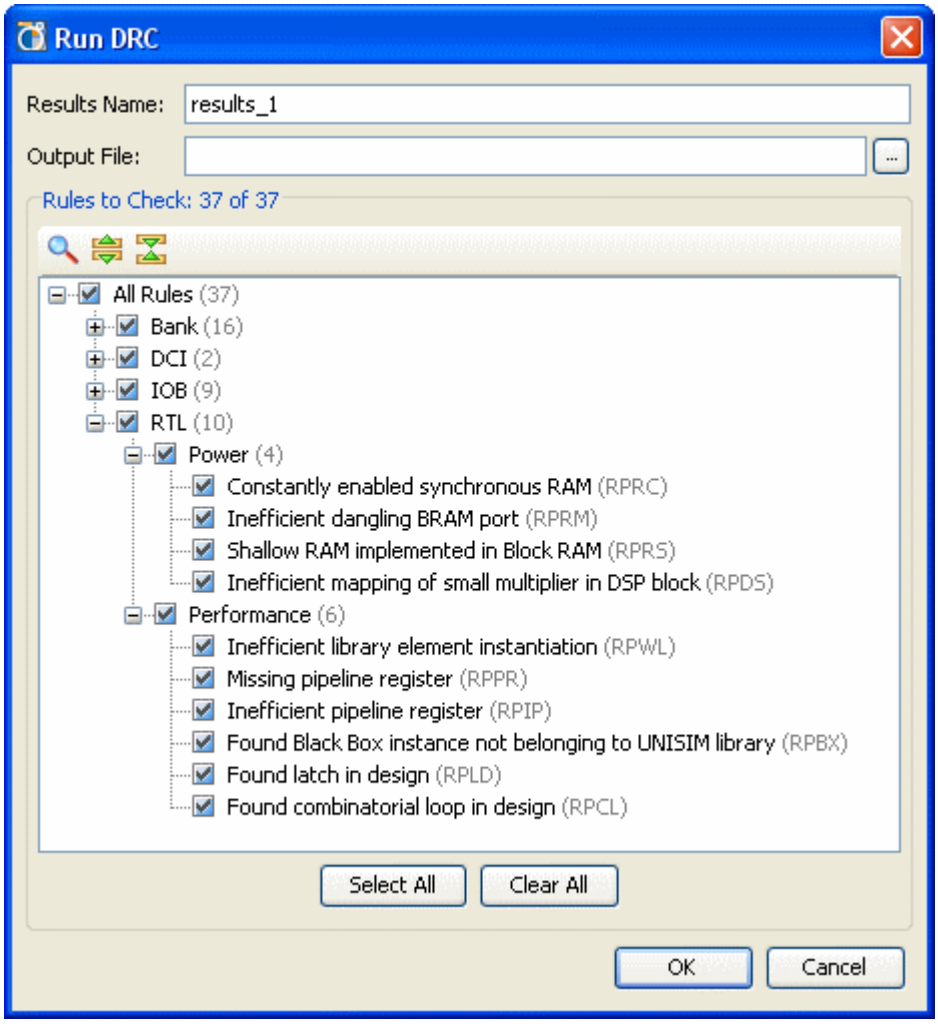

Figure 26: Running RTL DRCs

The DRC Results view opens (Figure 27).

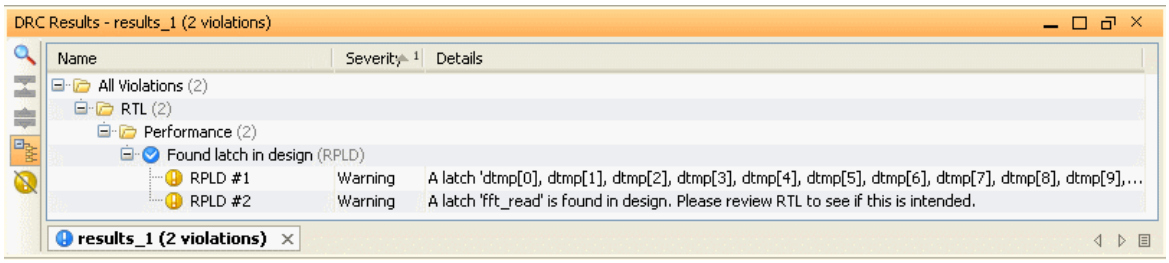

Figure 27: Viewing RTL DRC Results

The RTL Results viewer shows Errors, Warnings, and Informational messages as follows:

- Errors with a red icon
- Warnings with an orange icon
- Informational messages with a yellow icon

5-1-3. Select the **RPLD #1** latch warning in the list.

Notice the Violation Properties view displays with information about the violation and links to select the offending logic objects.

5-1-4. In the Violation Properties view click the **dtmp[0]** link, and see that the logic object is selected in the RTL Netlist view.

5-1-5. In the RTL Netlist view, select the **Show Source** popup menu command (or press the **F7** key) to invoke the Text Editor.

5-1-6. Close the DRC Results view and any open Text Editor views.

5-1-7. Close the RTL Design by selecting the X button in the RTL Design view header, and then click **OK** in the confirmation dialog box.

Step 6: Selecting IP from the Xilinx IP Catalog Step 6

PlanAhead is integrated with the CORE Generator software tool to provide an IP Catalog with search and filtering capabilities to easily find the desired IP. Once located, you can customize, instantiate and implement the core directly from the PlanAhead environment. Access to the IP Catalog is provided in both the Project Manager and RTL Design environments.

6-1. Open the IP Catalog and explore the search options.

6-1-1. Select **IP Catalog** from the Flow Navigator.

6-1-2. Expand some of the IP categories.

6-1-3. Select an IP and explore the available toolbar buttons and pop-up menu commands (Figure 28).

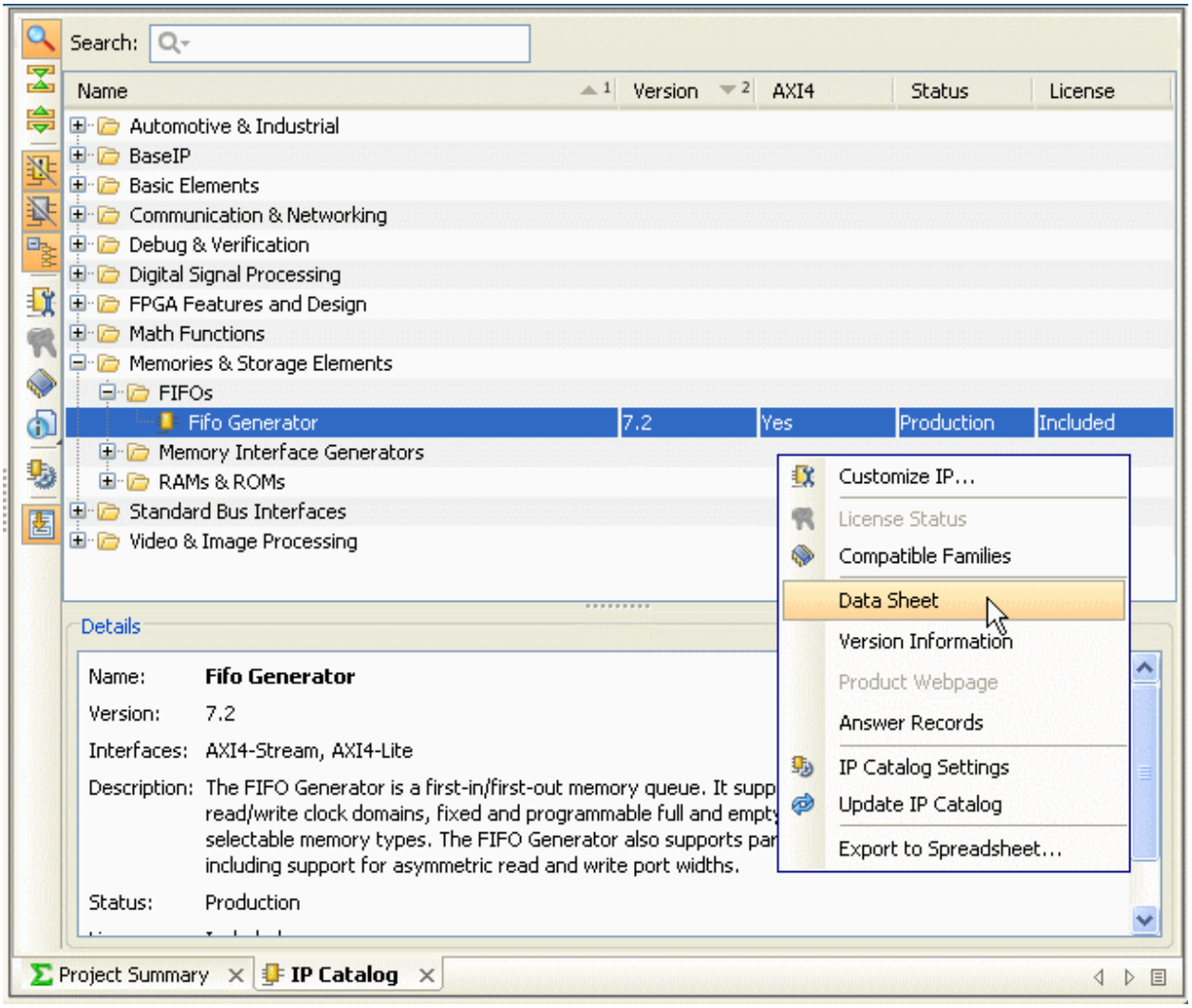

Figure 28: Browse the IP Catalog

6-1-4. Notice the Details for the selected IP are displayed at the bottom of the view. By default, only the IP that is current and available for the device family selected is displayed.

Note: To view all IP, toggle the Hide Superseded and Discontinued IPs button  and Hide incompatible IPs button  . To view a flattened list of IP, toggle the Group by Category toolbar button  .

6-1-5. Type **fir** in the Search field at the top of the view.

6-1-6. Select a FIR Compiler IP and click the Data Sheet button  .

6-1-7. Wait a few moments for the datasheet to open and then examine it and close the PDF viewer.

6-1-8. Clear the Search field to expand the Catalog list.

Step 7: Customizing and Instantiating IP Step 7

7-1. Customize a simple adder IP.

7-1-1. Click the Group by Category button.

7-1-2. Expand the **Math Functions > Adders & Subtracters** folder.

7-1-3. Double-click on the Adder Subtracter to run the Customize IP command.

This will invoke the CORE Generator tool and present the customization interface for the IP selected. Various IP have different types of interfaces (Figure 29).

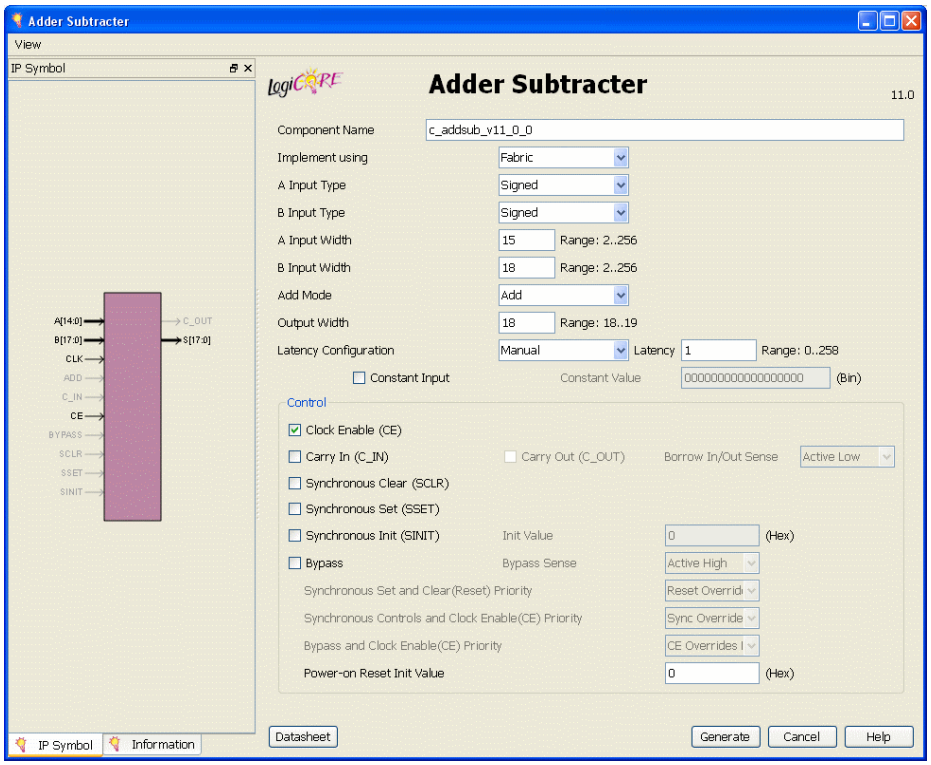

Figure 29: Customize IP using CORE Generator

7-1-4. In the B Input Width field, type **18**.

7-1-5. Click **Generate**.

Clicking the Generate button has a different effect when launched from PlanAhead than when running Core Generator standalone. In standalone mode, CORE Generator automatically launches XST to synthesize the IP core. When launched from PlanAhead, the synthesis step is not run automatically, which enables you to instantiate and configure the core in your RTL before launching synthesis. You can synthesize the IP at any time or launch synthesis on the design and the IP are synthesized first automatically.

7-2. Instantiate the adder IP.

7-2-1. In the Sources view, click the Collapse All button and then expand the **IP** folder.

7-2-2. Expand the IP folder and then expand the **c_addsub_v11_0_0** IP.

7-2-3. Double-click the `c_addsub_v11_0_0.veo` file to view the instantiation template in the Text Editor (Figure 30).

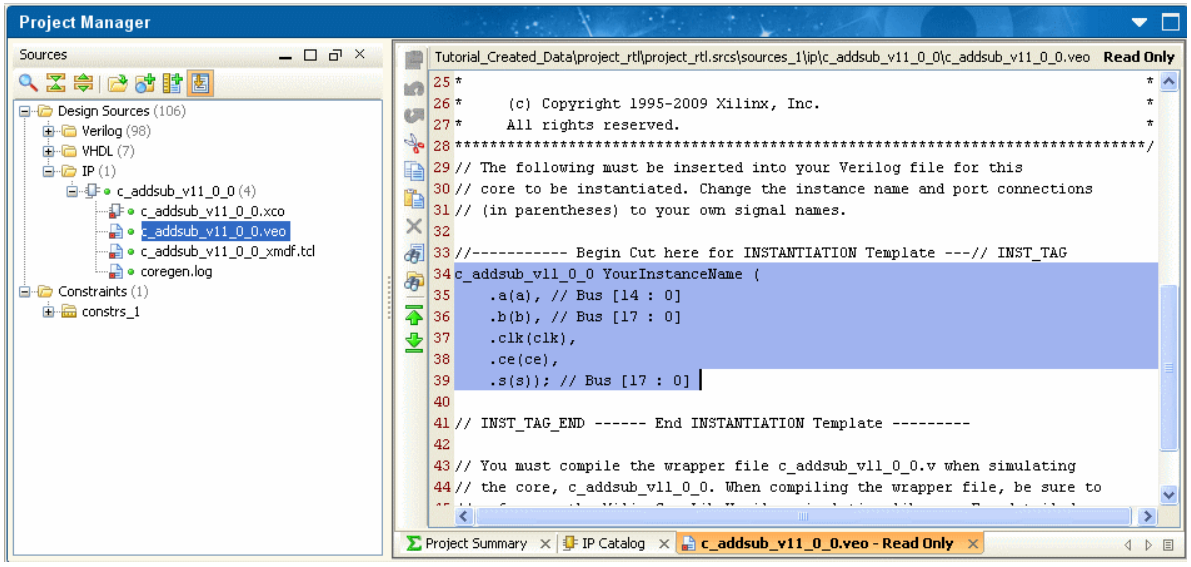

Figure 30: Viewing the Instantiation Template

7-2-4. Select the text in the Text Editor, as shown above, and select the Copy button .

7-2-5. In the Sources view, expand the Verilog folder.

7-2-6. Scroll and double-click the `top.v` file to open it in the Text Editor and scroll to the bottom of the file just before the `endmodule` text.

7-2-7. Select the line just above the `endmodule` statement and select the Paste button .

7-2-8. Change the *YourInstanceName* text in the template to `my_adder` (Figure 31).

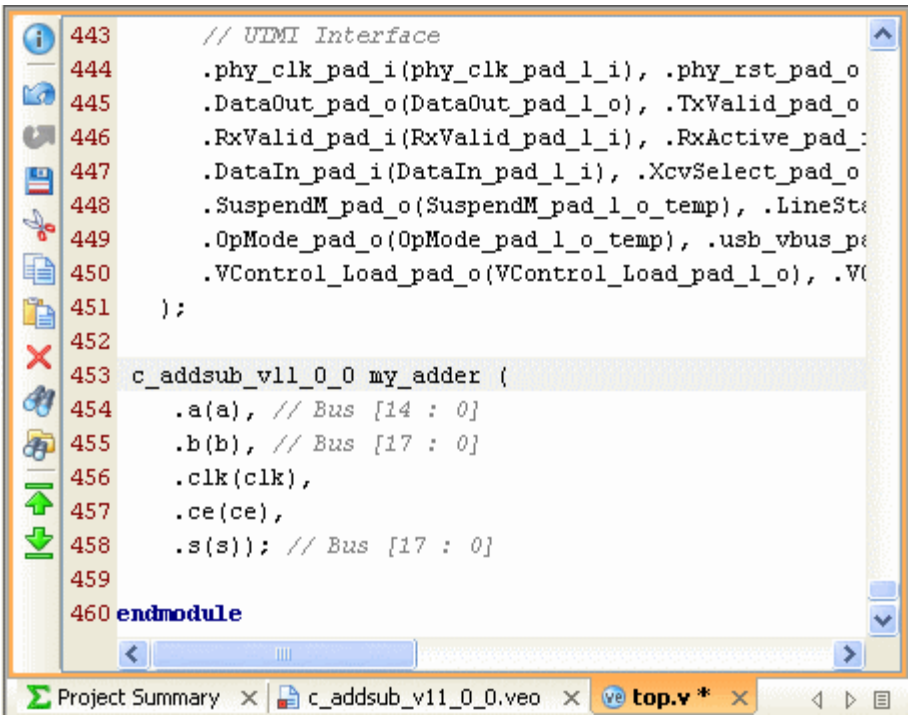

Figure 31: Instantiate IP in Your Design

7-2-9. Change the .clk port definition to use the existing cpuClk clock signal (Figure 32).

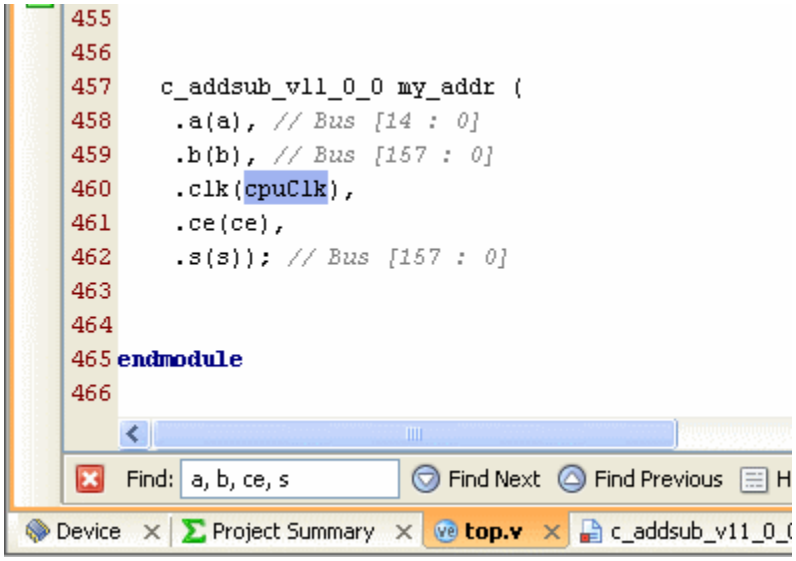

Figure 32: Modify the clk signal to use the cpuClk clock signal

7-2-10. Scroll to the top of the top.v file and add the IP ports (a, b, ce, s) to the module port definition (Figure 33).

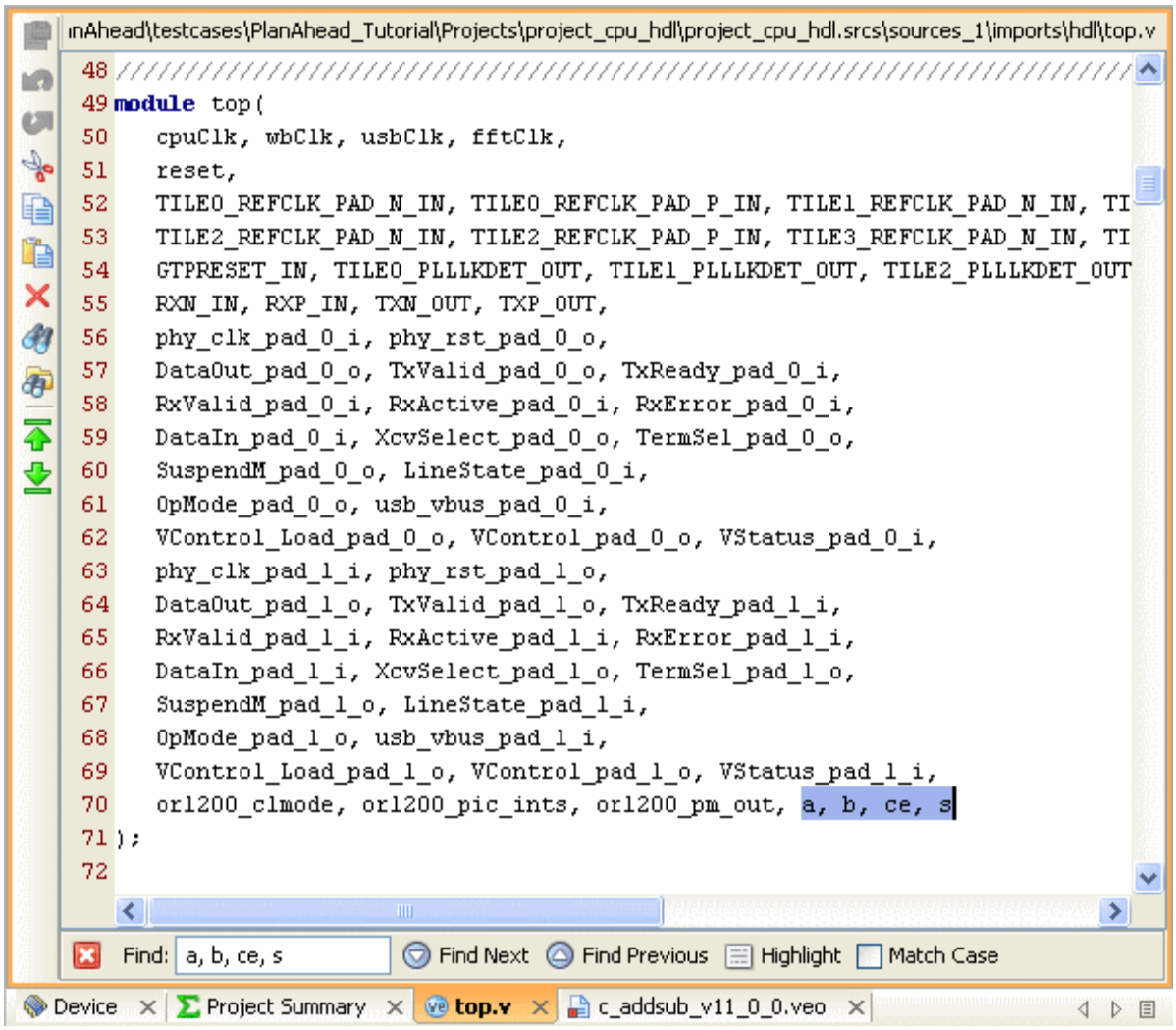

Figure 33: Add IP Ports to the top-level module port list

7-2-11. Define the new ports in the top.v file. Add the following text as shown in Figure 34.

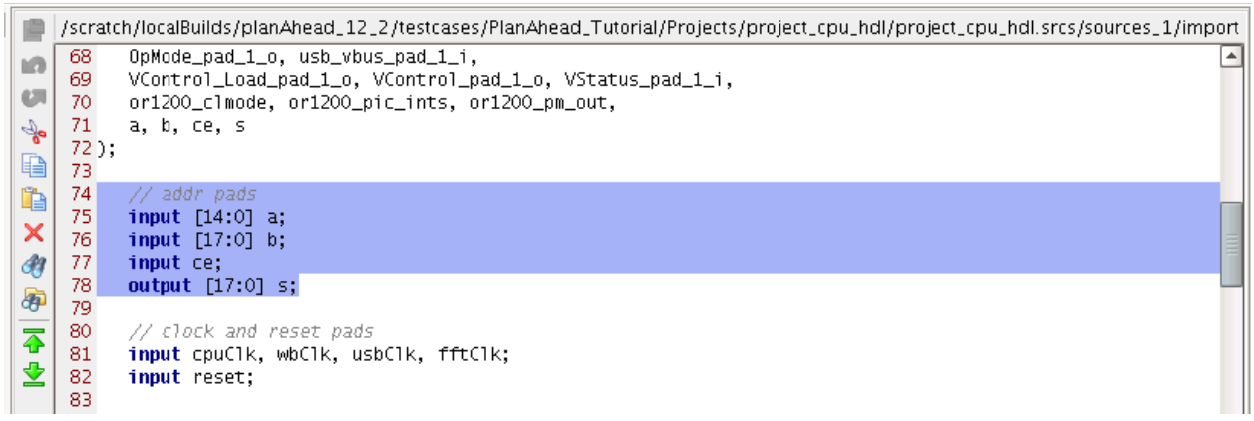

Figure 34: Define IP Ports in top.v file

7-2-12. The top.v file should look exactly like it does in Figure 34.

7-2-13. To close the `top.v` file, click the X icon in the tab and select **Yes** to save changes.

7-2-14. To close the `.veo` template file, click the X button in the tab.

7-2-15. To close the IP Catalog, click the X button in the tab.

Step 8: Generating IP Step 8

8-1. Generate the IP and explore the logic in the Schematic.

8-1-1. In the Sources view, right-click the `c_addsub_Vxx_x` top-level file, and select **Generate IP**. Wait for the IP to synthesize.

After the IP generates, notice the Compilation Messages view displays information about the IP compilation

The RTL Design needs to be opened.

8-1-2. In the Flow Navigator, click the **RTL Design** button.

8-1-3. In the RTL Netlist view, expand and select the **my_adder** module.

Note: If the a black box icon appears in the RTL Netlist, review the Elaboration messages and go back and check the top.v file for errors. Rerun the Elaborate command until the IP appears in the RTL Netlist.

8-1-4. From the toolbar or popup menu, select the **Schematic** command

8-1-5. In the Schematic view, double-click on the instance to expand the inside logic (Figure 35).

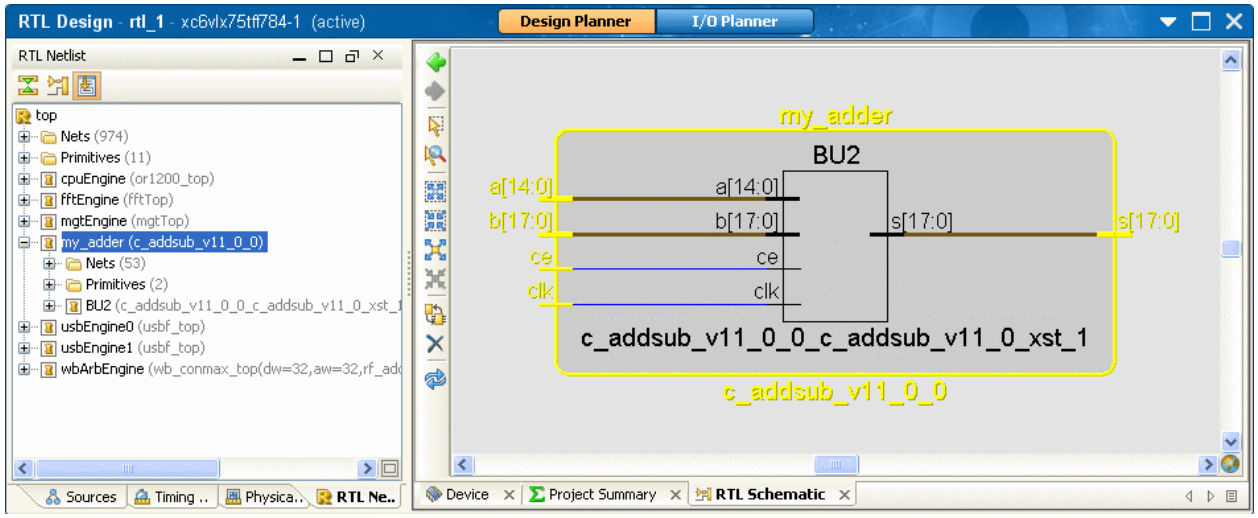

Figure 35: Analyzing the IP Logic in the Schematic

8-1-6. Close the Schematic view.

8-1-7. Select **File > Exit**. If prompted, click **No** to save and **OK** to close PlanAhead.

Conclusion

In this tutorial, you used a small RTL project to examine the PlanAhead RTL development and analysis environment. You started by creating an RTL project, explored RTL sources and the RTL editor. You elaborated the RTL design, and explored the analysis capabilities, which included examining the RTL logic hierarchy, RTL schematic exploration, searching for logic types, reviewing RTL resource and power estimates and running RTL DRCs. You then examined the Xilinx IP Catalog, and customized, instantiated, and implemented a small adder IP core.