

RTL Design and IP Generation Tutorial

PlanAhead Design Tool

UG675(v14.1) May 8, 2012





Notice of Disclaimer

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of the Limited Warranties which can be viewed at <http://www.xilinx.com/warranty.htm>; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in Critical Applications: <http://www.xilinx.com/warranty.htm#critapps>.

©Copyright 2012 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.

Table of Contents

Software Requirements.....	5
Hardware Requirements.....	5
Tutorial Design Description.....	6
Locating Tutorial Design Files	6
Step 1: Creating a New RTL Project	7
Opening the PlanAhead Tool.....	7
Creating a New RTL Project.....	8
Adding Directories and Files	10
Adding a Constraints File	10
Selecting a Default Part	11
Step 2: Using the Sources View and the Text Editor	13
Exploring the Sources View and Project Summary	13
Setting VHDL Library Names	13
Configuring Simulation Source Files.....	15
Exploring the Sources View Commands.....	17
Using the Text Editor to View Source File Content.....	17
Creating a New RTL Source File and Importing a Template.....	17
Step 3: Running Behavioral Simulation	19
Running Behavioral Simulation on the bft Module	19
Setting Compilation Order and Disabling Unused Sources	20
Step 4: Elaborating and Analyzing the RTL Design.....	22
Elaborating and Opening the RTL Design.....	22
Examining the RTL Logical Netlist and Hierarchy	23
Examining the RTL Schematic	24
Using the Find Command to Locate RTL Block RAM Logic.....	26
Step 5: Estimating Resource Utilization.....	28
Examining the Resource Estimation Options.....	28
Examining Resource Estimates for RTL Instances	29
Step 6: Running RTL Design Rule Checks.....	31

Reporting DRCs.....	31
Step 7: Selecting IP from the Xilinx IP Catalog	33
Opening the IP Catalog and Exploring the Search Options	33
Step 8: Customizing and Instantiating IP	35
Customizing a Simple Adder IP	35
Instantiating the Adder IP	36
Conclusion.....	37

RTL Design and IP Generation Tutorial

This tutorial provides an overview of the Register Transfer Level (RTL) development and analysis environment, in which you will:

- Import RTL sources and review them using the Text Editor
- Run Behavioral Simulation on the bft module
- Run elaboration to compile the RTL
- Use a variety of RTL analysis features to explore your compiled RTL design. These include:
 - Analyze the RTL logic hierarchy using the RTL schematic
 - Estimate RTL resources
 - Run RTL Design Rules Checks (DRCs)
- Browse the Xilinx[®] IP Catalog, and customize and implement an Intellectual Property (IP) core in the design

Many of the PlanAhead[™] software analysis features are covered in more detail in other tutorials. Not every command or command option is covered.

The objective of this tutorial is to familiarize you with the RTL development and analysis process using the PlanAhead tool.

Software Requirements

The PlanAhead tool is installed with ISE[®] Design Suite software. Before starting the tutorial, be sure that the PlanAhead tool is operational, and that the tutorial design data is installed.

For installation instructions and information, see the ISE Design Suite: *Installation and Licensing Guide* (UG798) at http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_1/iil.pdf.

Hardware Requirements

Xilinx recommends a minimum of 2 GB of RAM when using the PlanAhead tool on larger devices. For this tutorial, a smaller 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

The small sample design used in this tutorial includes:

- A RISC processor CPU core
- A pseudo FFT
- Four gigabit transceivers (GTs)
- Two USB interfaces

The design targets an xc7k70t device. A small design is used to:

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

Locating Tutorial Design Files

Copy the files from the ISE software installation area:

`<ISE_install_area>/ISE_DS/PlanAhead/testcases/PlanAhead_Tutorial.zip`

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: Creating a New RTL Project

The PlanAhead tool 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.

Opening the PlanAhead Tool

Open the PlanAhead tool:

- On Windows, select the **Xilinx PlanAhead 14.1** desktop icon or **Start > All Programs > Xilinx Design Tools > ISE Design Suite 14.1 > 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.

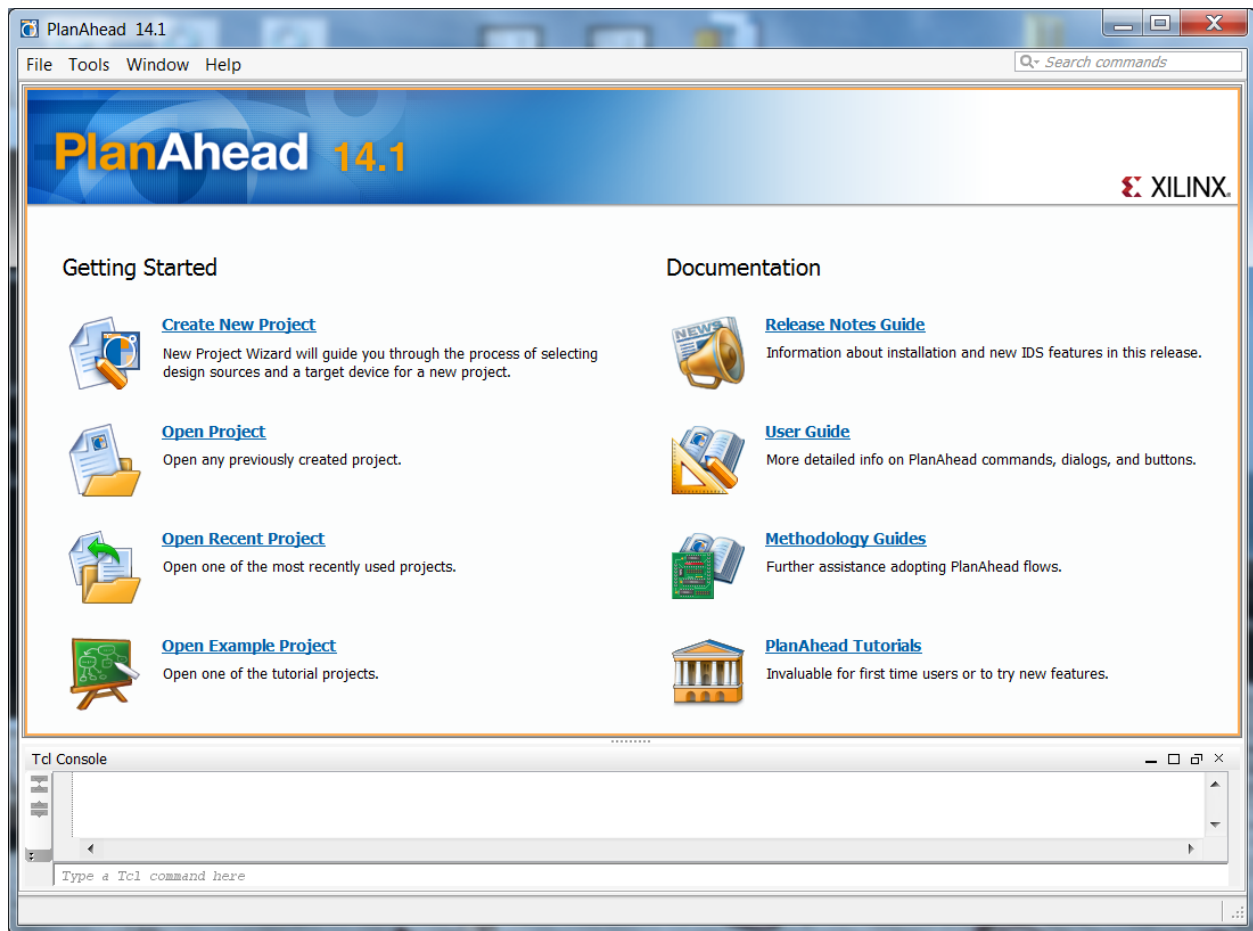


Figure 1: Getting Started Page

The PlanAhead Getting Started page contains links to open or create projects, and view the documentation.

Creating a New RTL Project

1. Create a new project called `project_rtl`, using the RTL source files in:
`<Extract_Dir>\PlanAhead_Tutorial\Sources\hdl` directory.
2. On the Getting Started page, select the **Create New Project** link.
3. In the **Create a New PlanAhead Project** confirmation dialog box, click **Next**.

The Project Name page of the New Project wizard opens:

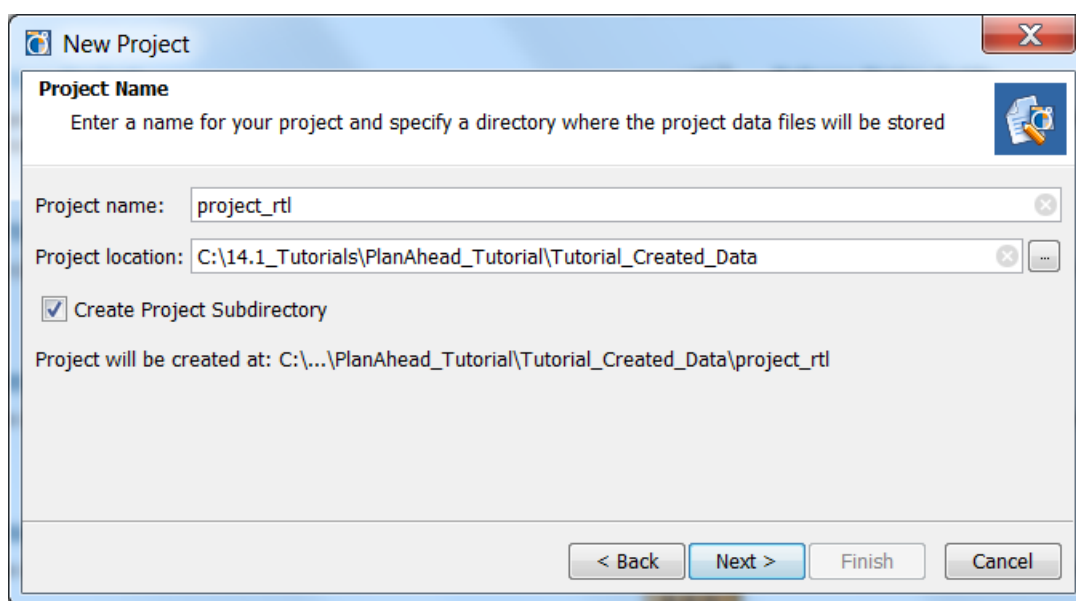


Figure 2: Entering the New Project Name

4. Browse to, and select
`<Extract_Dir>\PlanAhead_Tutorial\Tutorial_Created_Data`.
5. Enter the Project name: **project_rtl**, and click **Next**.

The Project Type page opens.

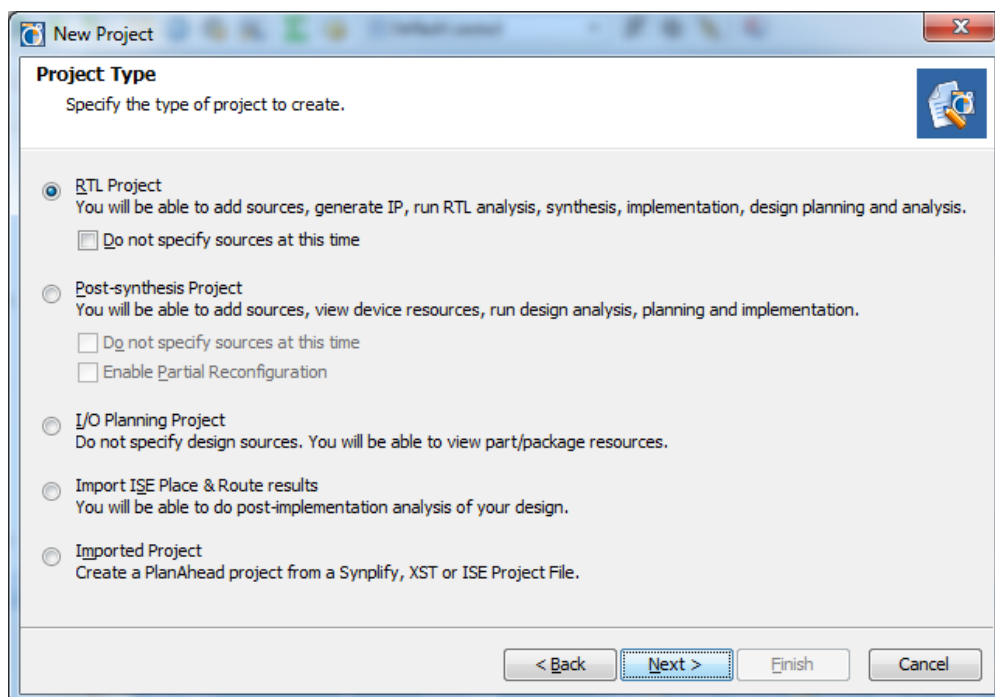


Figure 3: New Project: Design Source Dialog Box

6. Select Specify RTL Project, and click **Next**.

The Add Sources page opens:

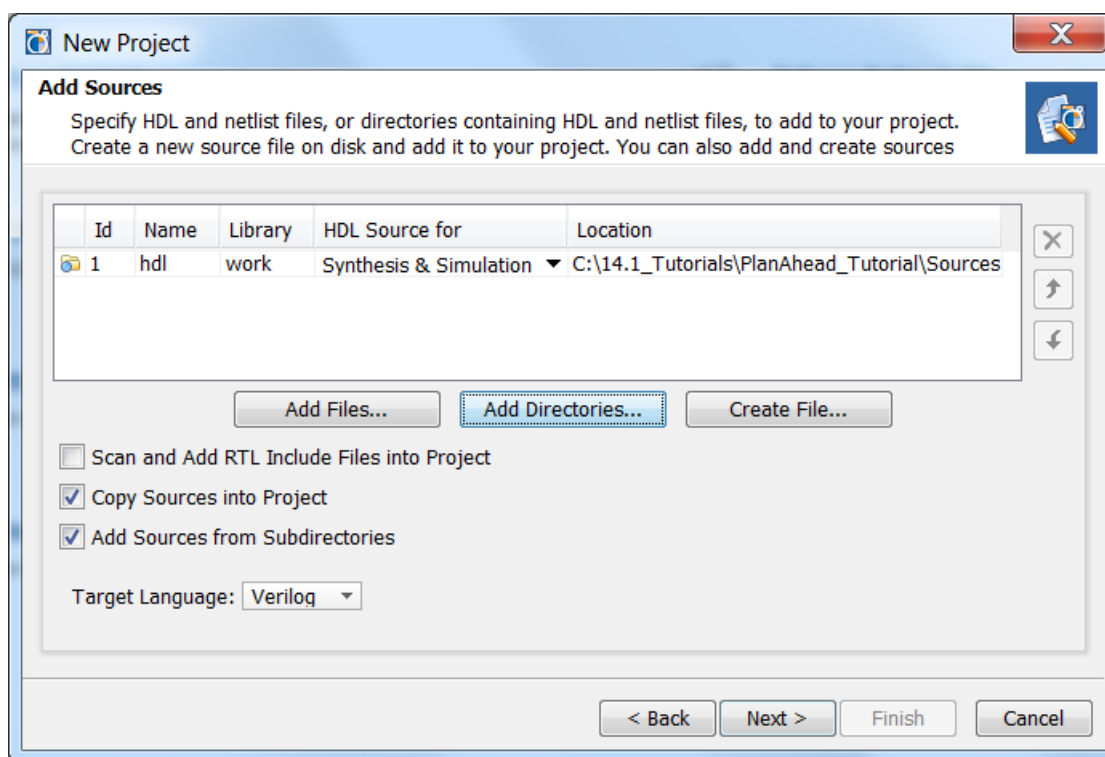


Figure 4: New Project: Add Sources Dialog Box

Adding Directories and Files

7. Click **Add Directories** , browse to , :
<Extract_Dir>/PlanAhead_Tutorial/Sources/hdl
8. Verify that the following checkboxes are selected:
 - a. **Copy Sources into Project.**
 - b. **Add Sources from Subdirectories.**
9. Be sure the page is identical to the previous figure (*New Project: Add Sources Dialog Box*), and click **Next**.
The Add Existing IP page opens.
10. Click **Next**.
The Add Constraints Files page opens:

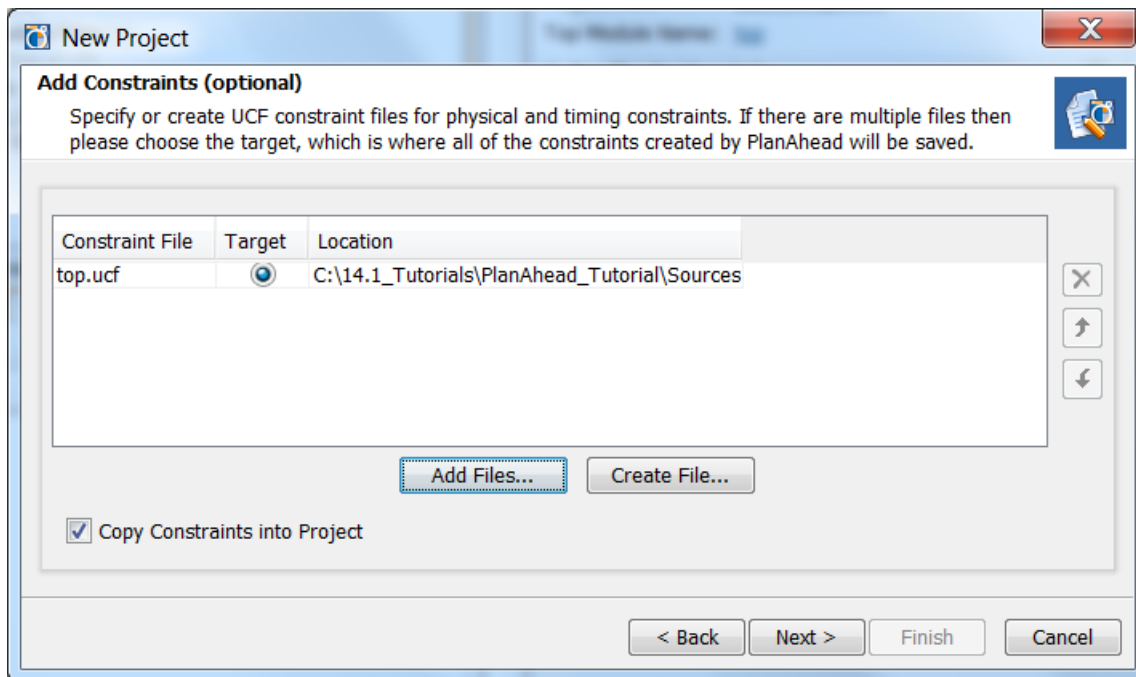


Figure 5: Selected Constraint File to Add to the Project

Adding a Constraints File

11. Click **Add Files** and browse to select the following file:
<Extract_Dir>/PlanAhead_Tutorial/Sources/top.ucf
12. Click **OK**.
13. Ensure the **Copy Constraints into Project** option is set to on, and click **Next**.
The Default Part page opens.

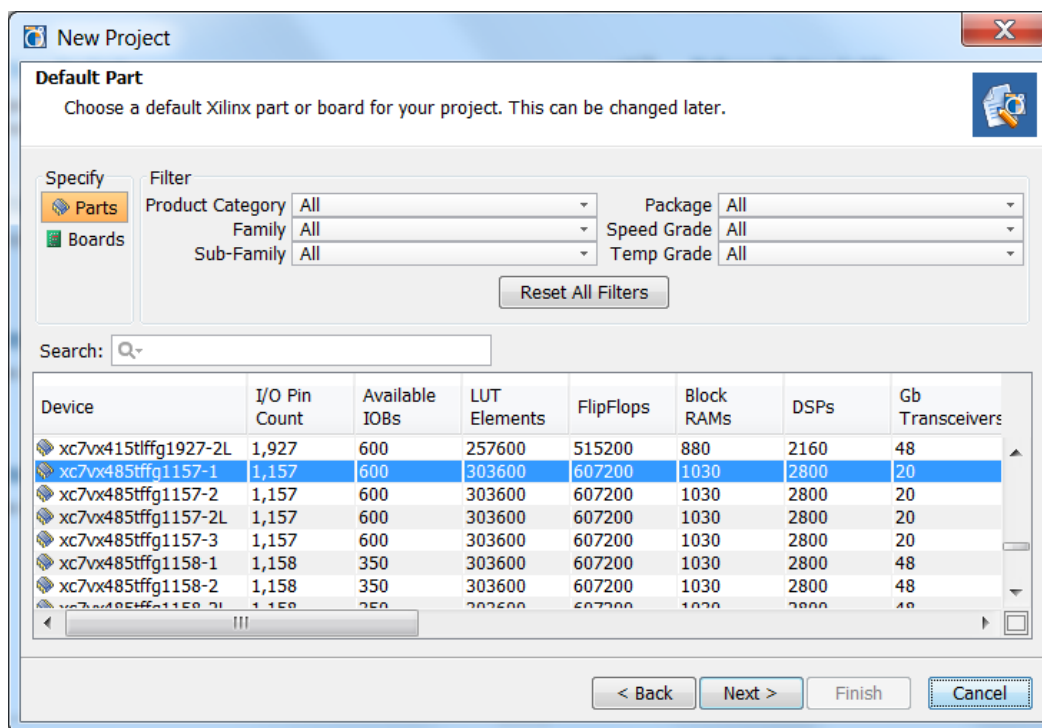


Figure 6: New Project: Default Part Dialog Window

Selecting a Default Part

14. In the Filter section, click the **Family** pull-down menu, and select **Kintex-7**. The list is filtered to only show Kintex®-7 devices
15. In the Search field type **70t**. The 70t devices are listed.

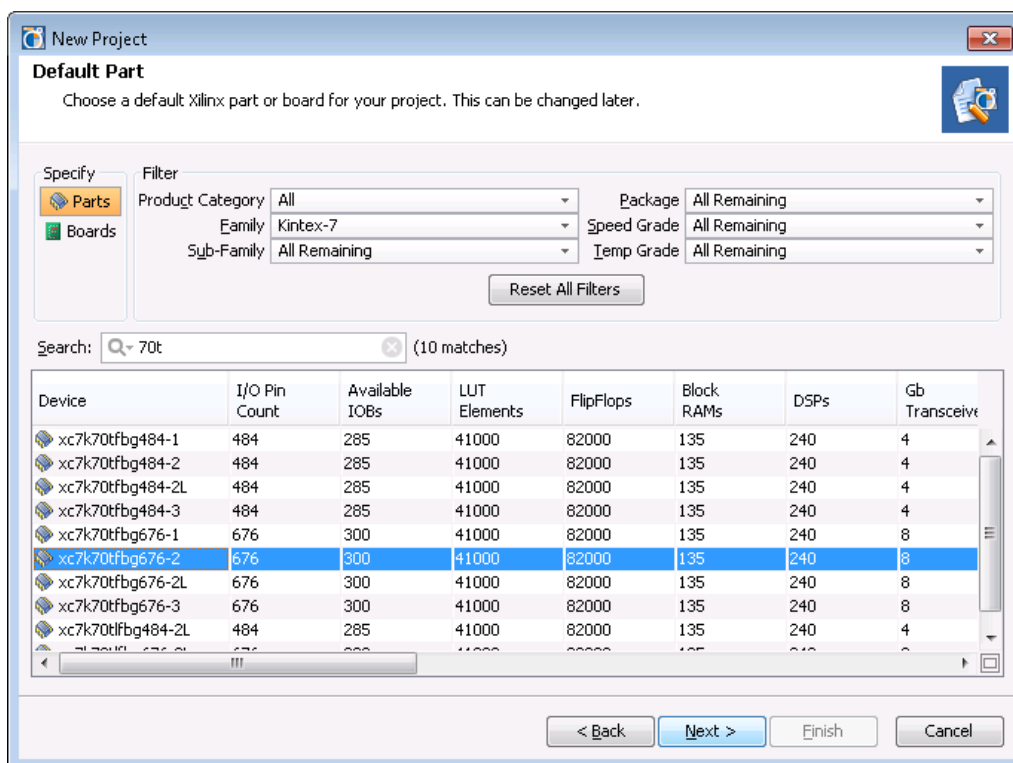


Figure 7: Selecting a Family and Default Part

16. Select **xc7k70tfbg676-2** and click **Next**.
17. Review the New Project Summary page, and click **Finish**.

The PlanAhead environment opens.

Step 2: Using the Sources View and the Text Editor

The PlanAhead tool 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. Use the supplied text editor to create or modify RTL sources.

Exploring the Sources View and Project Summary

1. Examine the information in the Project Summary. More information displays as the design progresses.
2. Examine the Sources view.

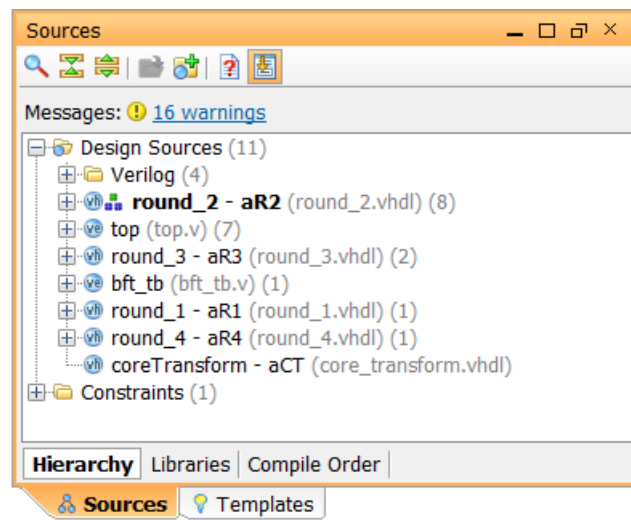


Figure 8: Viewing Sources Hierarchy

The source hierarchy is displayed by default. The tree does not show the `top` module as the top level of hierarchy, therefore there is an issue with the compilation of the design. The design has several VHDL files that have unique Library names. We need to set those in order for the design to compile and display the correct hierarchy.

Note: VHDL Library names can be set for individual files or for directories when using the Add Sources dialog box. When creating this project, we selected a single `hdl` directory which had a VHDL subdirectory.

Setting VHDL Library Names

3. Click the **Libraries** tab.
4. Expand the VHDL **Unreferenced** folder.
5. Use the Ctrl key to select `bft_package.vhdl`, `round_1-4.vhdl`, and `core_transform.vhdl`
6. Right-click to select **Set Library**

7. Enter **bftLib**.

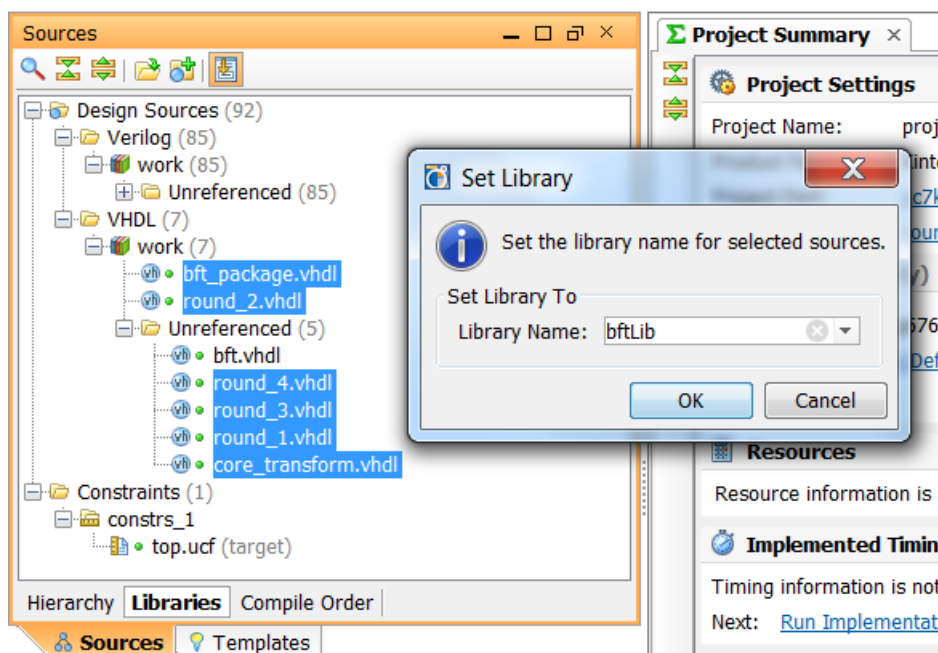


Figure 9: Setting VHDL Library Names

8. Click **OK**.

Verifying the bftLib VHDL Library

9. Verify that the bftLib folder has the following files shown under it:

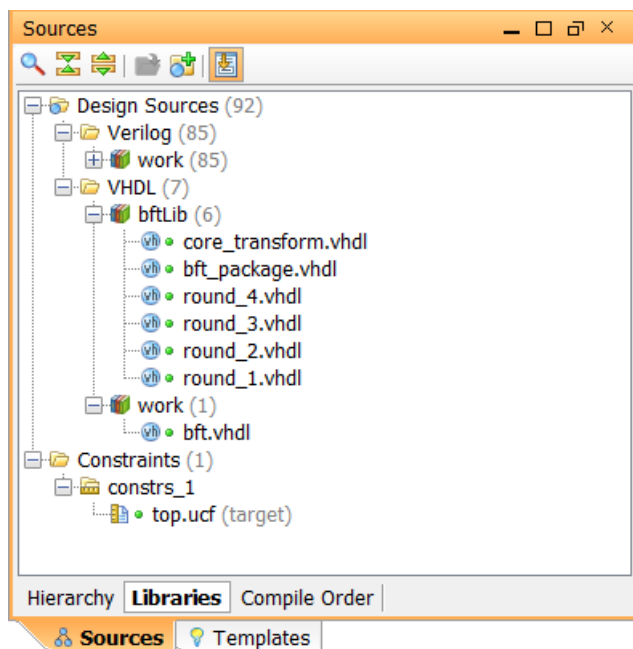



Figure 10: Verifying a VHDL Library

Configuring Simulation Source Files

10. In the Sources view, Libraries tab, expand the Verilog folder and the work subfolder.
11. Scroll to the bottom of the work list and expand the Unreferenced folder.
You should see the **bft_tb.v** source file. It is a simulation test bench and needs to be set to Simulation Only.
12. Select the bft_tb.v file and right-click to select **Move to Simulation Sources**.
13. Click the **Collapse All** button  in the Sources view header.
The file is now shown under the **Simulation-Only Sources** folder (it will become a reference file when it is set as the top file for simulation in a later step).
14. Expand the **sim_1** folder under Simulation-Only Sources.

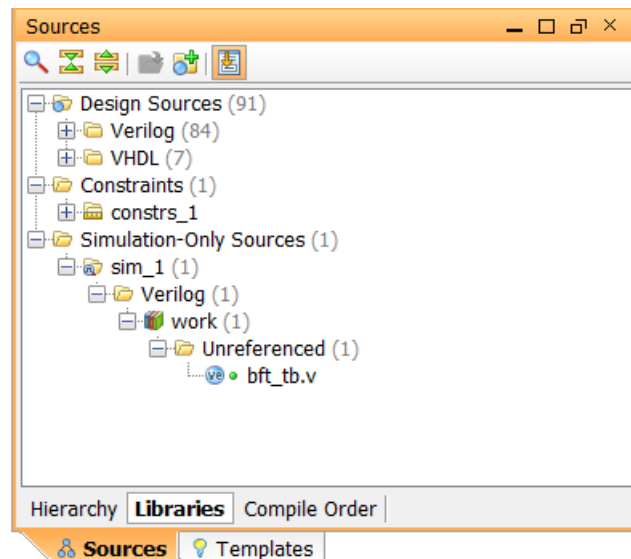


Figure 11: Viewing the Simulation-Only Sources

Verifying the Design Compilation Hierarchy

15. Click on the **Hierarchy** tab in the Sources view.
Verify that the design RTL hierarchy is now displayed properly, as shown below.
NOTE: The

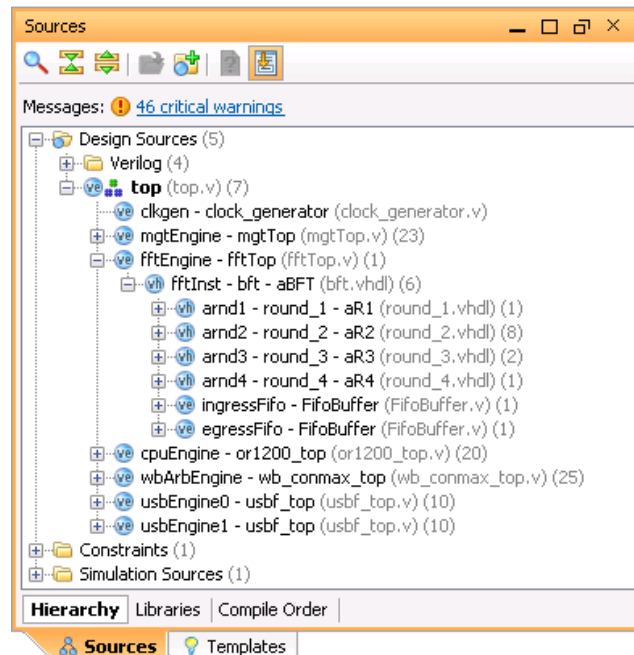



Figure 12: Viewing the RTL Design File Hierarchy

Defining Global Include Files

16. Click on the **Search** button  in the Sources view header.
17. Enter **timescale** in the Search field.
Notice that only files with that text appear.

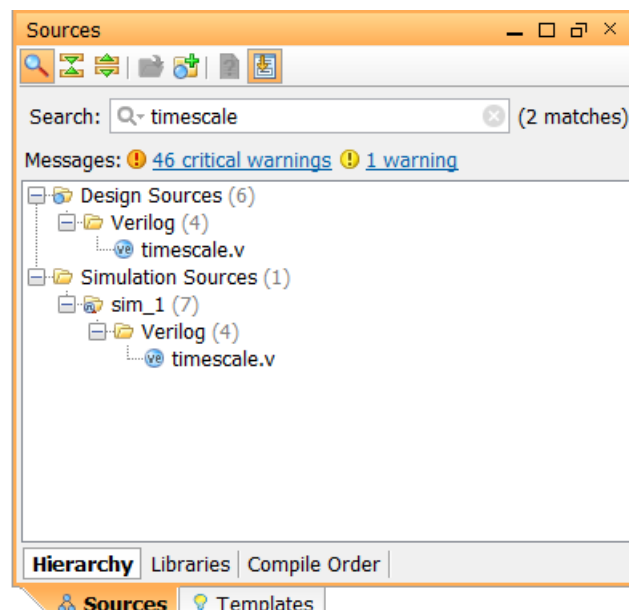



Figure 13: Viewing the RTL Design File Hierarchy

18. Select the Verilog timescale.v file and right click to select **Set Global Include**.
Notice that the file is now displayed under the new Global Include folder.
19. Click on the **Search** button  in the Sources view header to display all files.

Exploring the Sources View Commands

20. Expand and select one of the VHDL or Verilog files in the Sources view.
21. Right-click in the Sources view to review the available popup commands. To dismiss it, press the **Esc** key.

Using the Text Editor to View Source File Content

22. In the Sources view, double-click a source file to open it in the Text Editor.
23. Right-click in the Text Editor to view the available popup commands.
24. Close the Text Editor by clicking on the **X** in the view tab.

Creating a New RTL Source File and Importing a Template

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

25. In the Flow Navigator under Project Manager, select **Add Sources**.
26. In the Add Sources dialog box, select **Add or Create Design Sources**, and click **Next**.
27. Click **Create File...** button in the Add or Create Design Sources dialog box.

The Create Source File dialog box opens:

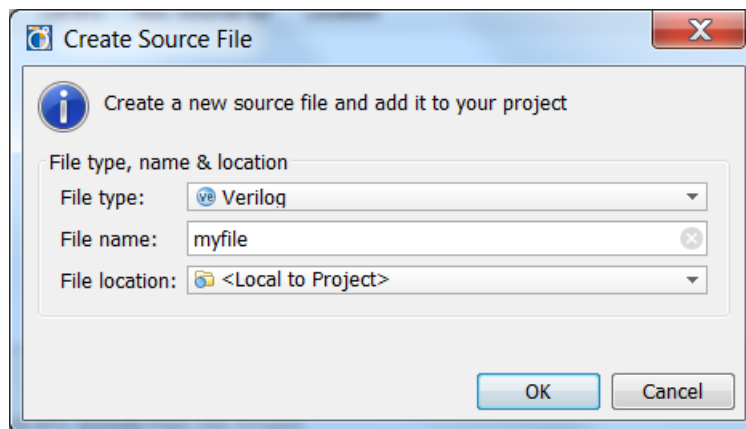


Figure 12: Create Source File Dialog Box

28. In the File name field, type **myfile**, and click **OK**.

29. In the **Add Sources** dialog box, click **Finish**.
The Define Modules dialog appears. This can be used to define a module and its ports. Click the **Cancel** button and **Yes** to confirm. The new empty file is now listed under the Non-module Files folder list in the Sources view.
30. In the Sources view, expand the **Non-module Files** folder and double-click **myfile.v** to open it in the Text Editor.
31. Click the **Templates** view tab (next to the Sources view).
32. Expand the **Verilog** subfolders to examine the types of templates available and select any Kintex-7 template.

The following figure shows the Verilog Templates folder.

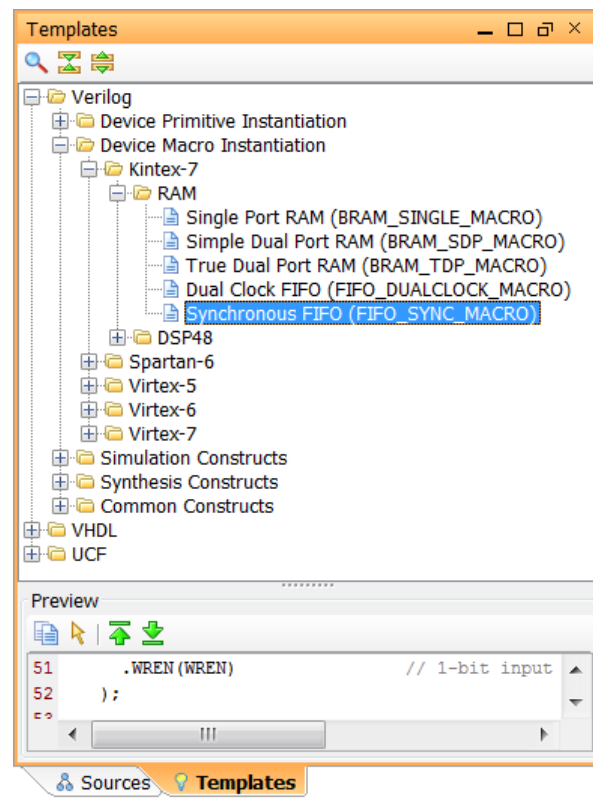


Figure 13: Examining the Templates in the Templates View

33. In the Text Editor, right-click and select **Insert Template**.
34. The template text is now inserted in the new source file.
35. Click the **X** button in the myfile.v view tab.
36. In the **Save Text Editor Changes** dialog box, click **No**.
37. Select the **Sources** view tab.
38. Click the **Collapse All** button in the Sources view.

Step 3: Running Behavioral Simulation

The Xilinx ISE Simulator (ISim) logic simulation environment is integrated with the PlanAhead tool. ISim can be used for behavioral or timing simulation. You can run behavioral logic simulation on the entire design, or an individual module.

Running Behavioral Simulation on the bft Module

1. In the Flow Navigator, select **Behavioral Simulation**.
2. Click the Simulation Top Module Name browser icon, and select **bft_tb**, and click **OK**.
3. Click **Launch**, and wait for ISim to open.

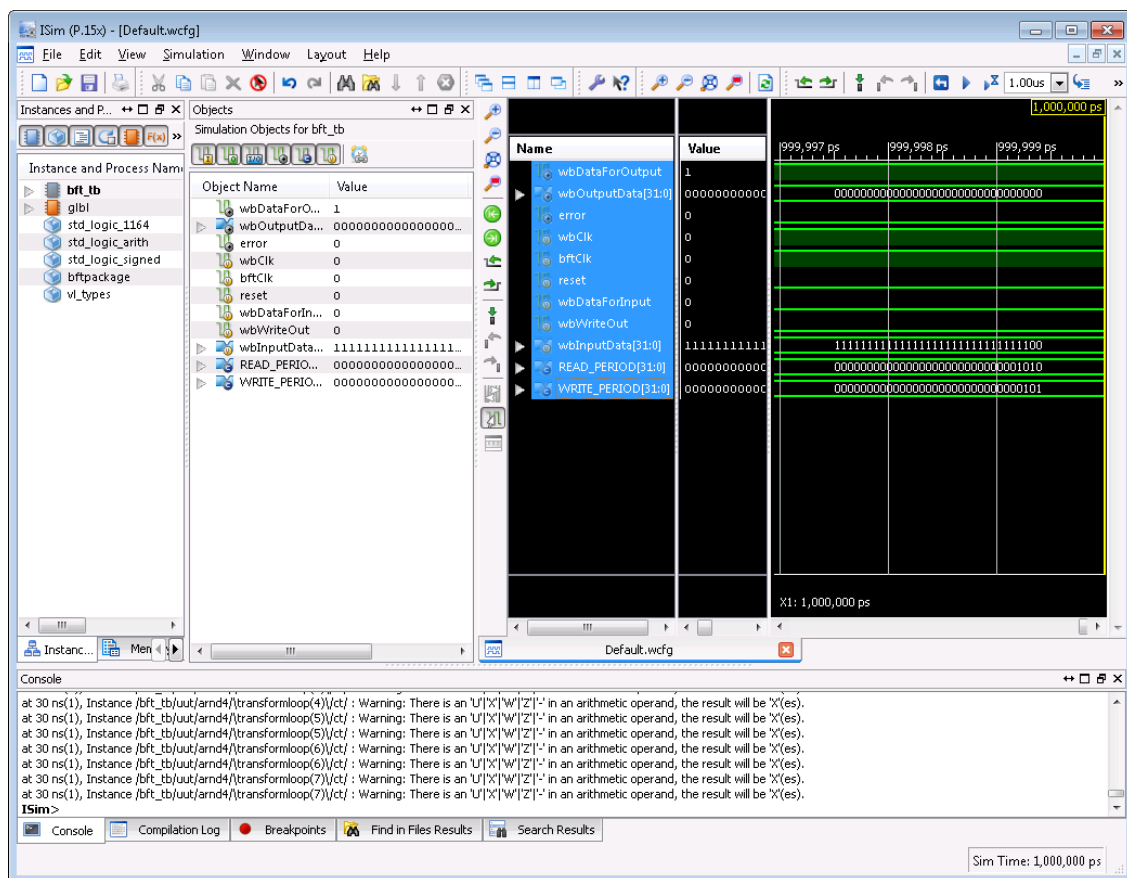


Figure 14: Launching ISim for Behavioral Simulation

4. Close the ISim window. Click **Yes** to confirm.

Setting Compilation Order and Disabling Unused Sources

The PlanAhead tool will automatically select a top module, order source files, and display source files based on compilation order. The top module can also be specified by the user. Files not needed in the design can be automatically or manually disabled.

5. In the Sources view, Hierarchy tab, expand **top** and select **mgtEngine**.

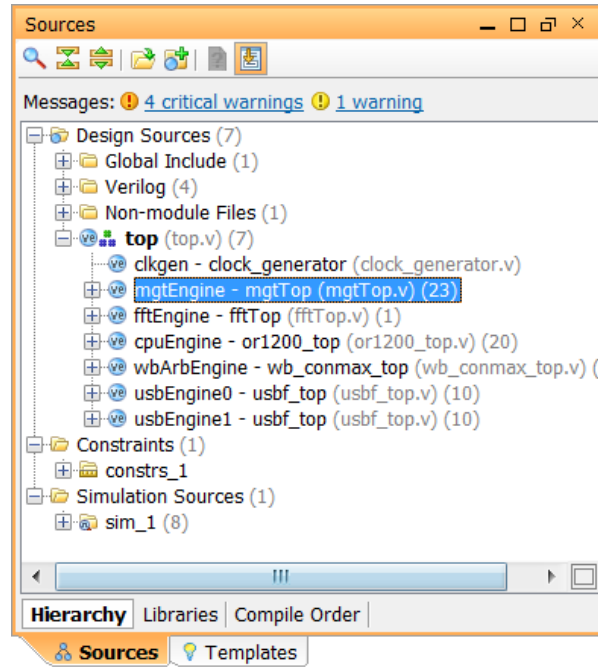


Figure 15: Specifying the Top Module in the Source Hierarchy View

6. Right-click, and select **Set as Top**
7. Select the Compile Order tab in the Sources view and see that the required source files and compile order have been updated based on the new top module. To manually enable or disable a file, select the file, right-click, and select **Enable File** or **Disable File**.

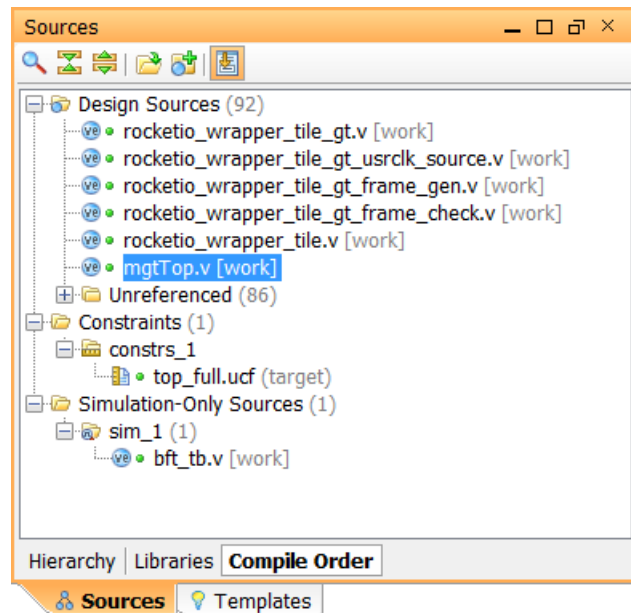


Figure 16: Viewing the Reordered Sources

8. Select the Hierarchy tab select **top**. Right-click, and select **Set as Top**.
9. The source file and compile order should be updated.
10. Right-click in the Sources view, select **Hierarchy Update**. See that three different options are available, ranging from fully automatic to completely manual file control.
11. Hit the **Esc** key to cancel the context menu.

Step 4: Elaborating and Analyzing the RTL Design

The PlanAhead tool provides RTL elaboration capabilities 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. Once elaborated, the RTL views enable cross-selection of logic objects. The RTL logic hierarchy is expanded and available for analysis. Opening the RTL Design from the Flow Navigator automatically elaborates the RTL design and displays the Design Analysis view layout.

- 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.

Elaborating and Opening the RTL Design

1. In the Flow Navigator, select **Open Elaborated Design** under RTL Analysis.
2. If prompted, click **OK** in the Warnings dialog.
The RTL Design is elaborated and opened.
3. Click the **Messages** view tab and click the **Collapse All** toolbar button
4. Expand to scroll through the **Elaborated Design** Warnings and Info messages.

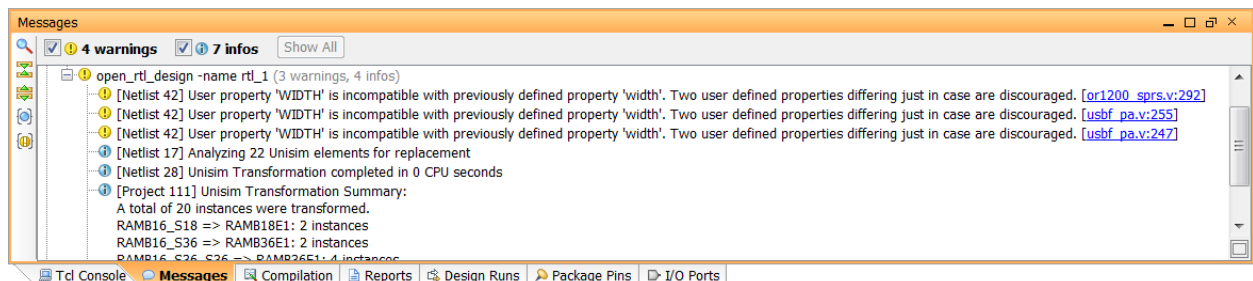


Figure 17: Viewing RTL Design Elaboration Messages

5. Click the checkbox, next to **7 infos** on the Messages view header to hide the Infos.
6. Review the Warning messages.
There are no Errors or Critical Warnings in the design. If there were, those categories would also appear in the Messages view header.
7. Click to enable the Info Messages again, or click the **Show All** button, in the Messages view header.

Examining the RTL Logical Netlist and Hierarchy

8. In the RTL Netlist view, expand the usbEngine0 instance by clicking the plus sign (+).
9. Select the **usbEngine0/u0** instance.
10. Right-click and select the **Go to Definition** popup command.

The RTL file `usb_g_utmi_if.v` opens in the Text Editor. This is the RTL code that defines the UTMI Interface module. The file opens to the line containing the `usb_g_utmi_if` module definition.

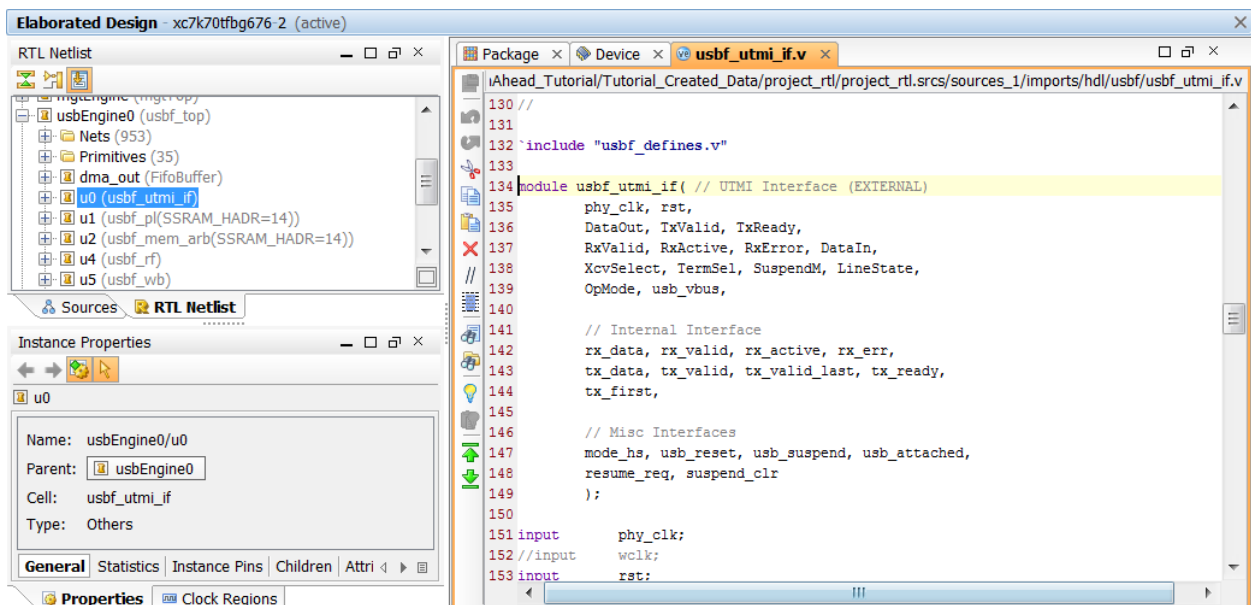


Figure 18: Viewing the Logical RTL Netlist

11. In the RTL Netlist view, right-click and select the **Go to Instantiation** popup command.
- The RTL file `usb_top.v` opens in the Text Editor. This is the file in which the UTMI Interface module is instantiated into the design. The file opens to the line containing the `usb_g_utmi_if` instance.
12. In the RTL Netlist view, right-click and select the **Show Hierarchy** popup command.

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

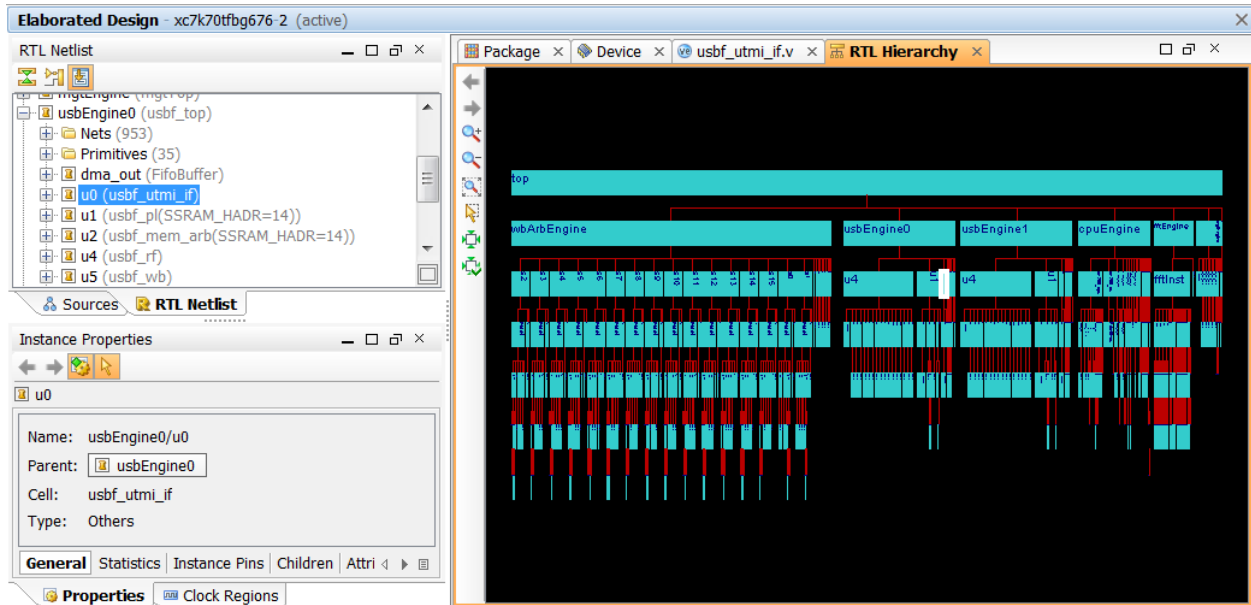


Figure 19: Displaying Modules in the RTL Hierarchy View

13. To close the RTL Hierarchy, click **X** on the view tab.
14. To close the Text Editor, click **X** on all open RTL files.

Examining the RTL Schematic

15. In the RTL Netlist view, expand and select the **usbEngine0/u0/u0** instance (the level below the previous selection).
16. In the RTL Netlist view, click the **Schematic** button or right-click and select the **Schematic** popup command.

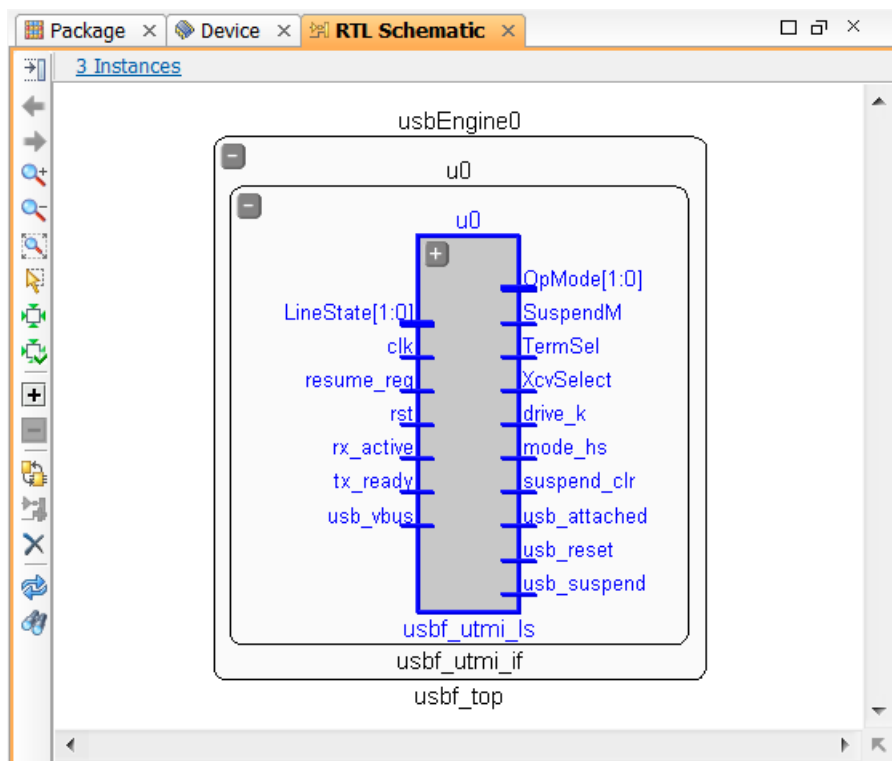


Figure 20: Viewing the RTL Schematic

17. Double-click the **LineState[1:0]** pin on the outside of the u0 module to expand the logic outward.
18. If needed, Zoom Fit the RTL Schematic view. The expanded logic is shown in the following figure.

Hint: Click and drag using the left mouse button in the RTL Schematic view from the lower right to the upper left to use the Zoom Fit command.

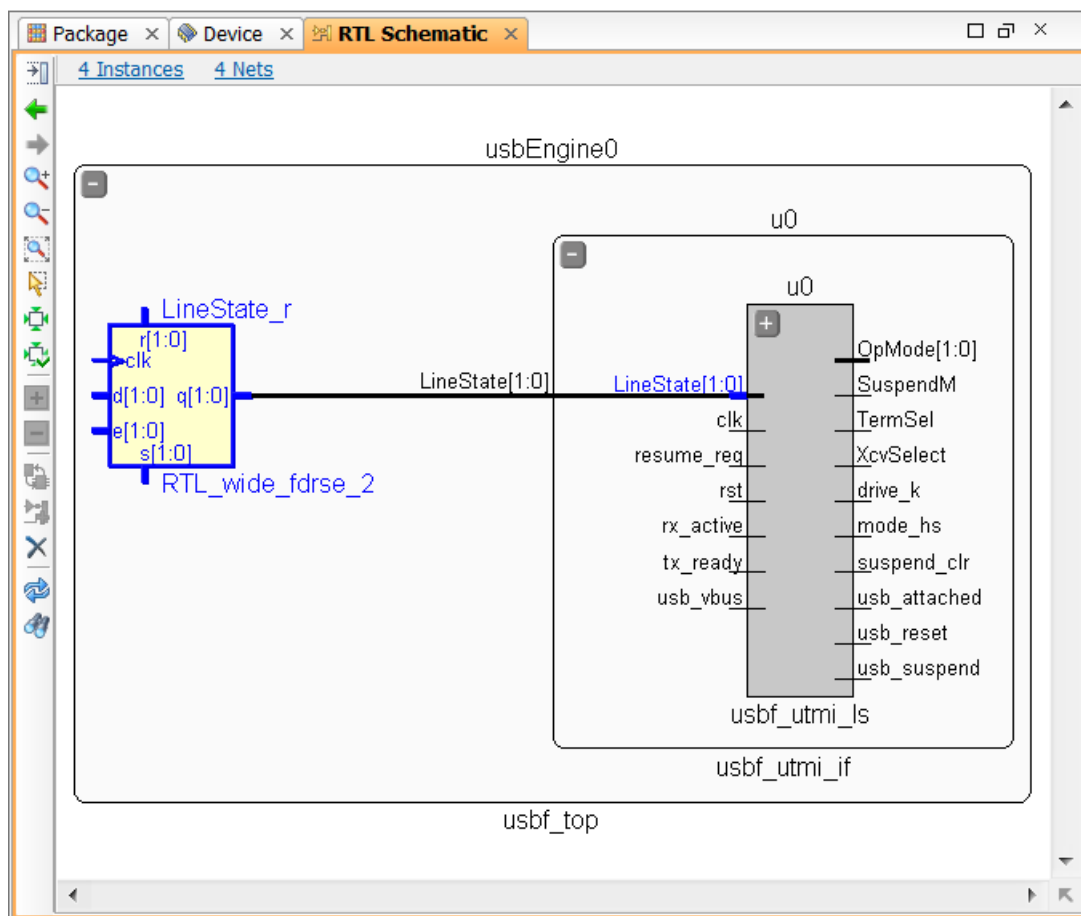




Figure 21: Expanding Logic in the RTL Schematic View

For additional schematic exploration capabilities, see the *Design Analysis and Floorplanning Tutorial: PlanAhead Design Tool (UG676)*.

19. Select the **LineState_r** instance on the left side of the RTL Schematic view.
20. In the RTL Schematic view, right-click and select the **Go to Instantiation** popup command to display the RTL file with the logic definition.
21. Close the Text Editor and the RTL Schematic.
22. In the RTL Netlist view, click the **Collapse All** button .

Using the Find Command to Locate RTL Block RAM Logic

23. Click the Find button in the main toolbar , or select **Edit > Find** to open the Find dialog box.

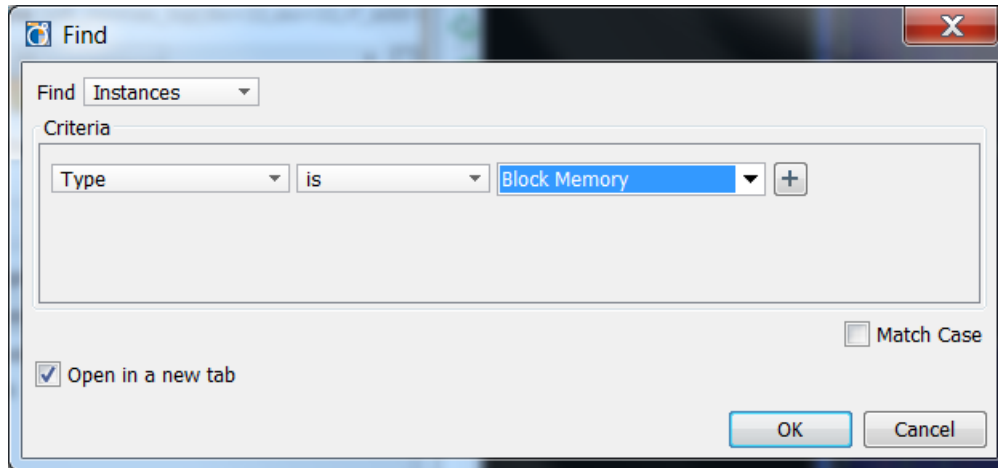


Figure 22: Searching for RTL Logic Using the Find Dialog Box

24. Examine the Find filter options.
25. Set the Criteria to **Type > is > Block Memory**, and click **OK**.

The Find Results view opens.

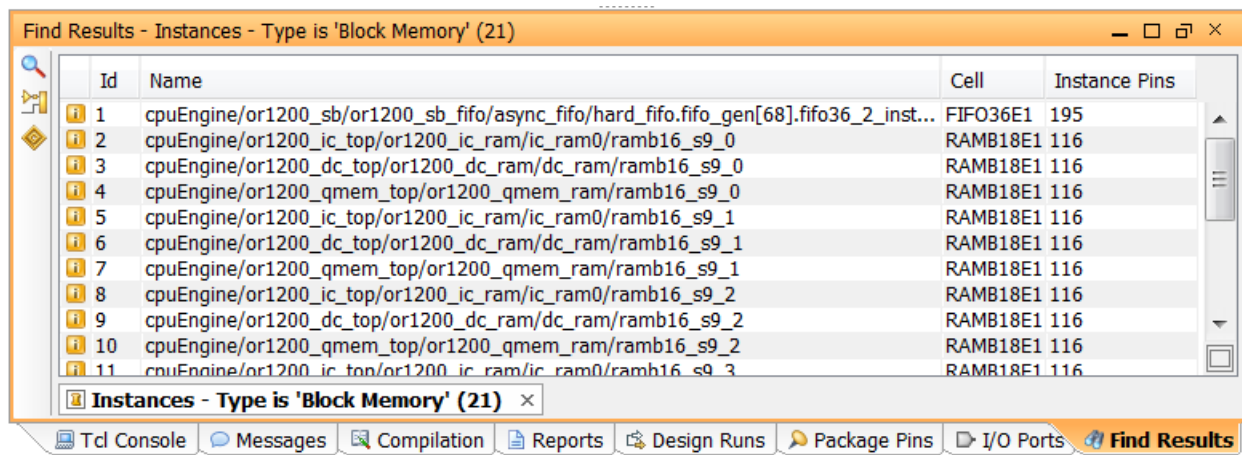


Figure 23: Find Results for RTL Block RAM Search

The Find Results view displays the results of the find operation.

26. Select one of the Block RAMs in the list; right-click and select **Go to Instantiation**.
The instance is selected in the RTL Netlist view and displayed in the Text Editor.
27. Close the Find Results view and the Text Editor.

Step 5: Estimating Resource Utilization

Examining the Resource Estimation Options

28. In the Flow Navigator, select the **Report Utilization** command.
The Resource Estimation view opens.

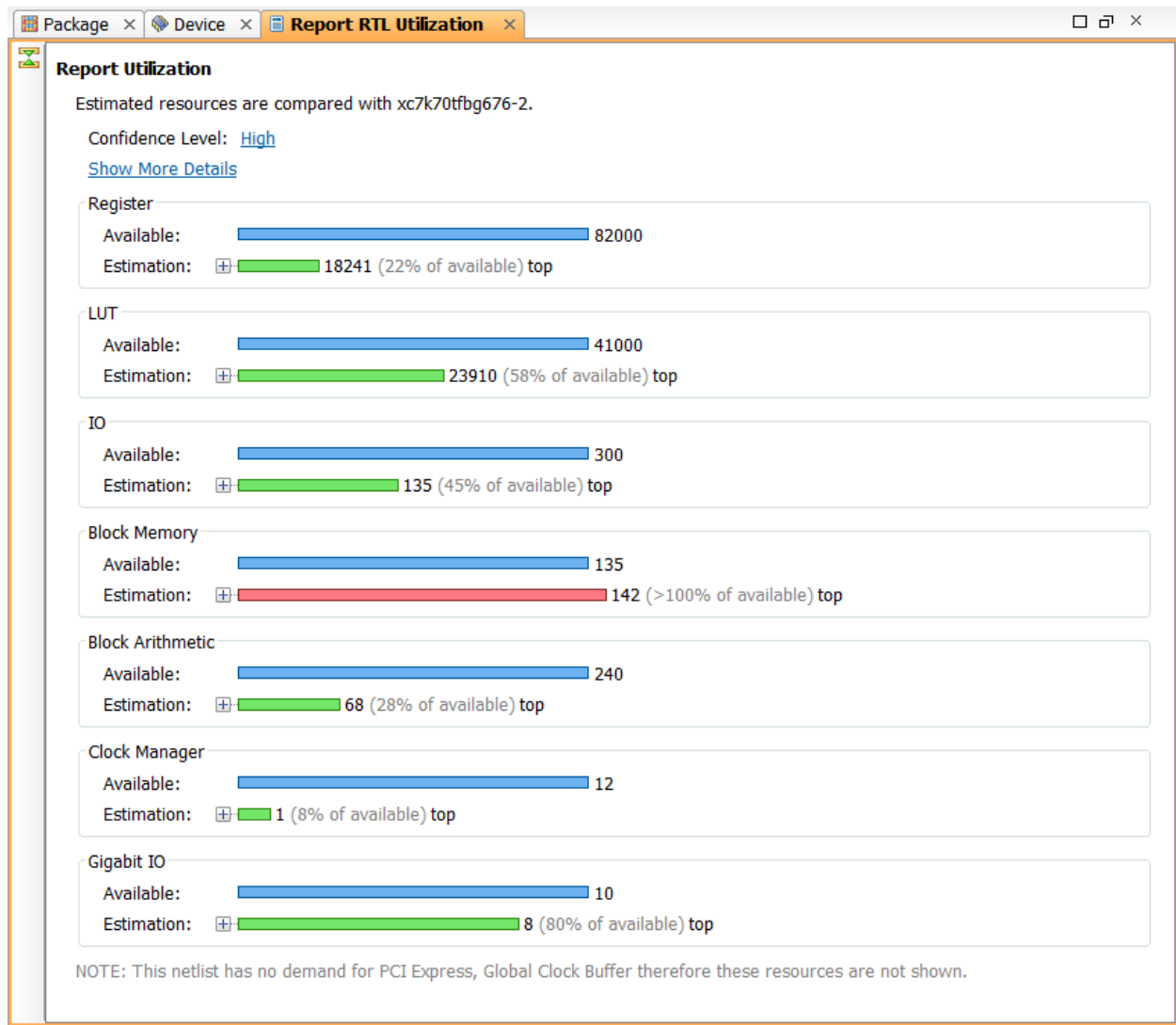


Figure 24: Viewing the RTL Resource Estimation

29. Expand the Block Memory Estimation tree to explore the hierarchical chart report.
Note: The pre-synthesis report shows the Block Memory over utilized in red. Synthesis can often pack Block Memory in LUTs, so we can ignore this for now.
30. Close the Report RTL Utilization view.

Examining Resource Estimates for RTL Instances

31. In the RTL Netlist view, select **top**.
32. In the Netlist Properties view, ensure that the **Statistics** tab is selected.

The RTL Macro Resources displays in the Netlist Properties view, as shown in the following figure.

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

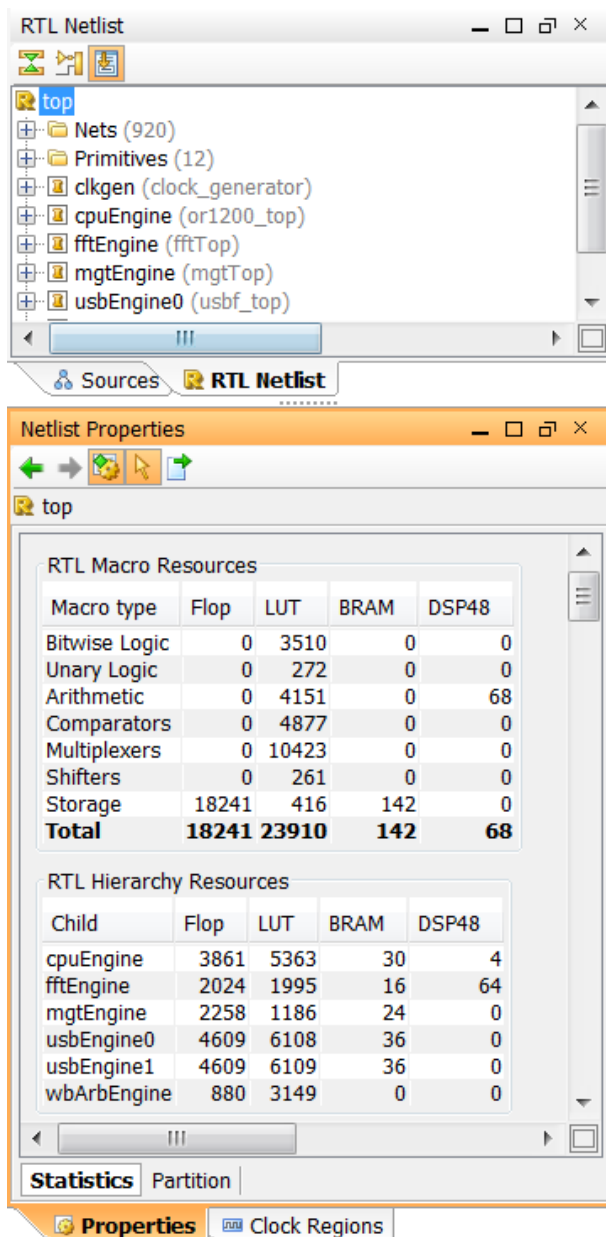


Figure 25: Viewing the RTL Resource Estimates

33. Scroll down the Netlist Properties.

34. Examine the properties, including:
 - RTL Macro Resources
 - RTL Hierarchy Resources
 - RTL Primitive Statistics
 - RTL Memory Resources
 - Net Boundary Statistics
 - Clock Report
35. In the RTL Netlist view, select other modules and examine the same estimates for the selected module.

Step 6: Running RTL Design Rule Checks

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

Reporting DRCs

1. From the Flow Navigator or the **Tools** menu, select **Report DRC**.
2. In the **Run DRC** dialog box, expand and examine the RTL rules, and click **OK**.

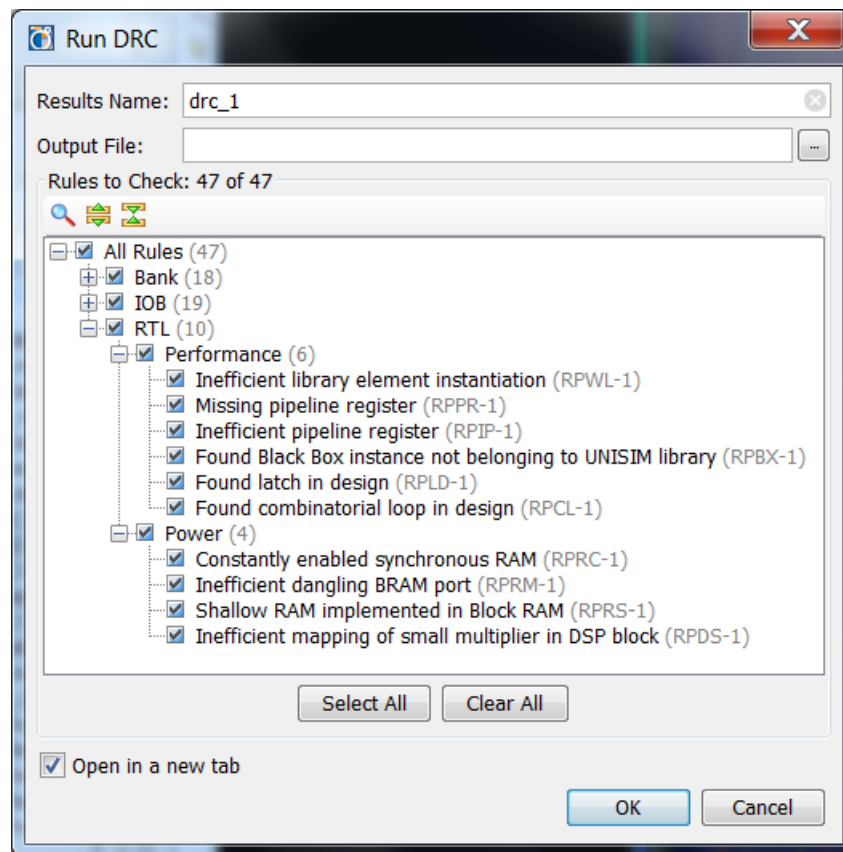


Figure 28: Running RTL DRCs

The DRC Results view opens.

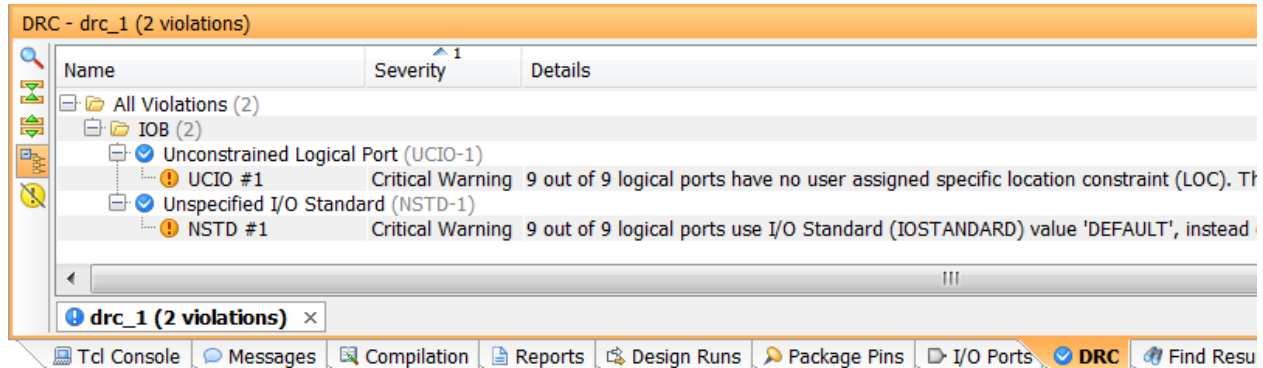


Figure 29: Viewing the RTL DRC Results

The DRC Results viewer color-codes messages as follows:

- Errors with a red icon
 - Critical Warnings with an orange icon
 - Warnings with a yellow icon
 - Informational messages with a blue icon
3. Select the **UCIO #1** latch warning in the violations list.
The Violation Properties view displays with information about the violation and links to select the offending logic objects.
 4. Close the DRC Results view.
 5. Close the RTL Design by clicking the Close **X** icon in the Elaborated Design banner. Click **OK** in the confirmation dialog box if needed.

Step 7: Selecting IP from the Xilinx IP Catalog

The PlanAhead tool is integrated with the CORE Generator™ tool to provide an IP Catalog with search and filtering capabilities. This allows you to easily find the desired IP. You can customize, instantiate and implement the core directly from the PlanAhead tool. You can access the IP Catalog from the Project Manager and Elaborated Design environments.

Opening the IP Catalog and Exploring the Search Options

1. Select **IP Catalog** from the Flow Navigator.
2. Expand some of the IP categories.
3. Select any IP and explore the available toolbar buttons and popup menu commands.

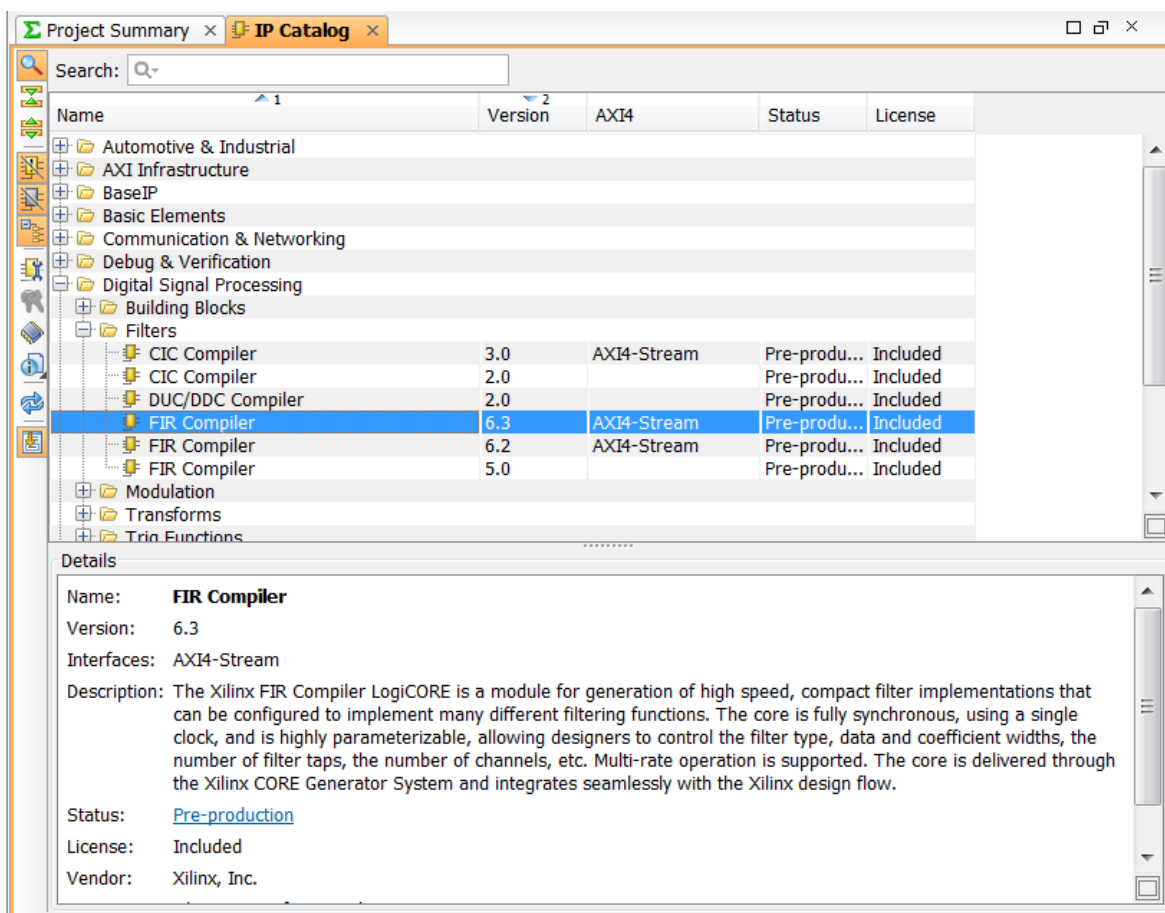






Figure 30: Browsing the IP Catalog

The Details for the selected IP display at the bottom of the view. By default, only the IP that is current and available for the selected device family is displayed.

4. To view all IP, toggle the Hide Superseded and Discontinued IPs  button and the Hide Incompatible IPs button .
5. To view a flattened list of IP, toggle the **Group by Category** toolbar button .
6. Type **fir** in the Search field at the top of the view.
7. Select a FIR Compiler IP, right-click to select **Data Sheet**.
8. Examine the Data Sheet, and then close the PDF viewer.
9. Clear the Search field to expand the Catalog list.

Step 8: Customizing and Instantiating IP

Customizing a Simple Adder IP

1. Click the **Group by Category** button .
2. Click the **Collapse All** button.
3. Expand the **Math Functions > Adders & Subtractors** folder.
4. Double-click the top **Adder Subtractor** to run the Customize IP command.

This opens the CORE Generator™ tool and displays the customized interface for the selected IP. Different IP have different types of interfaces.

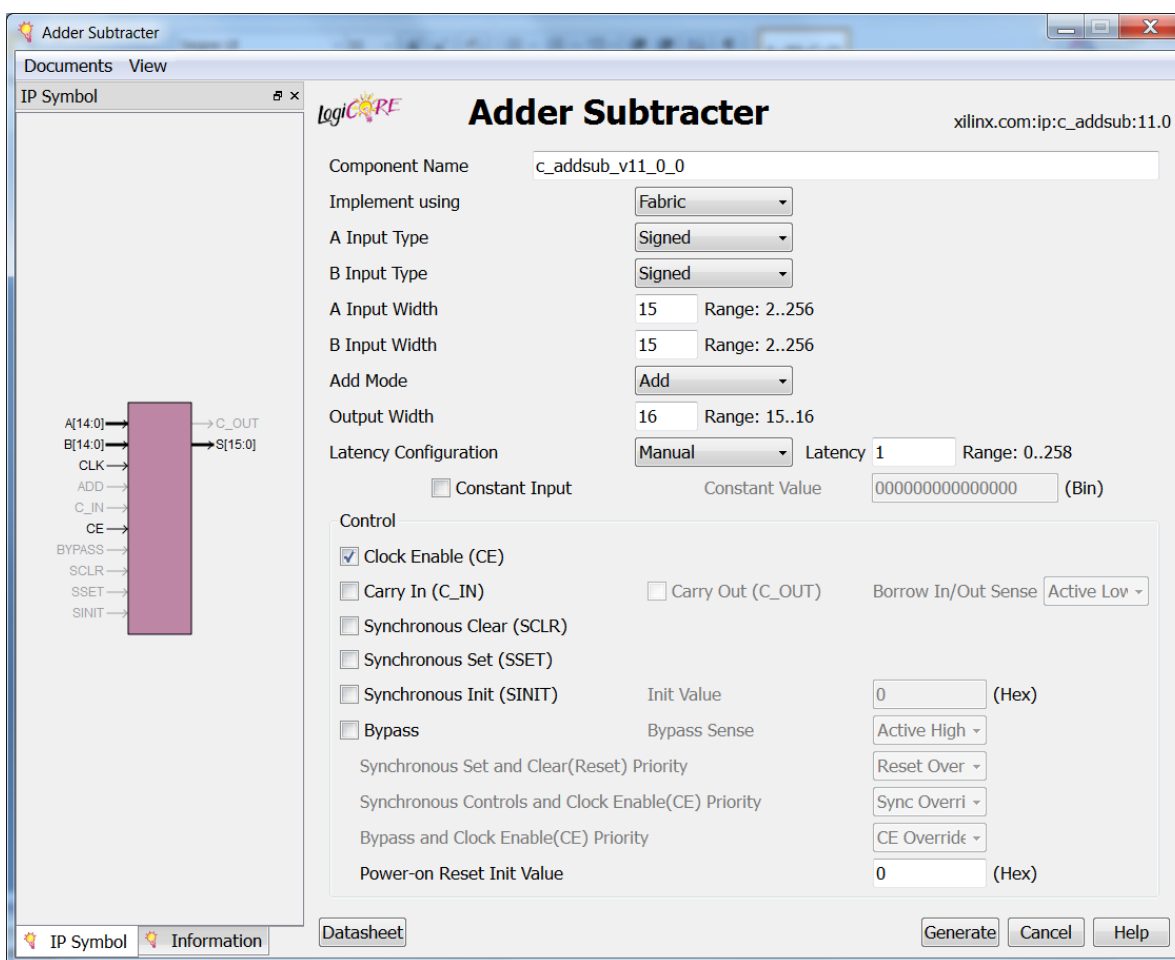


Figure 31: Customizing IP With the CORE Generator Software

5. In the B Input Width field, type **18**.
6. Click Generate.

Clicking Generate has a different effect when launched from PlanAhead than when run from CORE Generator standalone.

- In standalone mode, the CORE Generator™ software automatically launches XST to synthesize the IP core.
- When launched from the PlanAhead tool, the synthesis step is not run automatically, which lets you instantiate and configure the core in your RTL before launching synthesis. You can synthesize the IP at any time or launch synthesis on the entire design for which the IP is synthesized first.

Instantiating the Adder IP

7. In the Sources view, select the **IP Sources** tab and expand the **c_addsub_v11_0_0** IP.
8. Expand the **instantiation_template** folder and double-click the **c_addsub_v11_0_0.v** file to view the instantiation template in the Text Editor.

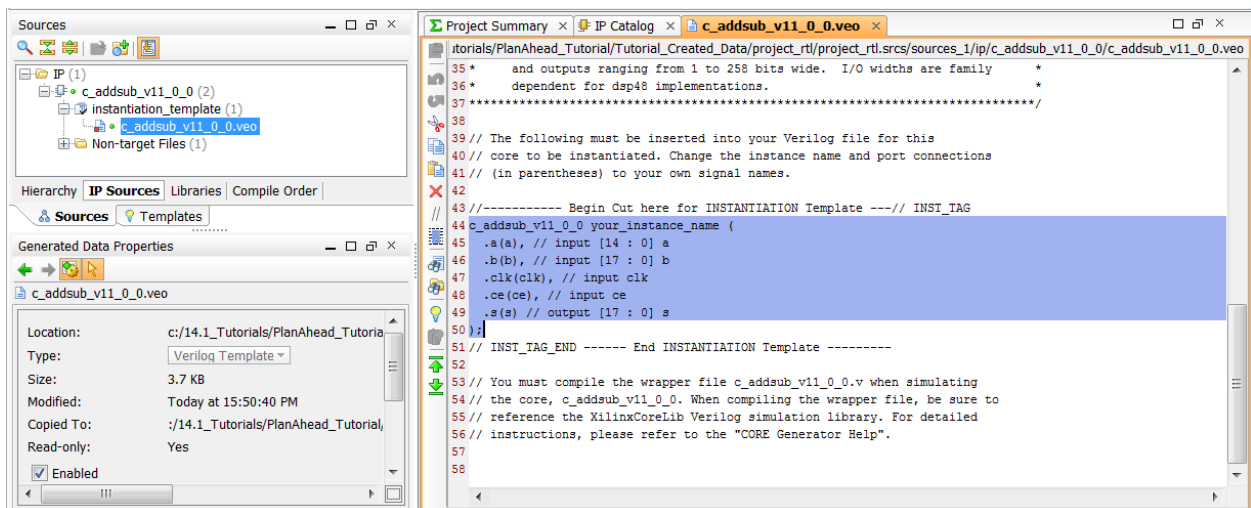


Figure 42: Viewing the Instantiated Template

9. Select the text in the Text Editor, as shown above, and click the **Copy Text** button.
The copied text can easily be pasted into any RTL source file to instantiate the IP. Once done, the IP can easily be generated through synthesis by selecting the Generate IP popup command.
10. To close the **.veo** template file in the Text Editor.
11. Close the IP Catalog.
12. 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.
- Started by creating an RTL project, explored RTL sources and the RTL editor.
- Ran behavioral simulation, 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 running RTL DRCs.
- Examined the Xilinx IP Catalog, and customized, and instantiated a small adder IP core.