See all versions of this document

Vivado Design Suite Tutorial

Power Analysis and Optimization

UG997 (v2021.2) January 19, 2022

XILINX

Revision History

01-19-2022: Released with Vivado® Design Suite 2021.2 without changes from 2021.1.

Section	Revision Summary
06/30/2021 Version 2021.1	
General updates	Editorial updates only. No technical content updates.

Table of Contents

Lab 1

Power Analysis and Optimization Tutorial

This tutorial introduces the power analysis and optimization use model recommended for use with the Xilinx® Vivado® Integrated Design Environment (IDE). The tutorial describes the basic steps involved in taking a small example design from RTL to implementation, estimating power through the different stages, and using simulation data to enhance the accuracy of the power analysis. It also describes the steps involved in using the power optimization tools in the design.

VIDEO: *The Vivado Design Suite Quick Take Video: Power Estimation and Analysis Using Vivado shows how the Vivado Design Suite can help you to estimate power consumption in your design and reviews best practices for getting the most accurate estimation.*

VIDEO: *The Vivado Design Suite QuickTake Video: Power Optimization Using Vivado describes the factors that affect power consumption in an FPGA, shows how the Vivado Design Suite helps to minimize power consumption in your design, and looks at some advanced control and best practices for getting the most out of Vivado power optimization.*

Software Requirements

This tutorial requires the latest Vivado Design Suite software is installed. For installation instructions and information, see the *Vivado Design Suite User Guide: Release Notes, Installation, and Licensing* (UG973).

For hardware power measurement of 7 series devices, the tutorial requires Texas Instruments Fusion Design Power Designer software, which can be downloaded from the following location: http://www.ti.com/tool/fusion_digital_power_designer

For hardware power measurement of UltraScale™ devices, the tutorial requires Maxim Digital Power Tool software, which can be downloaded from the following location:

https://www.maximintegrated.com/en/products/power/switching-regulators/MAXPOWERTOOL002.html

Hardware Requirements

Supported operating systems to run the Vivado Design Suite, and memory recommendations when using the Vivado tools, are described in the *Vivado Design Suite User Guide: Release Notes, Installation, and Licensing* (UG973).

Hardware Requirements for 7 Series Devices

- The hardware power measurements for 7 series devices (needed in Lab 4: Measuring Hardware Power Using the KC705 Evaluation Board), require a Xilinx Kintex®-7 FPGA KC705 Evaluation Kit. You can find information on the Evaluation Kit at this location: Xilinx Kintex®-7 FPGA KC705 Evaluation Kit
- For power measurements through TI Power Regulators (needed in Lab 4: Measuring Hardware Power Using the KC705 Evaluation Board), use the Texas Instruments USB Interface Adapter. You can find information on the USB Interface Adapter at this location:

  www.ti.com/lit/ml/sllu093/sllu093.pdf

Hardware Requirements for UltraScale Devices

- The hardware power measurements in UltraScale devices (needed in Lab 5: Measuring Hardware Power Using the KCU105 Evaluation Board), requires a Xilinx Kintex® UltraScale™ FPGA KCU105 Evaluation Kit. You can find information on the Evaluation Kit at the following location: Xilinx Kintex UltraScale FPGA KCU105 Evaluation Kit
- For power measurements through Maxim Digital Power Tool (needed in Lab 5: Measuring Hardware Power Using the KCU105 Evaluation Board), use the Maxim Power interface adapter. You can find information on the interface adapter at the following location: https://www.maximintegrated.com/en/products/power/switching-regulators/MAXPOWERTOOL002.html

Locating Tutorial Design Files

1. Download the reference design files from the Xilinx website:

   ug997-vivado-power-analysis-optimization-tutorial.zip

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

   This tutorial refers to the location of the extracted ug997-vivado-power-analysis-optimization-tutorial.zip file contents as `<Extract_Dir>`.

IMPORTANT! *You will modify the tutorial design data while working through this tutorial. Use a new copy of the original data each time you start this tutorial.*

The `ug997-vivado-power-analysis-optimization-tutorial.zip` file includes a readme file which contains the details and version history of the design files along with the folders of 7 series and UltraScale design files.

7 Series Tutorial Design Files

You can find a separate 7 series folder containing the 7 series tutorial design files in the contents of the zip file.

The following table describes the contents of the 7 series tutorial design files:

Directories/Files	Description
`/src`	Contains the design HDL and testbench for the functional simulation.
`/src/dut_fpga.v`	Top module for the design.

`/src/bram_tdp.v`
`/src/bram_top.v`
`/src/dut.v`
	Other design blocks - synthesized module.

`dut_fpga_kc705.xdc`	Contains clocking and timing constraints for the design.
`/src/testbench.v`	Testbench for simulating the design.

UltraScale Device Tutorial Design Files

You can find a separate UltraScale™ folder containing the UltraScale device tutorial design files in the contents of the zip file.

The following table describes the contents of the UltraScale device tutorial design files:

Table 1: **Example table**

Directories/Files	Description
`/src`	Contains the design HDL and testbench for the simulation.
`/src/dut_fpga.v`	Top module for the design.

`/src/dut.v`
`/src/Cascade_bram.v`
`/src/Noncascade_bram.v`
`/src/bram_top_cascade.v`
`/src/bram_top_noncascade.v`
`/src/bram_tdp_cas.v`
`/src/bram_tdp_noncas.v`
	Other design blocks.

`dut_fpga_kcu105.xdc`	Contains clocking and timing constraints for the design.
`/src/testbench.v`	Testbench for simulating the design.

Lab 2

Running Power Analysis in the Vivado Tools

Introduction

In this lab, you will learn about the Power Analysis and Optimization features in the Vivado® IDE. The lab will take you through the steps of project creation and power analysis at the synthesis stage, using the Vivado Report Power feature in vectorless mode. It will also demonstrate using the SAIF file generated from behavioral simulation for Vivado report power analysis.

You will analyze power in the Vivado IDE. Then you will examine some of the major features in the Power window and closely examine some power specific Tcl commands. You will also learn to create a Switching Activity Interchange Format (SAIF) file by simulating the design in the timing simulation stage using both the Vivado simulator and Questa Advanced Simulator.

You will also learn how to achieve Power Optimization after `opt_design` in the Vivado IDE. You will examine the power optimization report and selectively turn power optimizations ON or OFF on specific signals, nets, modules, or hierarchy.

Step 1: Creating a New Project

To create a project, use the New Project wizard to name the project, to add RTL source files and constraints, and to specify the target device.

Note: Throughout this tutorial, Xilinx® 7 series example design is used to explain the process of configuring, implementing, estimating the power through different stages, and using simulation data to enhance the accuracy of the power analysis. For UltraScale™ device design, most of the steps are similar to 7 series. Additional information, wherever necessary, is provided for UltraScale devices.

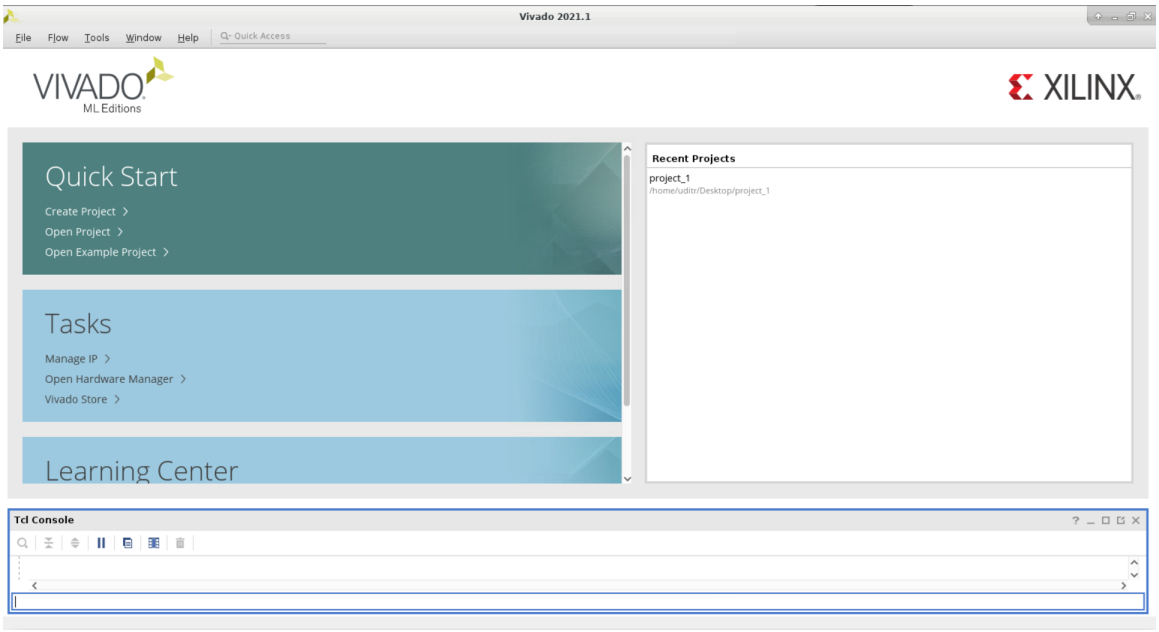
On Linux, do the following.

1. Go to the directory where the lab materials are stored:

   `cd <Extract_Dir>/7_series` (for 7 series devices) or

   `cd <Extract_Dir>/UltraScale` (for UltraScale devices)

2. Launch the Vivado IDE: `vivado`



   On Windows, do the following.

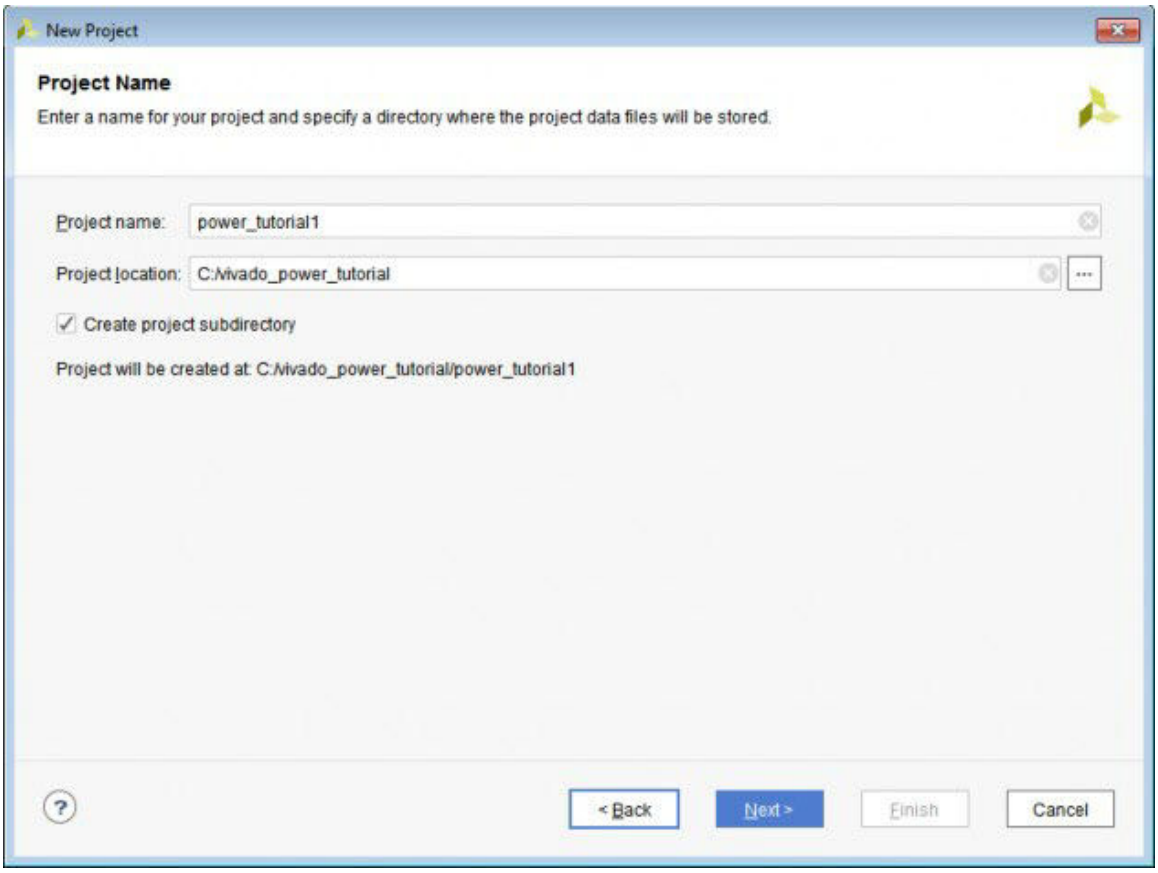
3. Launch the Vivado IDE by selecting **Start → All Programs → Xilinx Design Tools → Vivado 2021.x → Vivado 2021.x** (x denotes the latest version of Vivado 2021 IDE).

   As an alternative, click the Vivado 2021.x Desktop icon to start the Vivado IDE.

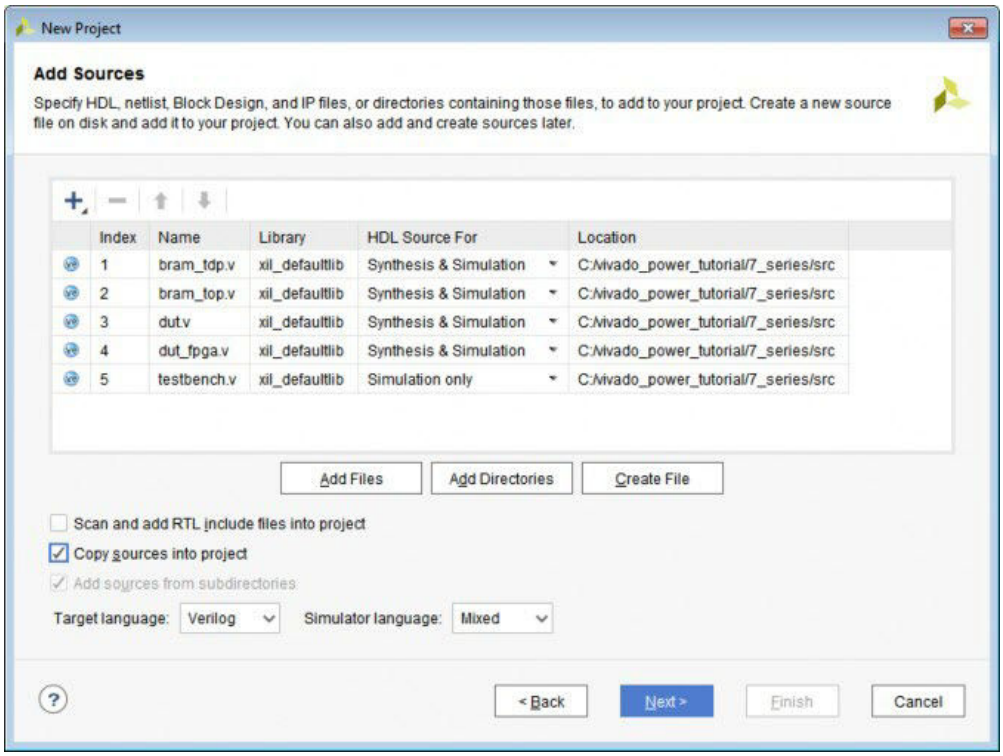
   The Vivado IDE Getting Started page contains links to open or create projects and to view documentation.

4. In the Getting Started page, click **Create New Project** to start the New Project wizard.

5. Click **Next** to continue to the next screen.



6. In the Project Name page, name the new project power_tutorial1 and enter the project location (`C:\Vivado_Power_Tutorial`). Make sure to check the **Create project subdirectory** option and click **Next**.
7. In the Project Type page, specify the type of project to create as **RTL Project**, make sure to uncheck the **Do not specify sources at this time** option, and click **Next**.
8. In the Add Sources page:
   a. Set Target Language to **Verilog** and Simulator language to **Mixed**.
   b. Click the **Add Files** button.
   c. In the Add Source Files dialog box, navigate to the `<Extract_Dir>/7_series/src` directory for 7 series devices or `<Extract_Dir>/UltraScale/src` for UltraScale devices.
   d. Select all of the Verilog (`.v`) source files, and click **OK**.
   e. In the Add Sources page, change the HDL Source For the `testbench.v` file to **Simulation only**.



f. Verify that the files are added and **Copy sources into project** is checked. Click **Next**.

9. In the Add Constraints (optional) page, click **Add Files** and select `dut_fpga_kc705.xdc` in the file browser. In the directory structure, you will find the `dut_fpga_kc705.xdc` file below the `/src` folder.

   For UltraScale devices, select `dut_fpga_kcu105.xdc` in the file browser. In the directory structure, you will find the `dut_fpga_kcu105.xdc` file below the `/src` folder.

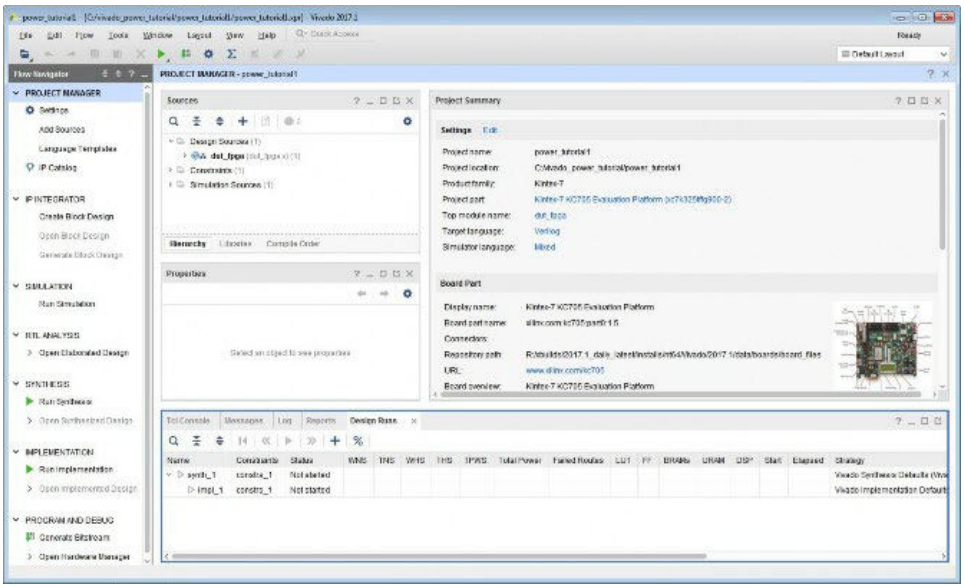
10. Click **Next** to continue.

11. In the Default Part page, click **Boards** and select Kintex-7 KC705 Evaluation Platform for 7 series or Kintex UltraScale KCU105 Evaluation Platform for UltraScale devices. Then click **Next**.

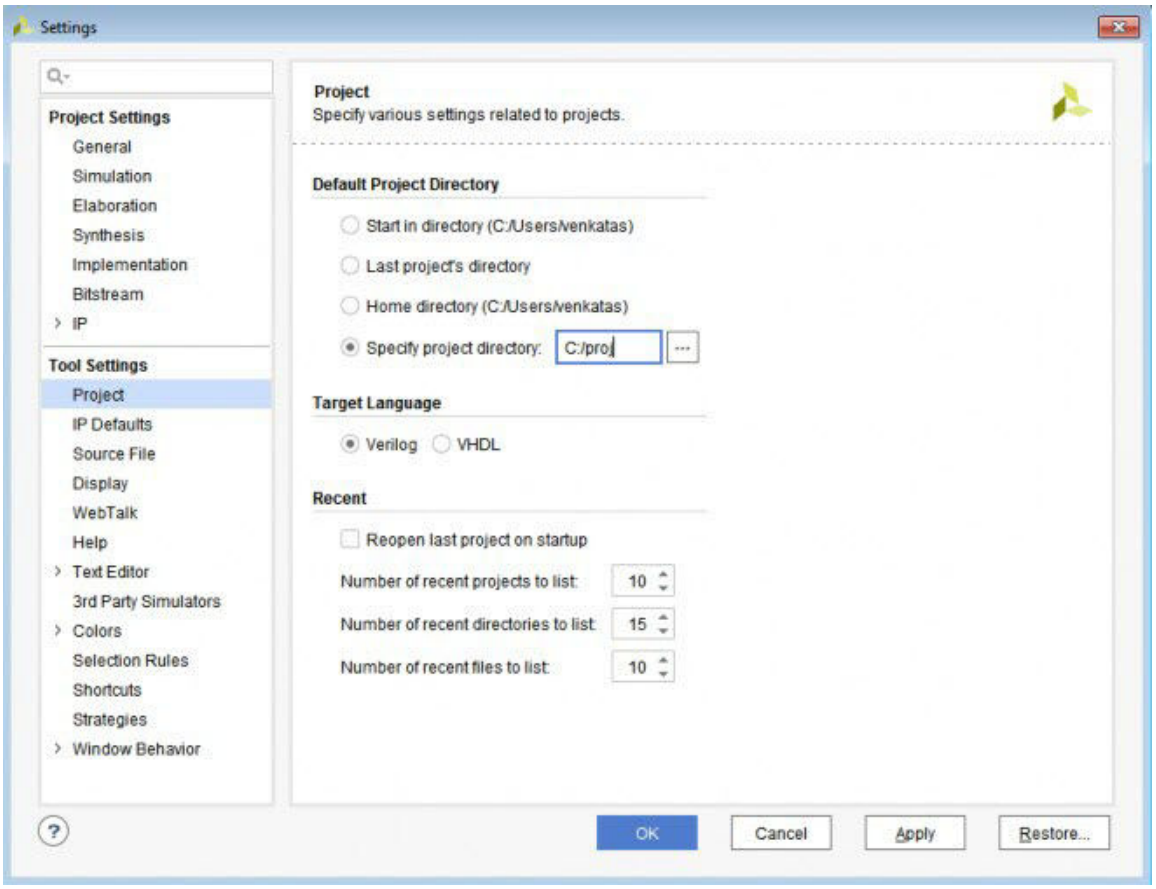
> **TIP:** *When you specify a board, you are also specifying the part you are targeting for your design, in this case an xc7k325tffg900-2 FPGA for 7 series or xcku040-ffva156-2-e FPGA for UltraScale devices.*

12. Review the New Project Summary page. Verify that the data appears as expected, per the steps above, and click **Finish**.

    Note: It might take a moment for the project to initialize in the Vivado IDE.



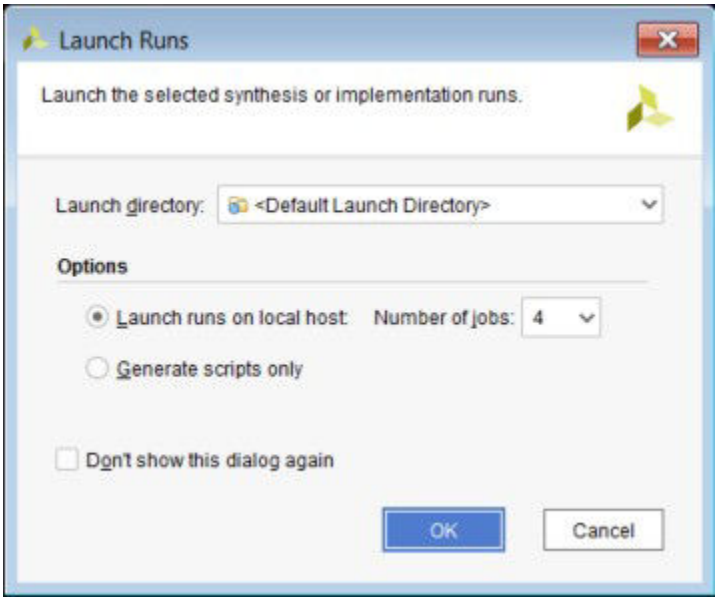
13. In the Settings dialog box (**Tools → Settings → Tool Settings → Project**), enter the tutorial project directory in the Specify project directory field, so that all reports are saved in the tutorial project directory. Then click **OK**.



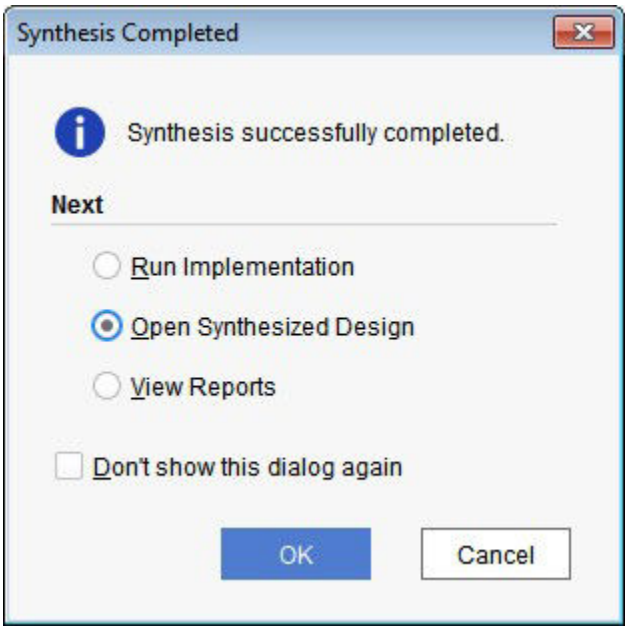
Now, the design is ready for synthesis.

Step 2: Synthesizing the Design

1. Click **Run Synthesis** in the Flow Navigator. In the Launch Runs dialog box that appears, click **OK**.



2. The Synthesis Completed dialog box appears after synthesis has completed on the design.



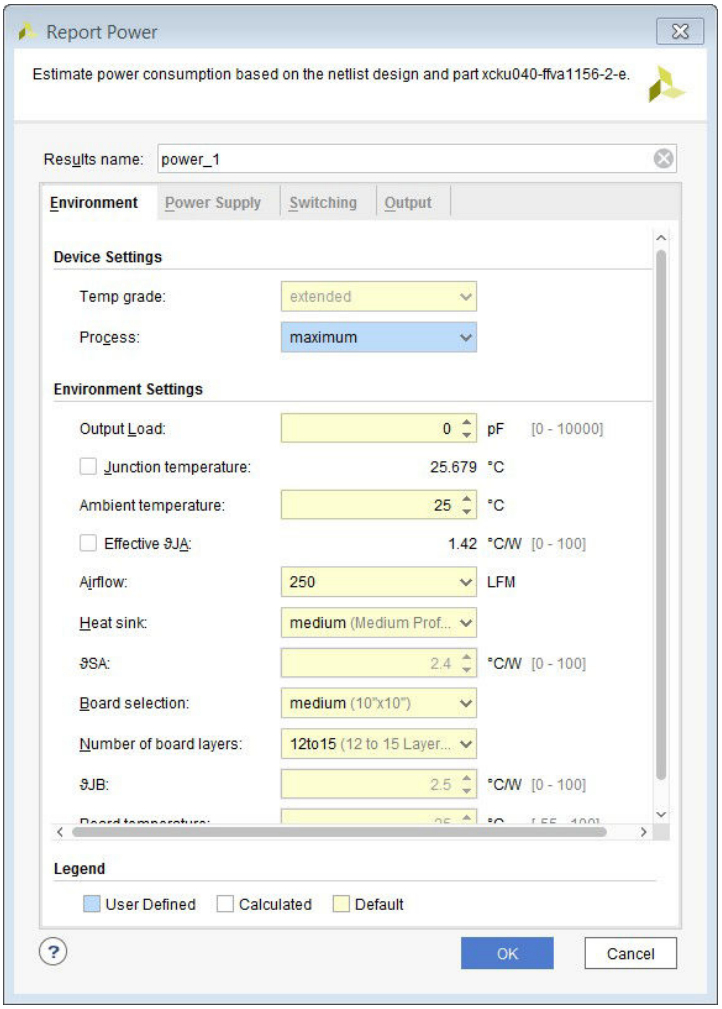
3. Open the synthesized design by selecting **Open Synthesized Design** in the Synthesis Completed dialog box and clicking **OK**.

Step 3: Setting Up the Report Power

The Vivado IDE allows you to specify input data to the Report Power tool to enhance the accuracy of the power analysis.

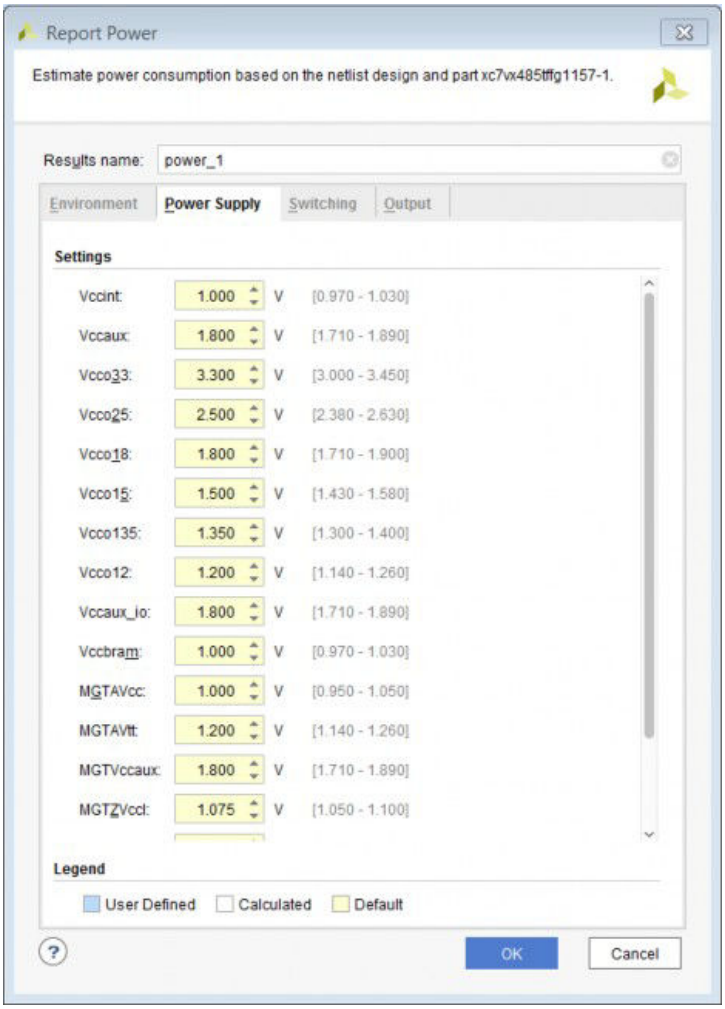
In the Vivado IDE, you can configure thermal, environmental, and power supply options to mimic the board level settings as closely as possible. For information on setting these options, see the *Vivado Design Suite User Guide: Power Analysis and Optimization* (UG907).

1. In the main menu bar, select **Reports → Report Power**.
2. Examine the Environment tab in the Report Power dialog box.



3. In the Environment tab, set Process to **maximum** for a worst case power analysis. Examine the Power Supply tab.

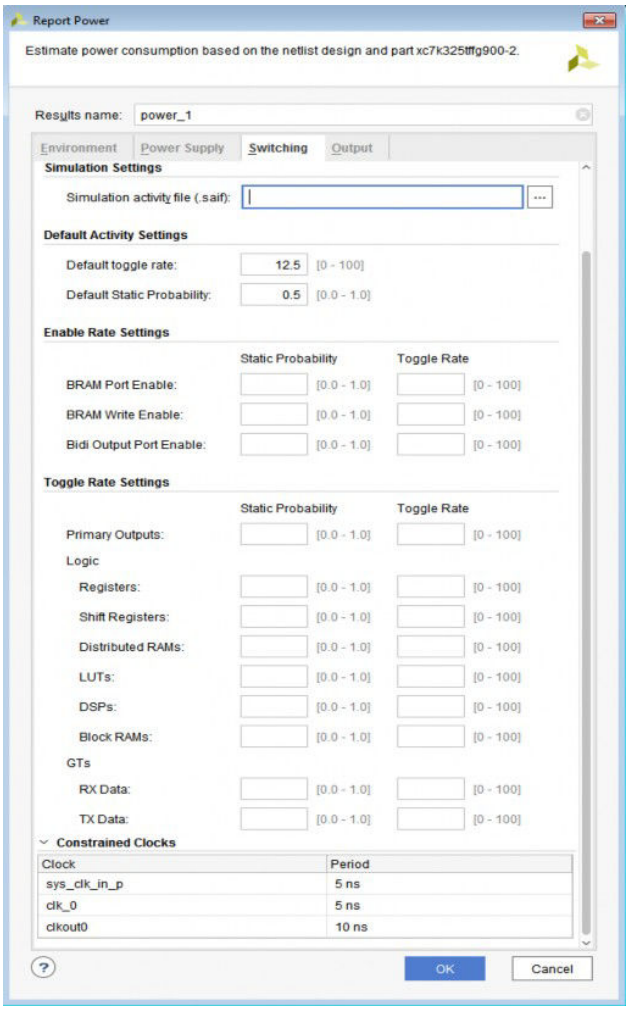
IMPORTANT! *By default, Vivado Report Power uses nominal values for voltage supply sources. Voltage is a large factor contributing to both static and dynamic power. For the most accurate analysis, ensure that actual voltage values are entered for each supply. Similarly, ensure temperature and other environmental factors match actual operating conditions.*



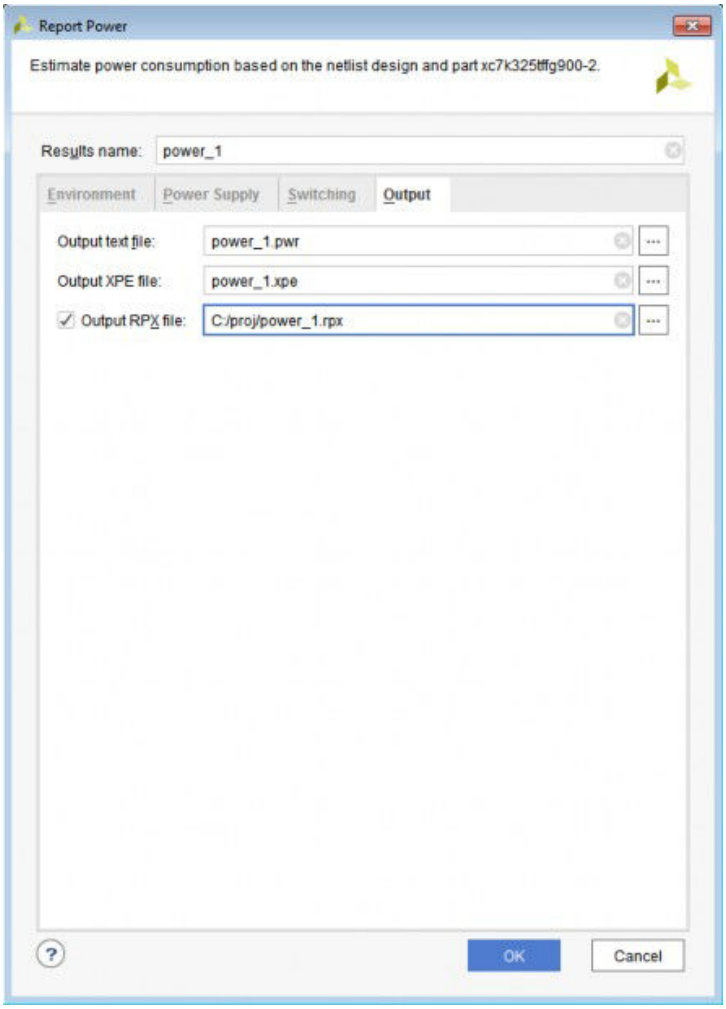
4. In the Switching tab, expand **Constrained Clocks** and examine the constrained clocks in the design.

IMPORTANT! *Make sure all the relevant clocks in the design are constrained. All the design clocks must be defined using* `create_clock` *or* `create_generated_clock` *XDC constraints, so that Report Power recognizes the clocks.*

Default toggle rate is set to 12.5% and Default Static Probability is set to 0.5. This will be applied to primary input ports (non-clock) and block box outputs.



5. In the Output tab of the Report Power dialog box, specify the **Output text file** as `power_1.pwr`.
6. Specify the Output XPE file as `power_1.xpe`. After creating this file when Report Power runs, you can import the file and results into the Xilinx Power Estimator. For information on importing the file in to the Xilinx Power Estimator, see the *Xilinx Power Estimator User Guide* (UG440).
7. Specify the RPX file to write the results of the Report Power command. The saved RPX file can be reloaded using the **Reports → Open Interactive Report** command to provide interaction/cross-probing with the open design.



Legends in Report Power Tool

The following legends appear consistently in the Report Power tool:

- **Constraint:** Displays when the nets are defined as clock with timer constraints. The defined frequency of a clock determines the switching activity.
- **Stimulation:** Displays when the nets with switching activities are derived from simulation's `.saif` file.
- **User Defined:** Displays when the nets with user set switching activities are derived from `set_switching_activity power` Tcl command.
- **Estimated:** Displays when the nets with switching activities are generated by `report_power` vectorless propagation engine.
- **Default:** Displays when the nets include default switching activities. If you use `set_switching_activity` on input port nets or on internal nets before running `report_power` (vectorless propagation), the report tool displays the default.

Step 4: Running Report Power

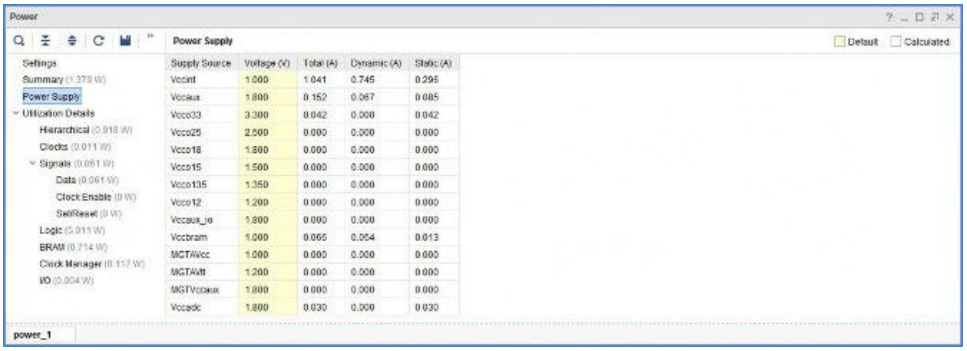
1. Click **OK** on the Report Power dialog box.

   This runs the `report_power` command.

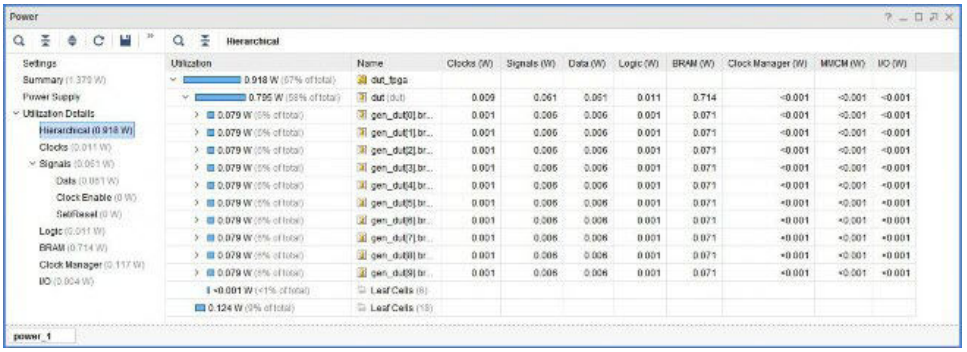
2. Examine the power report, power_1, generated in the Power window in the Vivado IDE.

   ***Note*:** Due to continuous accuracy improvements in the Vivado tools, the actual power numbers you see might be slightly different than the ones that appear in the following figures.

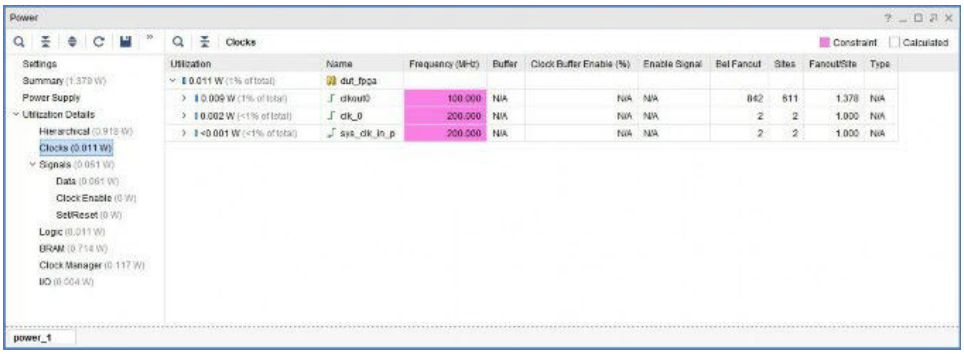
3. Examine the power breakdown in the power report by block type (Logic, BRAM, I/O, etc.).
4. Examine the power supply breakdown in the Power Supply view.



5. Examine the hierarchical breakdown of the power in the **Utilization Details → Hierarchical** view.



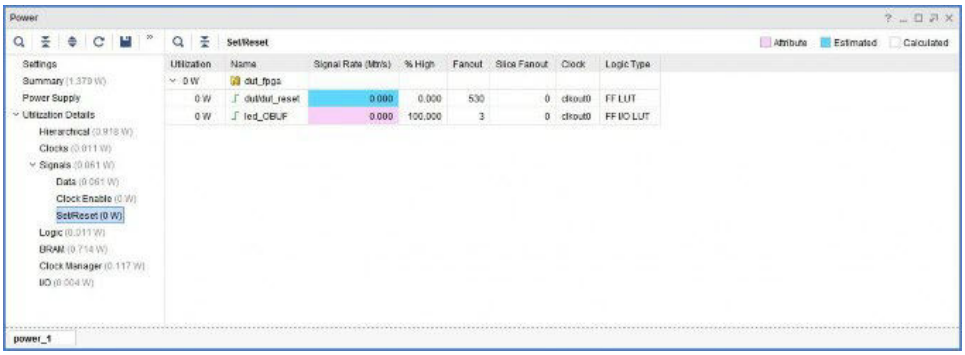
6. Examine the Clocks view and the various Signals views (Data, Clock Enable, and Set/Reset).



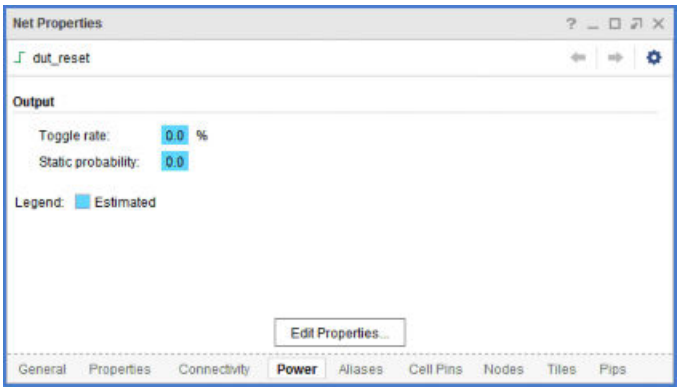
Step 5: Viewing the Power Properties

This step shows how you can get the display of static probability and toggle rate for a signal in property window.

1. Note the total power (Total On-Chip Power) in the Power Report Summary view.
2. Click the **Set/Reset** item in the Power Report.
3. Click on the **dut/dut_reset** signal.



4. Note that there is a Power view in the Net Properties window that displays net properties for the `dut/dut_reset` signal. Click on **Load Power Properties** to get the power information the first time.

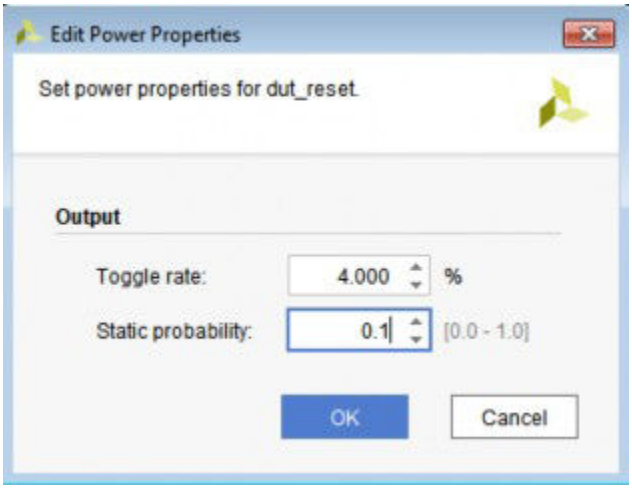


5. Note the Toggle rate is 0% and the Static probability is 0 for the `dut/dut_reset` signal, which indicates that reset is always deasserted in the design.

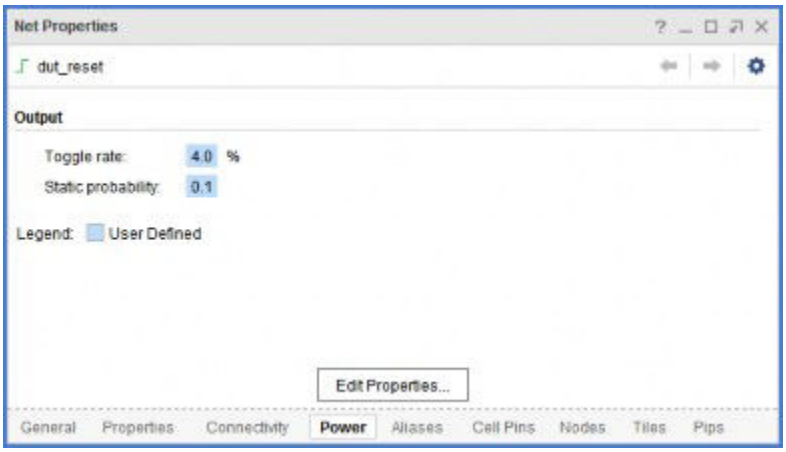
Step 6: Editing Power Properties and Refining the Power Analysis

Assume the reset is asserted for 10% of the cycles in this design. Switching activity can be set accordingly to re-estimate the power.

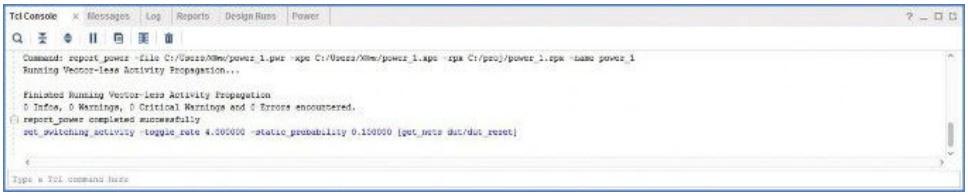
1. In the Net Properties window, click the **Edit Properties** button.
2. In the **Edit Power Properties** dialog box, change the Toggle rate to 4% and the Static probability to 0.1.



3. Click **OK**.
4. In the Net Properties window, observe that the Toggle Rate and Static Probability values turn a different color to indicate that they are user defined.

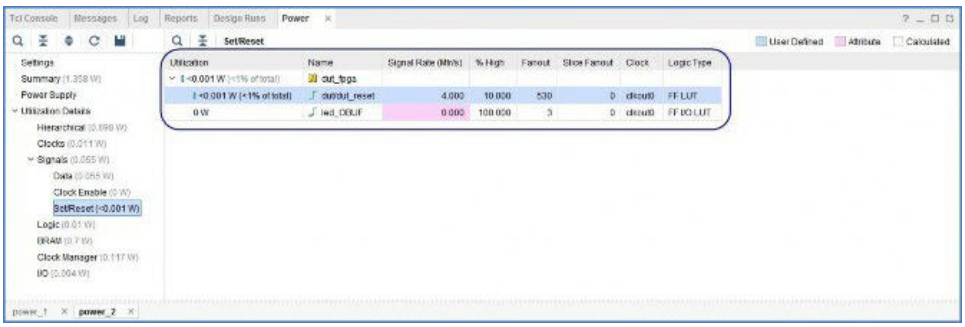


You can also observe the equivalent Tcl command executed in the Tcl Console.



5. Rerun Report Power (**Reports → Report Power**).
6. Change the Output text File and Output XPE File in the Output tab to **power_2.pwr** and **power_2.xpe** respectively.
7. In the Switching tab, set Switching Activity for Resets: to None. Then click **OK**.
8. In the Power window, note the change in total power reported in the power_2 report compared to the power_1 report. The total power has decreased due to the change in the Signal Rate for the `dut/dut_reset` signal. Because the signal is a reset signal, an increase in its activity will significantly reduce the activity of other signals in the design. The Signal Rate of the `dut/dut_reset` signal is now color coded as being User Defined in both the properties window and the Set/Reset view of the Power Report.

   You can also observe the equivalent Tcl command executed in the Tcl Console.



Xilinx recommends you to double-check the signal rates and percentage high (%High) values of high impact I/O ports, control signals (such as resets and clock enables) and high fanout nets. This is an opportunity to guide the Report Power tool to the right estimation scenario.

See the *Vivado Design Suite User Guide: Power Analysis and Optimization* (UG907) for more information on switching activity.

TIP: *In Tcl, use the* `set_switching_activity` *command to change the signal rate and static probability of signals and use* `report_switching_activity` *to query the values that were set on the signals.*

```
set_switching_activity -signal_rate 4 -static_probability 0.1 [get_nets
dut/dut_reset]
      report_switching_activity [get_nets dut/dut_reset]
```

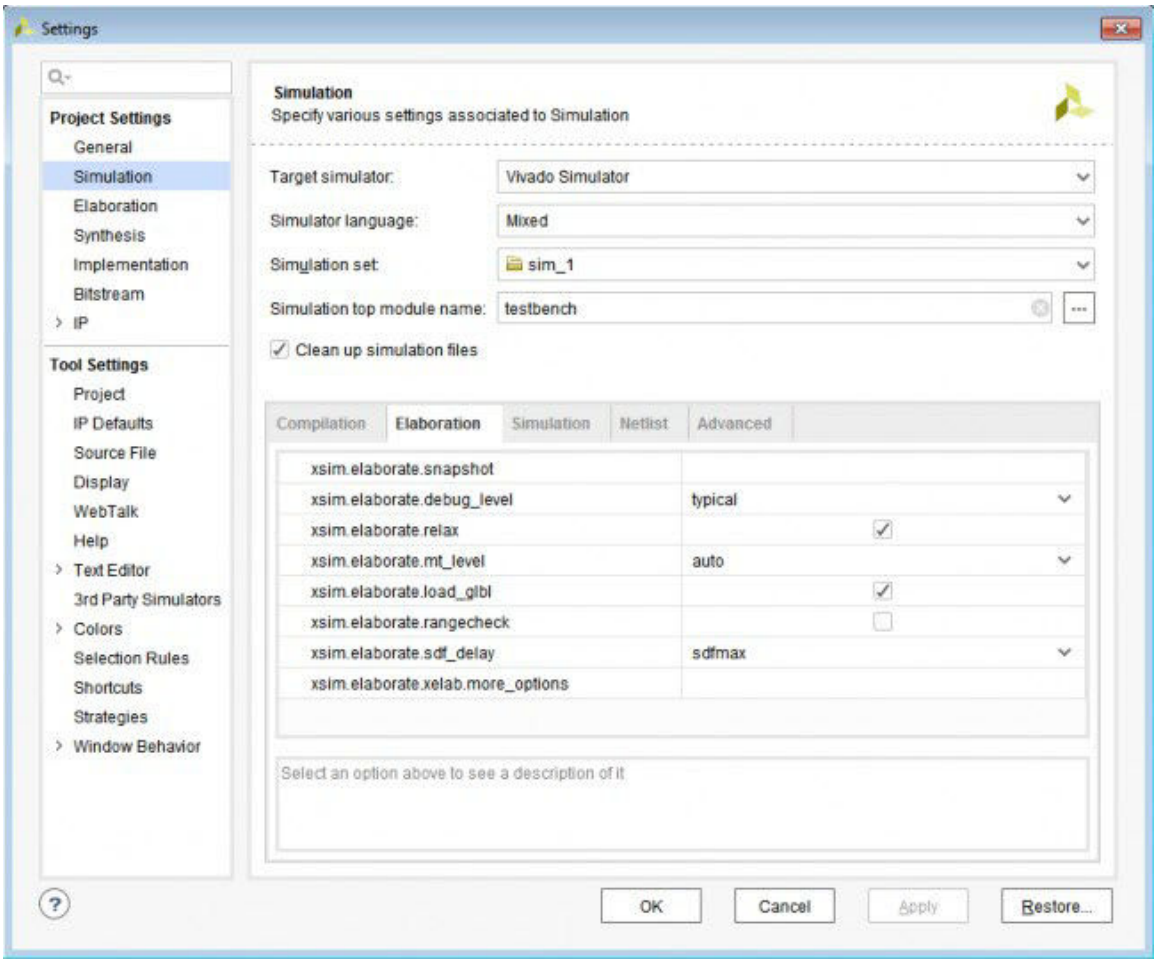
IMPORTANT! *Switching activity can also be specified in terms of toggle rate. Toggle rate is always associated with a clock. The primary ports can be associated with a specific clock using the* `set_input_delay` *and* `set_output_delay` *commands. If no clock association is found, Report Power will associate the ports with respect to the capturing clock.*

*For a clock of 100 MHz and a toggle rate of 4, the equivalent signal rate will be 4 MTr/s (signal_rate = toggle_rate * Freq = 4 * 100 MHz).*

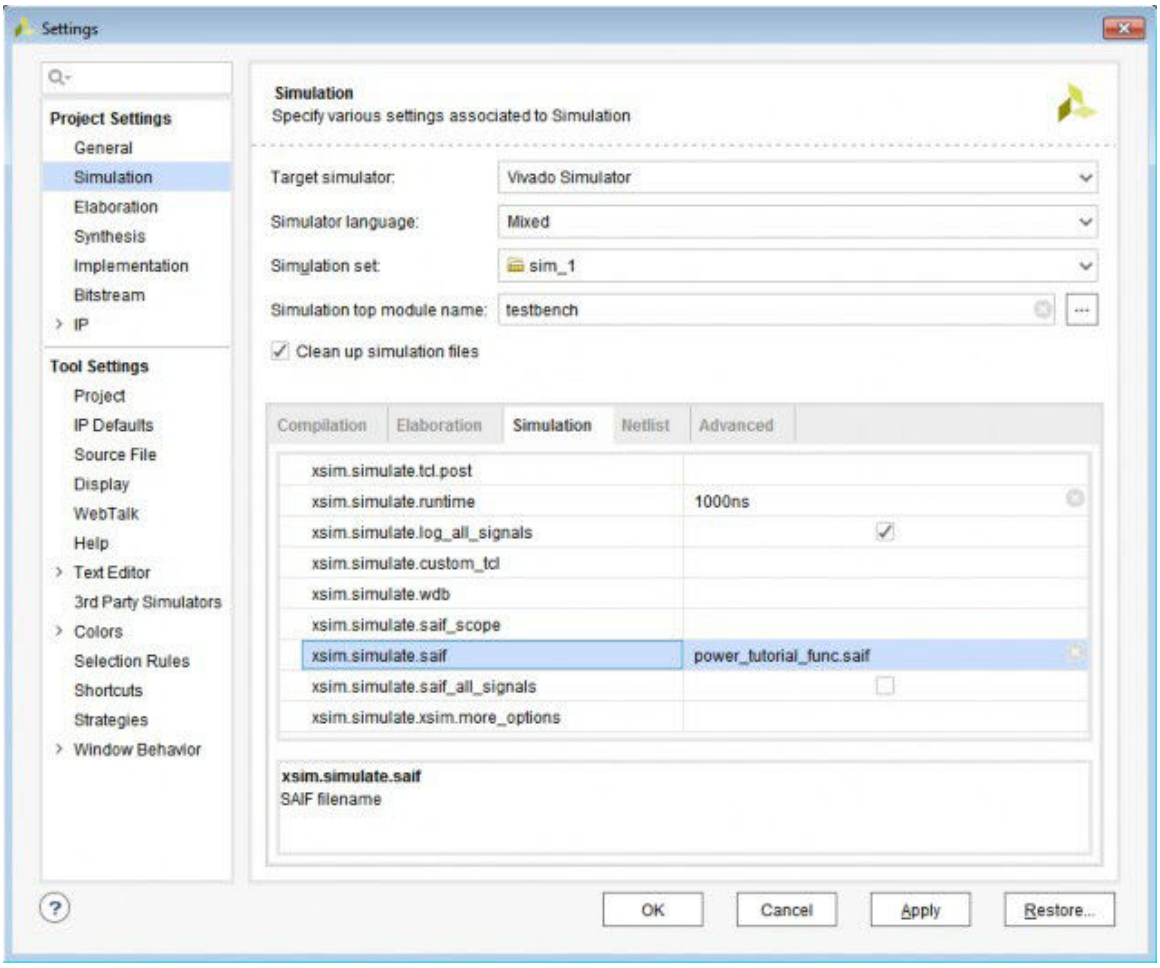
Step 7: Running Functional Simulation with SAIF Output

Now that you have created a Vivado Design Suite project for the tutorial design, you can set up and launch the Vivado simulator to run post-synthesis functional simulation. Simulation will generate a switching activity values file (SAIF) that will enable you to do more accurate power estimation on your design.

1. In the Flow Navigator, click **Settings** to open the Settings dialog box and set the simulation properties in Simulation section.
2. In the Simulation section of Settings dialog box, note that the following Simulation defaults are automatically set for you based on the design files:
   - Simulator language: **Mixed**
   - Simulation set: **sim_1**
   - Simulation top-module name: **testbench**
3. In the Elaboration tab of Simulation section, make sure the xsim.elaborate.debug_level is set to **typical**, which is the default value.



4. In the Simulation tab enter the SAIF file name as power_tutorial_func.saif for xsim.simulate.saif. Observe that the xsim.simulate.runtime is 1000 ns.
5. Click **OK**.

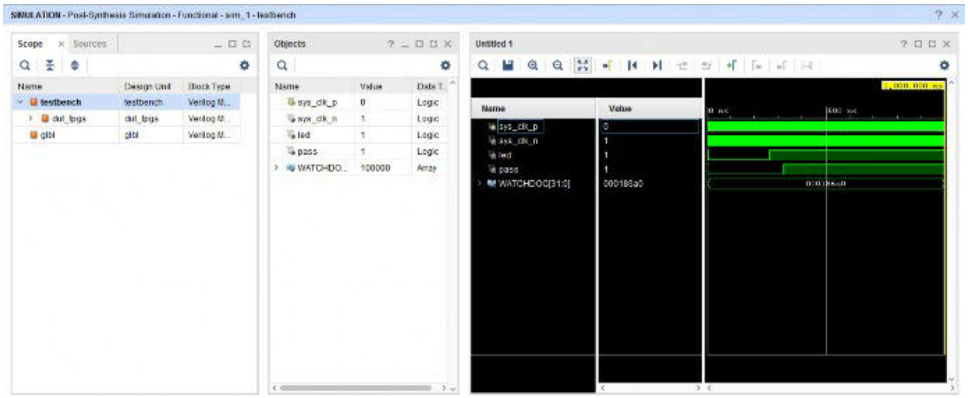


With the simulation settings properly configured, you can launch the Vivado simulator to perform a post-synthesis functional simulation of the design.

***Note*:** The power reporting and analysis are not performed at the RTL level. They are performed at the gate level.

6. In the Flow Navigator, click **Run Simulation → Run Post-Synthesis Functional Simulation**.

When you launch the Run Post-Synthesis Functional Simulation command, the Vivado simulator is invoked to run the simulation.



After the simulation completes, click **x** at the top right corner to close the simulation window.

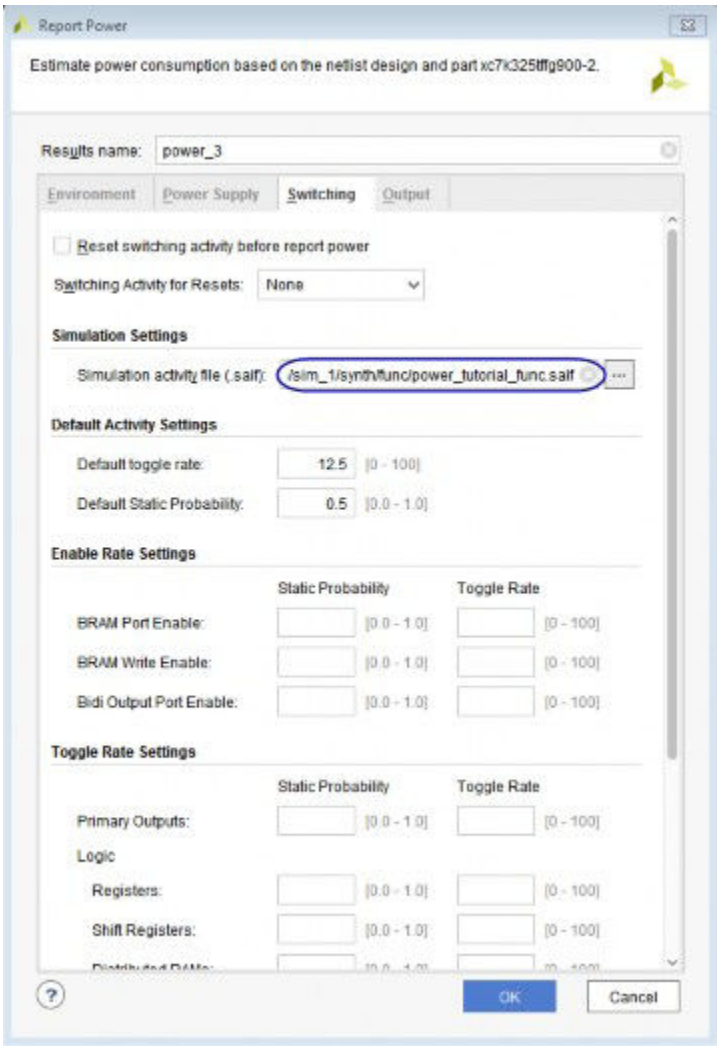
Step 8: Incorporating SAIF Data into Power Analysis

The SAIF output file requested in the simulation run is generated in the project directory. This SAIF file is used to further guide the power analysis algorithm.

1. Ensure the SAIF file requested is generated. Check to see that the SAIF file requested in the simulation settings prior to running simulation appears in this directory:

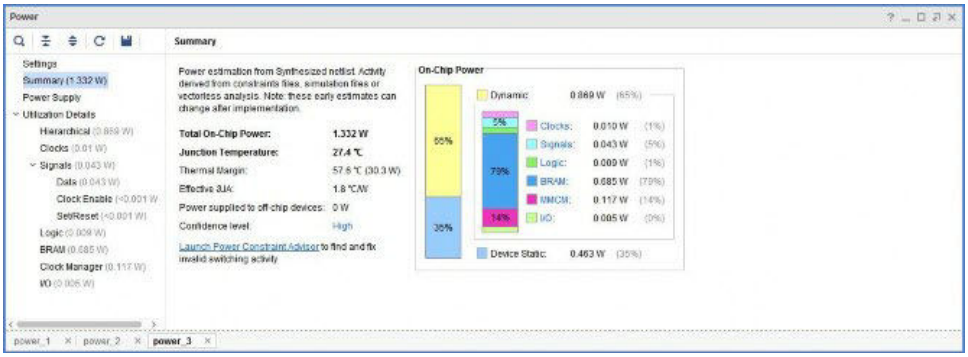
   ```
   <project_directory>/power_tutorial1/power_tutorial1.sim/sim_1/
   synth/ func/power_tutorial_func.saif
   ```

2. In the Flow Navigator window, click on **Open Synthesized Design** to expand options.
3. From the Synthesized Design options, select **Report Power**.
4. In the **Report Power** dialog box, set the Results name to **power_3**.
5. In the Output tab of Report Power dialog box, make the following changes:
   - Set the Output text File to **power_3.pwr**
   - Set the Output XPE File to **power_3.xpe**
6. In the Environment tab of Report Power dialog box, make sure that the Process is set to **maximum**.
7. In the Switching tab of Report Power dialog box, specify the SAIF file location.



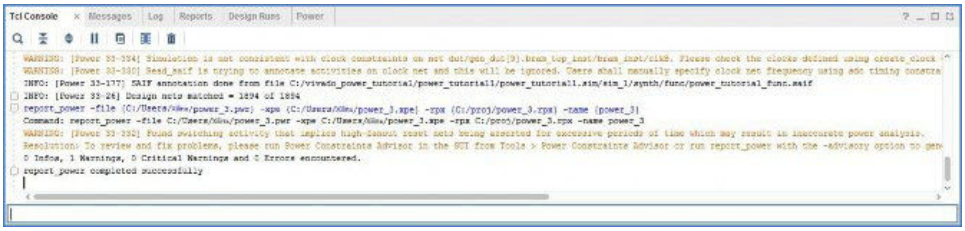
8. Click **OK** in the Report Power dialog box.

   The `report_power` command runs, and the Power Report power_3 is generated in the Power window.

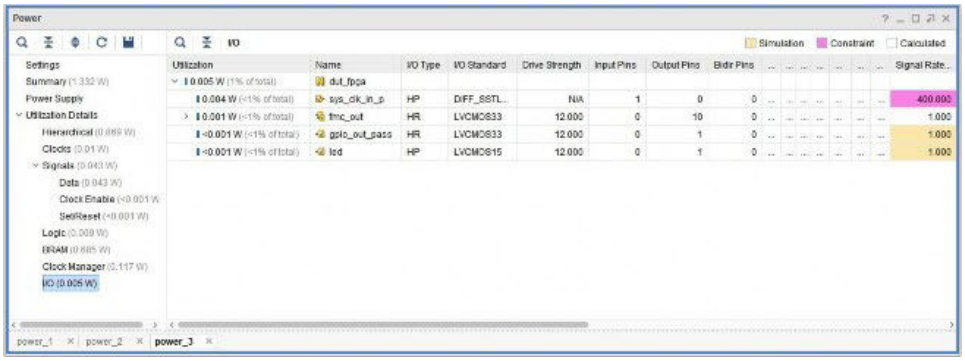


Note: The SAIF annotation results are displayed in the Tcl Console. Make sure that all the design nets are matched with simulation nets, to achieve better accuracy by including Simulation data. For 7 series devices, the number of design nets and simulation nets may vary due to various reasons. The most common reason is that their hierarchical separators are different. Sometimes, the simulation nets may be lower down in the hierarchy level. However, they should match 100%.

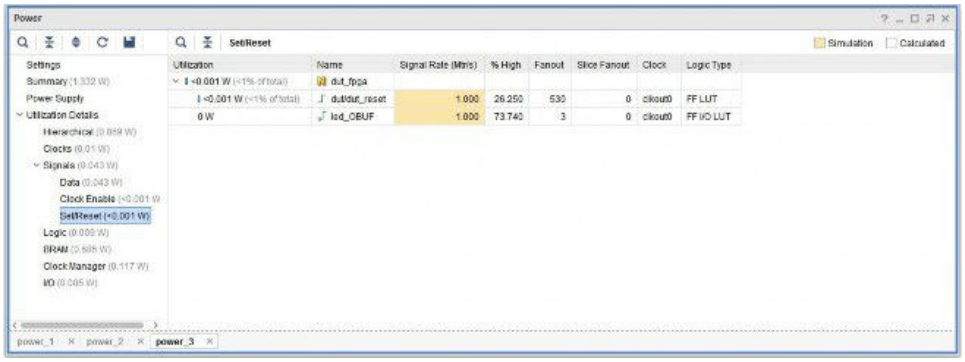
Example: INFO: [Power 33-26] Design nets matched = 1894 of 1894



9. Go to the I/O view in the Power window. Note that all the I/O port activity data has been set from simulation data we specified. The data is color coded to indicate activity rates read from the simulation output file.



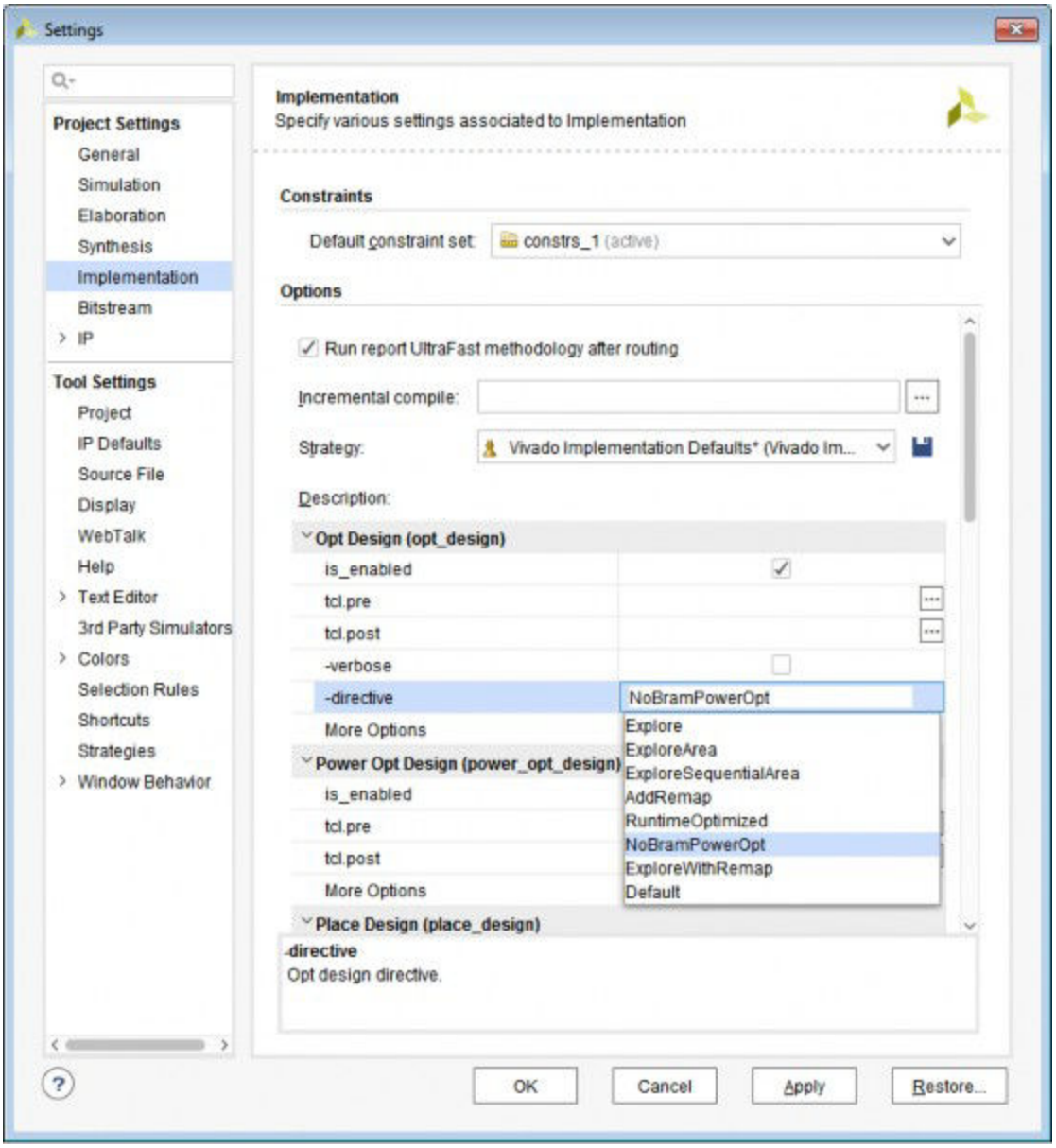
10. Note the difference in total power numbers (Total On-Chip Power in the Summary view) between a pure vectorless run in the power_1 results versus with the post synthesis functional simulation data in the power_3 results. Also note that the `dut/dut_reset` signal rates are overwritten by simulation SAIF data.



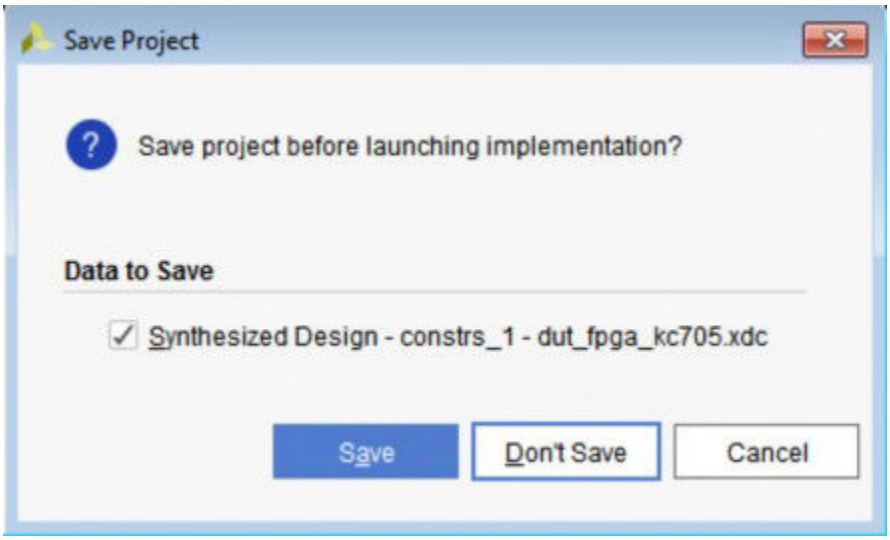
Step 9: Implementing the Design

This tutorial helps you understand power analysis with and without power optimization. In this step, you will run Implementation without power optimization.

1. In the Flow Navigator, right-click **Implementation** and select **Implementation Settings**.
2. In the Opt Design settings, select the **NoBramPowerOpt** option for –directive and click **OK**.



3. In the Flow Navigator, click **Run Implementation**.
4. When Save Project dialog box is displayed to save the project before launching implementation, click **Don't Save**.



Conclusion

In this lab, you have learned how to set the power analysis in the Vivado. In lab 2, you will learn about the timing simulation and its effect on the power analysis.

Lab 3

Running Timing Simulation and Estimating Power

Introduction

In this lab, you will learn about generating a SAIF file after running a timing level simulation using the Vivado® simulator and Questa Advanced Simulator. The lab will take you through the steps for SAIF file creation, running timing simulation, and estimating power using the SAIF data.

Step 1: Configuring and Running the Timing Simulation using Vivado Simulator

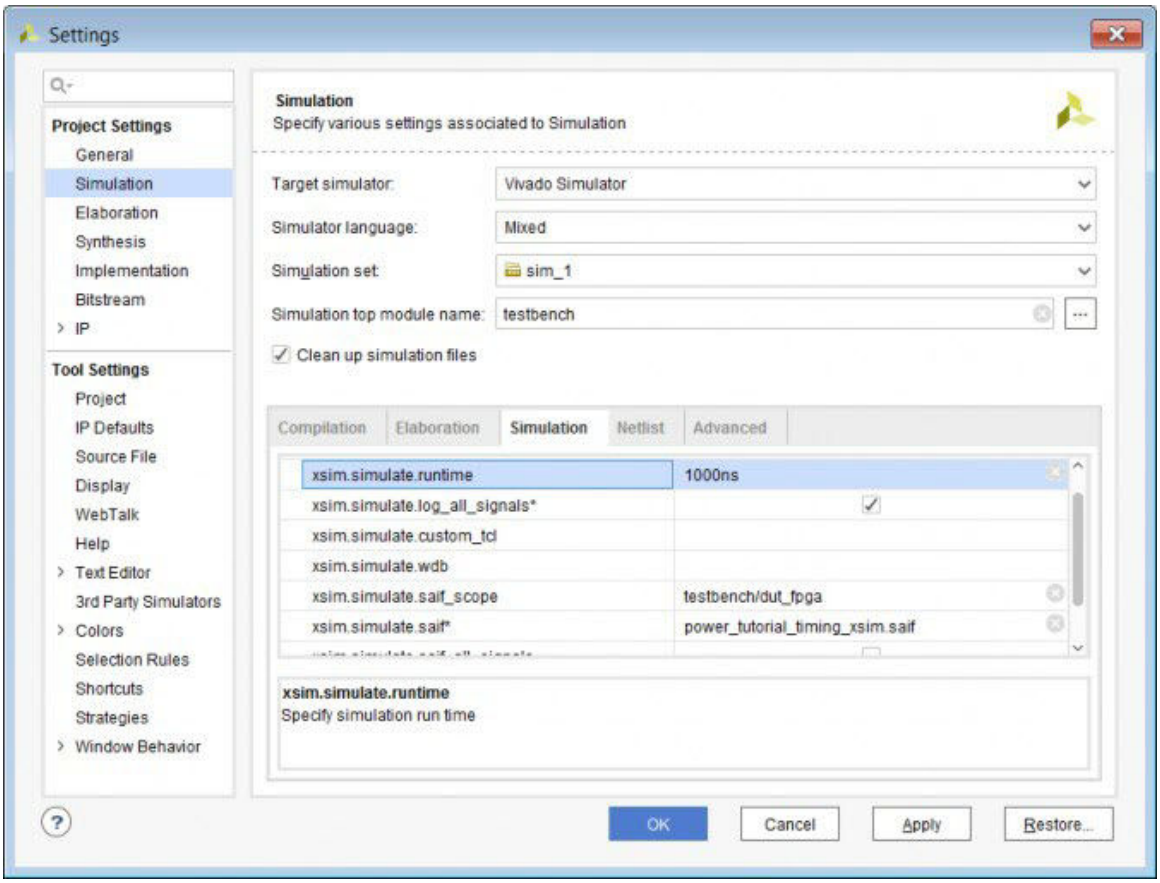
1. In the Implementation Complete dialog box, select **Open Implemented Design** and click **OK** to open the implemented design. When prompted to save the project before opening an implemented design, click **Don't Save**.

   Now you are ready to set up and launch the Vivado simulator to run post implementation timing simulation. You will set the timing simulation properties in the Vivado IDE, then run the timing simulation.

2. In the Flow Navigator, click **Settings** and select the **Simulation** to set the timing simulation properties. In the Settings dialog box, note that the following defaults are automatically set:

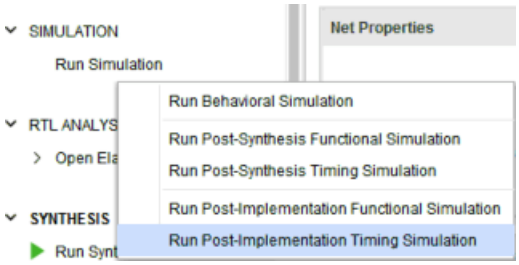
   - Simulation set: **sim_1**
   - Simulation top-module name: **testbench**

3. In the Elaboration tab, make sure that debug_level is set to **typical**, which is the default value.
4. In the Simulation tab, set the SAIF file name xsim.simulate.saif to **power_tutorial_timing_xsim.saif**.
5. Set the xsim.simulate.saif_scope to **testbench/dut_fpga**.
6. Observe that the simulation run time xsim.simulate.runtime is 1000ns.
7. Click **OK**.



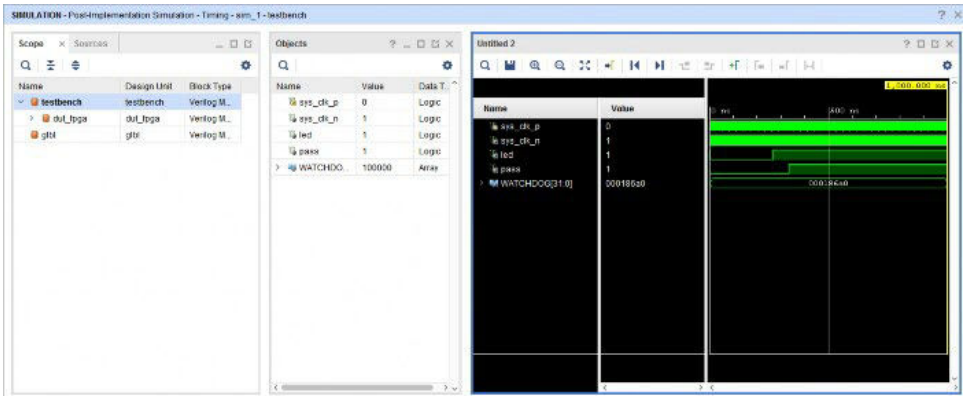
With the simulation settings properly configured, you can launch the Vivado simulator to perform a timing simulation of the post implemented design.

8. In the Flow Navigator, click **Run Simulation → Run Post-Implementation Timing Simulation**.



9. After the Vivado simulator has finished simulating the design, ensure that the SAIF file requested has been generated. Check to see that the SAIF file requested in the simulation settings prior to running simulation appears in this directory:

```
<project_directory>/power_tutorial1/power_tutorial1.sim/ sim_1/
impl/timing/power_tutorial_timing_xsim.saif
```



Step 2: Running Report Power in Vectorless Mode

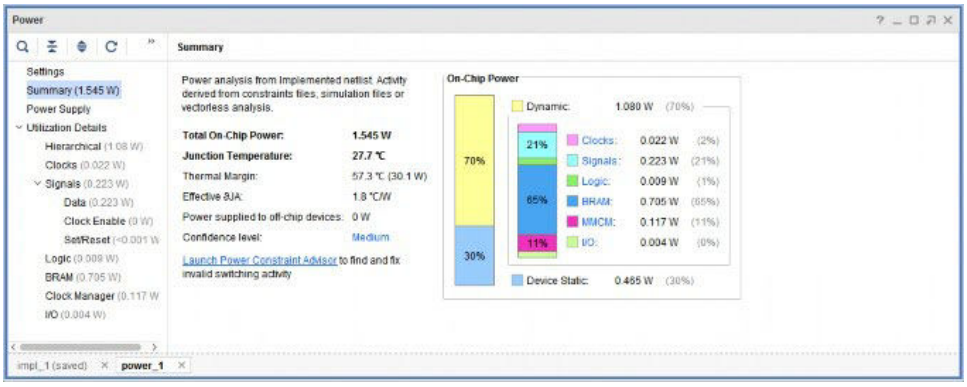
1. In the Flow Navigator, select **Open Implemented Design → Report Power** to open the Report Power dialog box.

   You can also select **Reports → Report Power** from the main menu.

2. In the Report Power dialog box, Environment tab, make sure the Process is set to **maximum** and click **OK**.

   The Report Power command creates a power report under the power_1 tab in the Power window.

3. Note the total power (Total On-Chip Power) in the power report Summary page.



   Vectorless analysis is done based on default switching activity specification on the primary ports and the design clocks.

   Refer to the *Vivado Design Suite User Guide: Power Analysis and Optimization* (UG907) for more information on vectorless power analysis.

Step 3: Running Report Power with Vivado Simulator SAIF Data

The project directory contains the SAIF output file requested in the previous timing simulation run. We use this SAIF file to further guide the power analysis algorithm.

1. From the main menu, select **Reports → Report Power**.
2. In the Report Power dialog box, specify the SAIF file location in the Switching tab.

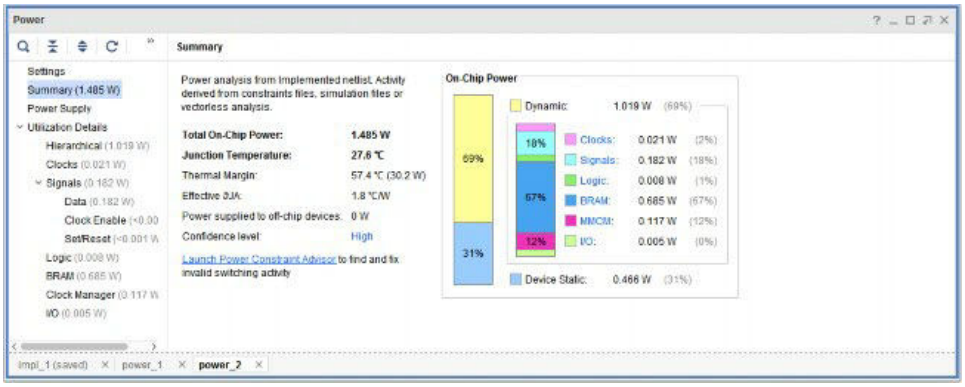
   The SAIF file, which was requested in the simulation settings prior to running timing simulation, should appear here:

   ```
   <project_directory>/power_tutorial1/power_tutorial1.sim/ sim_1/
   impl/timing/power_tutorial_timing_xsim.saif
   ```

3. Click **OK** in the Report Power dialog box.

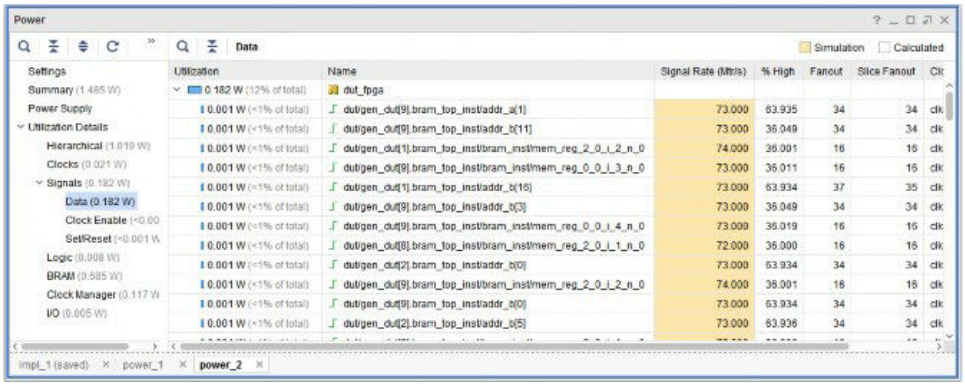
   After the Report Power command completes, the Power windows displays power report power_2.

   In the Tcl console, observe that the SAIF file is read successfully and that 100% of the design nets are matched. This assures you that the generated SAIF file is correct and matched with all design nets.



4. Note the change in total power (Total On-Chip Power in the Summary view) in the power_2 report compared to the power_1 report. The total power estimated in the report generated with SAIF file data will be different than the total power estimated in the vectorless run (power_1 results).
5. Examine the summary and block level (On-Chip Power) power distribution in the Summary view of the power report.
6. Go to the **Utilization Details → Signals → Data** view in the power report. Note that all the Signal Rate data has been set from simulation data the SAIF file provided.

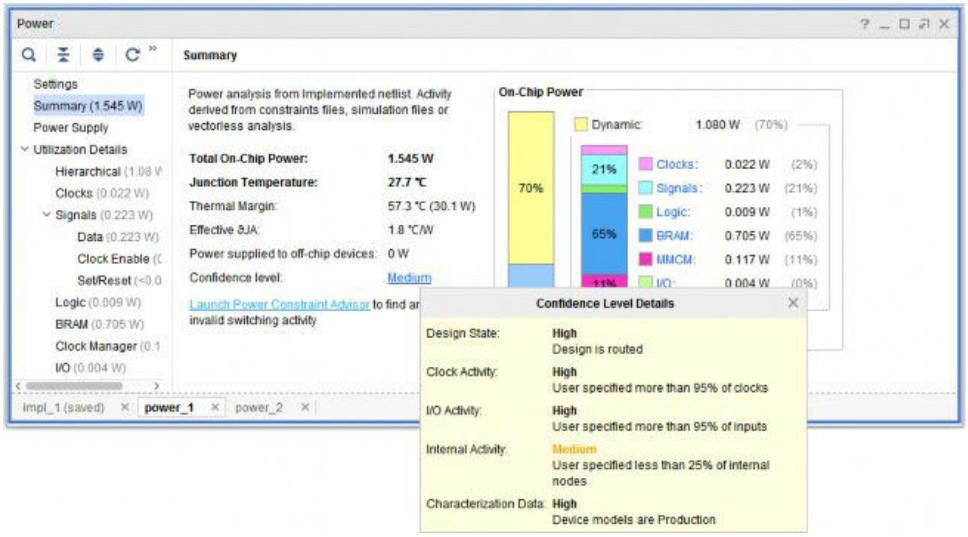
   The data is color coded to indicate activity rates read from the simulation output file.



7. In the Summary view of the power_1 report (the report generated by the vectorless analysis), click on **Confidence level** (the following figure).

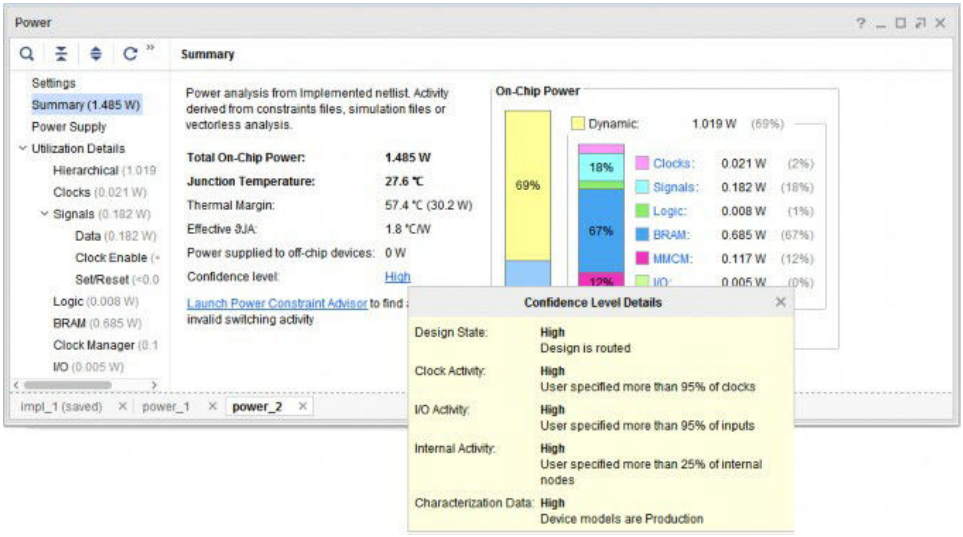
   The Confidence Level is a measurement of the accuracy and the completeness of the input data that the Report Power uses while performing power analysis.

   Notice that the Confidence Level is High for the vectorless analysis because less than 25% of internal nodes are user specified for **Internal Activity**.



8. In the Summary view of the power_2 report (the report generated by the analysis for which you specified a SAIF file as input), click on **Confidence level** (the following figure).

   Notice that the Confidence Level has increased to High, because more than 25% of internal nodes are user specified for **Internal Activity**.



Generating a SAIF File using Questa Advanced Simulator

The following steps will take you through the process of SAIF file creation, running timing simulation, and estimating power using the SAIF data using Questa Advanced Simulator.

IMPORTANT! *Make sure the Vivado Design Suite knows where to pick up the Questa Advanced Simulator tool. You can either:*

Manually set the path to ModelSim/Questa Advanced Simulator using the `$PATH` *environment variable*

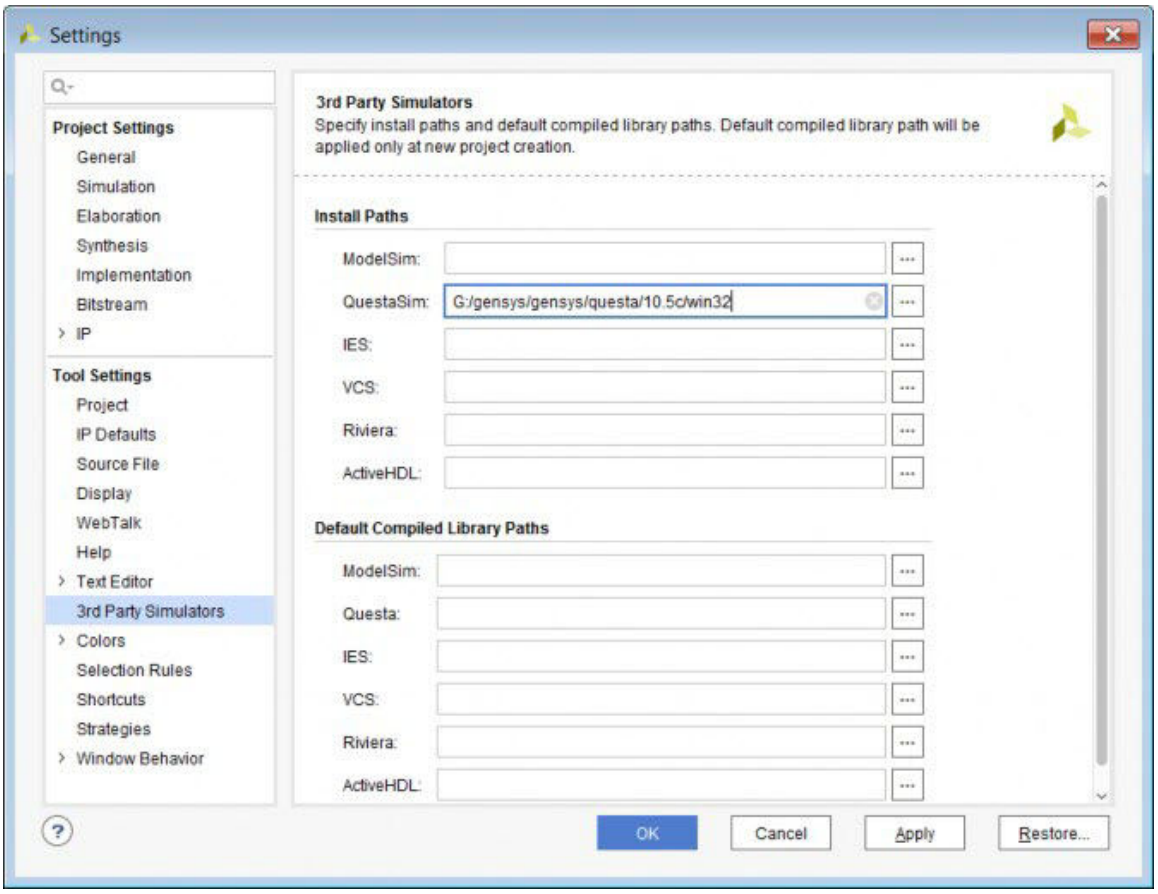
or

In the Vivado IDE, click ***Tools → Settings → Tool Settings****, and define the path to the Questa Advanced Simulatoron the 3rd Party Tools page.*

Make sure the Default Compiled Library Paths points to a valid location for the compiled Xilinx simulation libraries.

To create new compiled libraries:

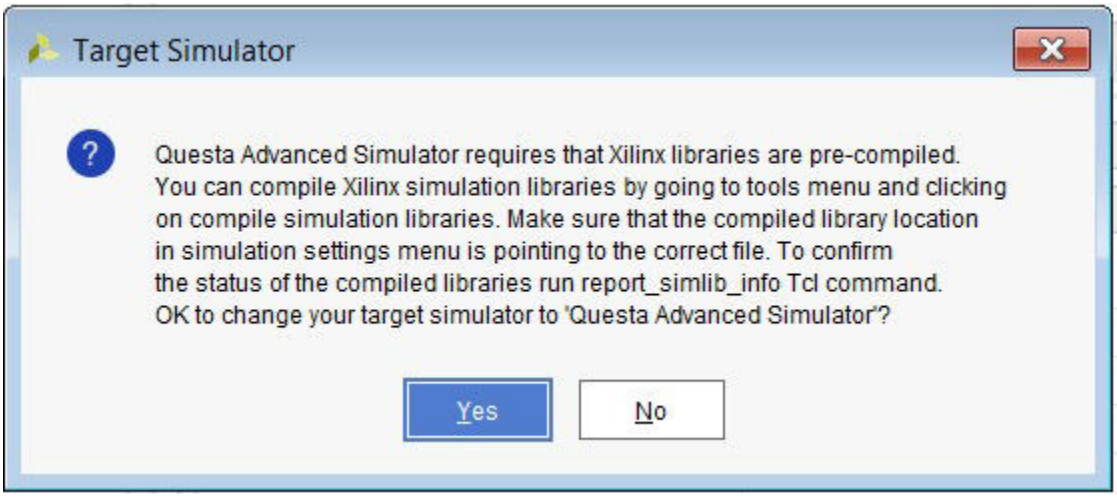
1. In the 3rd Party Simulators page, specify the compiled library path for Questa Advanced Simulator in the **Questa** field under Default Compiled Library Paths. Enter the **Compiled library location** specified during the compiled library generation. It should point to the `compile_simlib` directory.
2. Click **OK** to define the path and generate compiled libraries.



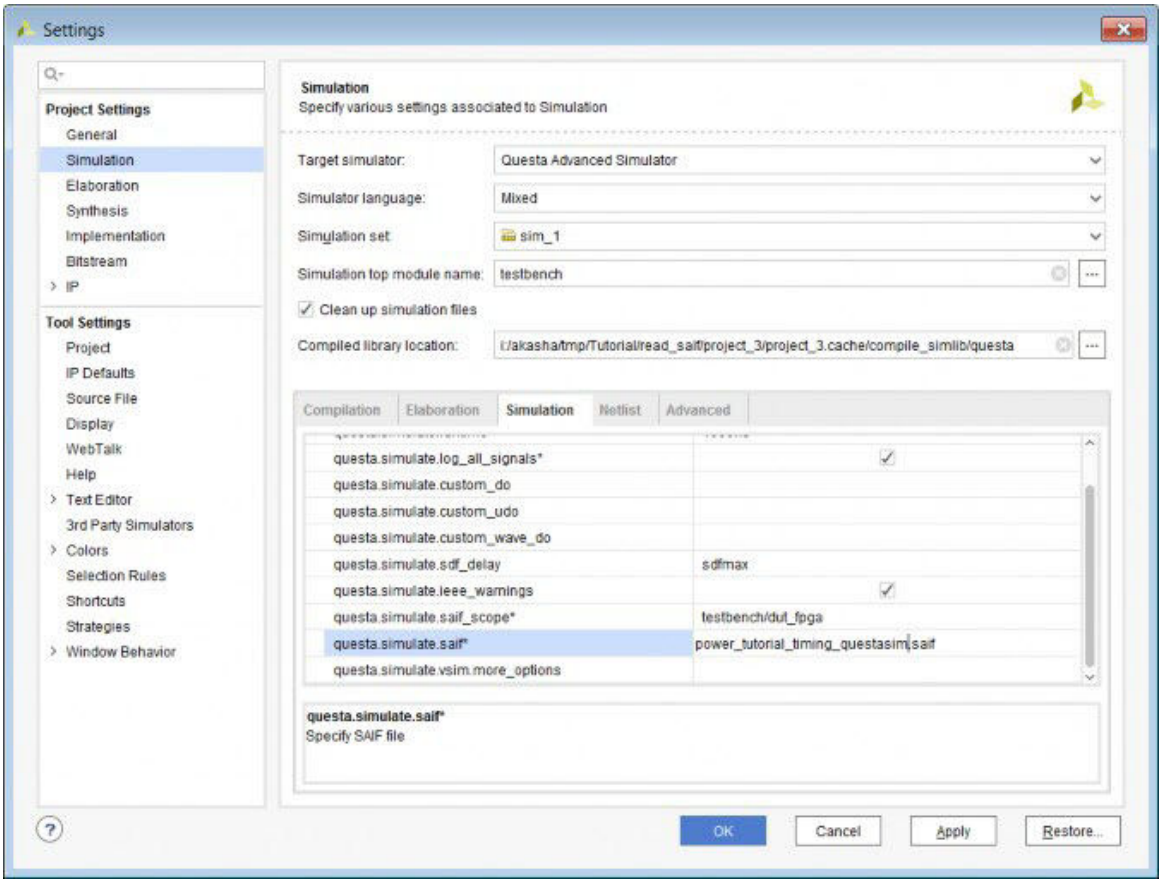
Step 1: Configuring and Running Timing Simulation in Questa Advanced Simulator

Now you are ready to set up and launch the Questa Advanced Simulator to run post-implementation timing simulation. You will set the timing simulation properties in the Vivado IDE, and run the timing simulation

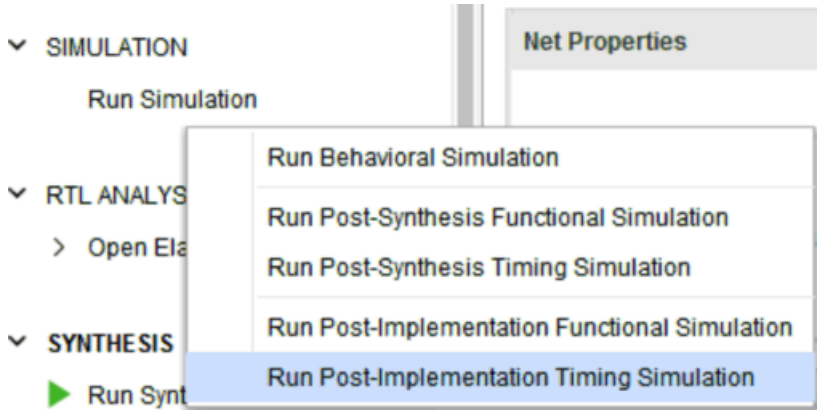
1. In the Flow Navigator, right-click **Simulation** to select **Simulation Settings**. Set the timing simulation properties.
2. In the Simulation Settings tab, set the Target simulator to **Questa Advance Simulator**.
3. Click **Yes** to change your target simulator to Questa Advanced Simulator.



4. Set the questa.simulate.saif to **power_tutorial_timing_questasim.saif**.
5. Set the questa.simulate.saif_scope to **testbench/dut_fpga**.
6. Note that the questa.simulate.runtime is **1000ns**.



7. Click **OK**. With the simulation settings properly configured, you can launch the Questa Advanced Simulator to perform a timing simulation of the design.
8. In the Flow Navigator, click **Run Simulation → Run Post-Implementation Timing Simulation**.



A separate Questa Advanced Simulator GUI opens and starts simulating the design.

9. After the Questa Advanced Simulator has finished simulating the design, make sure the SAIF file requested has been generated. Check to see that the SAIF file requested in the simulation settings prior to running simulation appears in this directory:

   `<project_directory>/power_tutorial1/power_tutorial1.sim/ sim_1/impl/timing/power_tutorial_timing_questasim.saif`

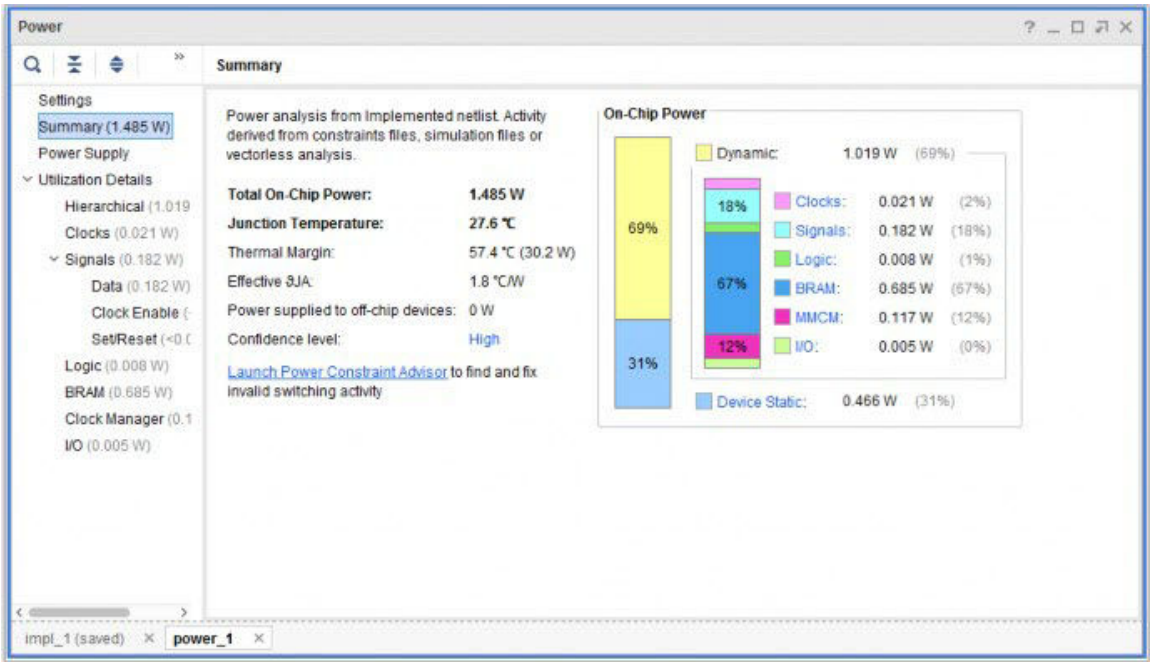
Step 2: Running Report Power in Vectorless Mode

> **IMPORTANT!** *If SAIF based `report_power` has already been run in this session, run the `reset_switching_activity -all` command in the Tcl Console. This will clear the SAIF data in the power engine from the earlier runs.*

1. Close any open Report Power views.
2. In the Flow Navigator, select **Implemented Design → Report Power** to open the Report Power dialog box.

   Alternatively, select **Reports → Report Power** in the main menu.
3. In the Report Power dialog box, make the following settings:
   - Specify the Results name as **power_1**.
   - In the Environment tab, set the Process to **maximum**.
   - In the Switching tab, leave the Simulation activity file empty.
4. Verify that all the input settings are correct and click **OK**.

   The Report Power command creates a power report under the power_1 tab in the results windows area. Note that the total power for vectorless analysis runs with default switching rates.



Step 3: Running Report Power with Questa Advanced Simulator SAIF Data

The SAIF output file requested in the simulation run has been generated under the project directory. We use this SAIF file to further guide the power estimation algorithm.

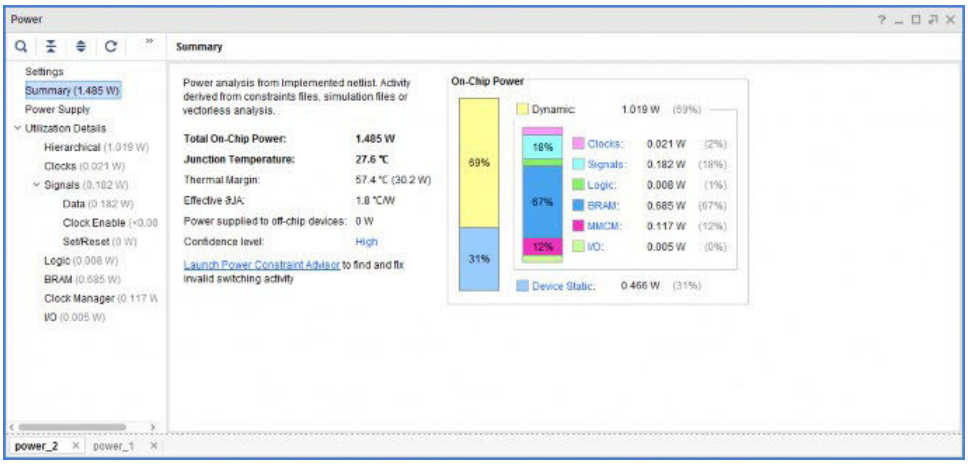
1. In the main menu bar, select **Reports → Report Power**.
2. In the Report Power dialog box, specify the SAIF file location in the Switching tab.

   The SAIF file, which was requested in the simulation settings prior to running simulation, should appear here:

   ```
   <project_directory>/power_tutorial1/power_tutorial1.sim/ sim_1/
   impl/timing/power_tutorial_timing_questasim.saif
   ```

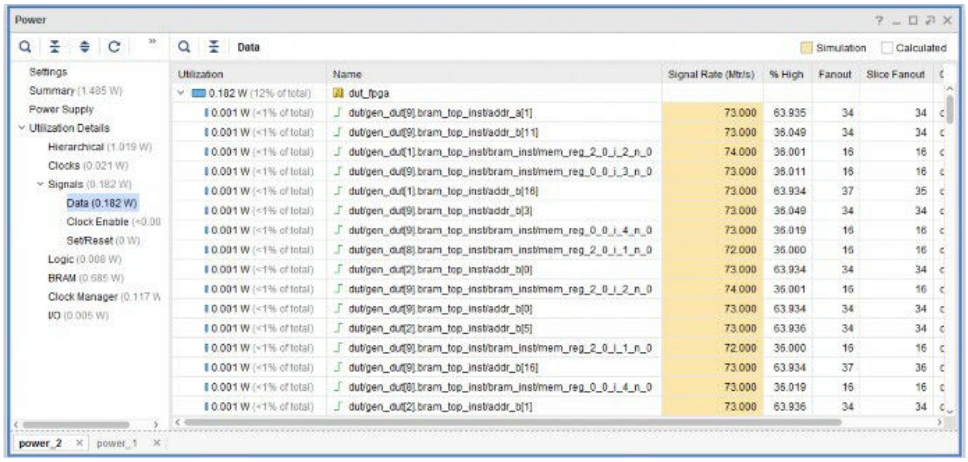
3. Click **OK** in the Report Power dialog box.

   The Report Power command runs, and the Power Report power_2 is generated in the Power tab of the results windows area.



4. In the Tcl console, observe the read_saif results. This shows the percentage of design nets matched with simulation SAIF. This is important for accurate power analysis.
5. Go to the **Signals → Data** view in the Power Report and scroll to the right. Note that all the Signal Rate data is set from simulation SAIF data that you provide.

   The data is color coded to indicate activity rates read from the Simulation output file.



6. Note the change in total power (Total On-Chip Power in the Summary view) in the power_2 report compared to the power_1 report. The total power estimated in the report generated with SAIF file data will be different than the total power estimated in the vectorless run (power_1 results).

Conclusion

In this lab, you have learned how to generate a SAIF file after running a timing level simulation using a Vivado Simulator and Questa Advanced Simulator.

In Lab 3, you will learn about basic hardware power measurement technique using the KC705 Evaluation Board and correlating the hardware power numbers with the numbers generated by Vivado Report Power.

Lab 4

Measuring Hardware Power Using the KC705 Evaluation Board

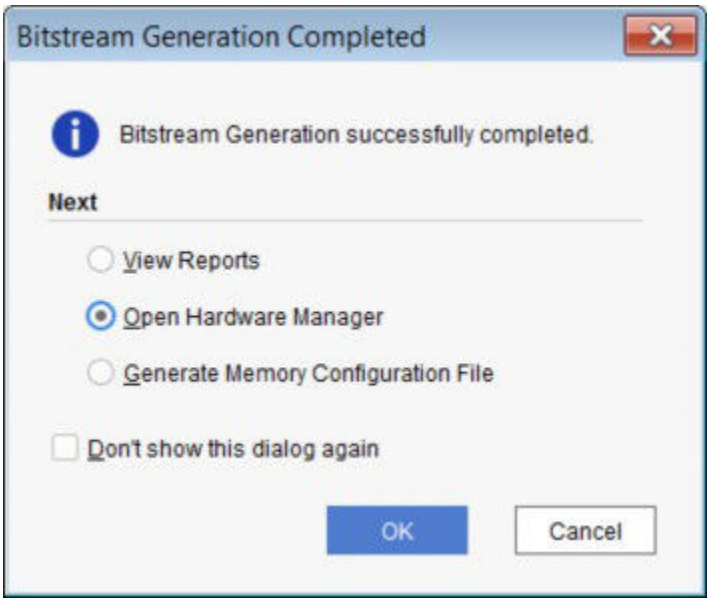
Introduction

In this lab, you will learn about basic hardware power measurement technique and correlating the hardware power numbers with the numbers generated by Vivado® Report Power using KC705 evaluation board for 7 series devices. The lab will take you through the steps for setting up the hardware measurement, programing a bit file using Vivado Hardware Manager and power measurement through Texas Instruments (TI) Fusion Design Software. It also includes Junction Temperature reading from Vivado System Monitor.

Step 1: Generating a Bit File from the Implemented Design (Non-Power Optimization)

1. In the Vivado Design Suite, open the 7 series implemented design.
2. In the Flow Navigator, click **Generate Bitstream**.

   The Bitstream Generation Completed dialog box appears after the bitstream has been generated.

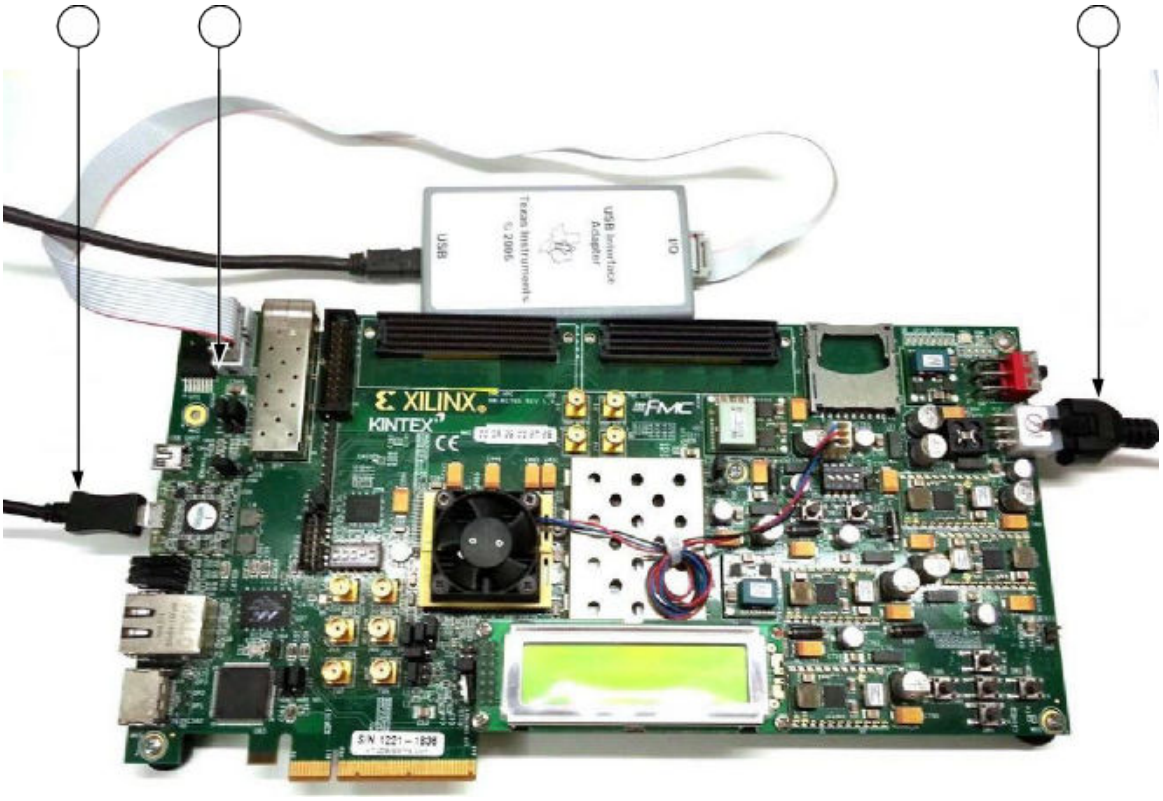


3. Select **Open Hardware Manager** in the Bitstream Generation Completed dialog box and then click **OK** to open the Hardware Manager.

Step 2: Setting Up the KC705 Evaluation Board

IMPORTANT! *This project is created for the KC705 Rev 1.0 Evaluation Board. The pin constraints are set based on this Evaluation Board. If you are using any other revisions, update the XDC file* `dut_fpga_kc705.xdc` *with the correct pin constraints.*

1. Connect the Digilent cable (or Platform USB Cable) for programming.
2. Connect the TI USB Interface Adapter to the PMBus port on the KC705 Evaluation Board.
3. Connect the Power cable.

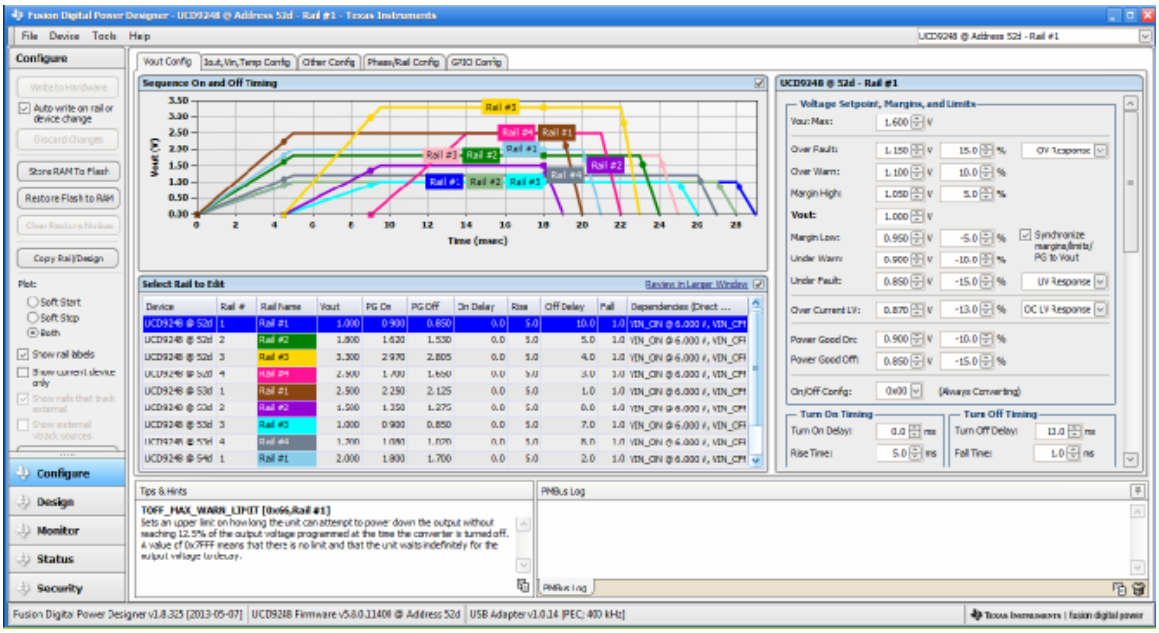


4. Install the TI Fusion Digital Power Designer software on the PC from this location.

Step 3: Setting Up the Fusion Digital Power Designer Software

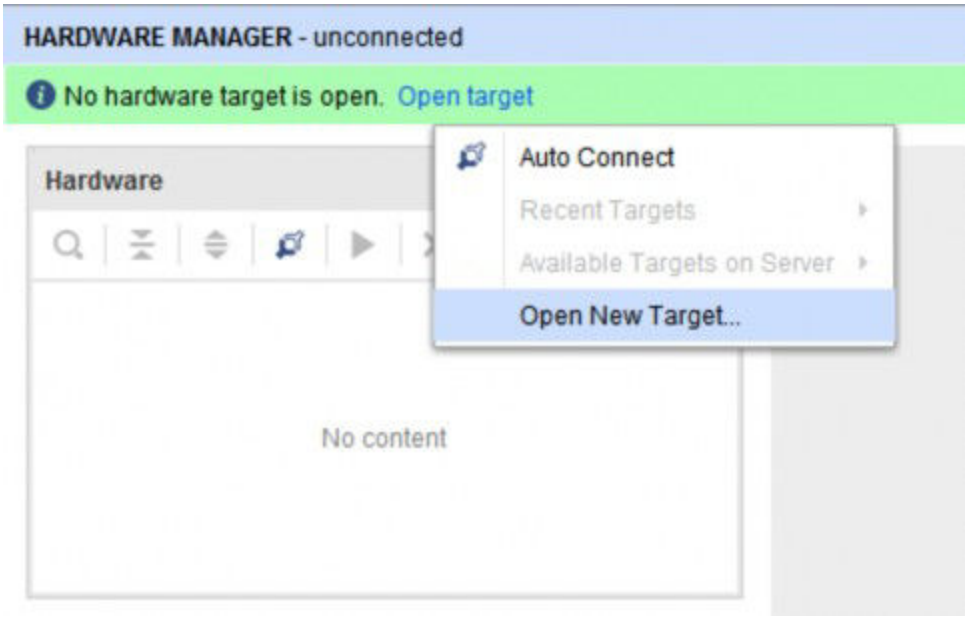
1. Power ON the KC705 Evaluation Board.
2. Open the **Fusion Digital Power Designer**.

   The software detects the USB adapter and brings up the GUI.



Step 4: Programming the Bitstream

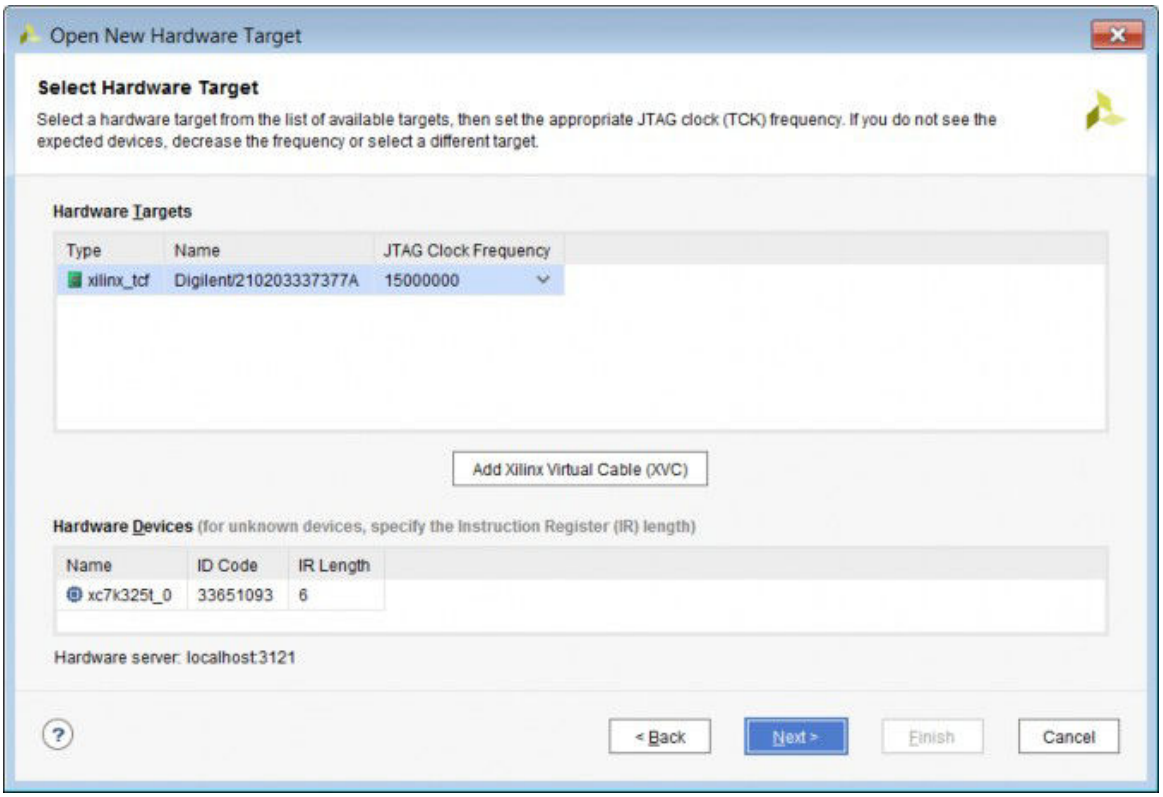
1. Power up the KC705 Evaluation Board.
2. In the Vivado Hardware Manager, click **Open Target** in the green alert bar and select **Open New Target**.



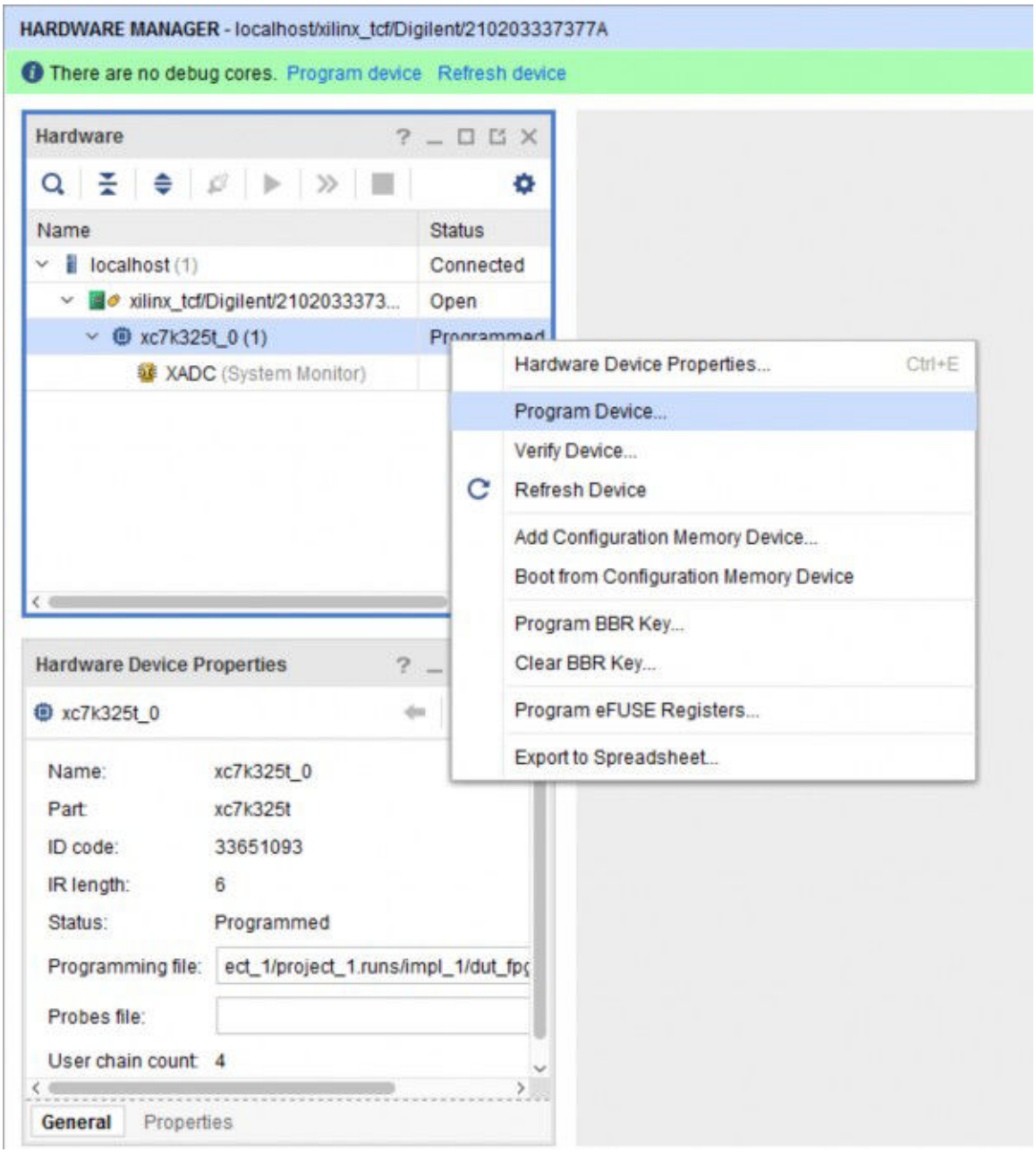
3. In the Open New Hardware Target wizard, click **Next** to go to the Hardware Server Settings.

4. Select the server to which the board is connected.
   - If the board is connected to the local PC, select **Local server** and click **Next**.
   - If you are connecting to a remote server, see Connecting to a Hardware Target Using hw_server in the *Vivado Design Suite User Guide: Programming and Debugging* (UG908).

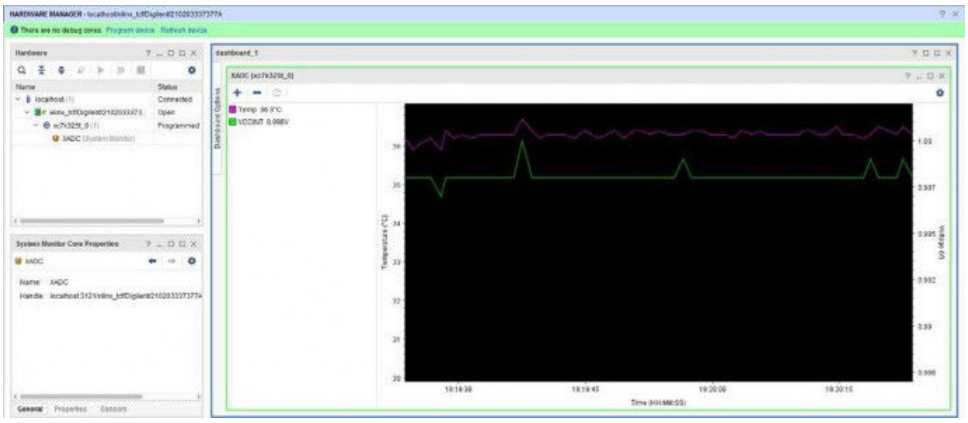
   When the hardware is detected successfully, the part information will be displayed in the Open New Hardware Target wizard.
5. Verify the part information, then click **Next** then click **Finish**.



6. In the Hardware window, right-click the part and select **Program Device**.

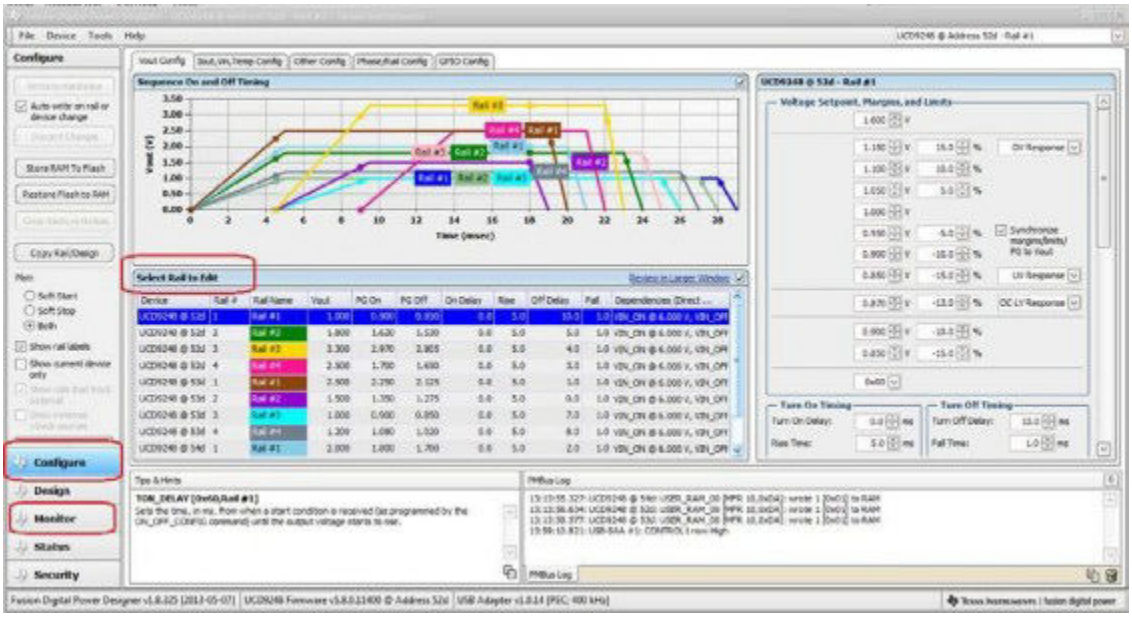


7. Select the bit file `<project_dir>/power_tutorial1/power_tutorial1.runs/impl_1/ dut_fpga.bit` and click **Program**.
8. After the program completes successfully, select **XADC (System Monitor)** in the Hardware window, right-click and select **Dashboard**, and then select **New Dashboard**.
9. Click **OK**. The System Monitor window opens and plots die temperature (junction temperature) in the graph window.

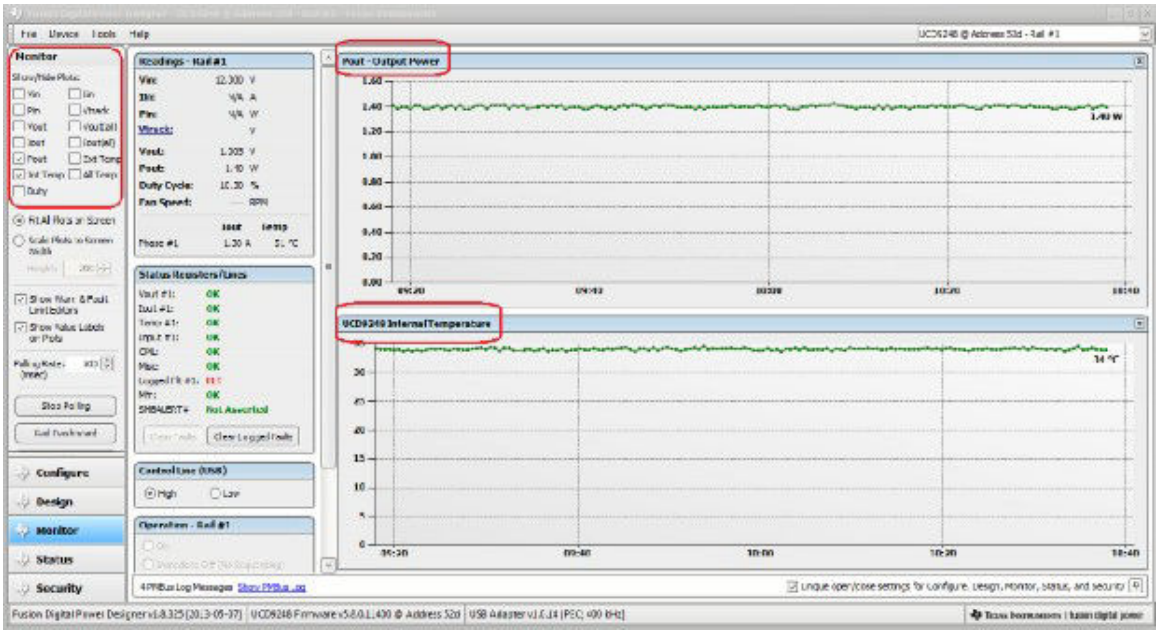


Step 5: Measuring the Hardware Power Rails

1. In the Fusion Digital Power Designer, select a rail in the Configure view and click **Monitor**.



2. Configure the parameters to be monitored. An Output Power graph will be plotted in the Monitor window.



3. Repeat the steps above to monitor the power information for each rail supplied to the device. Note that rail information is displayed in terms of regulator address.

		KC705 (UCD9248)			
device	**rail**	**purpose**	**voltage**	**IOUT_CAL_GAIN**	**IOUT_CAL_OFFSET**
52	1	VCCINT	1.0V	0xEBDC	0x8000
	2	VCCAUX	1.8V	0xEBDC	0x8000
	3	VCC3V3	3.3V	0xEBDC	0x8000
	4	VADJ	2.5V	0xEBDC	0x8000
53	1	VCC2V5	2.5V	0xEBDC	0x8000
	2	VCC1V5	1.5V	0xEBDC	0x8000
	3	MGT_AVCC	1.0V	0xEBDC	0x8000
	4	MGT_AVTT	1.2V	0xEBDC	0x8000
54	1	VCCAUX_IO	1.8V	0xEBDC	0x8000
	2	VCCBRAM	1.0V	0xEBDC	0x8000
	3	MGT_VCCAUX	1.8V	0xEBDC	0x8000
	4	*N/A*	*N/A*	*N/A*	*N/A*

4. Note the Junction Temperature value either from the Vivado Hardware Manager or from the Fusion Digital Power Designer.

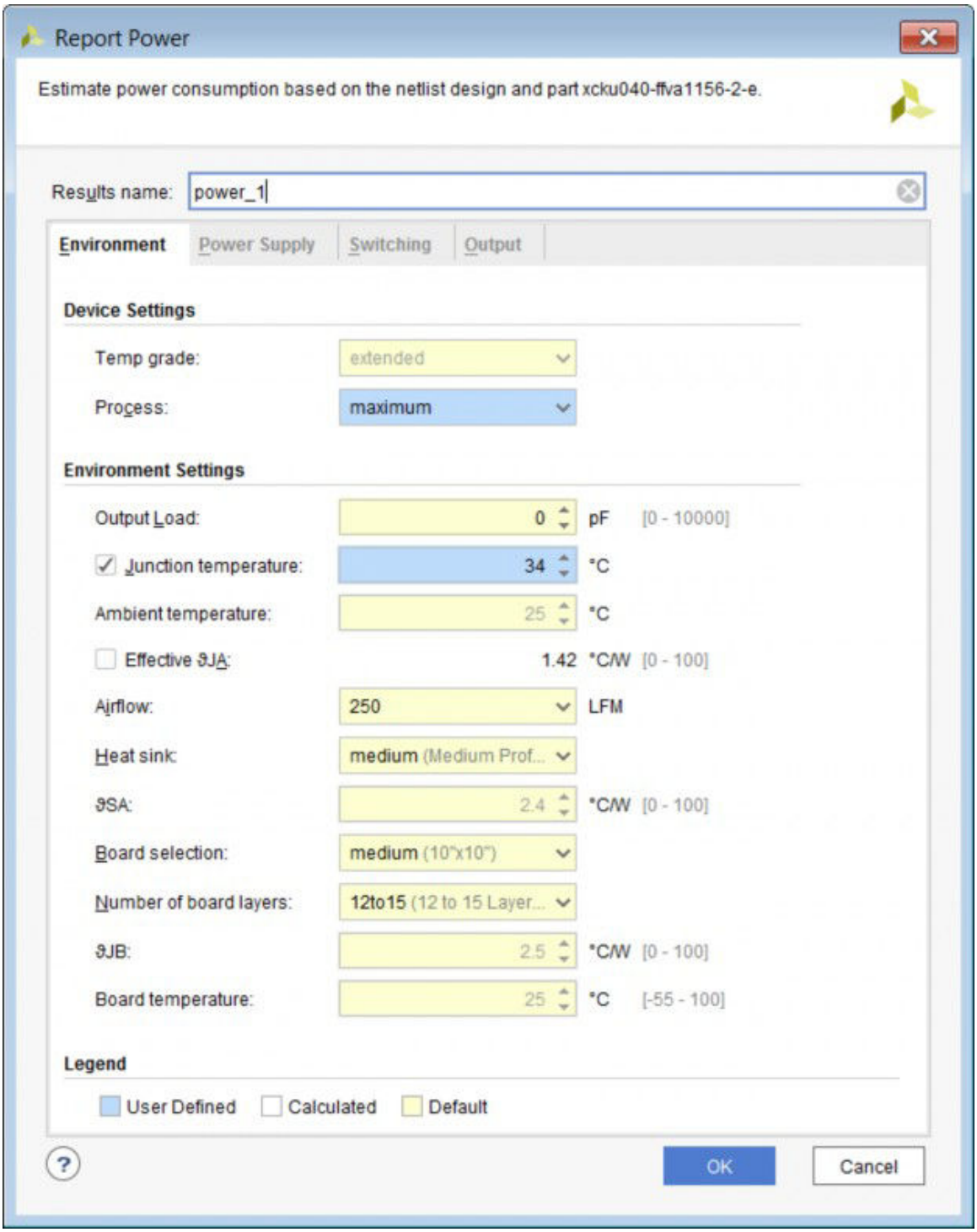
Step 6: Estimating Vectorless Power with Junction Temperature

For further Power Analysis, you can use the measured Junction Temperature and other thermal settings to feed into Vivado Report Power for better accuracy.

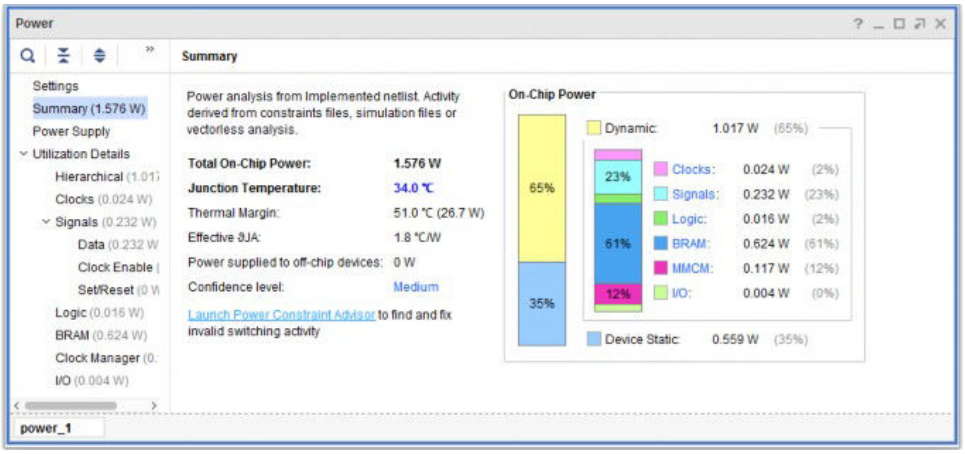
1. In the Vivado Design Suite, open the tutorial project and click **Open Implemented Design** to display the implemented design.
2. In the Tcl Console, run the following command to reset any user defined or SAIF file defined settings:

   ```
   reset_switching_activity -all
   ```

3. From the main menu, select **Reports → Report Power**.
4. In the Environment tab of the Report Power dialog box, enter the **Junction Temperature** value supplied by the hardware power measurement.
5. Set the Process to **maximum**.
6. In the Switching tab, make sure the Simulation activity file (.saif) is blank.
7. Click **OK**.



8. In the Power Report, observe that the power numbers increase slightly as compared to the vectorless power analysis using a default junction temperature value. Note that the Junction Temperature is now color coded as being user defined in the Power Report.



9. Similarly, you can overwrite the Junction Temperature setting and do a SAIF based power analysis. Note the power numbers measured and estimated on non-power optimized design.

Conclusion

In this lab, you have completed a Vivado Report Power analysis on post-synthesis and post-implementation netlist designs without Power Optimization. You also experimented with hardware power measurement using the KC705 Evaluation Board and with reading Junction Temperature for software analysis.

In lab 4, you will learn to experiment with hardware power measurement using the KCU105 Evaluation Board and with reading Junction Temperature for software analysis.

Lab 5

Measuring Hardware Power Using the KCU105 Evaluation Board

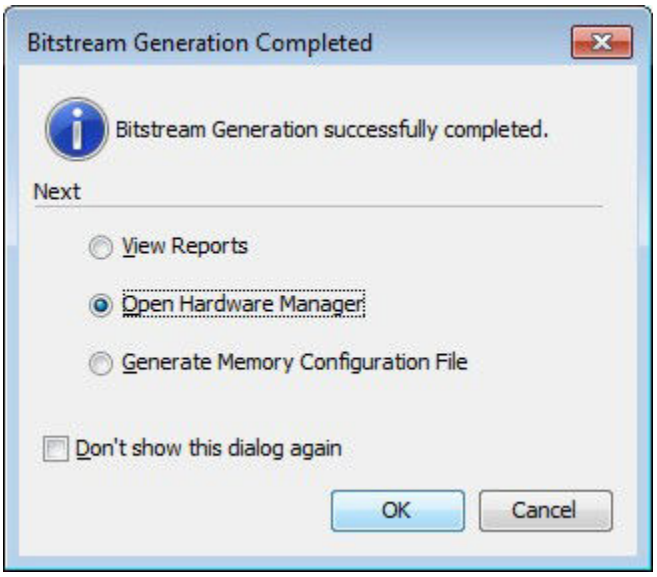
Introduction

In this lab, you will learn about the basic hardware power measurement technique and correlating the hardware power numbers with the numbers generated by Vivado® Report Power using the KCU105 evaluation board for UltraScale+™ devices. The lab will take you through the steps for setting up the hardware measurement, programing a bit file using the Vivado Hardware Manager and power measurement through the Maxim Digital Power Tool. It also includes the Junction Temperature reading from the Vivado System Monitor.

Step 1: Generating a Bit File from the Implemented Design

1. In the Vivado Design Suite, open the UltraScale™ Implemented design.
2. In the Flow Navigator, click **Generate Bitstream**.
3. When prompted to Save project before generating bitstream, click **Don't Save**.

   The Bitstream Generation Completed dialog box appears after the bitstream has been generated.

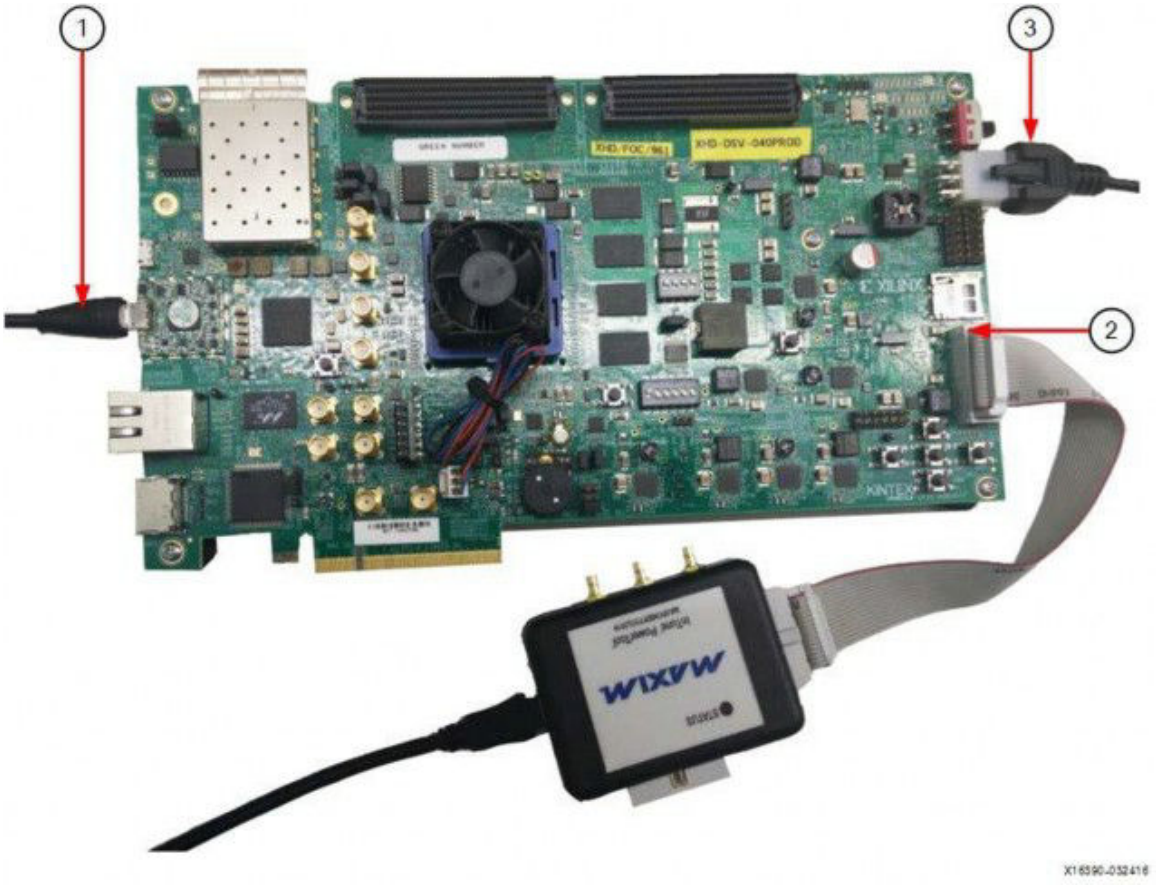


4. Select **Open Hardware Manager** in the Bitstream Generation dialog box and click **OK** to open the Hardware Manager.

Step 2: Setting up the KCU105 Evaluation Board

IMPORTANT! *This project is created for the KCU105 Rev B Evaluation Board. The pin constraints are set based on this Evaluation Board. If you are using any other Revisions, update the XDC file* `dut_fpga_kcu105.xdc` *with the correct pin constraints.*

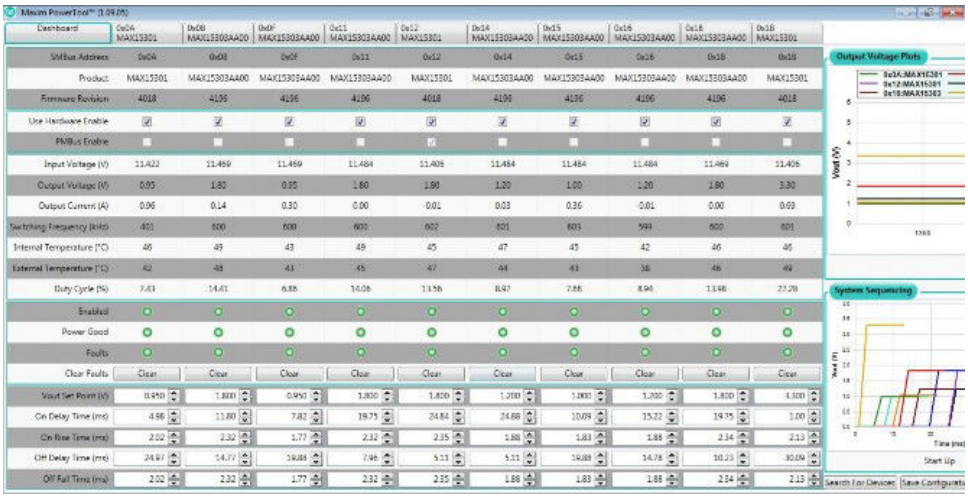
1. Connect the Digilent cable (or platform USB Cable) for programming.
2. Connect the MAXPOWERTOOL002# Interface Adapter to the PMBus port on the KCU105 Evaluation Board.
3. Connect the power cable.



4. Install the Maxim Digital Power Designer software on the PC from this location.

Step 3: Configuring the Maxim Digital Power Tool Software

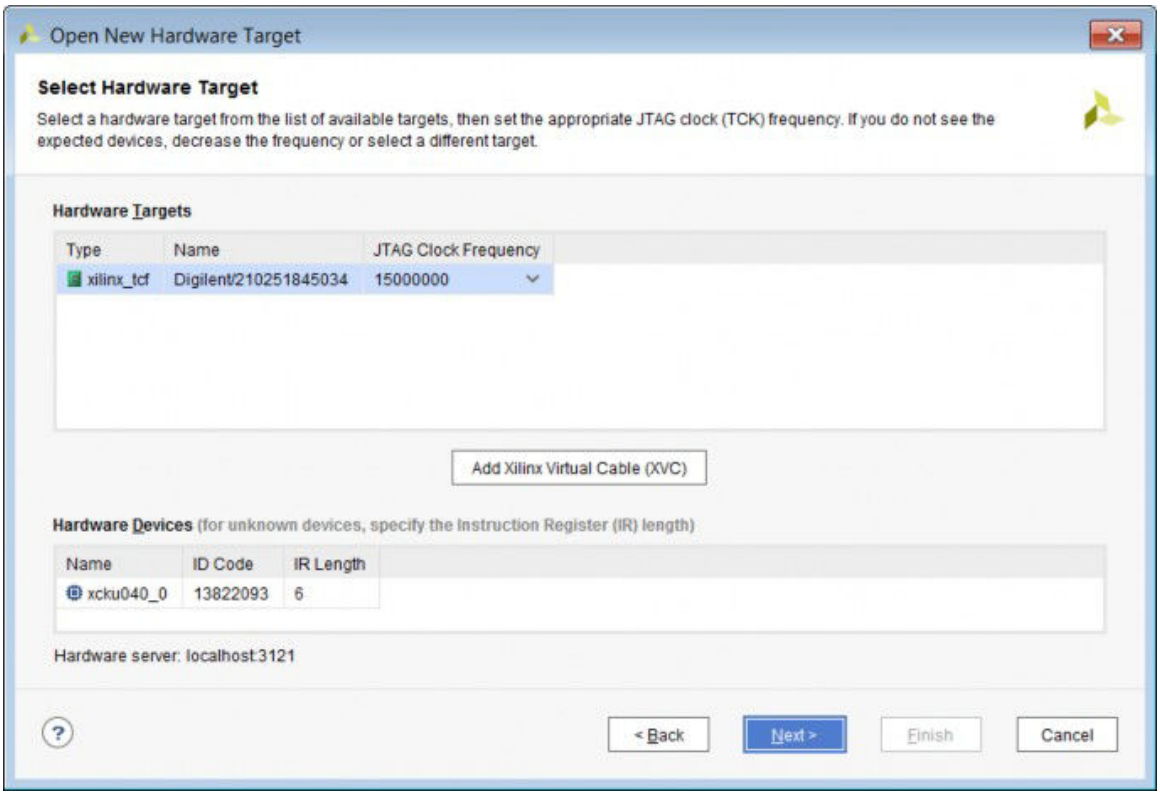
1. Power on the KCU105 Evaluation Board.
2. Open the Maxim Digital Power Tool. The software detects the Interface adapter and brings up the GUI.



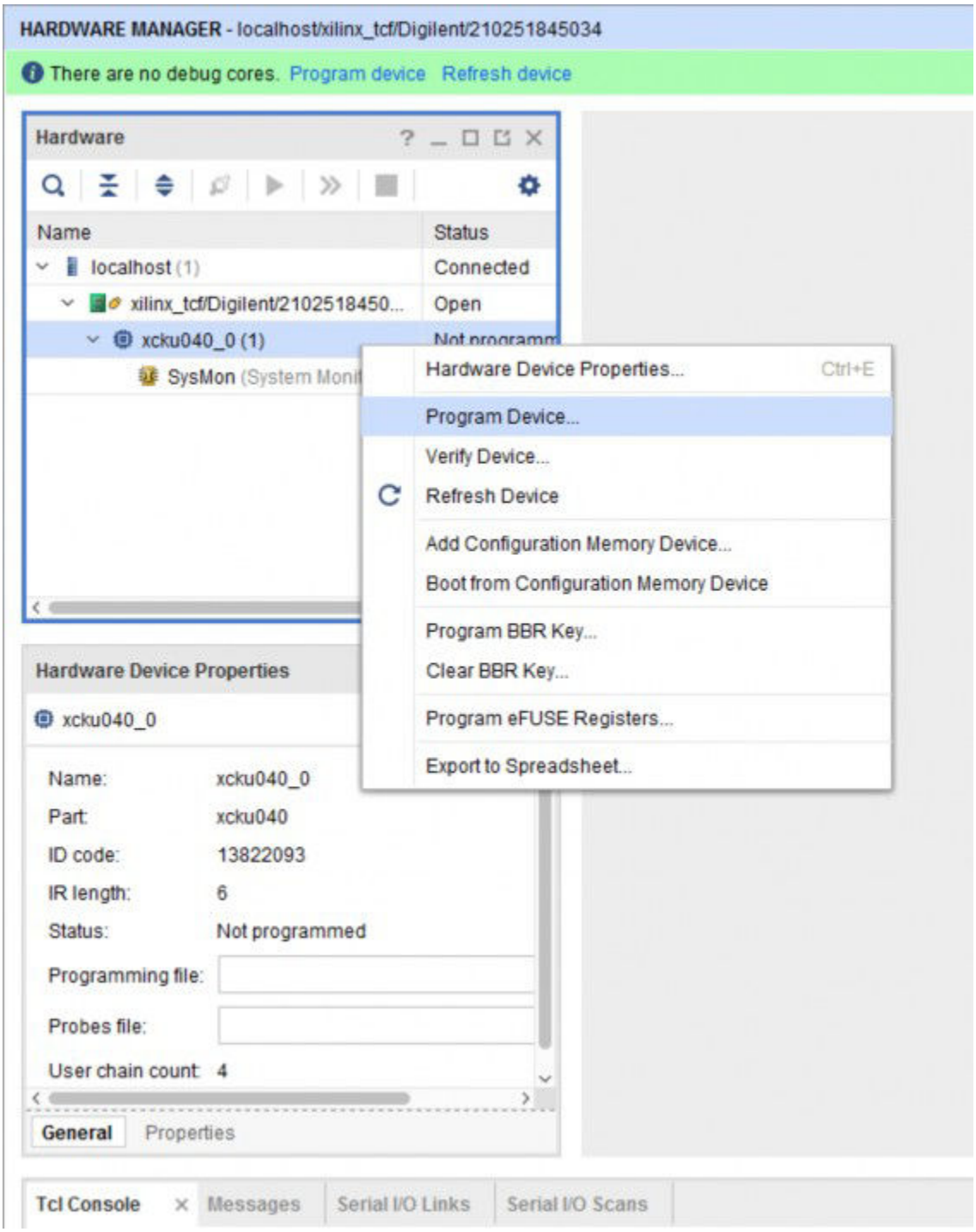
Step 4: Programming the Bitstream

1. In the Vivado Hardware Manager, click **Open Target** in the green alert bar and select **Open New Target**.
2. In the Open New Hardware Target wizard, click **Next** to go to the Hardware Server Settings page.
3. Select the server to which the board is connected.
   - If the board is connected to the local PC, select **Local Server** and click **Next**.
   - If you are connecting to a remote server, see Connecting to a Hardware Target Using `hw_server` in the *Vivado Design Suite User Guide: Programming and Debugging* (UG908) .

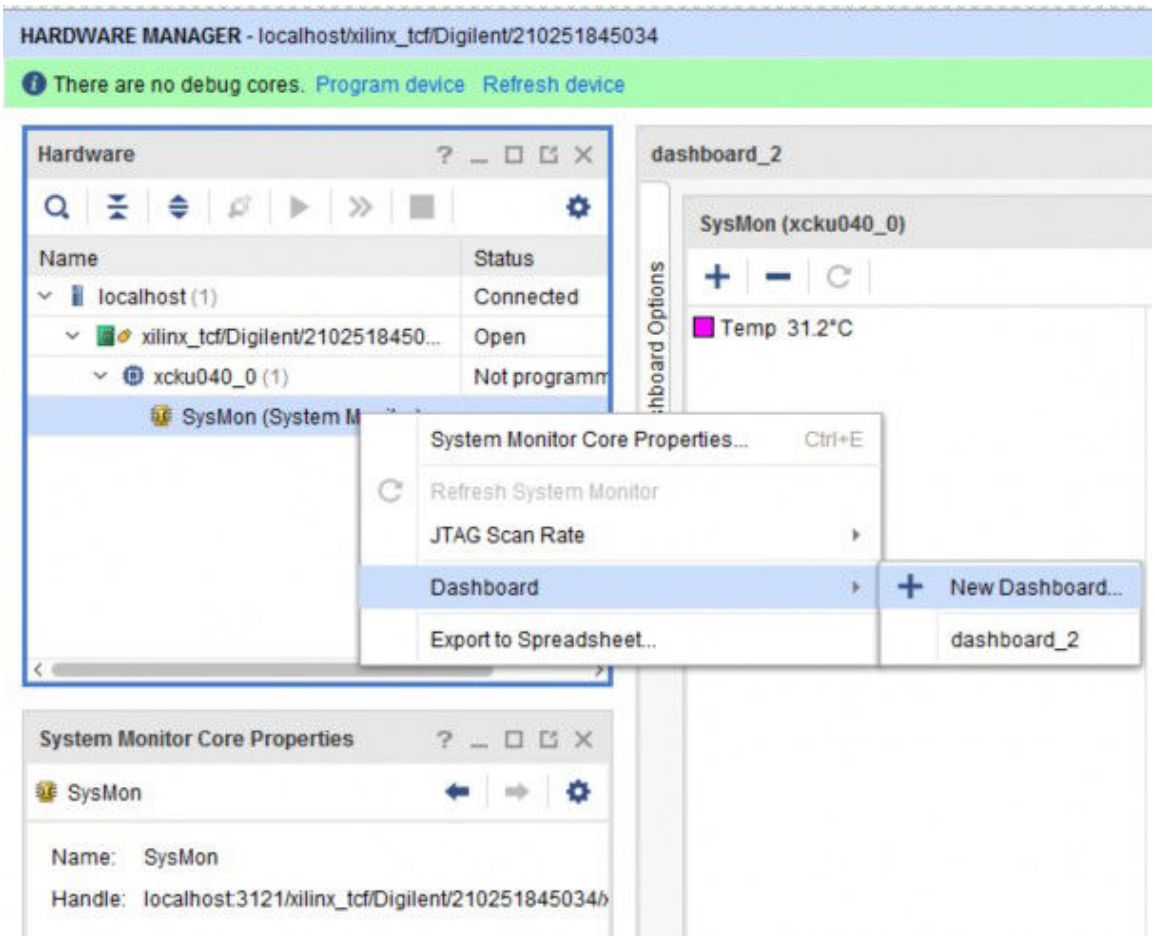
   When the hardware is detected successfully, the part information will be displayed in the Open New Hardware Target dialog box.
4. Verify the part information, then click **Next** and **Finish**.



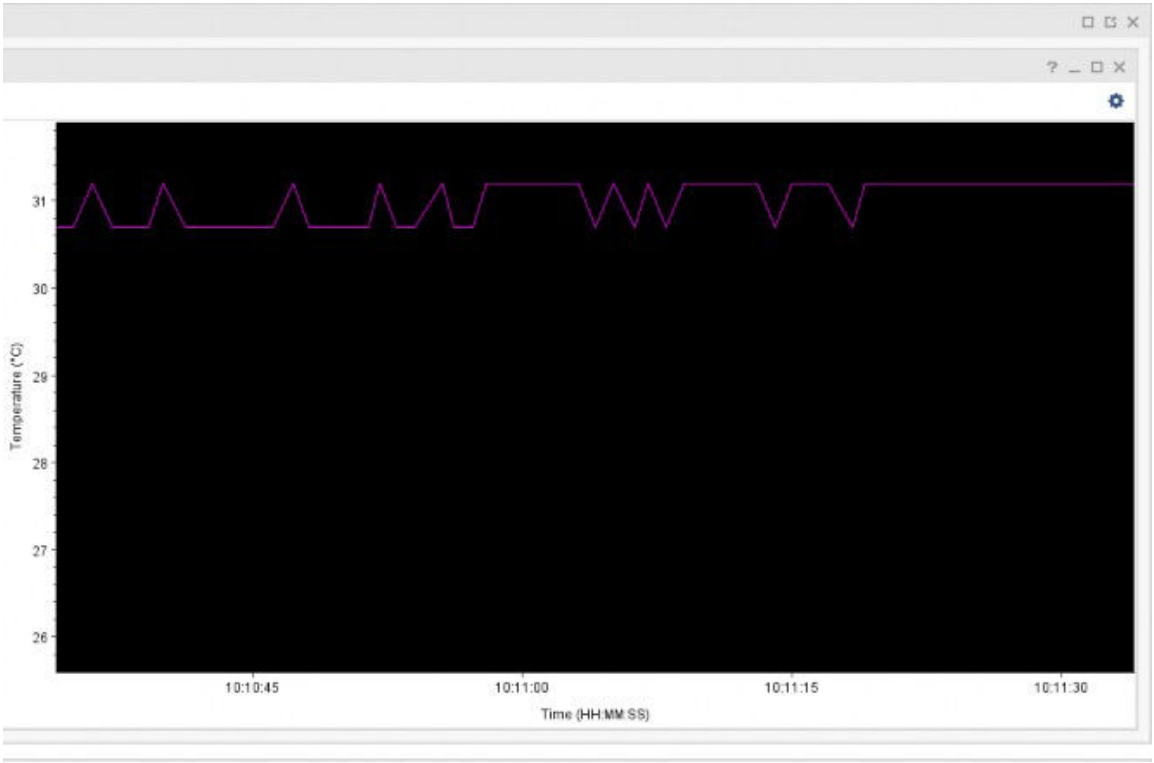
5. In the Hardware Devices window, right-click the part and select **Program Device**.



6. Select the bit file from the implementation runs directory of the project created in Lab 2 for the UltraScale™ design (`<project_dir>/power_tutorial2/power_tutorial2.runs/impl_1/ dut_fpga.bit`) and click **Program**.
7. After the program completes successfully, select **XADC (System Monitor)** in the Hardware window, right-click and select **Dashboard**, and then select **New Dashboard**.

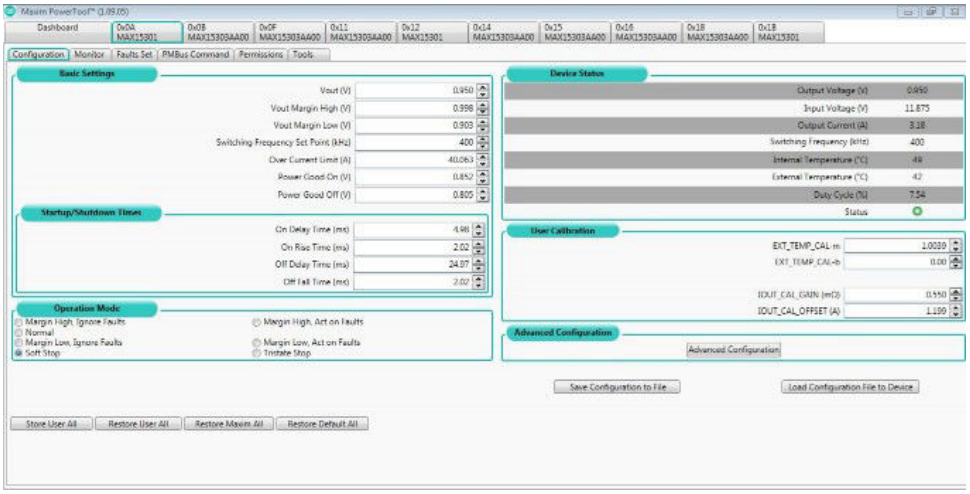


8. Click **OK**. The System Monitor window opens and plots die temperature (junction temperature) in the Graph Window.

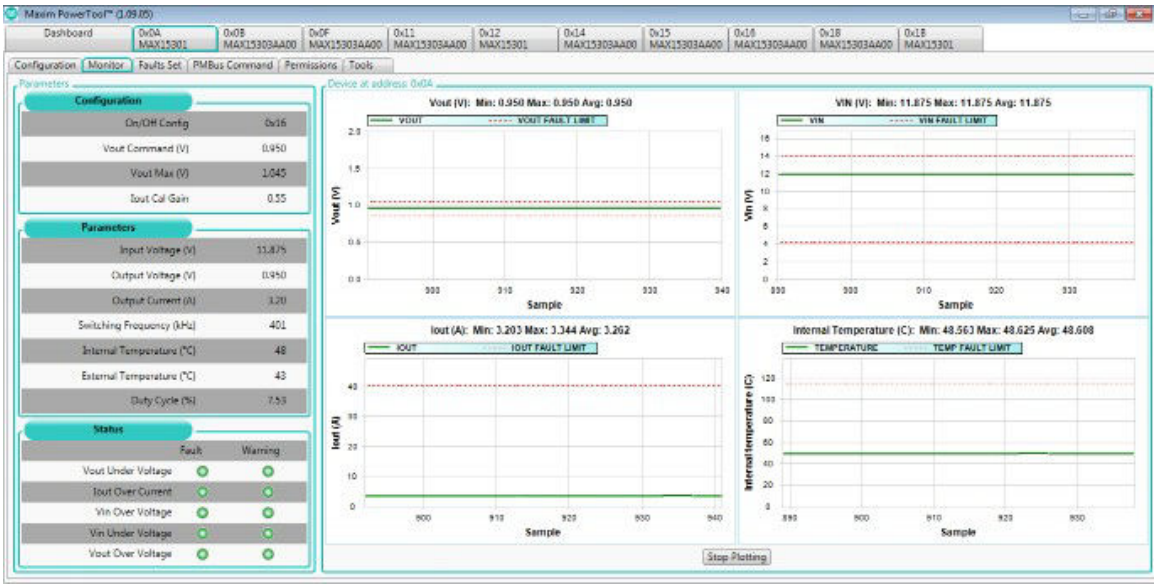


Step 5: Measuring the Hardware Power Rails

1. In the Maxim Digital Power GUI, select a rail to monitor the power information.



2. In the Configuration tab, you can observe the basic settings and device status.
3. Click **Monitor** tab to observe the voltage and current plots.



4. Repeat the steps mentioned above to monitor the power information for each rail supplied to the device.

   ***Note*:** The rail information is displayed in terms of Regulator address.

Table 2: **Rail Information**

RAIL	VOLTAGE	PMBUS ADDR
VCCINT	0.95V	0x0A
VCCAUX	1.8V	0x0B
VCCBRAM	0.95V	0x0F
VCC1V8	1.8V	0x11
VADJ_1V8	1.8V	0x12
VCC1V2	1.2V	0x14
MGTAVCC	1V	0x15
MGTAVTT	1.2V	0x16
MGTAVCCAUX	1.8V	0x18
UTIL_3V3	3.3V	0x1B

5. Note that the junction temperature value from the Vivado Hardware Manager (System Monitor).

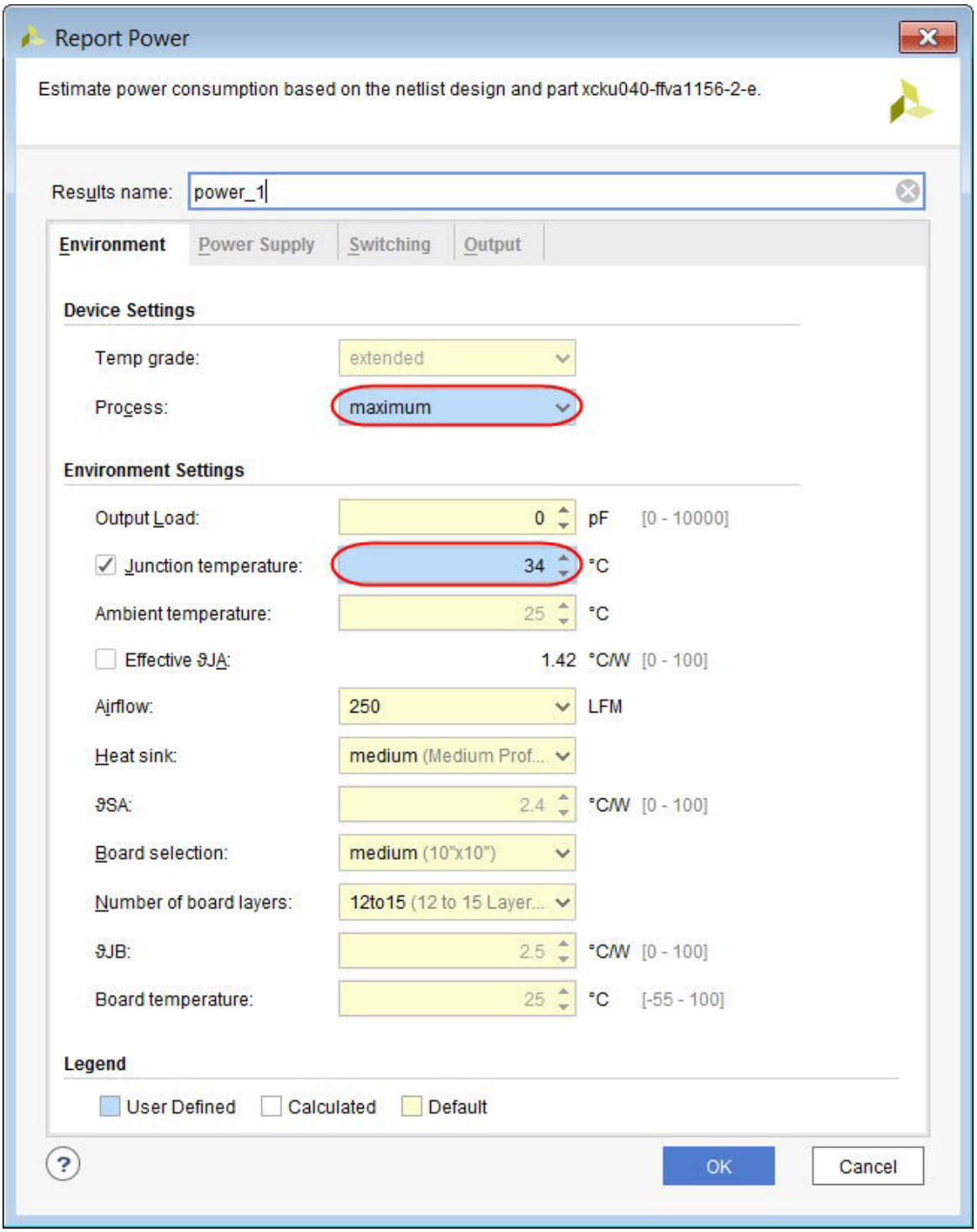
Step 6: Estimating the Vectorless Power with Junction Temperature

For further Power Analysis, you can use the measured Junction Temperature and other thermal settings to feed into Vivado Report Power for better accuracy.

1. In the Vivado Design Suite, open the tutorial project and click **Open Implemented Design** to display the implemented design.
2. In the Tcl Console, run the following command to reset any user defined or SAIF file defined settings.

   ```
   reset_switching_activity -all
   ```

3. In the main menu bar, select **Reports → Report Power**.
4. In the Environment tab of Report Power dialog box, enter the Junction Temperature value supplied by the hardware power measurement.
5. Set the Process to **maximum**.
6. In the Switching tab, make sure that the Simulation activity file (.saif) is blank.
7. Click **OK**.



8. In the Power Report, observe that the power numbers increase slightly as compared to the vectorless power analysis using a default junction temperature value.

   Note that the Junction Temperature is now color coded as being user defined in the Power Report.

9. Similarly, you can overwrite the Junction Temperature setting and do a SAIF based power analysis.

Conclusion

In this lab, you have learned to experiment with hardware power measurement using the KCU105 Evaluation Board.

In lab 5, you will learn about using the Power Optimization features in the Vivado IDE.

Lab 6

Performing Power Optimization

Introduction

In this lab, you will learn about using the Power Optimization features in Vivado® for 7 series devices. The lab will take you through the steps for invoking Power Optimization after synthesizing the design. It will also guide you on how to use the power optimization report, make decisions and selectively turn off power optimization on signals, blocks, and hierarchies.

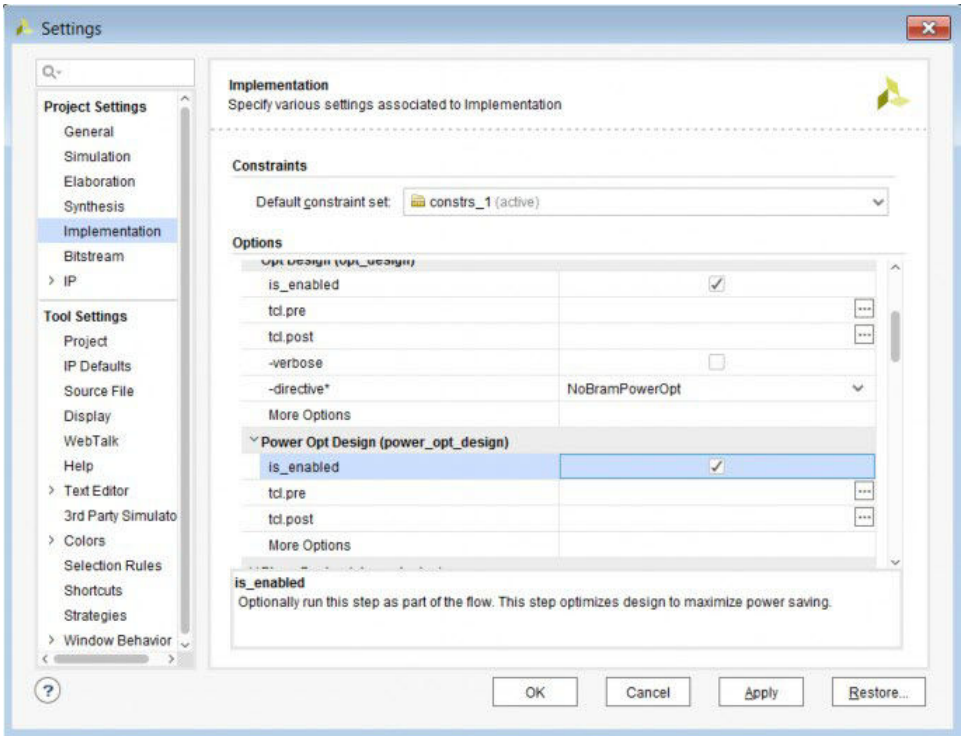
TIP: *When you run Implementation on your design, the Vivado tools may perform block RAM power optimizations by default during opt_design. These optimizations will not affect performance, and will have little impact on area and run time. In the previous Lab, the default block RAM power optimization was disabled (Step 9 of Lab 1) by setting a NoBramPowerOpt directive to opt_design.*

Step 1: Setting Up Options to Run Power Optimization

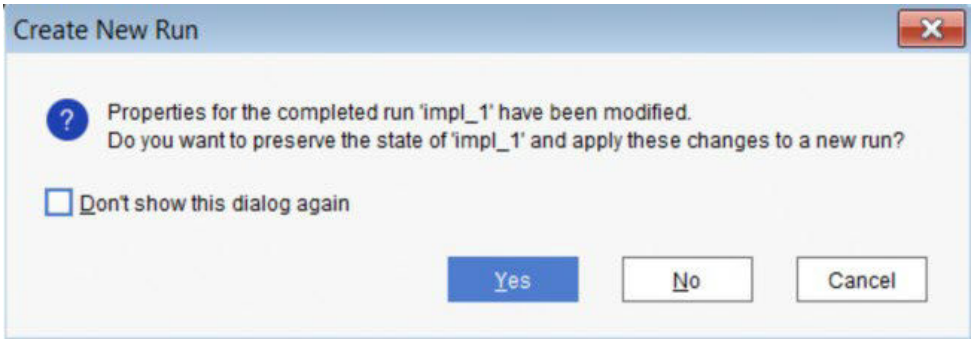
1. In the Flow Navigator, right-click **Implementation** and select **Implementation Settings**.
2. In the Project Settings dialog box, select **Implementation** tab to make the following settings:
   - In the Opt Design settings, set the **-directive** option to **Default**.

     Block RAM optimization runs in the Default setting for Opt Design during Implementation. Block RAM optimization was disabled in the previous lab. It is now re-enabled when the design runs Power Optimization.
   - In the Power Opt Design settings, check the **is_enabled** box.

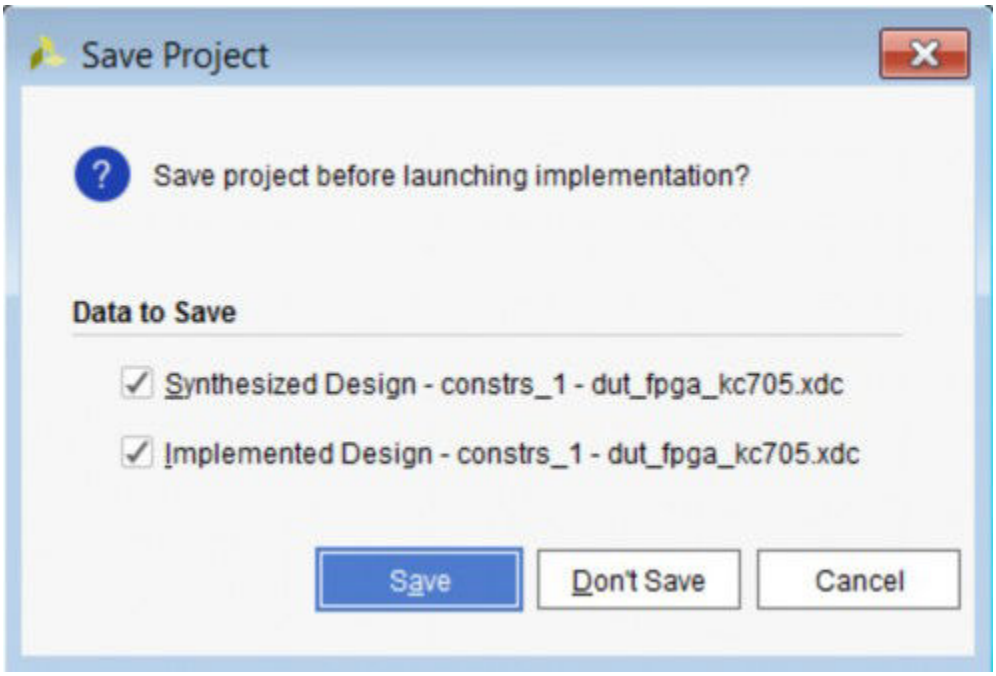
     This ensures Power Optimization runs after `opt_design`. Enabling the **Power Opt Design** option prior to `place_design` results in a complete power optimization to be performed. This option yields the best possible power saving from the Vivado tools.



3. Click **OK.**
4. In the Create New Run dialog box, click **Yes** to Properties for the completed run 'impl_1' have been modified. Do you want to preserve the state of 'impl_1' and apply these changes to a new run?.



5. In the Create Run dialog box, set the **Run Name** to `impl_2`.
6. Click **OK**.
7. In the Flow Navigator, select **Run Implementation**. Click **Don't Save** when the Save Project window pops up to save both Synthesis and Implementation constraints.

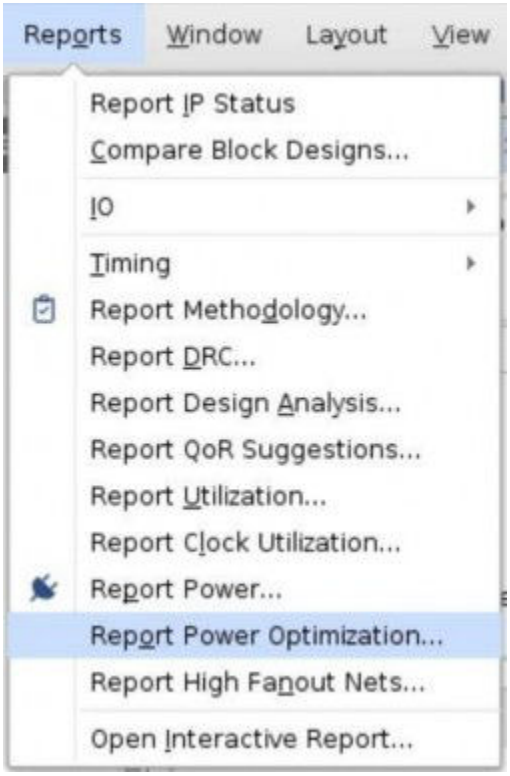


You are running Implementation with Power Optimization turned on.

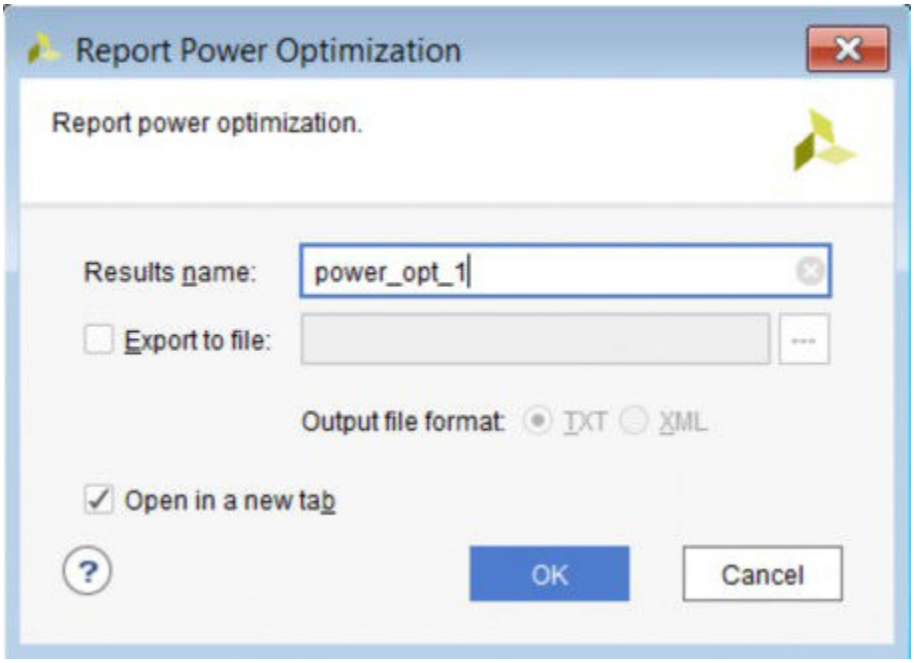
8. In the Implementation Completed dialog box, select **Open Implemented Design** and click **OK**. Click **Don't Save** when the Save Project window pops up to save both Synthesis and Implementation constraints.

Step 2: Running report_power_opt to Examine User/Design Specific Power Optimizations

1. In the Flow Navigator, select **Implemented Design**.
2. From the main menu, select **Reports → Report Power Optimization**.



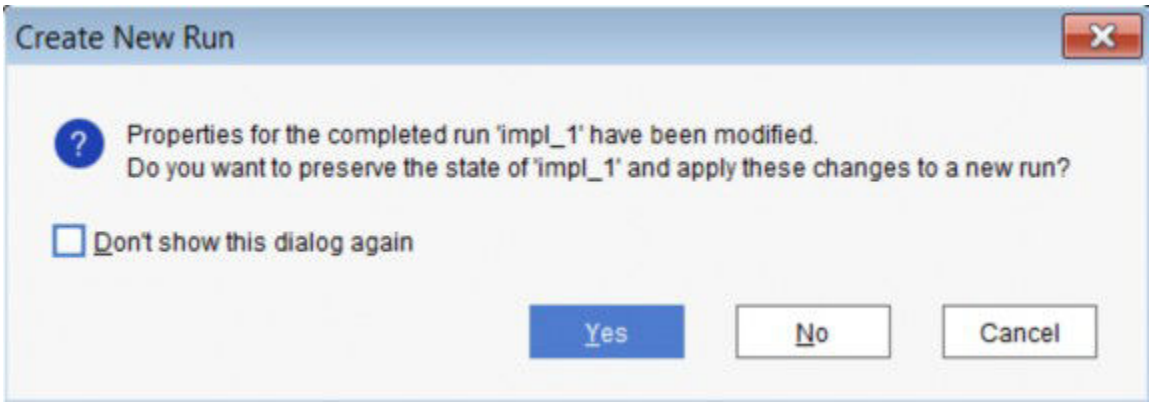
The Report Power Optimization dialog box appears, as shown in the following figure.



3. Enter `power_opt_1` for the Results name.
4. Ensure that the **Open in a new tab** option is checked.
5. Click **OK**. Alternatively, execute the following command in the Tcl Console:

```
report_power_opt -name power_opt_1
```

6. Observe the report power_opt_1 is generated in the Power Opt window. When the report opens, the Summary view is displayed in the report.
7. In the Summary view, note that 50% of the block RAMs are clock gated by the tool during power optimization.

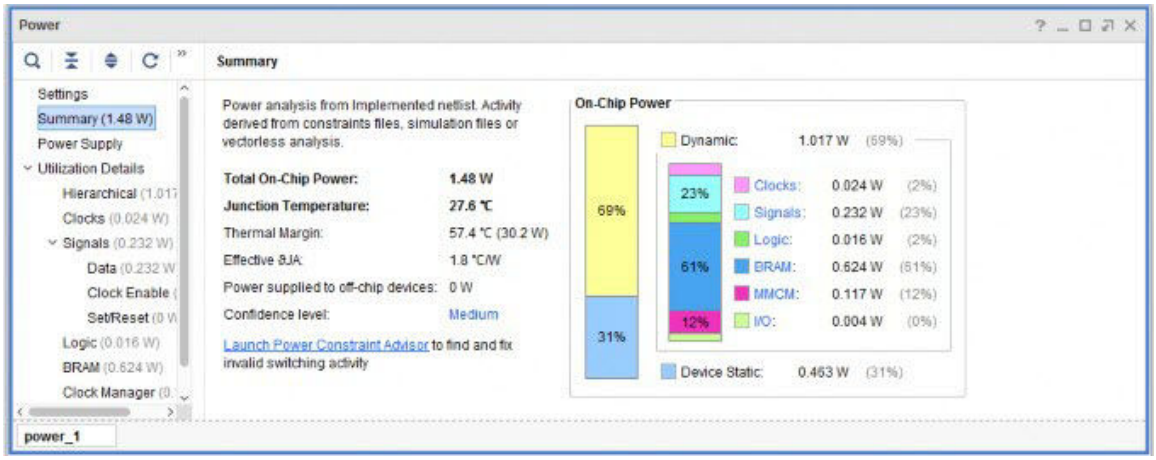


8. In the Power Optimization Report, select **Hierarchical Information → BRAMs → Tool Gated BRAMs** and observe the block RAM cells and its CE ports which are gated by the tool during the power optimization.

Step 3: Running report_power to Examine Power Savings

1. In the main menu bar, select **Reports → Report Power**.
2. In the Report Power dialog box, make the following settings
   - Specify the Results name as `power_1`.
   - In the Environment tab, make sure the Process is set to **maximum**.
3. Click **OK**. Alternatively, in the Tcl Console execute this Tcl command:
4. In the Summary view of the Power Report, observe an approximately 100-200mW power savings compared to the non-optimized power run in the previous lab.

   You can generate a bitstream to program the hardware and measure its power, to observe the power saving in hardware. See Lab 4: Measuring Hardware Power Using the KC705 Evaluation Board for hardware power measurement instructions.



Step 4: Turning Off Optimizations on Specific Signals and Rerunning the Implementation

In this step you will learn how to turn off the power optimization on specific block RAMs.

> **IMPORTANT!** *Power optimization works to minimize the impact on timing while maximizing power savings. However, in certain cases, if timing degrades after power optimization, you can identify and apply power optimizations only on non-timing critical clock domains or modules using the* `set_power_opt` *XDC command.*
>
> *See the Vivado Design Suite User Guide: Power Analysis and Optimization (UG907) for more information on the* `set_power_opt` *command.*

Assume that this block RAM is in the critical path:

```
dut/gen_dut[0].bram_top_inst/bram_inst/mem_reg_0_0
```

This step makes sure the tool does not gate this block RAM.

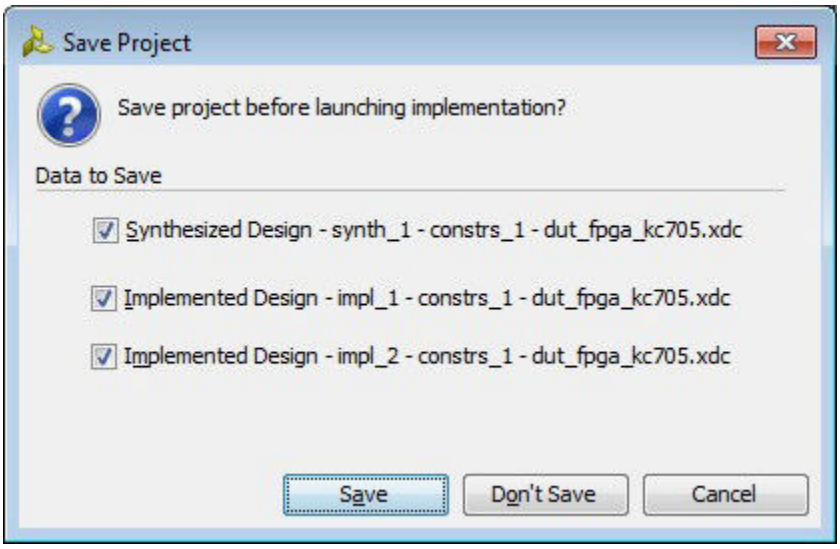
1. In the Tcl Console, type this command:

   ```
   set_power_opt -exclude_cells [get_cells dut/gen_dut[0].bram_top_inst/
   bram_inst/mem_reg_0_0]
   ```

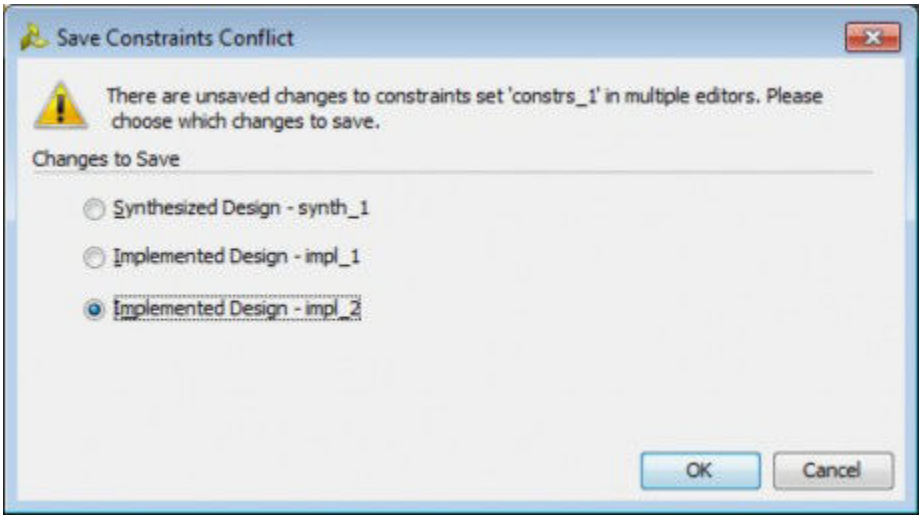
   This will prevent the tool from gating this block RAM.

2. From the Flow Navigator choose **Run Implementation**, which in turn reruns `power_opt_design`.

3. Click **Save** in the Save Project dialog box to save the synthesized design and implemented design constraints before launching implementation.



Also, select **Implemented Design – impl_2** in the Save Constraints Conflict dialog box to save the changes in constraints from the `set_power_opt` command.



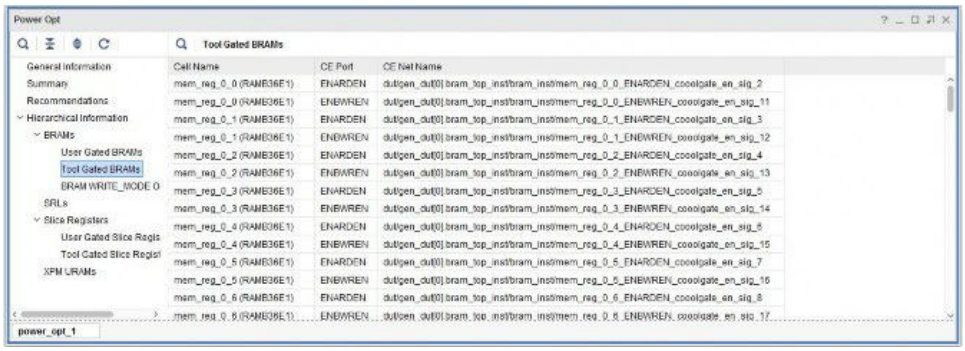
4. In the Implementation Completed dialog box, select **Open Implemented Design** and click **OK**.

Step 5: Running report_power_opt to Examine Tool Optimizations Again

1. In the main menu bar, select **Reports → Report Power Optimization**.
2. In the Report Power Optimization dialog box, type in the Results name as **power_opt_2**. Alternatively, execute this Tcl command in the Tcl Console:

```
report_power_opt -name power_opt_2
```

3. In the generated report power_opt_2 in the Power Opt window, display **Tool Gated BRAMs**.



Note that this block RAM is no longer in the list of Tool Gated BRAMs: `dut/gen_dut[0].bram_top_inst/bram_inst/mem_reg_0_0`

Step 6: Saving Power using UltraScale Block RAM in Cascaded Mode

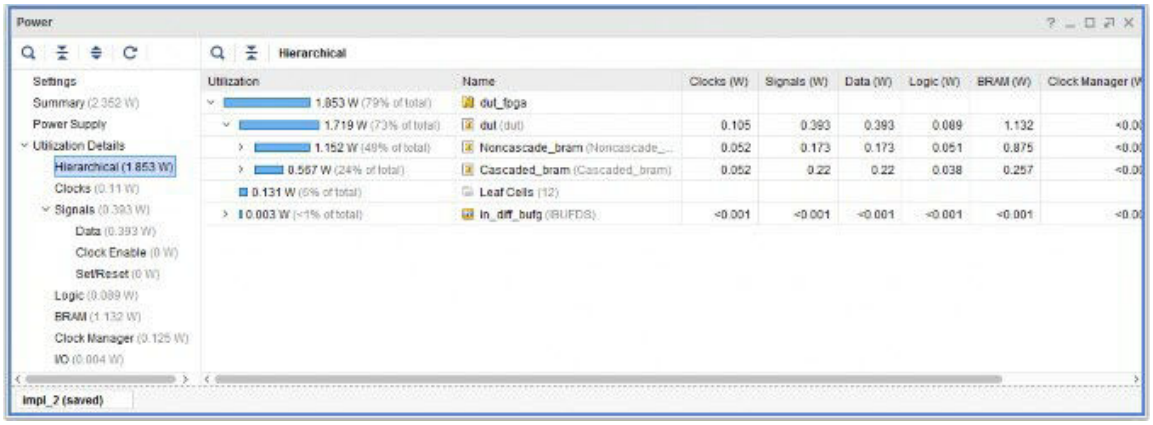
UltraScale architecture-based devices provide the capability to cascade the data out from one block RAM to the next block RAM serially. This will enable the devices to create a deeper block RAM in a bottom-up fashion. When used in cascaded mode, the power consumption is considerably low compared to the block RAM used in non-cascaded mode.

1. Run the steps mentioned in Step 1 shown in Lab 1.
   a. In the Add Source Files dialog box, add the source files in the `<Extract_Dir>/UltraScale/src` for UltraScale devices.
   b. In the Add Constraints (optional) page, click **Add Files** and select `dut_fpga_kcu105.xdc` in the file browser. In the directory structure, you will find the `dut_fpga_kcu105.xdc` file below the /src folder.
   c. Select the Kintex UltraScale KCU105 Evaluation Platform (xcku040-ffva156-2-e FPGA), click **Next**.
2. Review the New Project Summary page. Verify that the data appears as expected and click **Finish**.
3. In the Vivado Settings dialog box (**Tools → Options → General**), enter the tutorial project directory in the Specify project directory box, so that all reports are saved in the tutorial project directory. Then click **OK**.
4. Click **Run Synthesis** in the Flow Navigator.

   The Synthesis Completed dialog box appears after synthesis has completed on the design.
5. Select **Run Implementation** in the Synthesis Completed dialog box and click **OK**.

6. After the Implementation completes, click **Open Implemented Design**.
7. You can see the automatically generated power report impl_1 in the Power window, which shows as a saved report. This is an autogenerated vectorless power report.
8. Note the total power (Total On-Chip Power) in the power report Summary view.

9. Select **Hierarchical** view under **Utilization Details** on the left panel and observe the cascaded and non-cascaded block RAM power.



10. You can see 50% to 60% saving in cascaded block RAM compared to non-cascaded block RAM.
11. Use the same steps as specified in Step 1, Step 2, and Step 3 to perform SAIF based power analysis using Vivado Simulator.

Conclusion

In this tutorial, we have accomplished the following:

- Used the Report Power dialog box to verify and set device, thermal, and environmental conditions that contribute to power estimation.

- Synthesized the design and estimated the power after synthesis.
- Set switching activities on an I/O port and reran Report Power.
- Ran functional simulation using the Vivado simulator and generated a SAIF file that is input to Report Power for a more accurate power analysis.
- Implemented the design, ran post-implementation timing simulation using the Vivado simulator, and generated a SAIF file that is input to report power for a more accurate power analysis.
- Ran Questa Advanced Simulator post-implementation timing simulation and generated a SAIF file that is input to report power for a more accurate power analysis.
- Performed power measurement on the design implemented in a KC705 and KCU105 Evaluation Boards. Compared the hardware power numbers with the numbers generated by Vivado Report Power.
- Learned how to achieve power optimization as part of an implementation run.
- Examined the power optimization report and selectively turned off power optimizations on a cell in the design.
- Examined the power saving of UltraScale block RAMs in cascaded mode when compared to block RAMs in Non-cascaded mode.

XILINX

Appendix A

Additional Resources and Legal Notices

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see Xilinx Support.

Documentation Navigator and Design Hubs

Xilinx® Documentation Navigator (DocNav) provides access to Xilinx documents, videos, and support resources, which you can filter and search to find information. To open DocNav:

- From the Vivado® IDE, select **Help → Documentation and Tutorials**.
- On Windows, select **Start → All Programs → Xilinx Design Tools → DocNav**.
- At the Linux command prompt, enter `docnav`.

Xilinx Design Hubs provide links to documentation organized by design tasks and other topics, which you can use to learn key concepts and address frequently asked questions. To access the Design Hubs:

- In DocNav, click the **Design Hubs View** tab.
- On the Xilinx website, see the Design Hubs page.

Note: For more information on DocNav, see the Documentation Navigator page on the Xilinx website.

References

These documents provide supplemental material useful with this guide:

1. *Vivado Design Suite User Guide: Power Analysis and Optimization* (UG907)
2. *Vivado Design Suite User Guide: Programming and Debugging* (UG908)
3. *Vivado Design Suite User Guide: Release Notes, Installation, and Licensing* (UG973)
4. *Xilinx Power Estimator User Guide* (UG440)

Please Read: Important Legal Notices

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 Xilinx's limited warranty, please refer to Xilinx's Terms of Sale which can be viewed at https://www.xilinx.com/legal.htm#tos; 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 such critical applications, please refer to Xilinx's Terms of Sale which can be viewed at https://www.xilinx.com/legal.htm#tos.

AUTOMOTIVE APPLICATIONS DISCLAIMER

AUTOMOTIVE PRODUCTS (IDENTIFIED AS "XA" IN THE PART NUMBER) ARE NOT WARRANTED FOR USE IN THE DEPLOYMENT OF AIRBAGS OR FOR USE IN APPLICATIONS THAT AFFECT CONTROL OF A VEHICLE ("SAFETY APPLICATION") UNLESS THERE IS A SAFETY CONCEPT OR REDUNDANCY FEATURE CONSISTENT WITH THE ISO 26262 AUTOMOTIVE SAFETY STANDARD ("SAFETY DESIGN"). CUSTOMER SHALL, PRIOR TO USING OR DISTRIBUTING ANY SYSTEMS THAT INCORPORATE PRODUCTS, THOROUGHLY TEST SUCH SYSTEMS FOR SAFETY PURPOSES. USE OF PRODUCTS IN A SAFETY APPLICATION WITHOUT A SAFETY DESIGN IS FULLY AT THE RISK OF CUSTOMER, SUBJECT ONLY TO APPLICABLE LAWS AND REGULATIONS GOVERNING LIMITATIONS ON PRODUCT LIABILITY.

Copyright

© Copyright 2012-2021 Xilinx, Inc. Xilinx, the Xilinx logo, Alveo, Artix, Kintex, Spartan, Versal, Virtex, Vivado, 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.