Vivado Design Suite Tutorial

Power Analysis and Optimization

UG997 (v2024.2) November 13, 2024

AMD Adaptive Computing is creating an environment where employees, customers, and partners feel welcome and included. To that end, we're removing non-inclusive language from our products and related collateral. We've launched an internal

UG997 (v2025.2) November 20, 2025

language in our older products as we work to make these changes and align with evolving industry standards. Follow this link for more information.

Table of Contents

Lab 1

Power Analysis and Optimization Tutorial

This tutorial introduces the power analysis and optimization model recommended for use with the AMD 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 that 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).

Locating Tutorial Design Files

1. Download the reference design files:

   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 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 AMD Versal™ adaptive SoC and AMD UltraScale+™ design files.

UltraScale+ Tutorial Design Files

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

The following table describes the contents of the AMD UltraScale+™ tutorial design files:

Files	Description
`/src`	Contains the design HDL and test bench 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_zcu102.xdc`	Contains clocking and timing constraints for the design
`/src/testbench.v`	Test Bench for simulating the design

Versal Device Tutorial Design Files

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

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

Files	Description
`design.tcl`	Tcl script to create design using IPI

Lab 2

Running Power Analysis in the Vivado Tool

Introduction

In this lab, you learn about the Power Analysis and Optimization features in the AMD Vivado™ IDE. The lab walks you through the project creation and power analysis steps at the synthesis stage, using the Vivado Report Power feature in the vectorless mode. It also demonstrates using the Switching Activity Interchange Format (SAIF) file that is generated from the post-synthesis functional simulation for Vivado report power analysis.

You analyze the power of the design in Vivado IDE, and examine some of the major features in the Report Power window, and closely examine some power specific Tcl commands. You also learn how to simulate the design in the timing simulation stage using both the Vivado simulator and Questa Advanced Simulator to create a SAIF file.

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

Designing with Versal Devices

Step 1: Creating a New Project

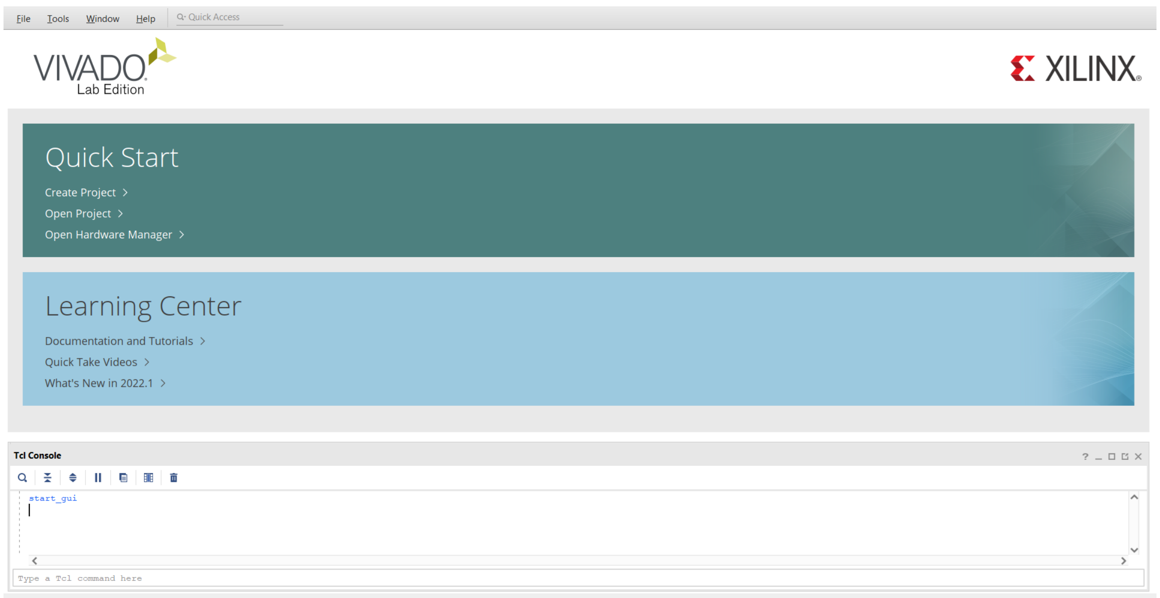
The IP integrator (IPI) is used to create the design.

On Linux, do the following.

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

   cd <Extract_Dir>/Versal (for Versal devices)

2. Launch Vivado IDE: vivado

   On Windows, do the following.

3. Launch the Vivado IDE by selecting **Start→ All Programs→ Xilinx Design Tools→ Vivado 2024.x→ Vivado 2024.x** (x denotes the latest version of Vivado 2024 IDE).

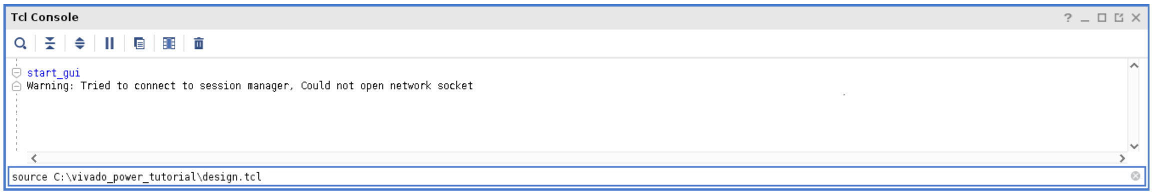
   As an alternative, click the Vivado 2024.x Desktop icon to start Vivado IDE.

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

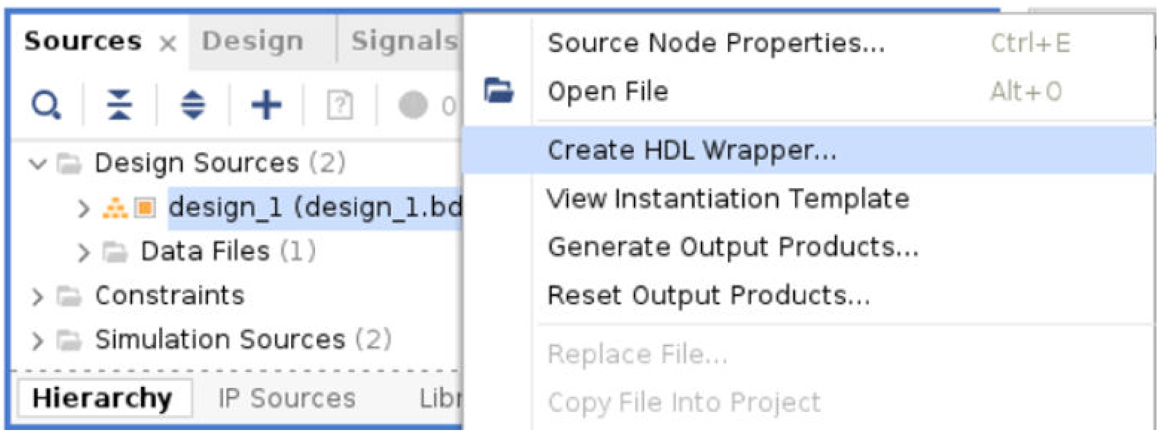
4. In the Getting Started page, click in **Tcl Console** to type the command.

5. Type the following command to generate a block design (BD): source <Extract_Dir/Versal/design.tcl.

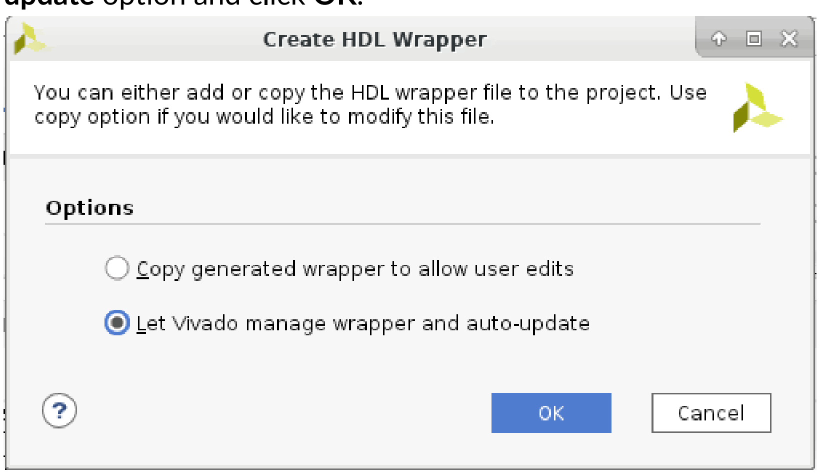
   ***Note*:** It might take a moment for the design to initialize in Vivado IDE.



6. When the block design is generated, you can find the block design file (`design_1.bd`) in the Sources window. A top-level HDL wrapper around the block design is needed because a BD source cannot be synthesized.
7. To generate the HDL wrapper:
   a. Right-click on your block design source file (`design_1.bd`) under Design Sources drop-down.
   b. Click **Create HDL Wrapper** option.



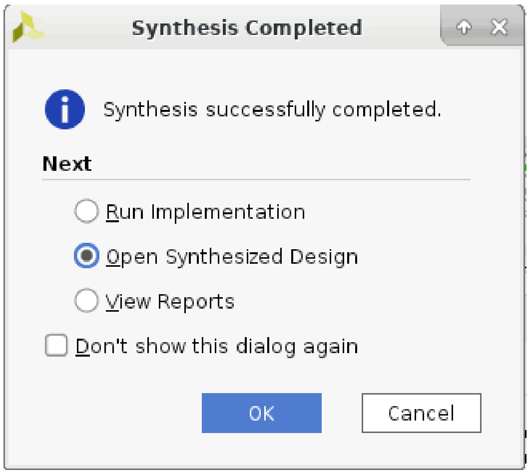
   c. In the Create HDL Wrapper dialog box, select the **Let Vivado manage wrapper and auto-update** option and click **OK**.



The design is now ready for synthesis.

Step 2: Synthesizing the Design

1. Click **Run Synthesis** in the Flow Navigator.
2. The Synthesis Completed dialog box is displayed after the synthesis is completed on the design.



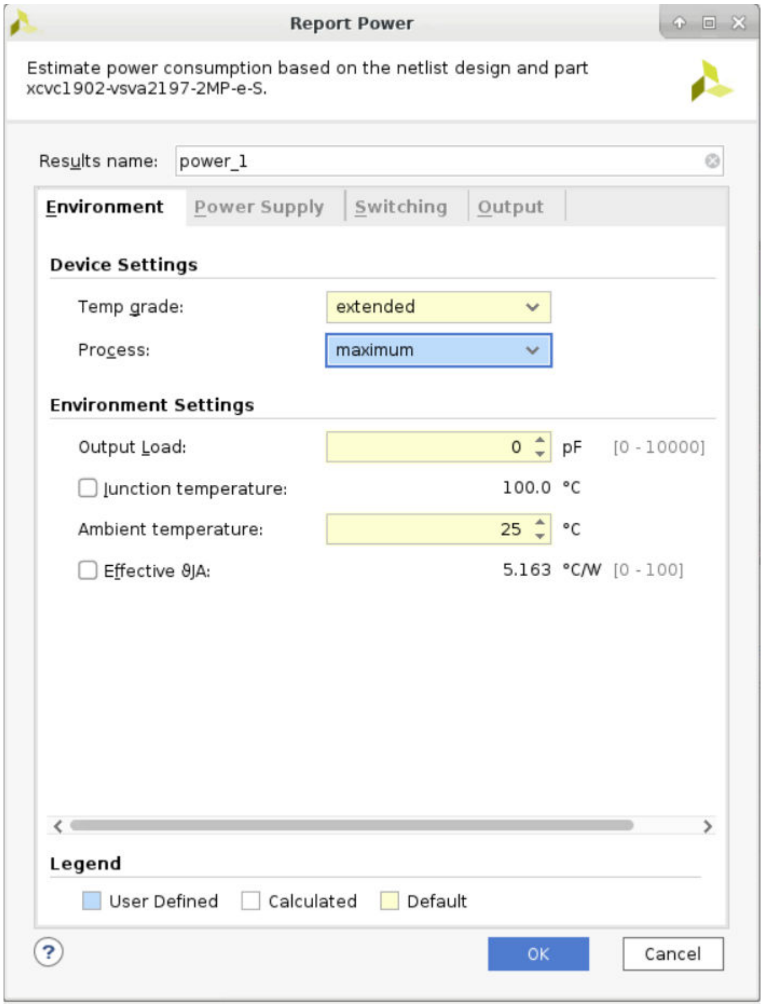
3. Select **Open Synthesized Design** in the Synthesis Completed dialog box and click **OK** to open the synthesized design.

Step 3: Report Power Setup

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

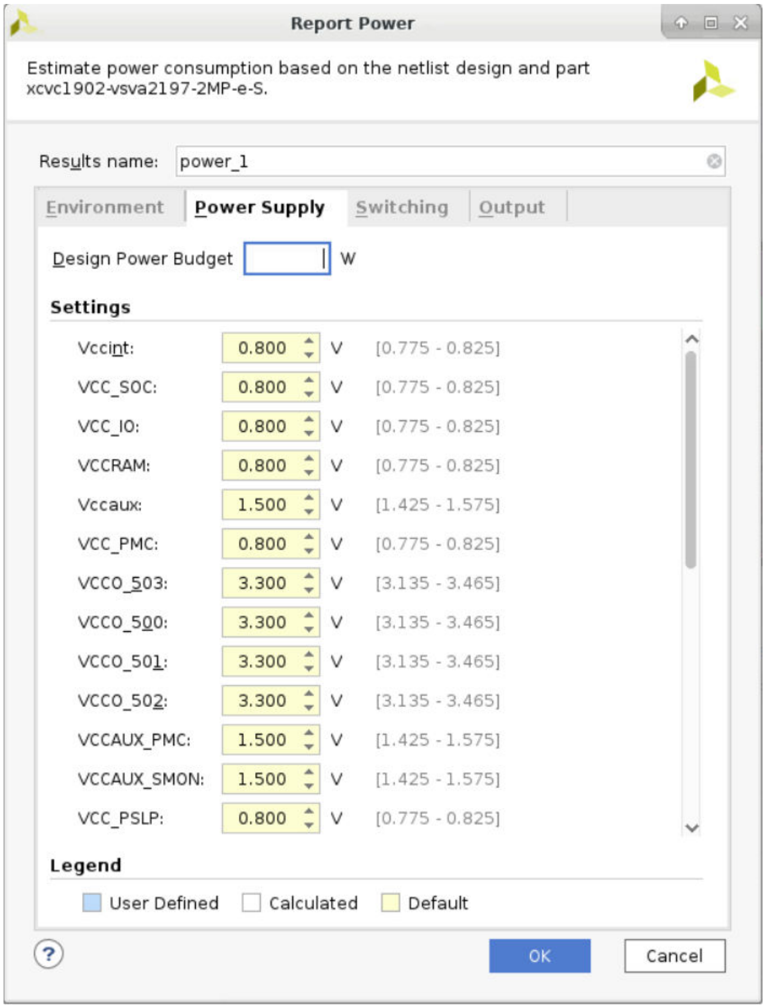
In the Vivado IDE, you can configure thermal, environmental, and power supply options to mimic the board level settings as closely as possible. For information on setting these options, see the *Vivado Design Suite User Guide: Power Analysis and Optimization* (UG907).

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



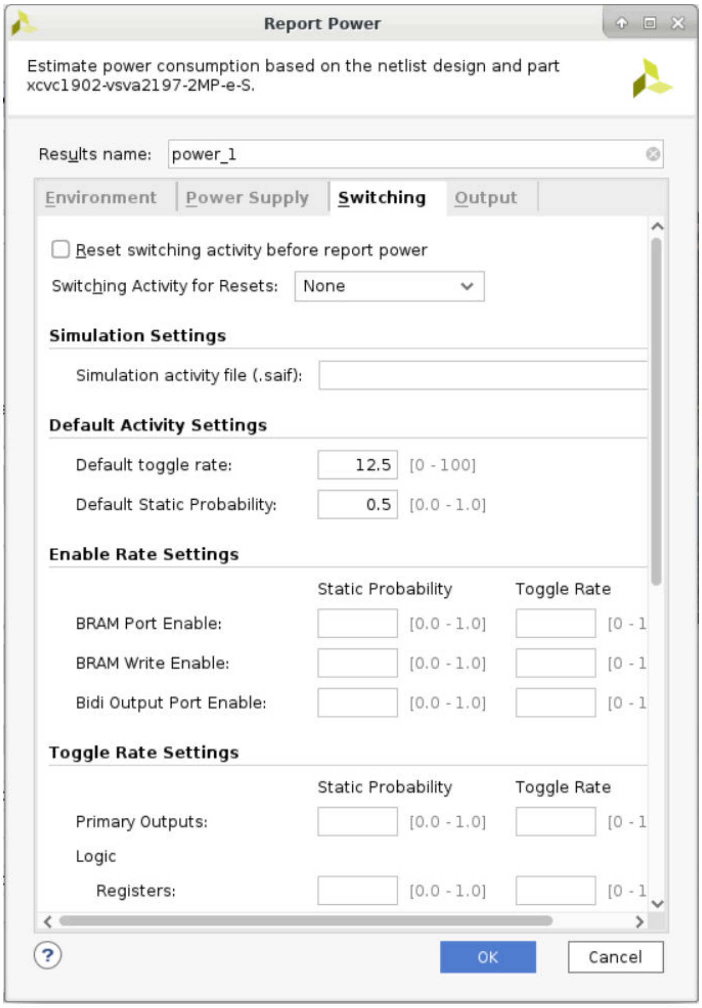
3. In the Environment tab, set **Process** to maximum for worst case power analysis. Examine the Power Supply tab.

> **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 that the temperature and other environmental factors match the actual operating conditions.*

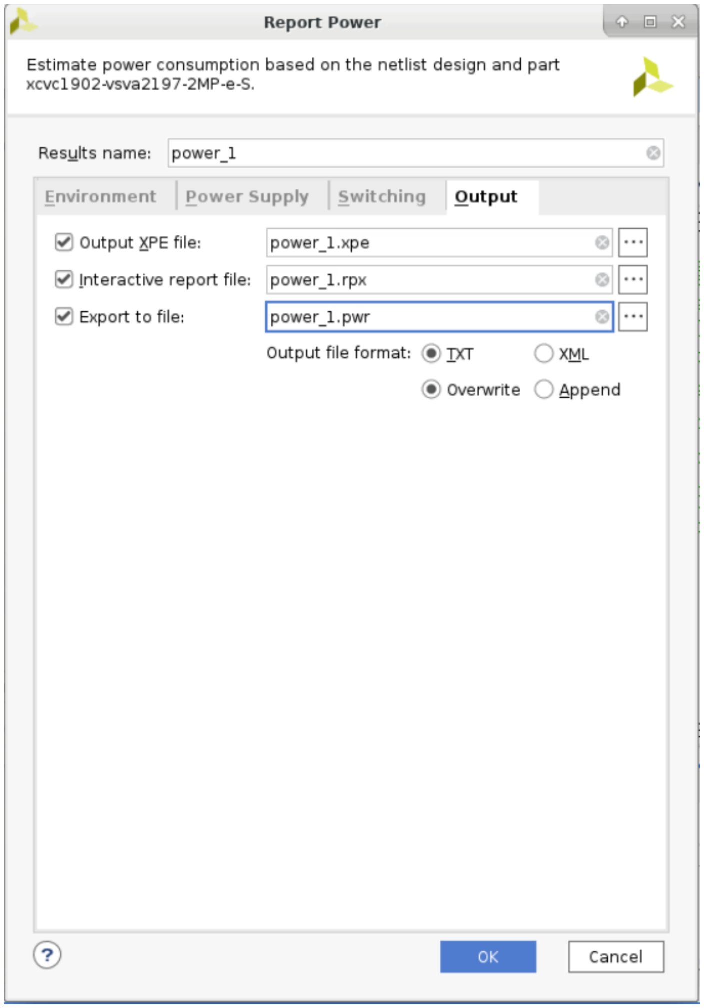


4. In the Switching tab, expand Constrained Clocks and examine the constrained clocks in the design.

> **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.*
>
> Make sure that the Default toggle rate is set to 12.5% and Default Static Probability is set to 0.5. This is applied to primary input ports (non-clock) and black box outputs.



5. In the Output tab of the Report Power dialog box, specify **Export to 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 of the file into the Xilinx Power Estimator. For information on importing the file into the Xilinx Power Estimator, see the *Xilinx Power Estimator User Guide* (UG440).
7. Specify the RPX file by setting Interactive report file as power_1.rpx 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 or 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 timing constraints. The defined frequency of a clock determines the switching activity.
- **Simulation:** 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.

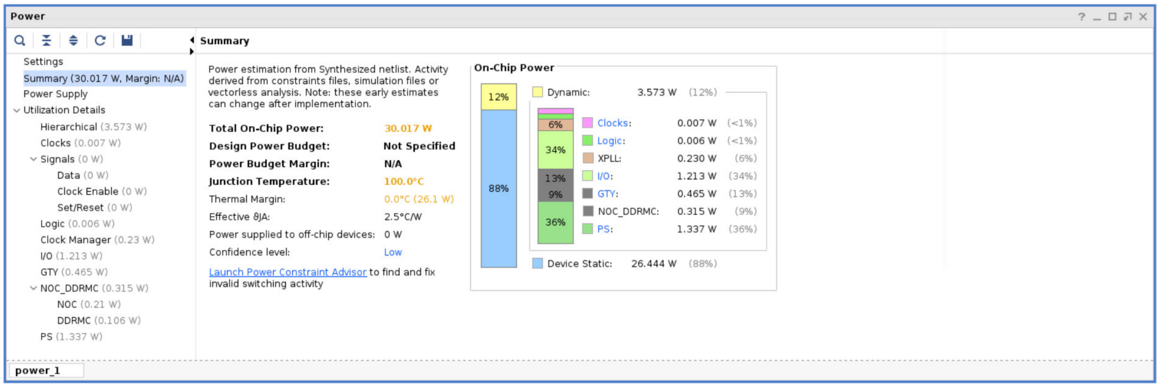
Step 4: Running Report Power

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

   This runs the `report_power` command.

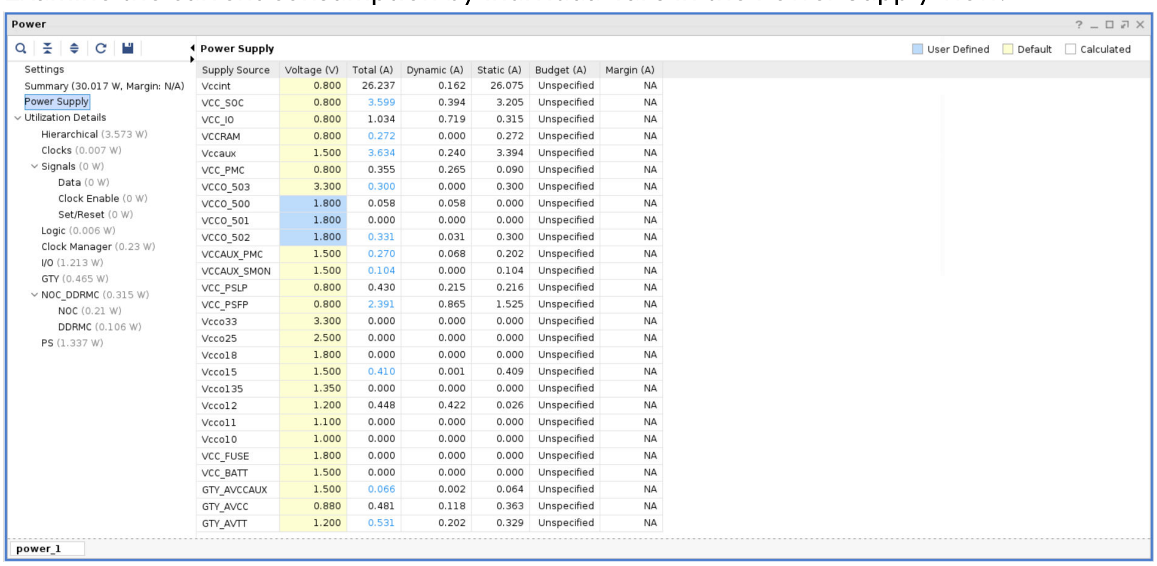
2. Examine the power report, power_1, that is generated in the Power window in Vivado IDE.

   ***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.



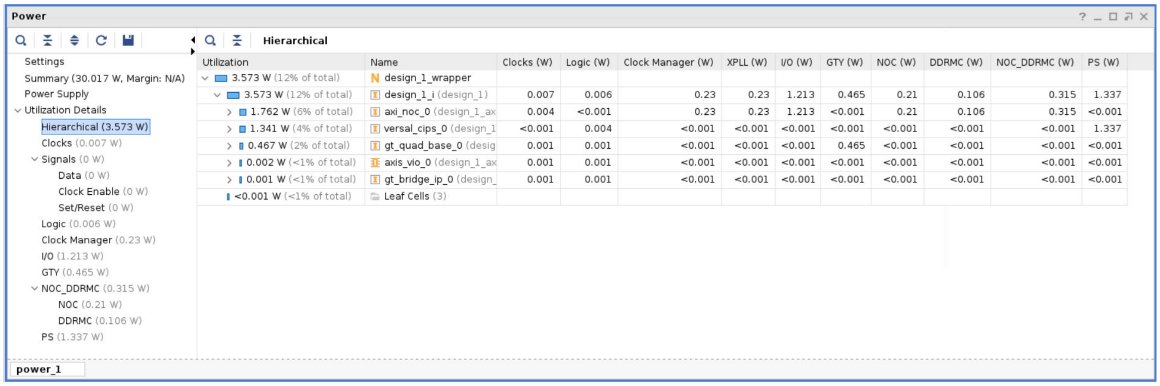
3. Examine the power breakdown in the power report by block type (Logic, GTY, I/O, and so on).

4. Examine the current consumption by individual rails in the Power Supply view.

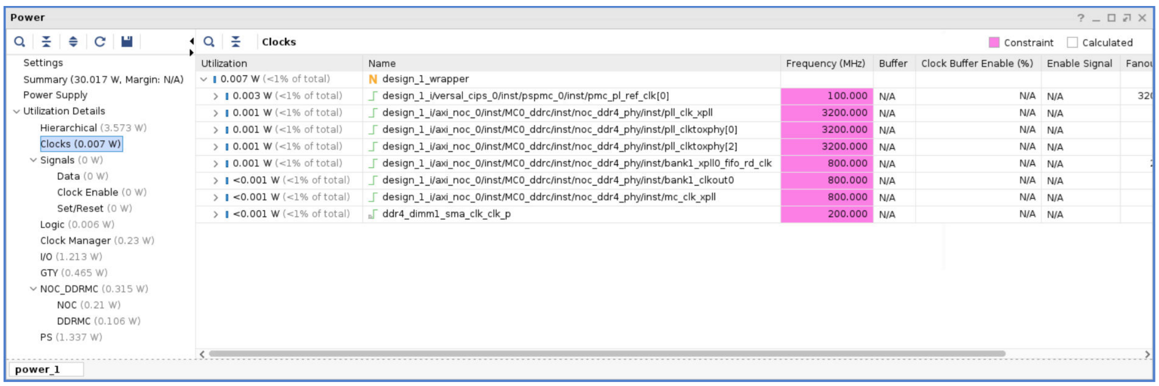


Supply Source	Voltage (V)	Total (A)	Dynamic (A)	Static (A)	Budget (A)	Margin (A)
Vccint	0.800	26.237	0.162	26.075	Unspecified	NA
VCC_SOC	0.800	3.599	0.394	3.205	Unspecified	NA
VCC_IO	0.800	1.034	0.719	0.315	Unspecified	NA
VCCRAM	0.800	0.272	0.000	0.272	Unspecified	NA
Vccaux	1.500	3.634	0.240	3.394	Unspecified	NA
VCC_PMC	0.800	0.355	0.265	0.090	Unspecified	NA
VCCO_503	3.300	0.300	0.000	0.300	Unspecified	NA
VCCO_500	1.800	0.058	0.058	0.000	Unspecified	NA
VCCO_501	1.800	0.000	0.000	0.000	Unspecified	NA
VCCO_502	1.800	0.331	0.031	0.300	Unspecified	NA
VCCAUX_PMC	1.500	0.270	0.068	0.202	Unspecified	NA
VCCAUX_SMON	1.500	0.104	0.000	0.104	Unspecified	NA
VCC_PSLP	0.800	0.430	0.215	0.216	Unspecified	NA
VCC_PSFP	0.800	2.391	0.865	1.525	Unspecified	NA
Vcco33	3.300	0.000	0.000	0.000	Unspecified	NA
Vcco25	2.500	0.000	0.000	0.000	Unspecified	NA
Vcco18	1.800	0.000	0.000	0.000	Unspecified	NA
Vcco15	1.500	0.410	0.001	0.409	Unspecified	NA
Vcco135	1.350	0.000	0.000	0.000	Unspecified	NA
Vcco12	1.200	0.448	0.422	0.026	Unspecified	NA
Vcco11	1.100	0.000	0.000	0.000	Unspecified	NA
Vcco10	1.000	0.000	0.000	0.000	Unspecified	NA
VCC_FUSE	1.800	0.000	0.000	0.000	Unspecified	NA
VCC_BATT	1.500	0.000	0.000	0.000	Unspecified	NA
GTY_AVCCAUX	1.500	0.066	0.002	0.064	Unspecified	NA
GTY_AVCC	0.880	0.481	0.118	0.363	Unspecified	NA
GTY_AVTT	1.200	0.531	0.202	0.329	Unspecified	NA

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: Implementing the Design

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

1. In the Flow Navigator, click **Run Implementation**.
2. When the Save Project dialog box is displayed before launching implementation, click **Don't Save**.

Designing with UltraScale+ Devices

Step 1: Creating a New Project

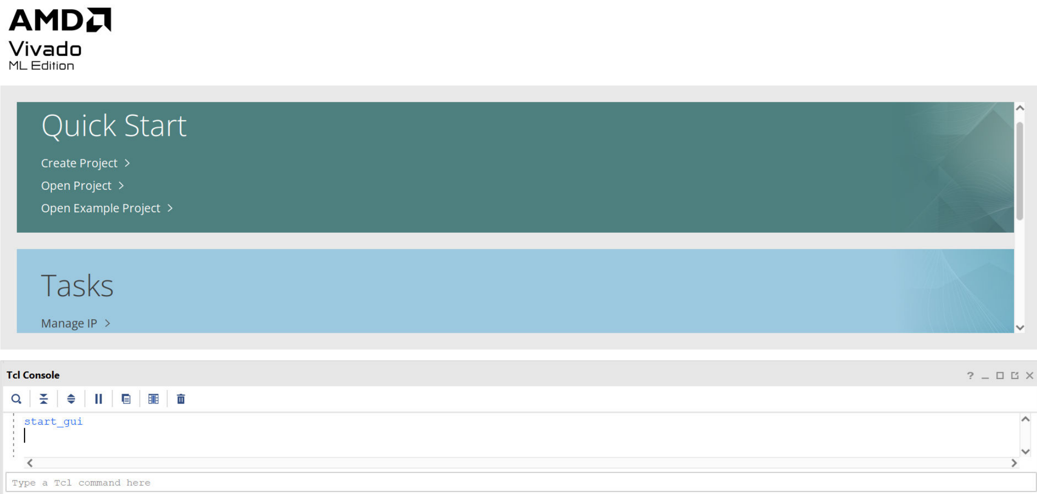
To create a project, use the New Project wizard to name the project, to add RTL source files and constraints, and to specify the target device.

On Linux, do the following.

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

   cd <Extract_Dir>/UltraScale+ (for AMD UltraScale+™ devices)

2. Launch Vivado IDE: vivado



   On Windows, do the following.

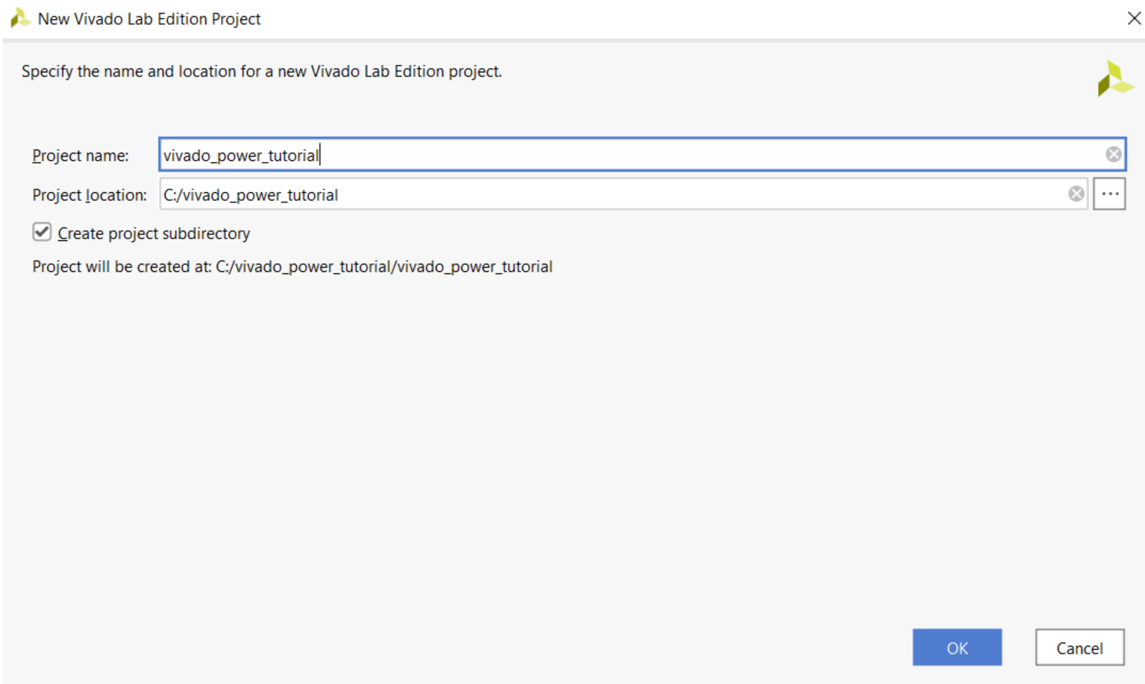
3. Launch the Vivado IDE by selecting **Start→ All Programs→ Xilinx Design Tools→ Vivado 2024.x→ Vivado 2024.x** (x denotes the latest version of Vivado 2024 IDE).

   As an alternative, click the Vivado 2024.x Desktop icon to start the Vivado IDE.

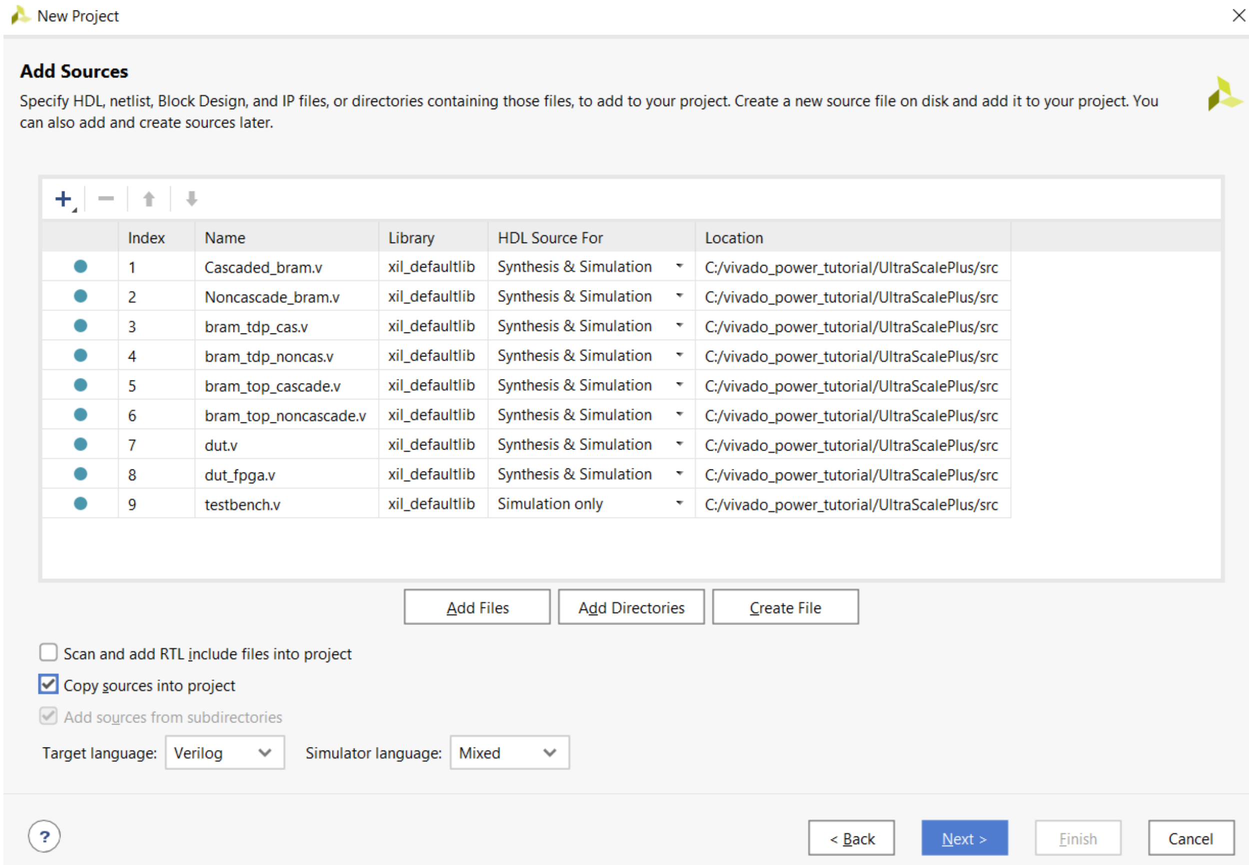
   The Vivado IDE Getting Started page contains links to open or create projects and to view documentation.

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 vivado_power_tutorial 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** and make sure to uncheck the **Do not specify sources at this time** option. Click **Next**.
8. In the Add Sources page, do the following.
    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>/UltraScale+/src directory.
    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_zcu102.xdc` in the file browser. In the directory structure, the `dut_fpga_zcu102.xdc` file is located in the `/src` folder.

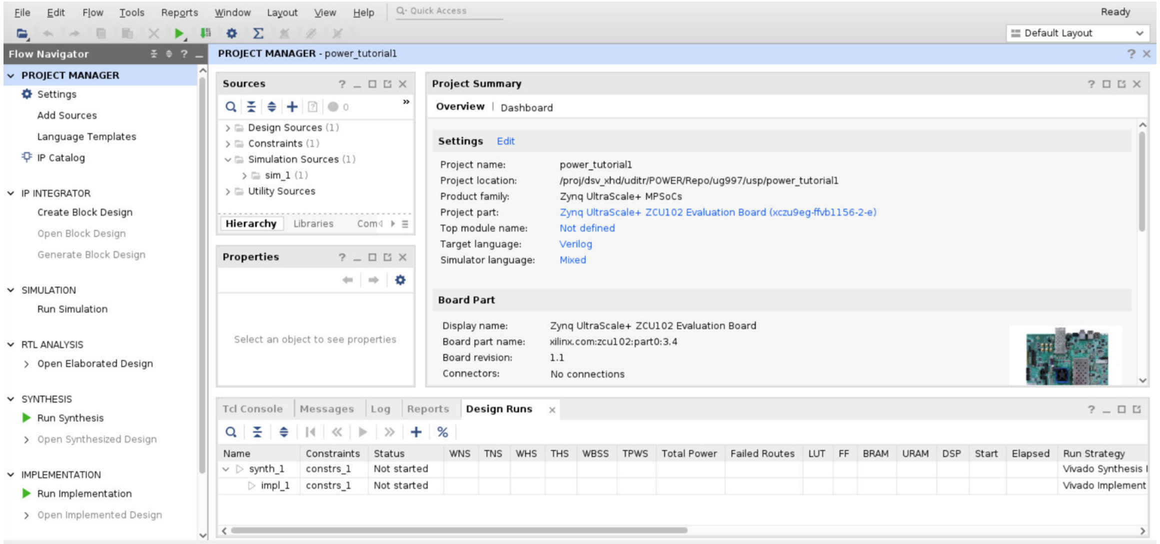
10. Click **Next** to continue.

11. In the Default Part page, click **Boards** and select Zynq UltraScale+ ZCU102 Evaluation Board.

TIP: *When you specify a board, you are also specifying the part you are targeting for your design, in this case an xczu9eg-ffvb1156-2-e FPGA UltraScale+ device.*

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

***Note*:** It can take sometime for the project to initialize in the Vivado IDE.

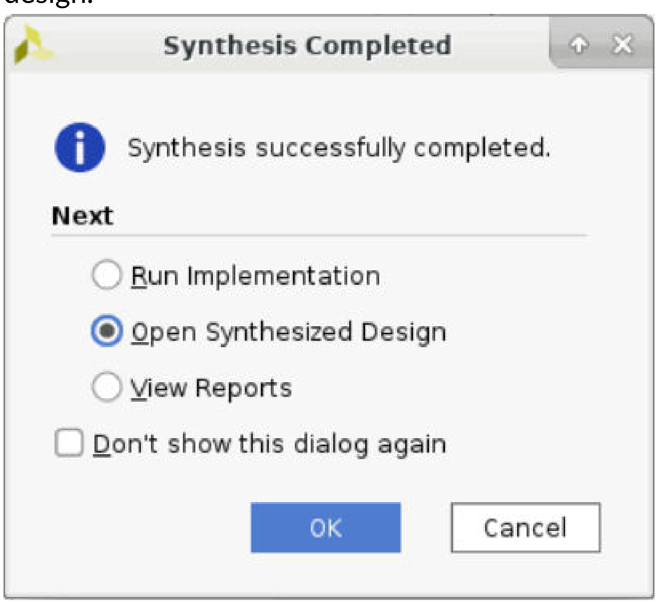


13. In the Settings dialog box (**Tools → Settings → Tool Settings → Project**), enter the tutorial project directory in the Specify project directory field, so that all reports are saved in the tutorial project directory. Next, click **OK**.

The design is now ready for synthesis.

Step 2: Synthesizing the Design

1. Click **Run Synthesis** in the Flow Navigator.
2. The Synthesis Completed dialog box is displayed after the synthesis is completed on the design.



3. Select **Open Synthesized Design** in the Synthesis Completed dialog box to open the synthesized design. Click **OK**.

Step 3: Report Power Setup

For information on setting up report power, refer to the Step 3: Report Power Setup section in Designing with Versal Devices.

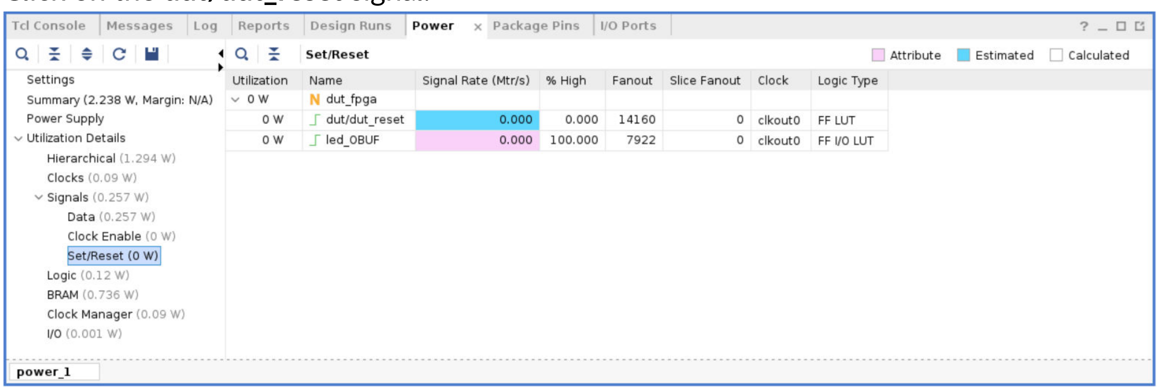
Step 4: Running Report Power

For information on running the report power, refer to the Step 4: Running Report Power section in Designing with Versal Devices.

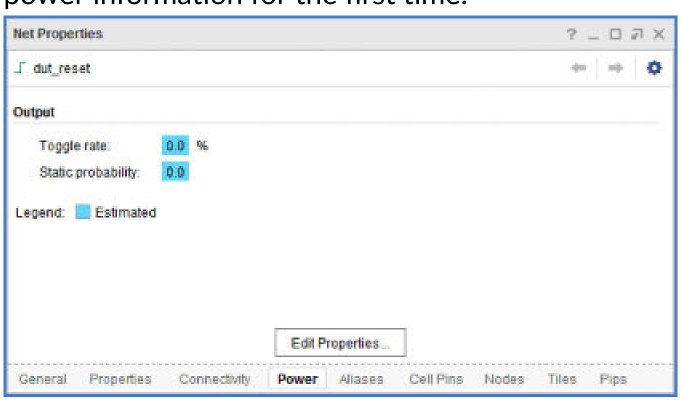
Step 5: Viewing the Power Properties

This step walks you through the process of obtaining the display of static probability and toggle rate for a signal in the 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 the net properties for the `dut/dut_reset` signal. Click on **Load Power Properties** to obtain the power information for the first time.

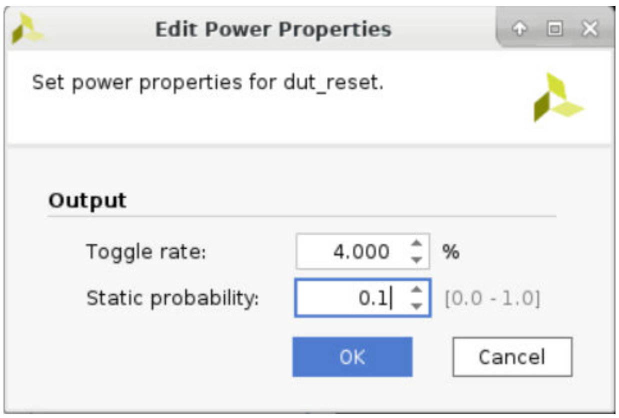


5. Note that the Toggle rate is 0% and the Static probability is 0 for the `dut/dut_reset` signal, indicating that reset is always de-asserted in the design.

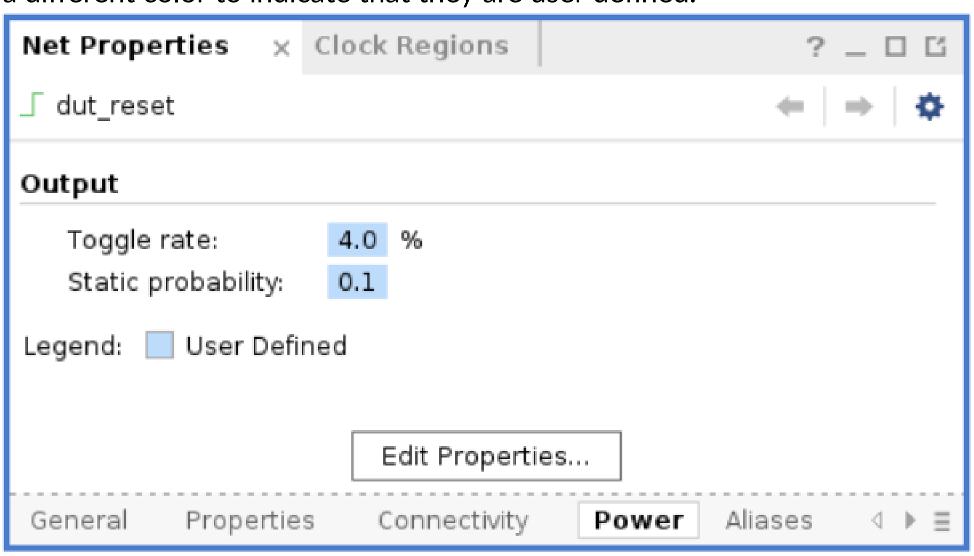
Step 6: Editing Power Properties and Refining the Power Analysis

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

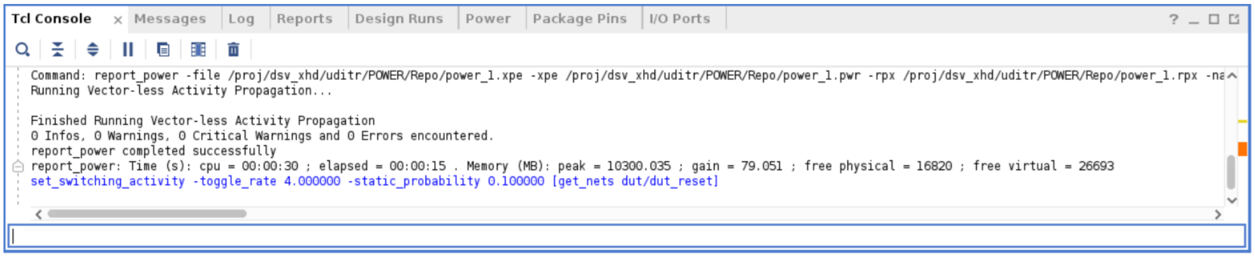
1. In the Net Properties window, click the **Edit Properties** button.
2. In the **Edit Power Properties** dialog box, change the Toggle rate to 4% and the Static probability to 0.1.



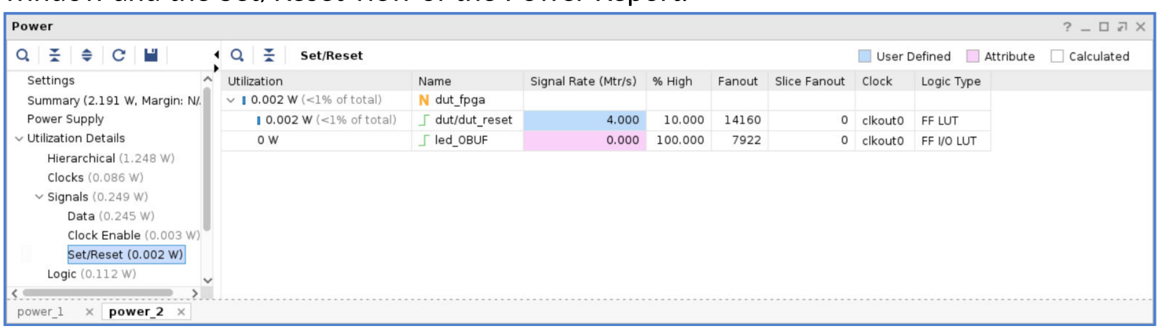
3. Click **OK**.
4. In the Net Properties window, observe that the Toggle Rate and Static Probability values turn a different color to indicate that they are user defined.



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



5. Re-run 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 and 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 significantly reduces 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.



AMD 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 towards the more accurate power estimation.

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

> **TIP:** *In the Tcl console, 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 are 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]
```

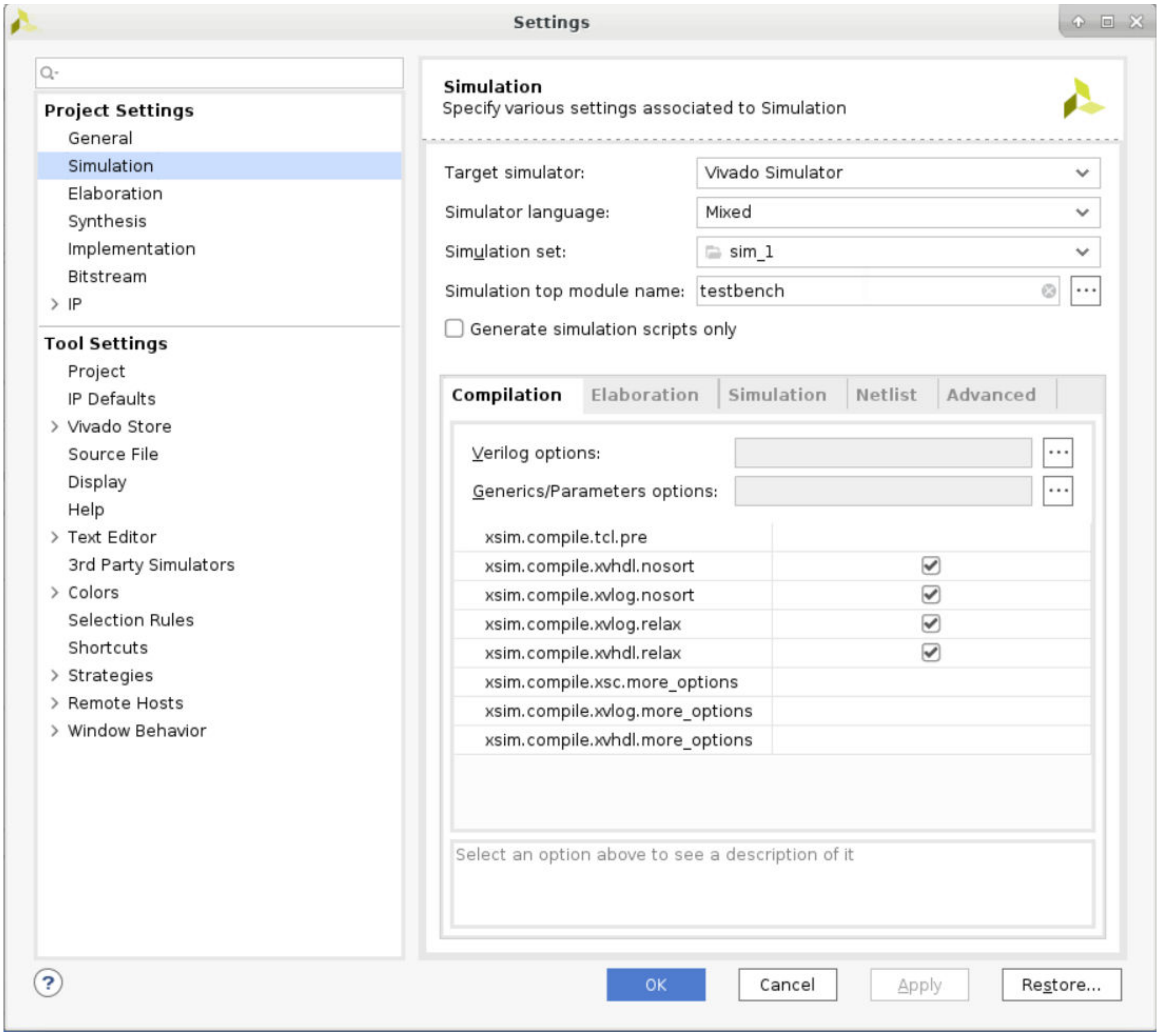
IMPORTANT! *Switching activity can also be specified in terms of toggle rate. Toggle rate is always associated with a clock. The primary ports can be associated with a specific clock using the* `set_input_delay` *and* `set_output_delay` *commands. If no clock association is found, Report Power associates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

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

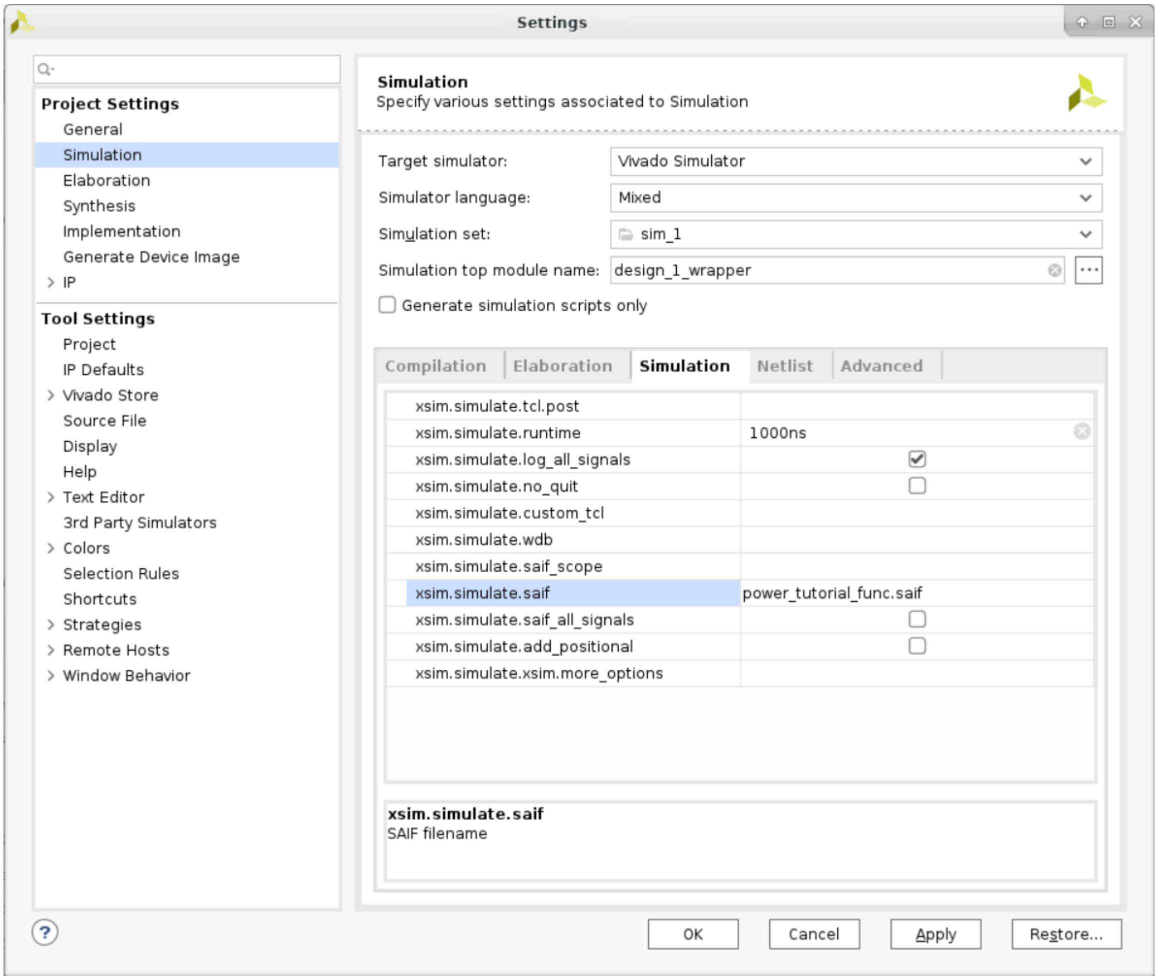
Step 7: Running Functional Simulation with SAIF Output

Now that you have created an AMD Vivado™ Design Suite project for the tutorial design, you can set up and launch the Vivado simulator to run post-synthesis functional simulation. Simulation generates a switching activity interchange format file (SAIF) that enables 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 the Simulation section.
2. In the Simulation section of Settings dialog box, 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**



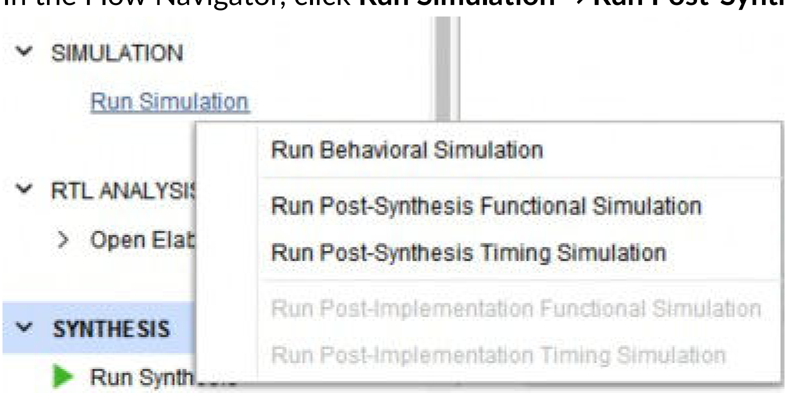
3. In the Elaboration tab of the Simulation section, make sure the xsim.elaborate.debug_level is set to **typical**, which is the default value.
4. In the Simulation tab, enter the SAIF file name as power_tutorial_func.saif for xsim.simulate.saif. Observe that the xsim.simulate.runtime is 1000 ns.
5. Make sure to check the xsim.simulate.log_all_signals box.
6. Click **OK**.



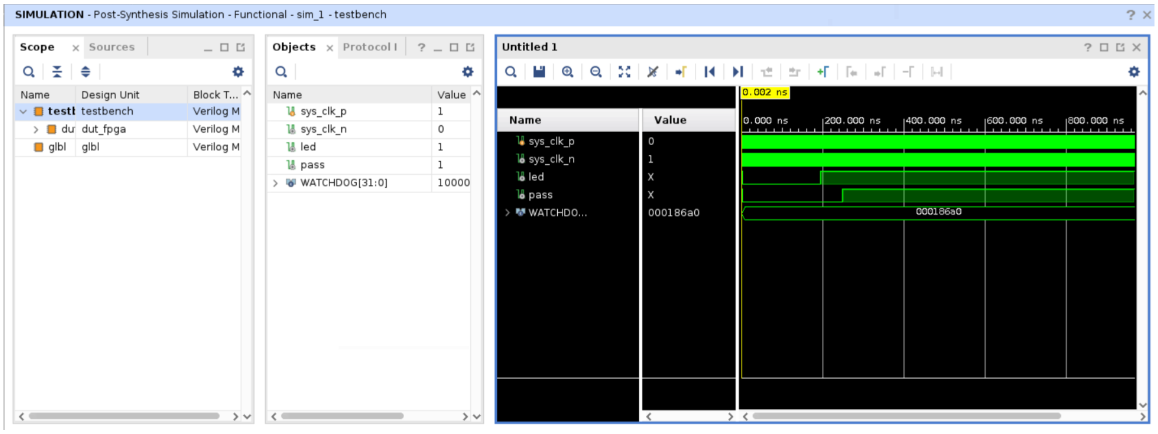
The simulation settings are now properly configured. You can launch the Vivado simulator to perform a post-synthesis functional simulation of the design.

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

7. 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.

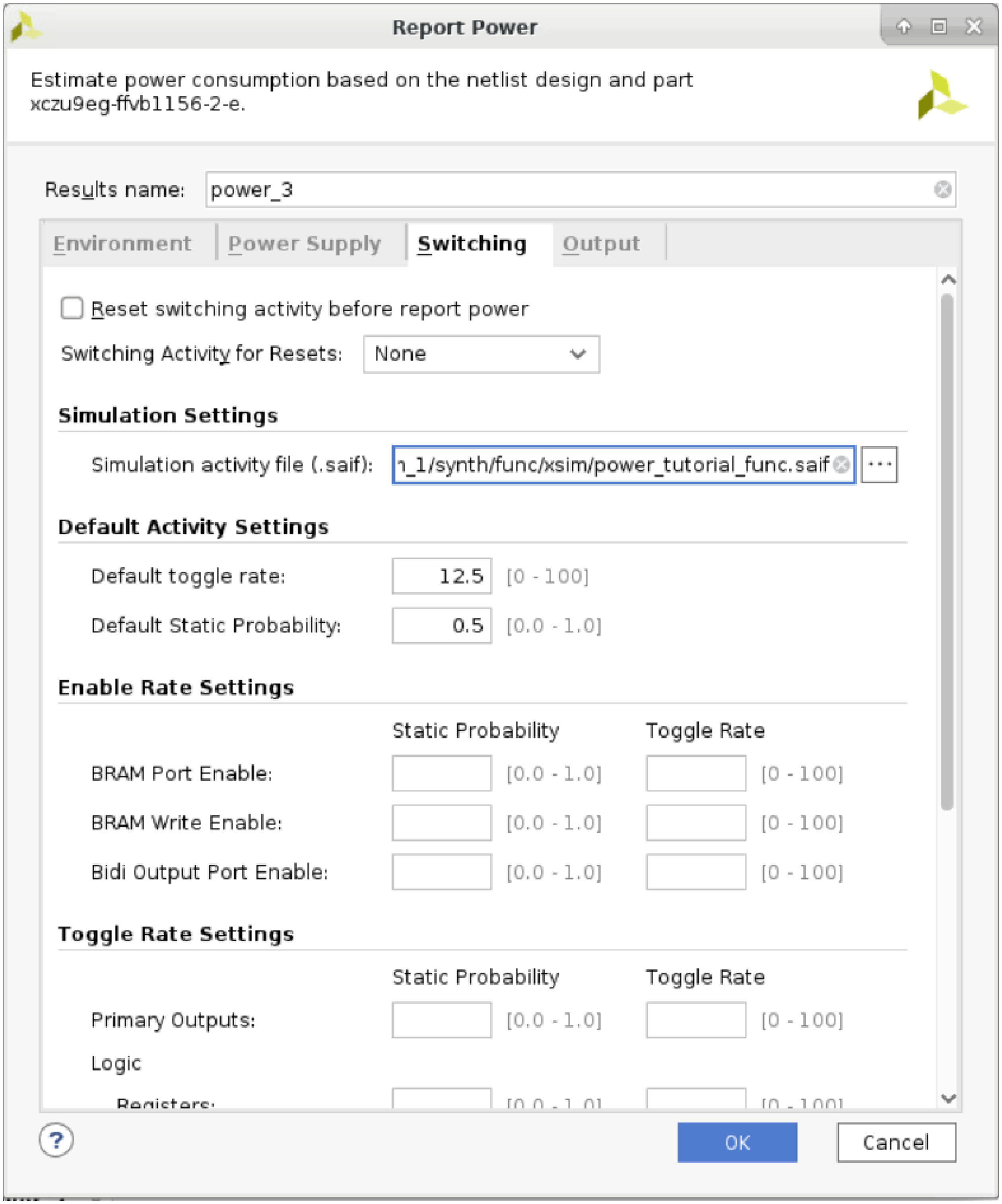
Step 8: Incorporating SAIF Data into Power Analysis

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

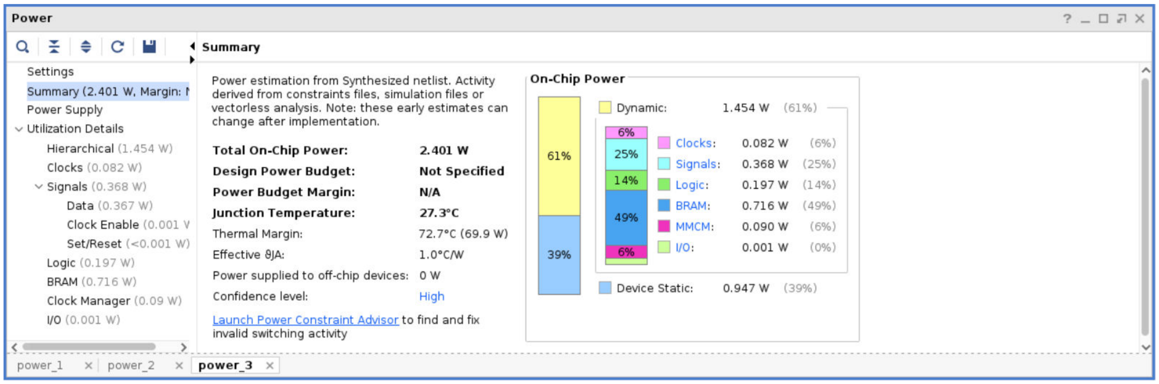
   <project_directory>/power_tutorial1/power_tutorial1.sim/sim_1/ synth/ func/power_tutorial_func.saif

2. In the Flow Navigator window, click **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.



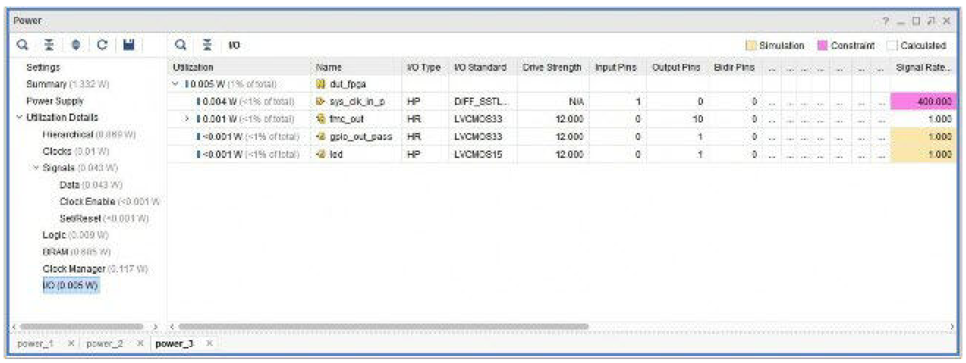
8. Click **OK** in the Report Power dialog box.

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

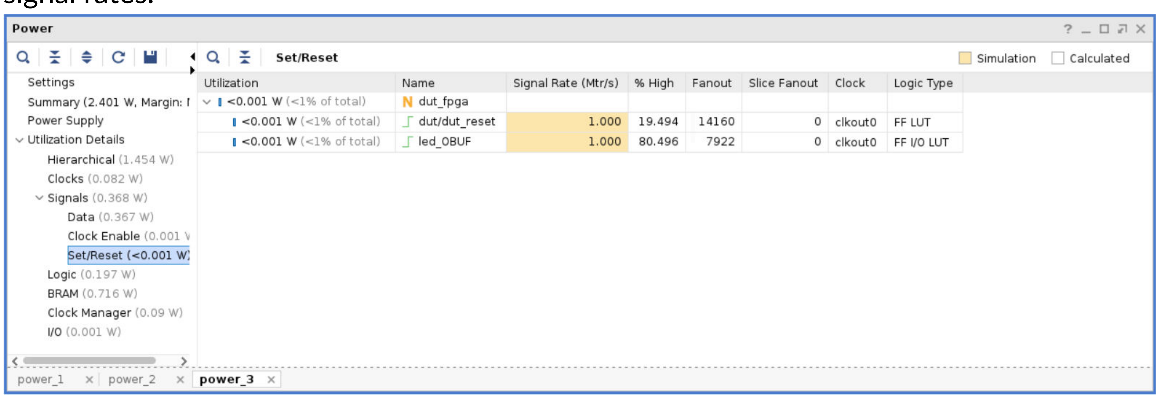


9. Go to the I/O view in the Power window.

   ***Note*:** All the I/O port activity data is set from simulation data we specified. The data is color coded to indicate activity rates read from the simulation output file.



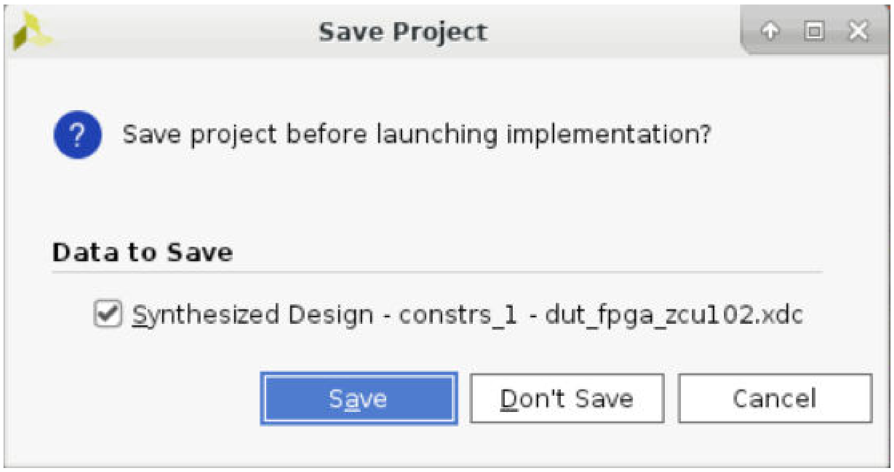
10. The difference in total power numbers (Total On-Chip Power in the Summary view) between a pure vectorless run in the power_1 results versus with the post synthesis functional simulation data in the power_3 results. Also, the SAIF data overwrites the `dut/dut_reset` signal rates.



Step 9: Implementing the Design

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

1. Run this command in the Tcl Console: `set_property STEPS.OPT_DESIGN.ARGS.DIRECTIVE NoBramPowerOpt [get_runs impl_1].`
2. In the Flow Navigator, click **Run Implementation**.
3. When the 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 tools. In Lab 3, you learn about the timing simulation and its effect on the power analysis.

> **IMPORTANT!** *The subsequent chapters discuss about AMD UltraScale+™ device only.*

Lab 3

Running Timing Simulation and Estimating Power

Introduction

In this lab, you learn about generating a SAIF file after running a timing level simulation using AMD Vivado™ simulator and Questa Advanced Simulator. The lab walks you through the steps to create a SAIF file, run timing simulation, and estimate power using the SAIF data.

***Note*:** From this lab onwards, AMD UltraScale+™ example design is used to explain the process of estimating the power through different stages, and using simulation data to enhance the accuracy of the power analysis.

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. Set the timing simulation properties in Vivado IDE, and run the timing simulation.

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

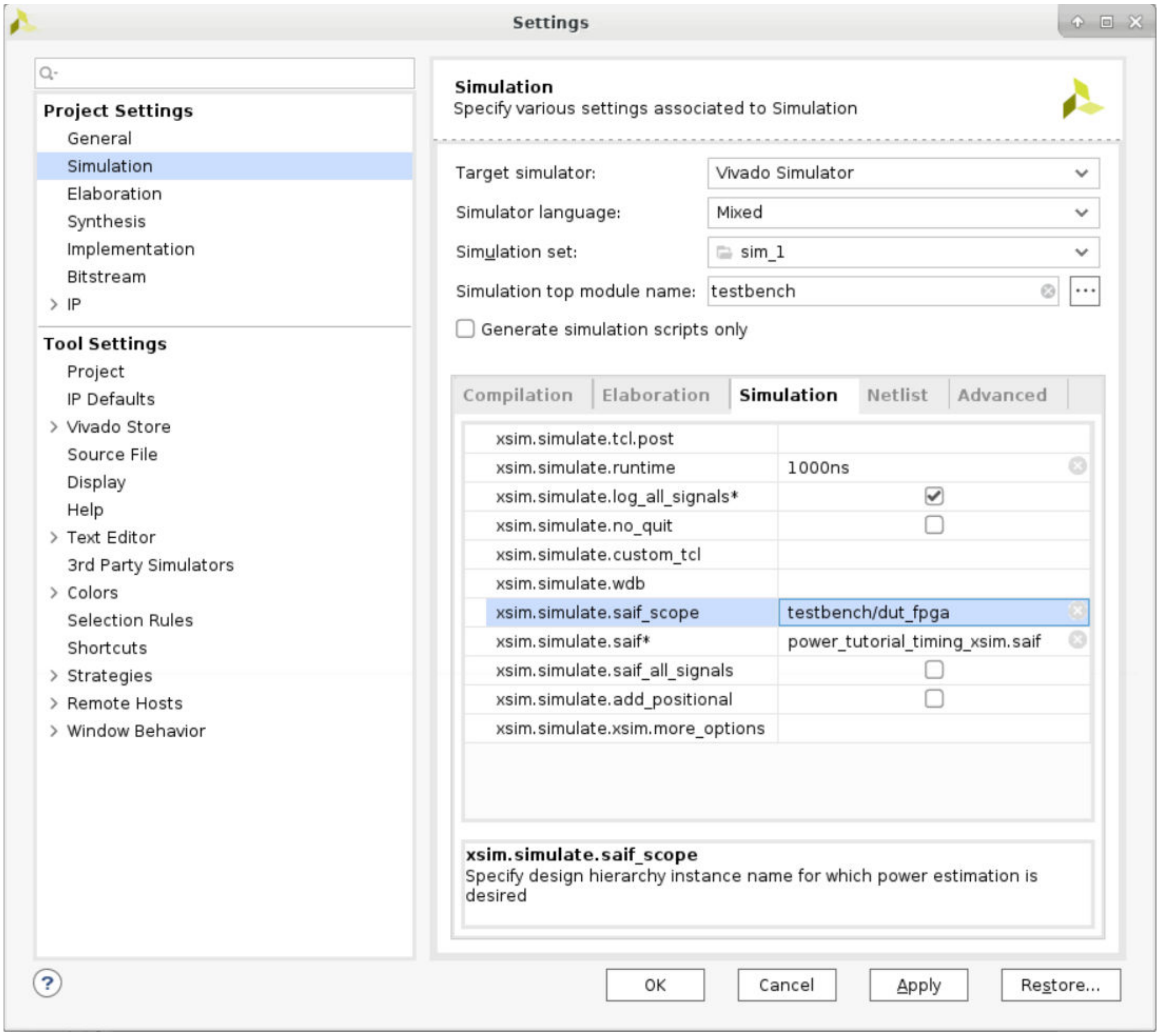
   - Simulation set: **sim_1**
   - Simulation top-module name: **testbench**

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**.

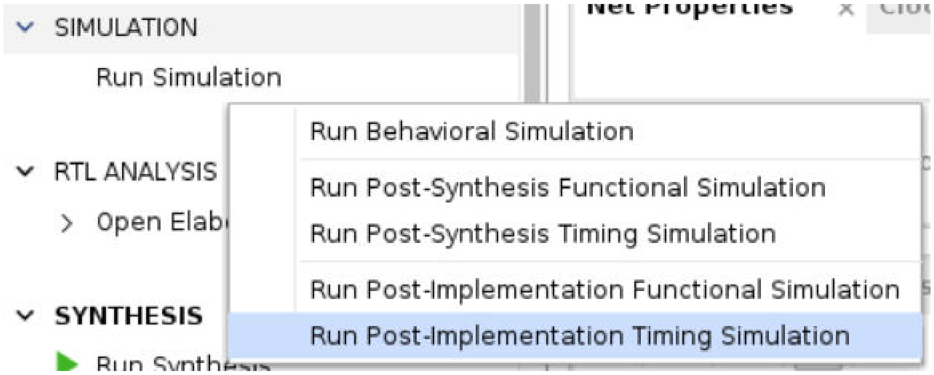
5. Set the xsim.simulate.saif_scope to **testbench/dut_fpga**.

6. Observe that the simulation run time xsim.simulate.runtime is 1000 ns.
7. Check **xsim.simulate.log_all_signals**.
8. Click **OK**.



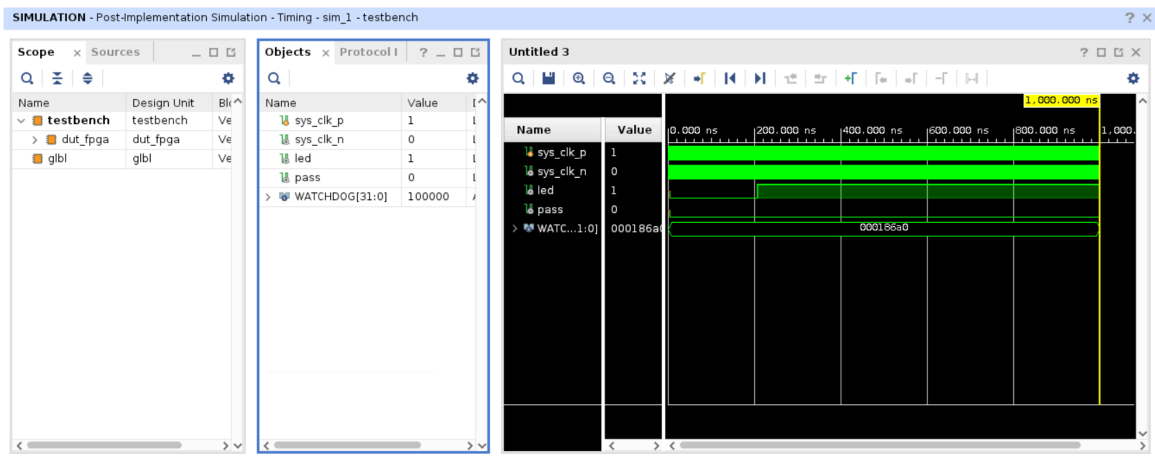
With the simulation settings properly configured, you can launch the Vivado simulator to perform a timing simulation of the post implemented design.

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



10. After the Vivado simulator finishes simulating the design, ensure that the requested SAIF file is generated. Check to see that the requested SAIF file 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

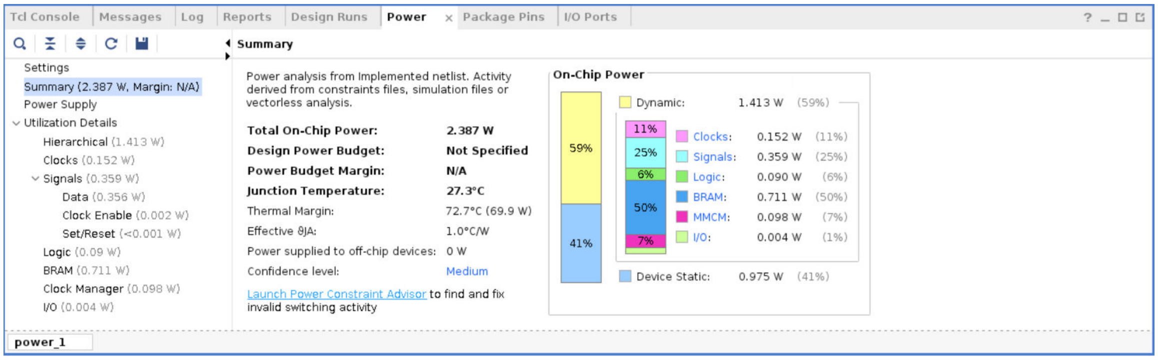
1. In the Flow Navigator, select **Open Implemented Design → Report Power** to open the Report Power dialog box.

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

2. In the Report Power dialog box, in the Environment tab, make sure that 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 in the Summary page.



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

Refer to *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 requested SAIF output file in the previous timing simulation run. The SAIF file is used 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.

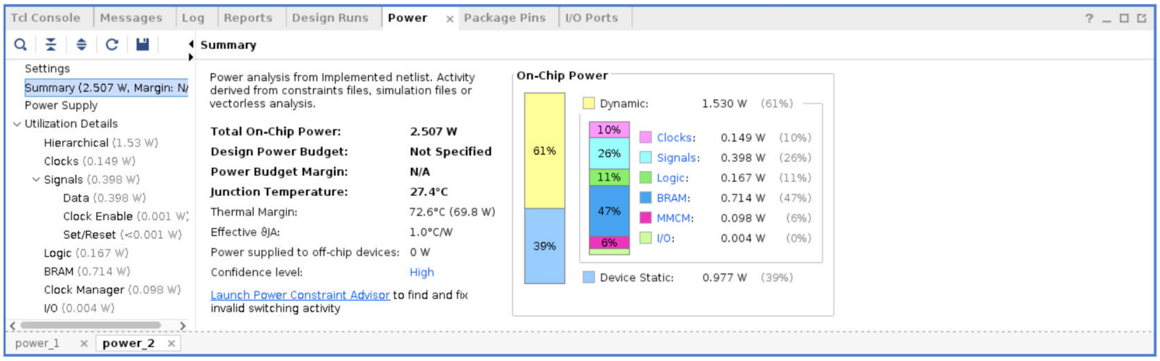
   The SAIF file, which was requested in the simulation settings prior to running timing simulation, should appear in this directory:

   ```
   <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 the power_2 power report.

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

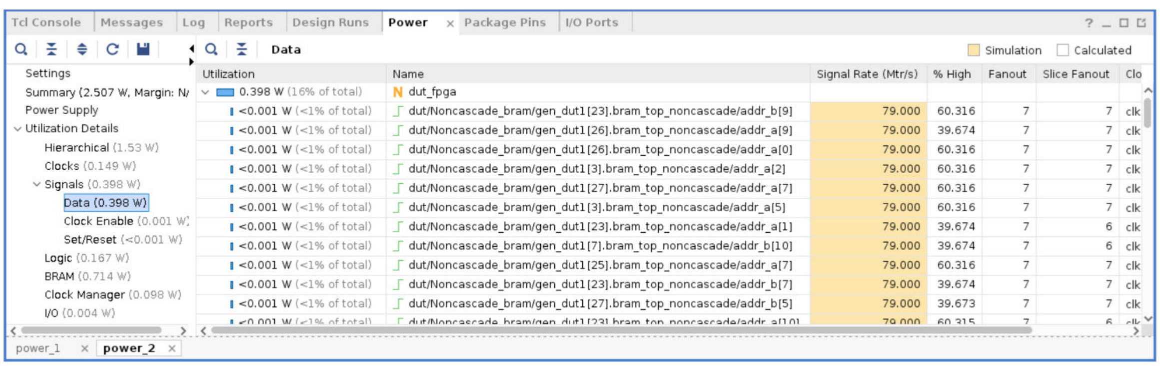


4. The change in total power (Total On-Chip Power in the Summary view) in the power_2 report compared to the power_1 report. The total power estimated in the report generated with SAIF file data is different than the total power estimated in the vectorless run (power_1 results).

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. All the Signal Rate data is set from simulation data that the SAIF file has provided.

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

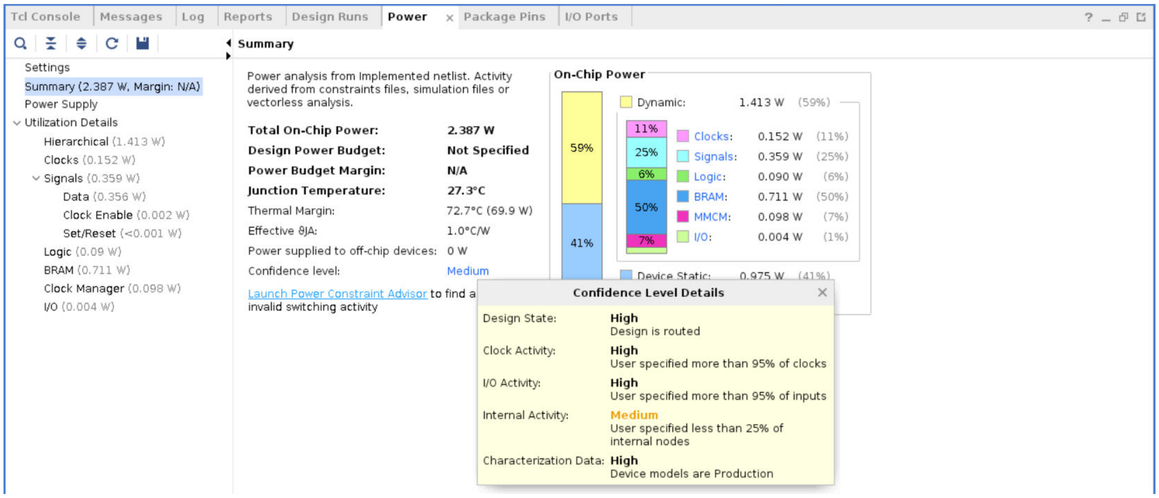


7. In the Summary view of the power_1 report (the report generated by the vectorless analysis), click **Confidence level**.

   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.

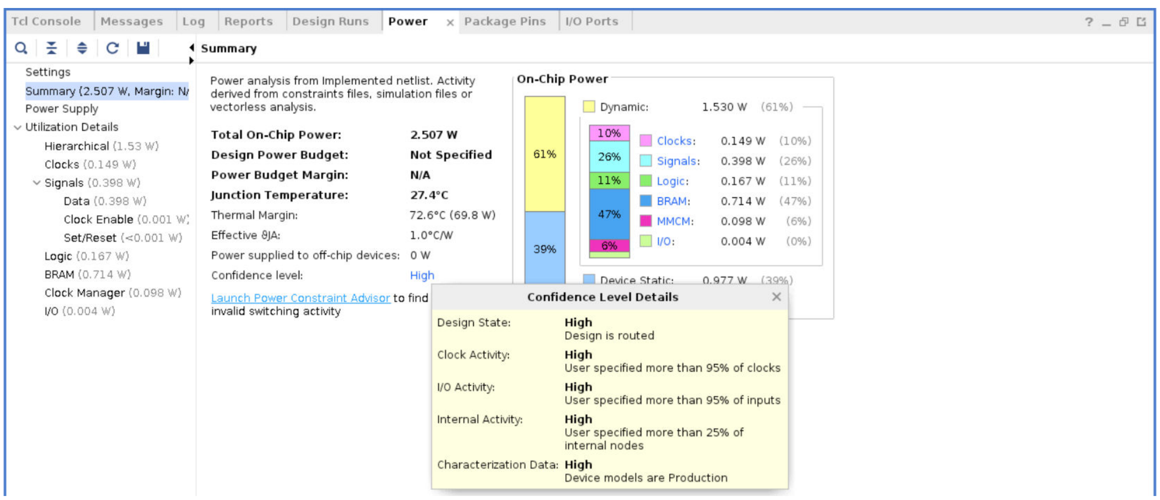
   ***Note*:** The

   Confidence Level is Medium 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 **Confidence level** (the following figure).

   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 takes you through the process of SAIF file creation, running timing simulation, and estimating power using SAIF data using the Questa Advanced Simulator.

> **IMPORTANT!** *Make sure that the AMD 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

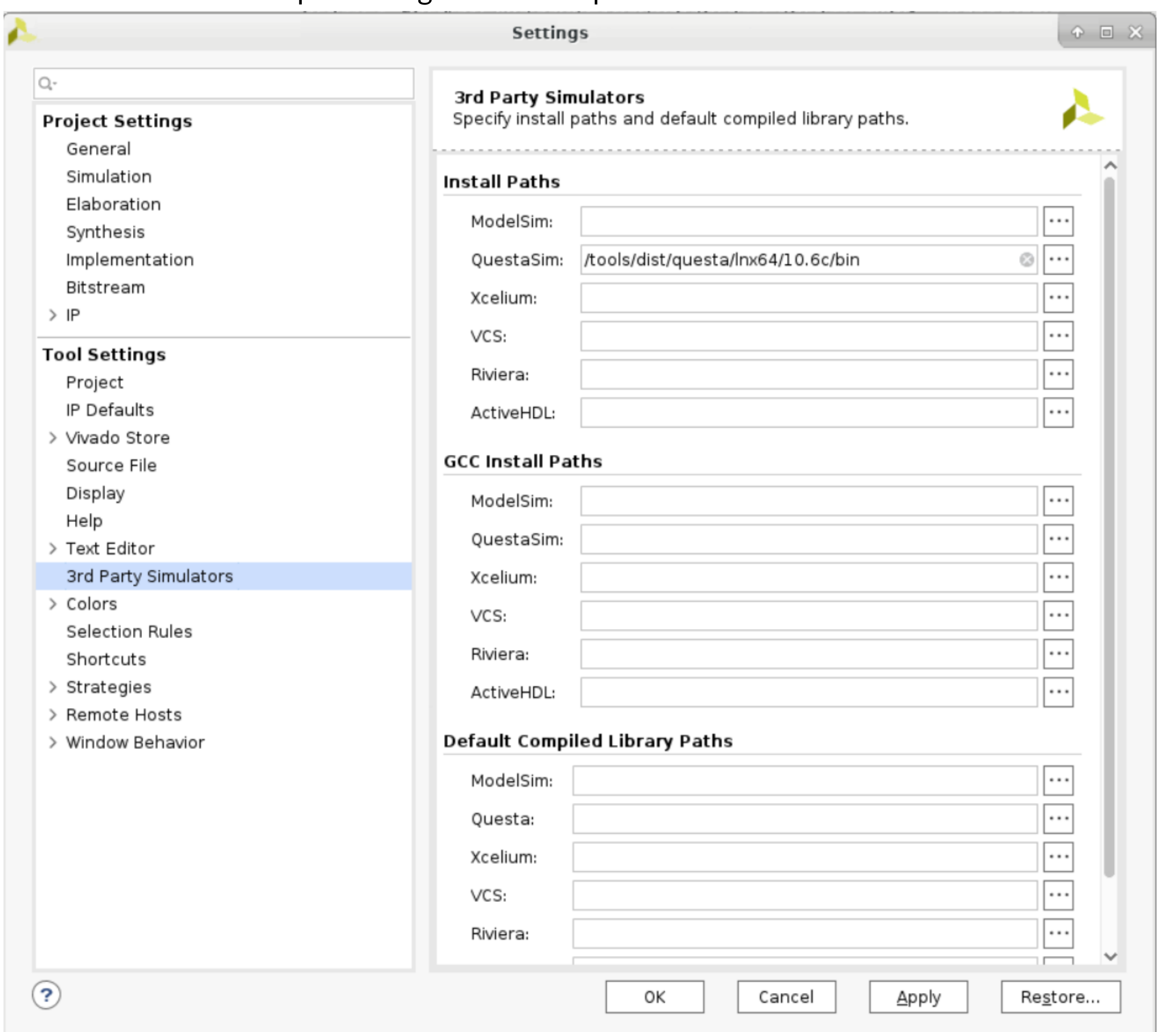
or

In the Vivado IDE, click **Tools → Settings → Tool Settings**, and define the path to the Questa Advanced Simulator on the Third Party Tools page.

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

To create new compiled libraries:

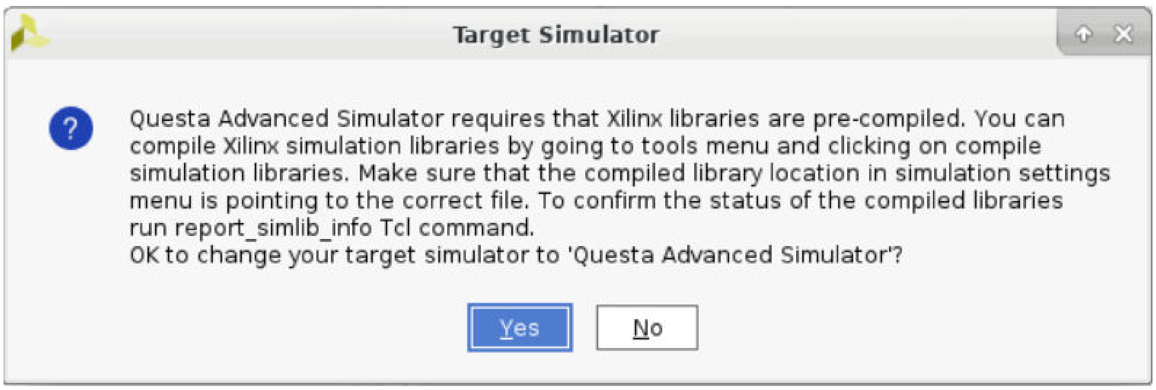
1. In the Third Party Simulators page, specify the compiled library path for Questa Advanced Simulator in the **Questa** field under Default Compiled Library Paths. Enter the **Compiled library location** specified during the compiled library generation. It should point to the `compile_simlib` directory.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



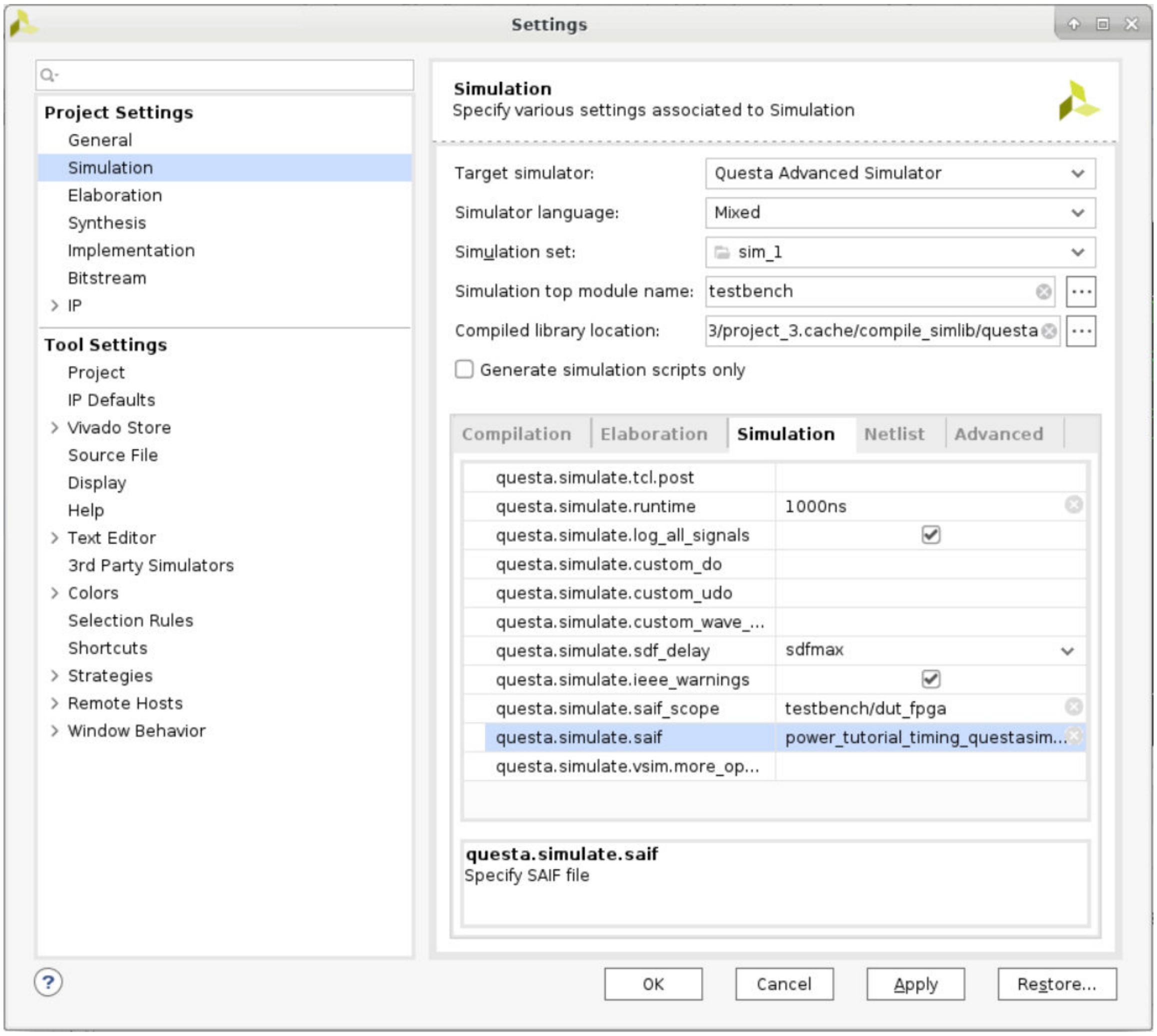
Step 1: Configuring and Running Timing Simulation in Questa Advanced Simulator

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

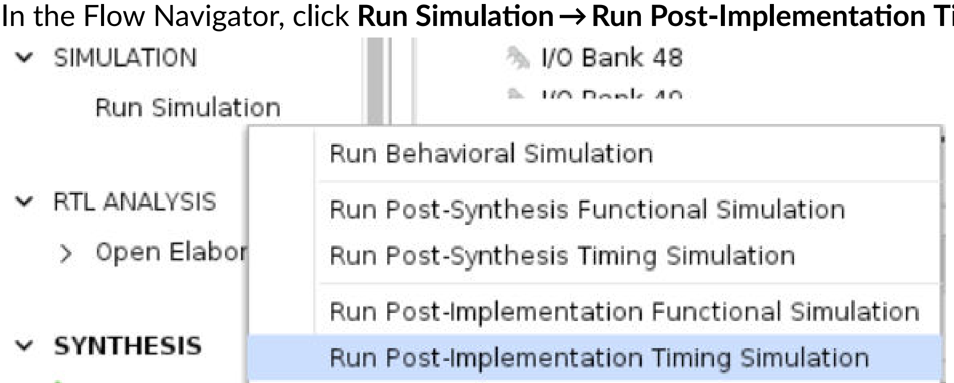
1. In the Flow Navigator, right-click **Simulation** to select **Simulation Settings**. Set the timing simulation properties.
2. In the Simulation Settings tab, set the Target simulator to **Questa Advance Simulator**.
3. Click **Yes** to change your target simulator to Questa Advanced Simulator.



4. Set **questa.simulate.saif** to power_tutorial_timing_questasim.saif.
5. Set **questa.simulate.saif_scope** to testbench/dut_fpga.
6. Make sure to check the questa.simulate.log_all_signals box.
7. The questa.simulate.runtime is 1000ns.



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



A separate Questa Advanced Simulator window opens and a design simulation starts.

10. After the Questa Advanced Simulator has finished simulating the design, make sure that the requested SAIF file 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/ impl/timing/power_tutorial_timing_questasim.saif`

Step 2: Running Report Power in Vectorless Mode

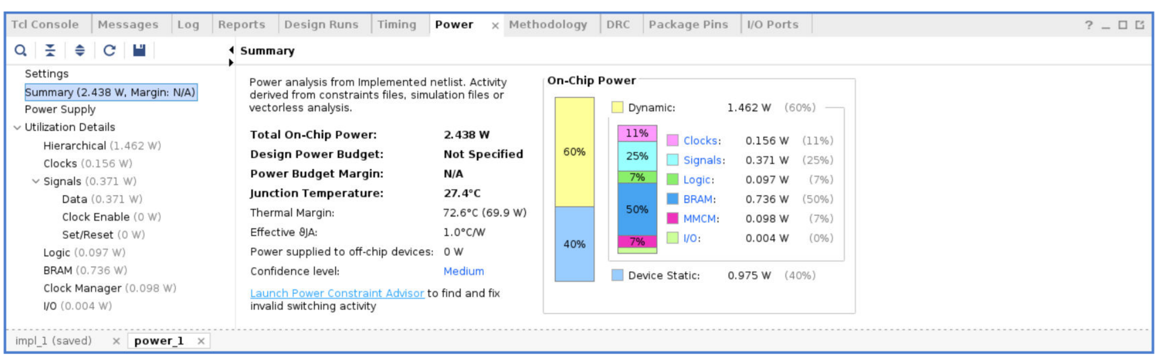
> **IMPORTANT!** *If SAIF based* `report_power` *has already been run in this session, run the* `reset_switching_activity -all` *command in the Tcl console. This clears the SAIF data in the power engine from the earlier runs.*

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*:** The total power for vectorless analysis runs with default switching rates.



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

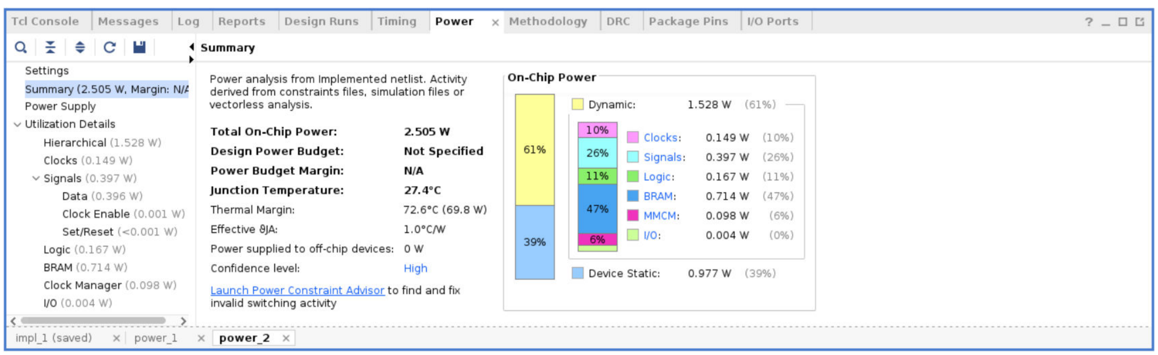
The SAIF output file requested in the simulation run is generated under the project directory. We use this SAIF file to further guide the power estimation algorithm.

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:

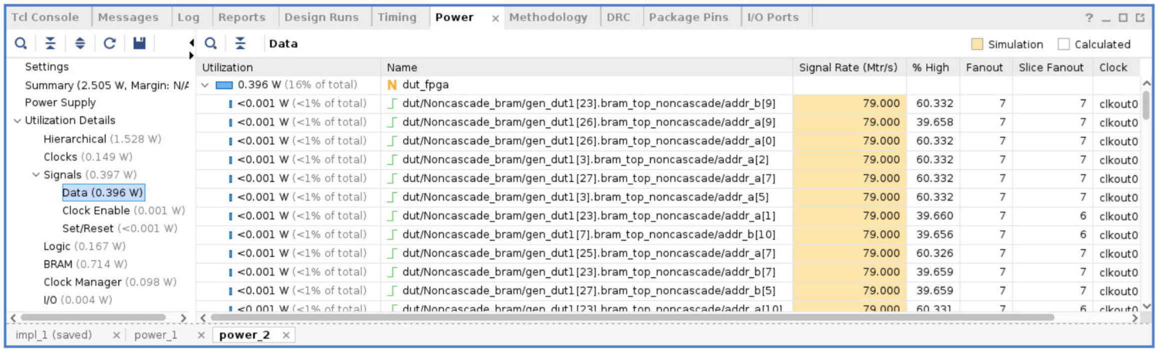
   `<project_directory>/power_tutorial1/power_tutorial1.sim/ sim_1/impl/timing/power_tutorial_timing_questasim.saif`
3. Click **OK** in the Report Power dialog box.

   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 4, you will learn about using the Power Optimization features in the Vivado IDE.

Lab 4

Performing Power Optimization

Introduction

In this lab, you will learn about using the Power Optimization features in AMD Vivado™ tools for AMD UltraScale+™ devices. The lab takes you through the steps for invoking Power Optimization after synthesizing the design. It also guides you on how to use the power optimization report, make decisions and selectively turn off power optimization on signals, blocks, and hierarchies.

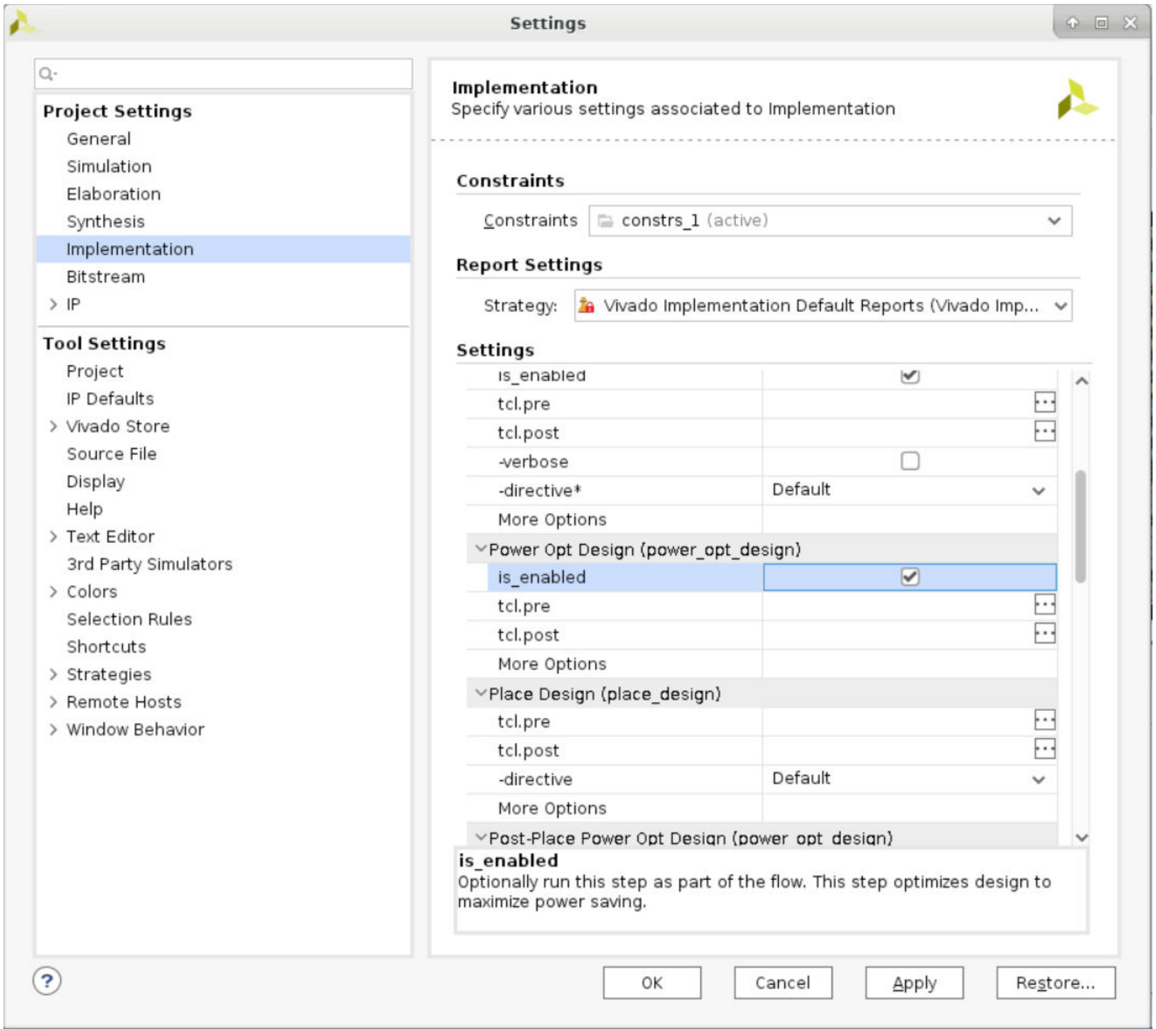
> **TIP:** *When you run Implementation on your design, the Vivado tools can perform block RAM power optimizations by default during* `opt_design`*. These optimizations do not affect performance, and have little impact on area and compile time. In the previous Lab, the default block RAM power optimization was disabled (Step 9 of Lab 2) 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.

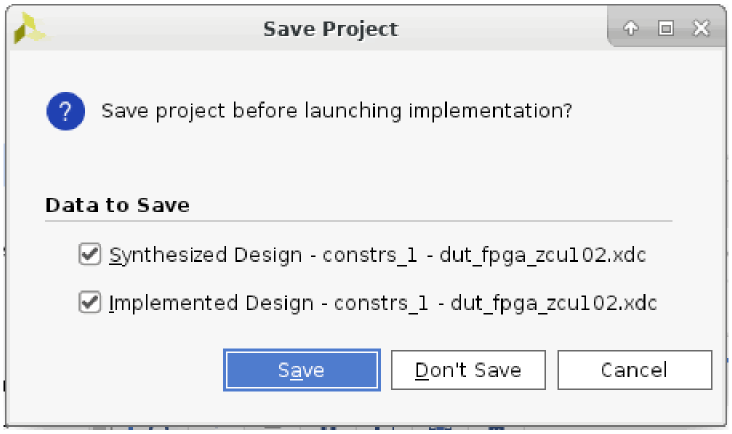
     This ensures Power Optimization runs after `opt_design`. Enabling the **Power Opt Design** option prior to `place_design` results in a complete power optimization to be performed. This option yields the best possible power saving from the Vivado tools.



3. Click **OK**.
4. In the Create New Run dialog box, click **Yes** to "Properties for the completed run 'impl_1' are 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.

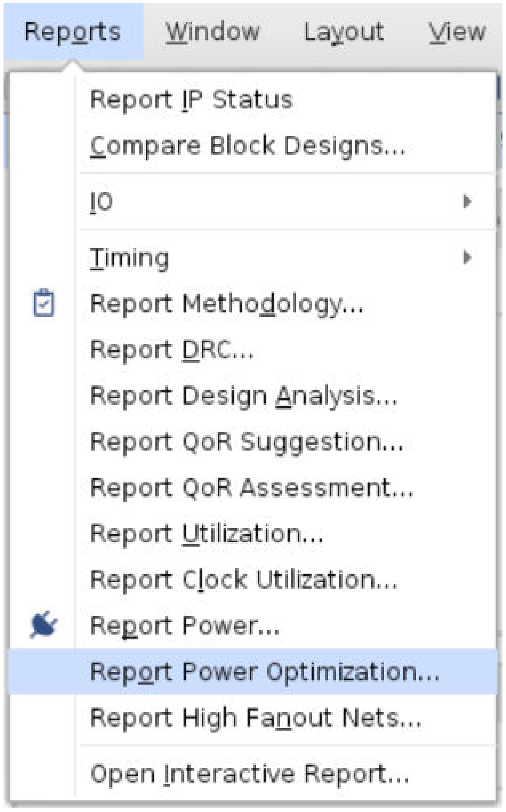


You are running Implementation with Power Optimization turned on.

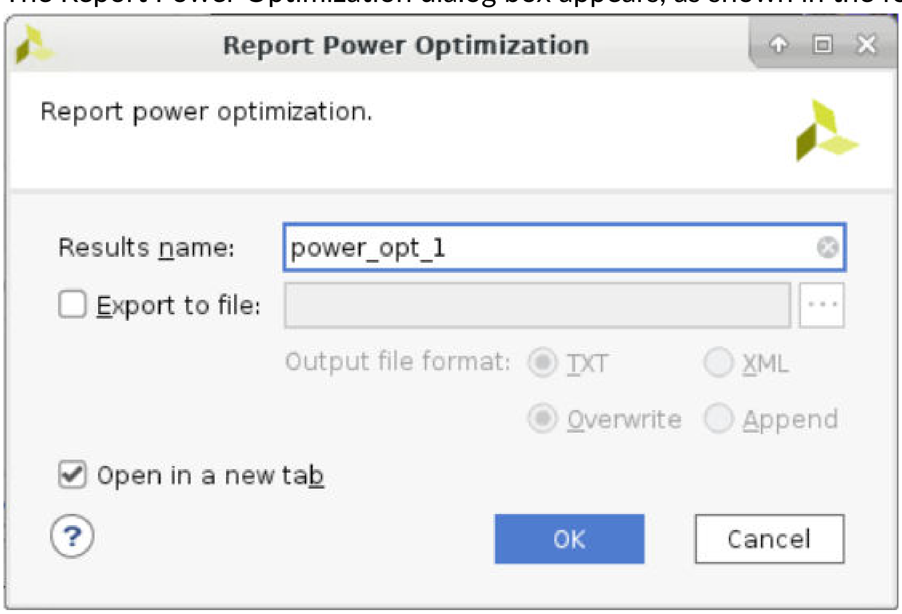
8. In the Implementation Completed dialog box, select **Open Implemented Design** and click **OK**. Click **Don't Save** when the Save Project window pops up to save both Synthesis and Implementation constraints.

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, the gated items are listed for all blocks. Under Hierarchical Information, block wise information of all gated instances are available.

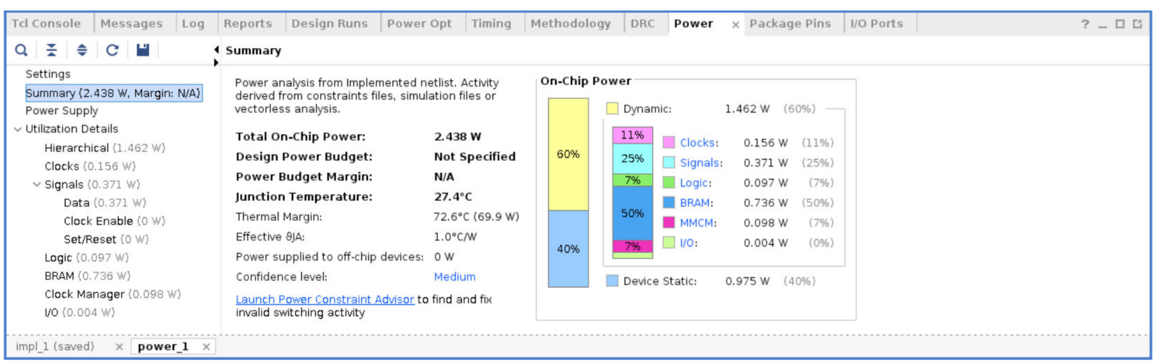
Step 3: Running report_power to Examine Power Savings

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:

   ```
   report_power -name power_1
   ```

4. In the Summary view of the Power Report, some 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.



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.

There are no tool gated blocks in this design, but assume that this block RAM is in the critical path:

`dut/Cascaded_bram/gen_dut[0].bram_top_cascade/bram_cas/mem_reg_bram_0`

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

1. In the Tcl Console, type this command:

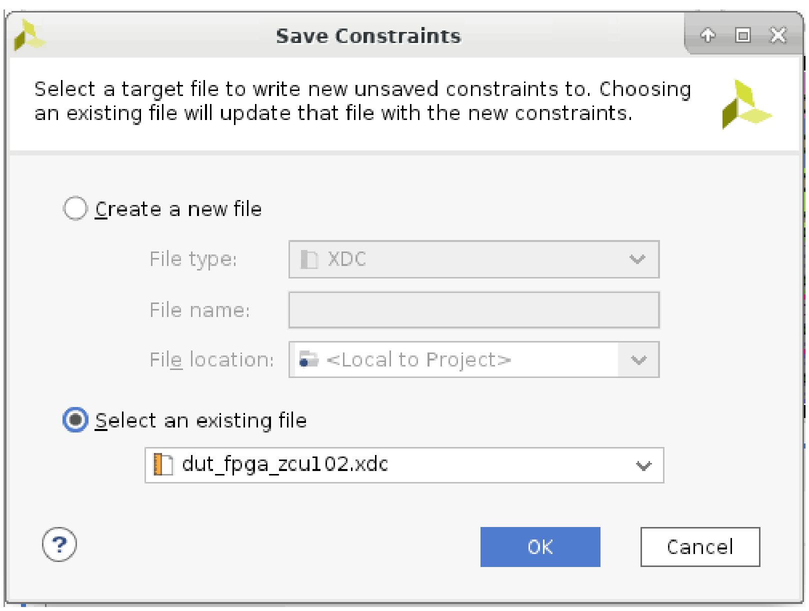
   ```
   set_power_opt -exclude_cells [get_cells dut/Cascaded_bram/
   gen_dut[0].bram_top_cascade/bram_cas/mem_reg_bram_0]
   ```

   This prevents 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.

   Click **OK** on the Save Constraints 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
```

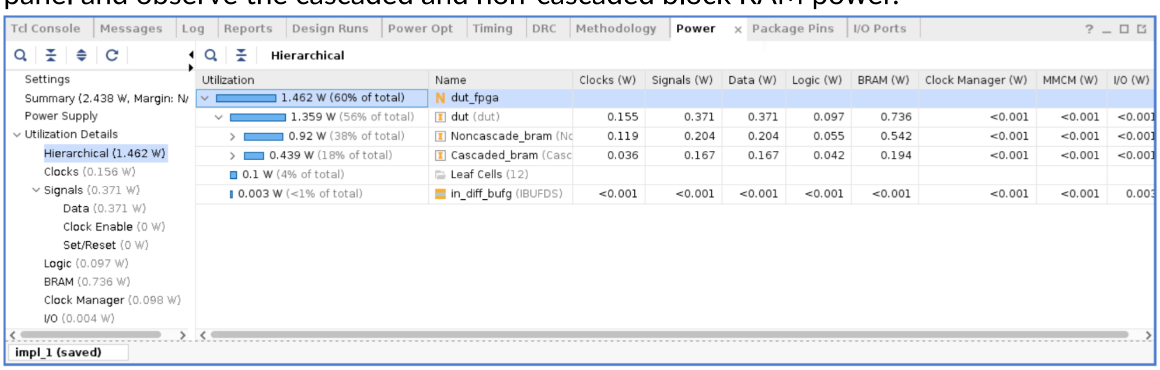
3. In the generated report power_opt_2, the excluded block RAM is no longer available in the list of **Tool Gated BRAMs**.

***Note*:** This block RAM is no longer in the list of Tool Gated BRAMs: `dut/Cascaded_bram/gen_dut[0].bram_top_cascade/bram_cas/mem_reg_bram_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 enables 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. To view this in power report, go to the **Hierarchical** view under **Utilization Details** on the left panel and observe the cascaded and non-cascaded block RAM power.



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 re-ran Report Power.
- Ran functional simulation using the Vivado simulator and generated a SAIF file that is data to feed 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 data to feed to Report Power for a more accurate power analysis.
- Ran Questa Advanced Simulator post-implementation timing simulation and generated a SAIF file that is data to feed to Report Power for a more accurate power analysis.
- Performed power measurement on the design implemented in a ZCU102 and VCK190 Evaluation Boards.
- 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.

Additional Resources and Legal Notices

Finding Additional Documentation

Technical Information Portal

The AMD Technical Information Portal is an online tool that provides robust search and navigation for documentation using your web browser. To access the Technical Information Portal, go to https://docs.amd.com.

Documentation Navigator

Documentation Navigator (DocNav) is an installed tool that provides access to AMD Adaptive Computing documents, videos, and support resources, which you can filter and search to find information. To open DocNav:

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

***Note*:** For more information on DocNav, refer to the *Documentation Navigator User Guide* (UG968).

Design Hubs

AMD 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.
- Go to the Design Hubs web page.

Support Resources

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

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)

Revision History

11-20-2025: Released with Vivado Design Suite 2025.2 without changes from 2024.2.

Section	Revision Summary
11/13/2024 Version 2024.2	
General updates	Editorial updates only. No technical content updates

Please Read: Important Legal Notices

UG997 (v2025.2) November 20, 2025

AUTOMOTIVE APPLICATIONS DISCLAIMER

Copyright

UG997 (v2025.2) November 20, 2025