Vivado Design Suite Tutorial

Power Analysis and Optimization

UG997 (v2022.2) October 19, 2022

Table of Contents

Lab 1

Power Analysis and Optimization Tutorial

This tutorial introduces the power analysis and optimization 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).

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 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 Versal® ACAP and 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 UltraScale+™ tutorial design files:

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_zcu102.xdc`	Contains clocking and timing constraints for the design.
`/src/testbench.v`	Testbench 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:

Table 1: **Example table**

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 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 will analyze power of the design in Vivado IDE. Then you will examine some of the major features in the Report Power window and closely examine some power specific Tcl commands. You will 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 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.

Designing with Versal

Step 1: Creating a New Project

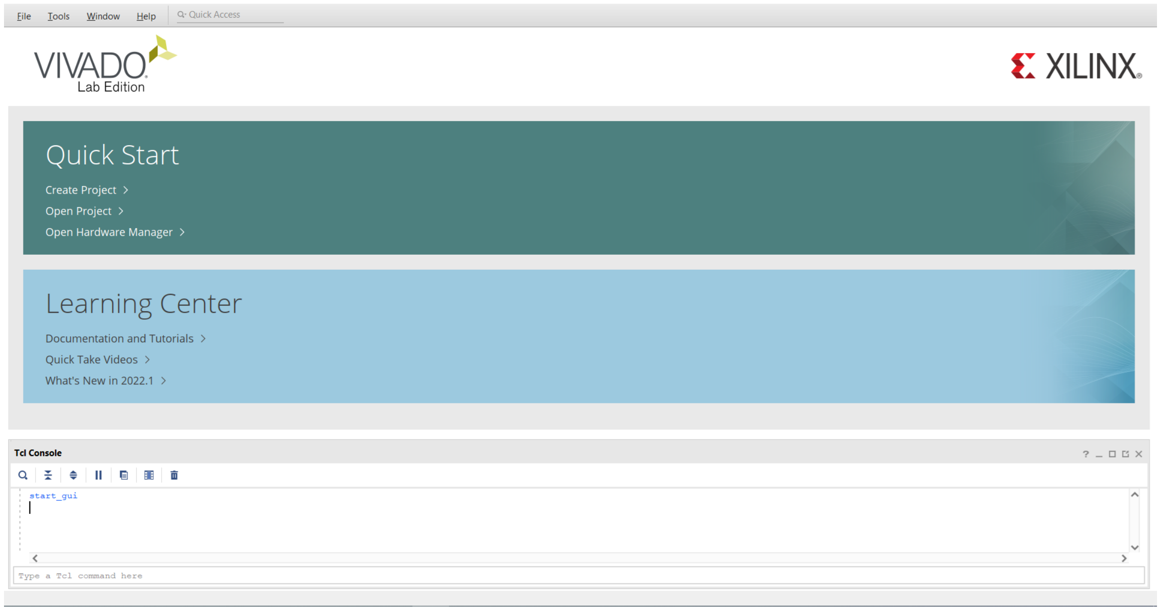
IP integrator (IPI) is used to create the design.

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`



```
start_gui
```

On Windows, do the following.

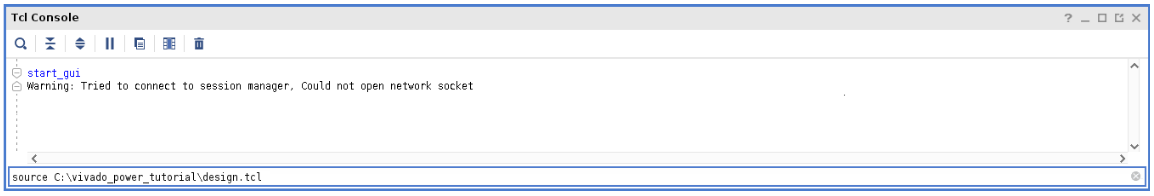
3. Launch the Vivado IDE by selecting **Start→ All Programs→ Xilinx Design Tools→ Vivado 2022.x→ Vivado 2022.x** (x denotes the latest version of Vivado 2022 IDE).

   As an alternative, click the Vivado 2022.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 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.



```
start_gui
Warning: Tried to connect to session manager, Could not open network socket
```

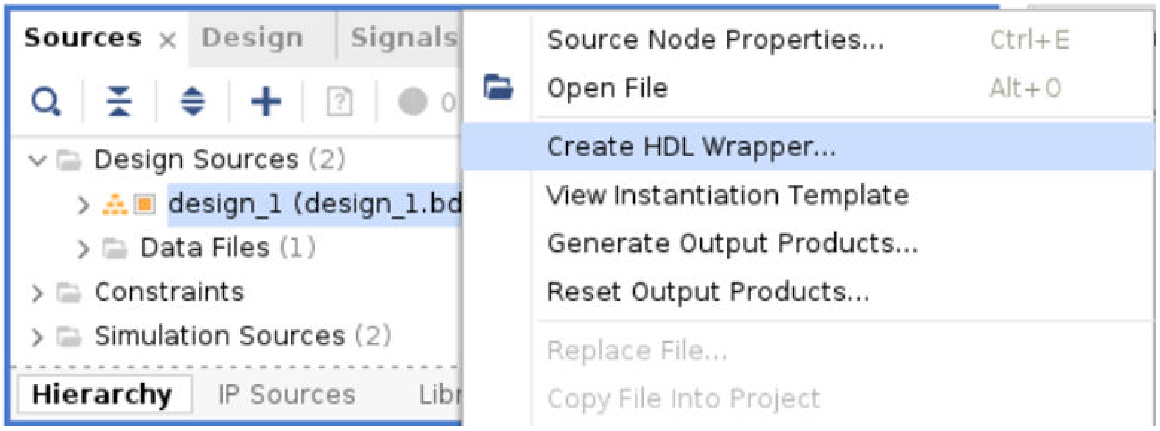
```
source C:\vivado_power_tutorial\design.tcl
```

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.

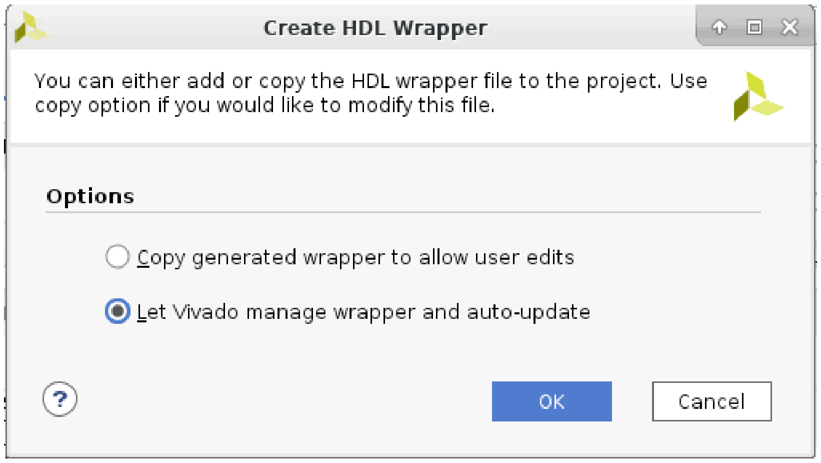
7. To generate HDL wrapper:

   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 **Let Vivado manage wrapper and auto-update** option and click **OK**.



You can either add or copy the HDL wrapper file to the project. Use copy option if you would like to modify this file.

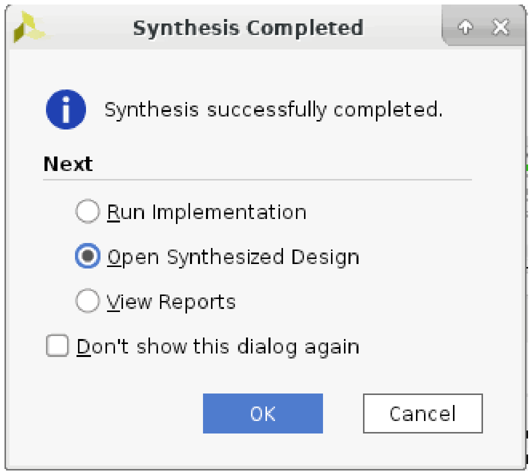
( ) Copy generated wrapper to allow user edits

[✓ Let Vivado manage wrapper and auto-update]

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 appears after synthesis is completed on the design.



Synthesis Completed

Synthesis successfully completed.

Next

Run Implementation [✓ Open Synthesized Design] View Reports

Don't show this dialog again

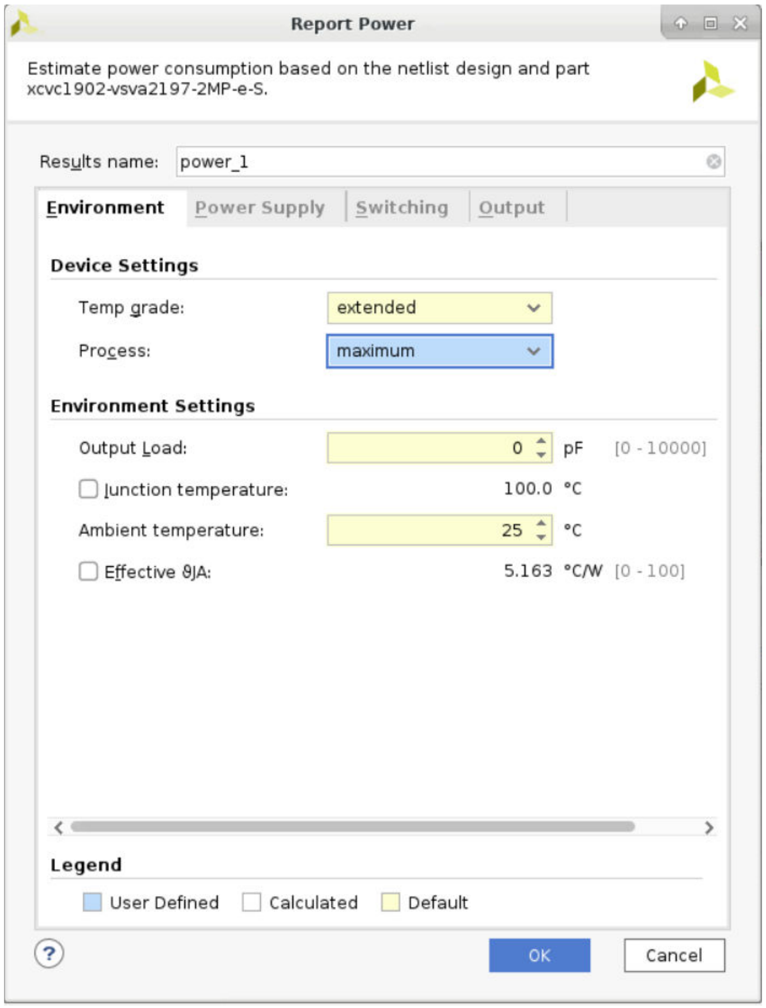
3. Select **Open Synthesized Design** in the Synthesis Completed dialog box and then click **OK** to open the synthesized design.

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 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.



Report Power

Estimate power consumption based on the netlist design and part xcvc1902-vsva2197-2MP-e-S.

Results name: power_1

Device Settings

Temp grade:	extended
Process:	maximum

Environment Settings

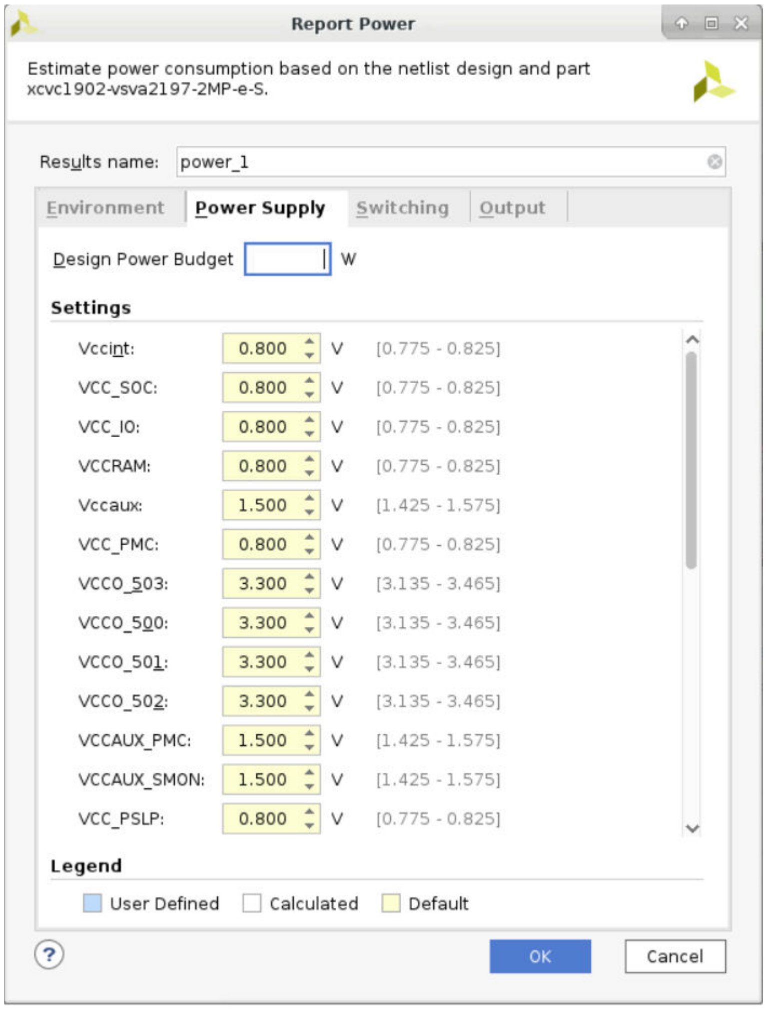
Output Load:	0	pF	[0 - 10000]
☐ Junction temperature:	100.0	°C	
Ambient temperature:	25	°C	
☐ Effective θJA:	5.163	°C/W	[0 - 100]

Legend

🟦 User Defined ☐ Calculated 🟨 Default

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.*



Report Power

Estimate power consumption based on the netlist design and part xcvc1902-vsva2197-2MP-e-S.

Results name: power_1

Design Power Budget [ ] W

Settings

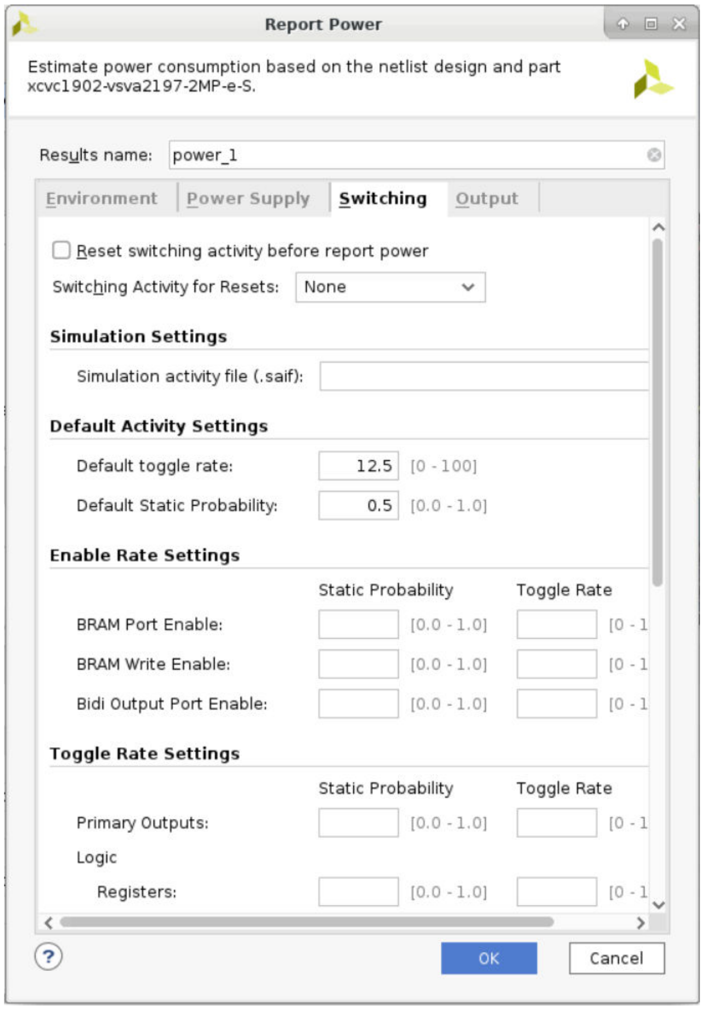
Vccint:	0.800	V	[0.775 - 0.825]
VCC_SOC:	0.800	V	[0.775 - 0.825]
VCC_IO:	0.800	V	[0.775 - 0.825]
VCCRAM:	0.800	V	[0.775 - 0.825]
Vccaux:	1.500	V	[1.425 - 1.575]
VCC_PMC:	0.800	V	[0.775 - 0.825]
VCCO_503:	3.300	V	[3.135 - 3.465]
VCCO_500:	3.300	V	[3.135 - 3.465]
VCCO_501:	3.300	V	[3.135 - 3.465]
VCCO_502:	3.300	V	[3.135 - 3.465]
VCCAUX_PMC:	1.500	V	[1.425 - 1.575]
VCCAUX_SMON:	1.500	V	[1.425 - 1.575]
VCC_PSLP:	0.800	V	[0.775 - 0.825]

Legend

🟦 User Defined ☐ Calculated 🟨 Default

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.*


Report Power

Estimate power consumption based on the netlist design and part xcvc1902-vsva2197-2MP-e-S.

Results name: power_1

☐ Reset switching activity before report power

Switching Activity for Resets: None

Simulation Settings

Simulation activity file (.saif):

Default Activity Settings

Default toggle rate: 12.5 [0 - 100]

Default Static Probability: 0.5 [0.0 - 1.0]

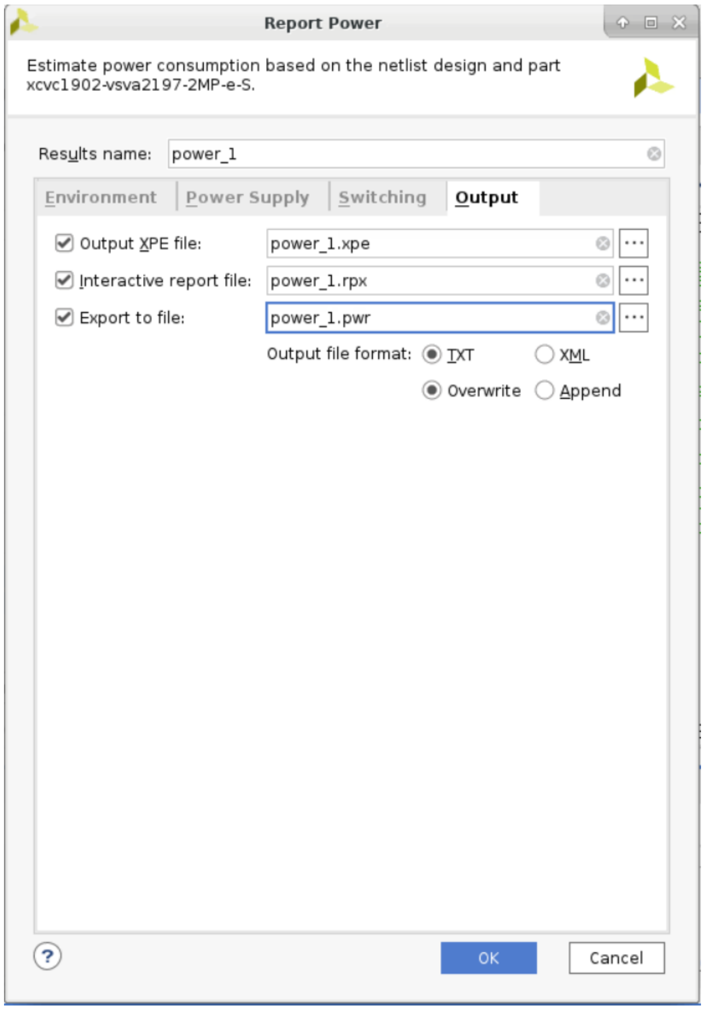
Enable Rate Settings

	Static Probability		Toggle Rate	
BRAM Port Enable:		[0.0 - 1.0]		[0 - 1
BRAM Write Enable:		[0.0 - 1.0]		[0 - 1
Bidi Output Port Enable:		[0.0 - 1.0]		[0 - 1

Toggle Rate Settings

	Static Probability		Toggle Rate	
Primary Outputs:		[0.0 - 1.0]		[0 - 1
Logic				
Registers:		[0.0 - 1.0]		[0 - 1

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 its results 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.


Report Power

Estimate power consumption based on the netlist design and part xcvc1902-vsva2197-2MP-e-S.

Results name: power_1

[✓] Output XPE file: power_1.xpe

[✓] Interactive report file: power_1.rpx

[✓] Export to file: power_1.pwr

Output file format: [✓ TXT] XML

[✓ Overwrite] Append

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.

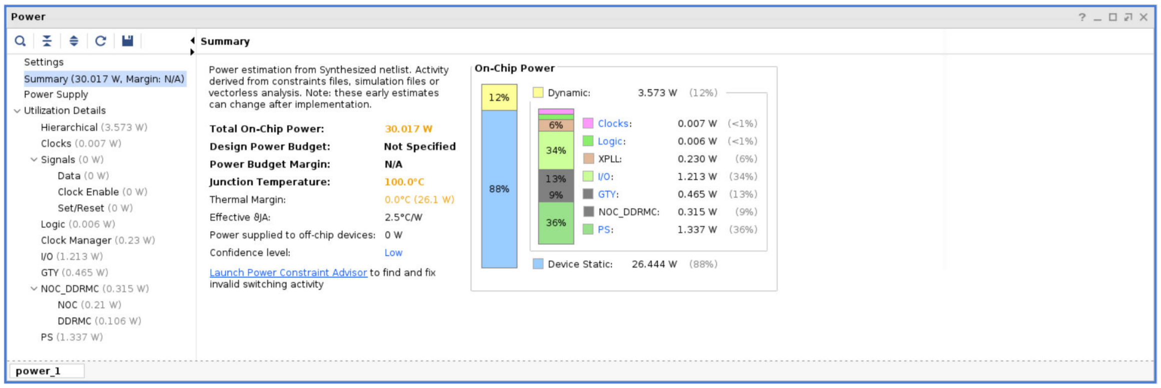
Step 4: Running Report Power

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 Vivado tools, the actual power numbers you see might be slightly different than the ones that appear in the following figures.



Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

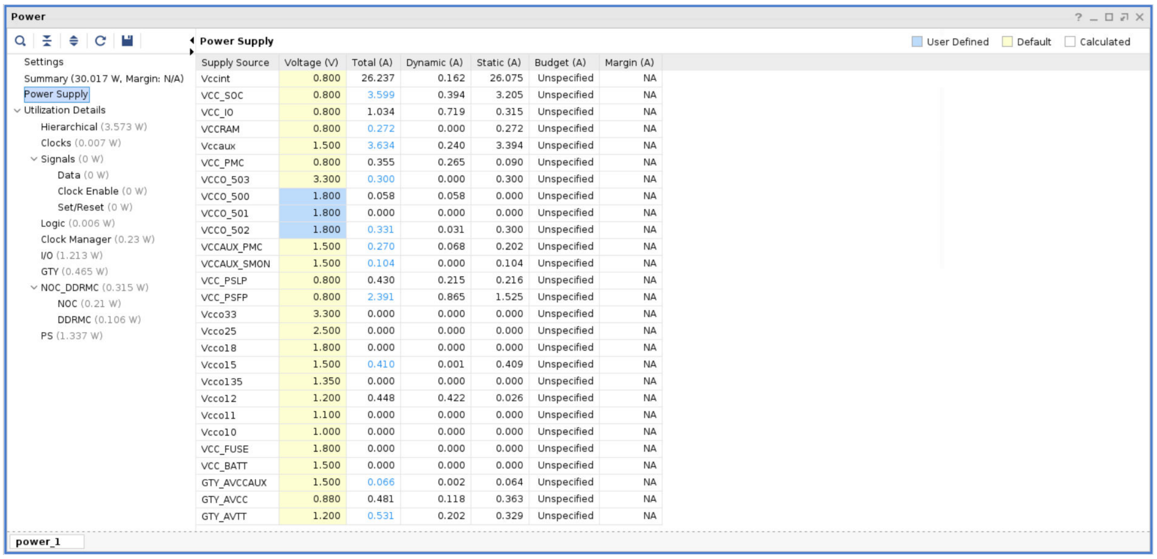
Total On-Chip Power:	30.017 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	100.0°C
Thermal Margin:	0.0°C (26.1 W)
Effective ϑJA:	2.5°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Launch Power Constraint Advisor to find and fix invalid switching activity

Dynamic:	3.573 W	(12%)
Clocks:	0.007 W	(<1%)
Logic:	0.006 W	(<1%)
XPLL:	0.230 W	(6%)
I/O:	1.213 W	(34%)
GTY:	0.465 W	(13%)
NOC_DDRMC:	0.315 W	(9%)
PS:	1.337 W	(36%)
Device Static:	26.444 W	(88%)

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

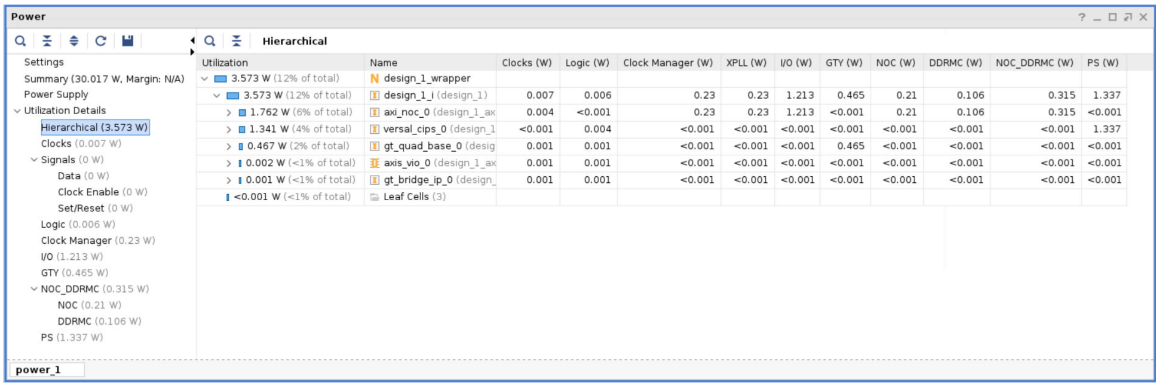
4. Examine the current consumption by individual rails in the Power Supply view.



Power Supply

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

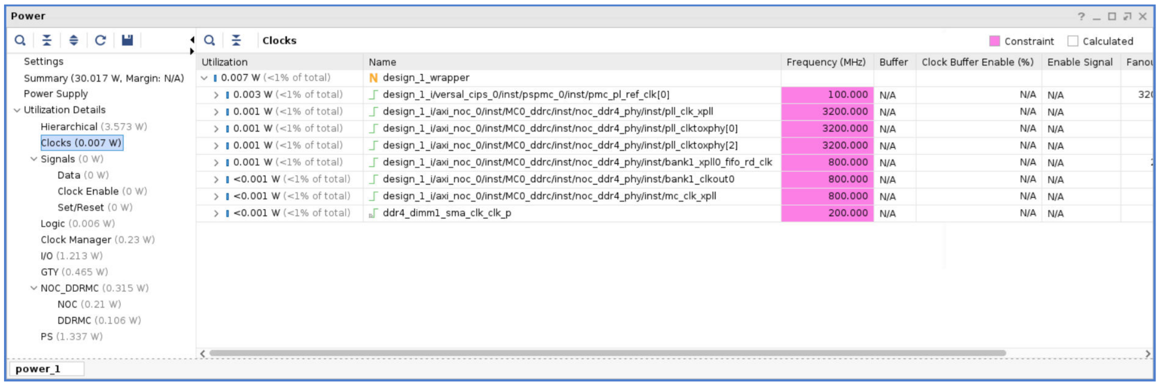
5. Examine the hierarchical breakdown of the power in the **Utilization Details → Hierarchical** view.



- Settings
- Summary (30.017 W, Margin: N/A)
- Power Supply
- Utilization Details
  - Hierarchical (3.573 W)
  - Clocks (0.007 W)
  - Signals (0 W)
    - Data (0 W)
    - Clock Enable (0 W)
    - Set/Reset (0 W)
  - Logic (0.006 W)
  - Clock Manager (0.23 W)
  - I/O (1.213 W)
  - GTY (0.465 W)
  - NOC_DDRMC (0.315 W)
    - NOC (0.21 W)
    - DDRMC (0.106 W)
  - PS (1.337 W)

Utilization	Name	Clocks (W)	Logic (W)	Clock Manager (W)	XPLL (W)	I/O (W)	GTY (W)	NOC (W)	DDRMC (W)	NOC_DDRMC (W)	PS (W)
3.573 W (12% of total)	design_1_wrapper										
3.573 W (12% of total)	design_1_i (design_1)	0.007	0.006	0.23	0.23	1.213	0.465	0.21	0.106	0.315	1.337
1.762 W (6% of total)	axi_noc_0 (design_1_ax	0.004	<0.001	0.23	0.23	1.213	<0.001	0.21	0.106	0.315	<0.001
1.341 W (4% of total)	versal_cips_0 (design_1	<0.001	0.004	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	1.337
0.467 W (2% of total)	gt_quad_base_0 (desig	0.001	0.001	<0.001	<0.001	<0.001	0.465	<0.001	<0.001	<0.001	<0.001
0.002 W (<1% of total)	axis_vio_0 (design_1_ax	0.001	0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
0.001 W (<1% of total)	gt_bridge_ip_0 (design_	0.001	0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
<0.001 W (<1% of total)	Leaf Cells (3)										

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



- Settings
- Summary (30.017 W, Margin: N/A)
- Power Supply
- Utilization Details
  - Hierarchical (3.573 W)
  - Clocks (0.007 W)
  - Signals (0 W)
    - Data (0 W)
    - Clock Enable (0 W)
    - Set/Reset (0 W)
  - Logic (0.006 W)
  - Clock Manager (0.23 W)
  - I/O (1.213 W)
  - GTY (0.465 W)
  - NOC_DDRMC (0.315 W)
    - NOC (0.21 W)
    - DDRMC (0.106 W)
  - PS (1.337 W)

Utilization	Name	Frequency (MHz)	Buffer	Clock Buffer Enable (%)	Enable Signal	Fanou
0.007 W (<1% of total)	design_1_wrapper					
0.003 W (<1% of total)	design_1_i/versal_cips_0/inst/pspmc_0/inst/pmc_pl_ref_clk[0]	100.000	N/A	N/A	N/A	32
0.001 W (<1% of total)	design_1_i/axi_noc_0/inst/MC0_ddrc/inst/noc_ddr4_phy/inst/pll_clk_xpll	3200.000	N/A	N/A	N/A	
0.001 W (<1% of total)	design_1_i/axi_noc_0/inst/MC0_ddrc/inst/noc_ddr4_phy/inst/pll_clktoxphy[0]	3200.000	N/A	N/A	N/A	
0.001 W (<1% of total)	design_1_i/axi_noc_0/inst/MC0_ddrc/inst/noc_ddr4_phy/inst/pll_clktoxphy[2]	3200.000	N/A	N/A	N/A	
0.001 W (<1% of total)	design_1_i/axi_noc_0/inst/MC0_ddrc/inst/noc_ddr4_phy/inst/bank1_xpll0_fifo_rd_clk	800.000	N/A	N/A	N/A	2
<0.001 W (<1% of total)	design_1_i/axi_noc_0/inst/MC0_ddrc/inst/noc_ddr4_phy/inst/bank1_clkout0	800.000	N/A	N/A	N/A	
<0.001 W (<1% of total)	design_1_i/axi_noc_0/inst/MC0_ddrc/inst/noc_ddr4_phy/inst/mc_clk_xpll	800.000	N/A	N/A	N/A	
<0.001 W (<1% of total)	ddr4_dimm1_sma_clk_clk_p	200.000	N/A	N/A	N/A	

Step 5: 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, click **Run Implementation**.
2. When the Save Project dialog box is displayed before launching implementation, click **Don't Save**.

Designing with UltraScale+

Step 1: Creating a New Project

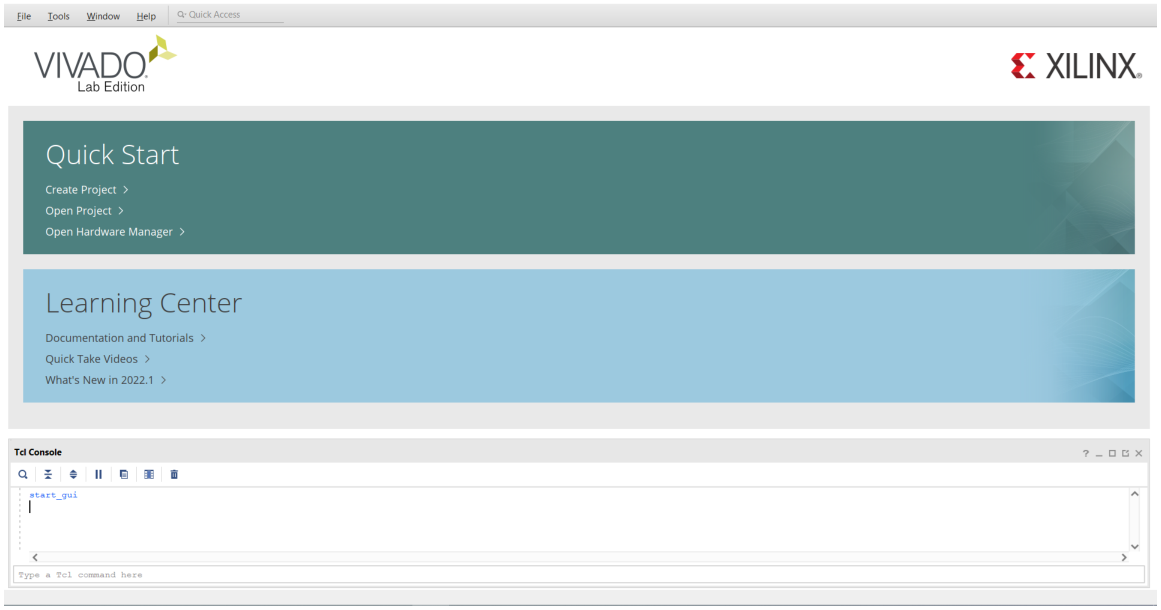
To create a project, use the New Project wizard to name the project, to add RTL source files and constraints, and to specify the target device.

On Linux, do the following.

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

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

2. Launch Vivado IDE: `vivado`



```
start_gui
```

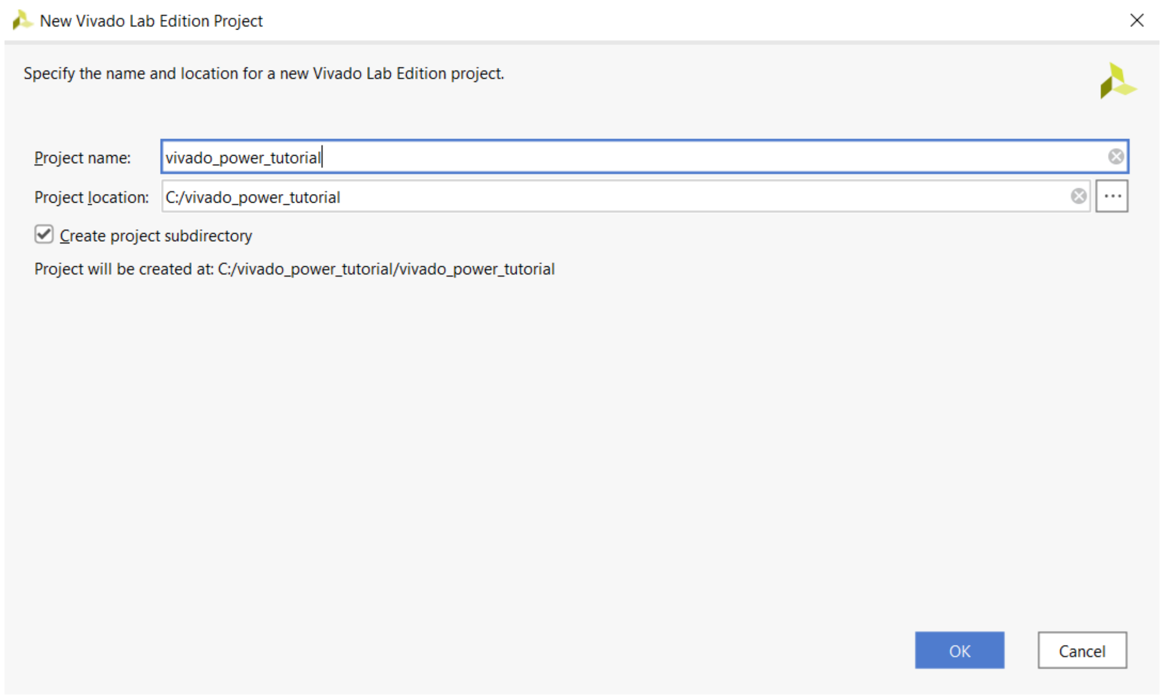
On Windows, do the following.

3. Launch the Vivado IDE by selecting **Start→ All Programs→ Xilinx Design Tools→ Vivado 2022.x→ Vivado 2022.x** (x denotes the latest version of Vivado 2022 IDE).

   As an alternative, click Vivado 2022.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.



Specify the name and location for a new Vivado Lab Edition project.

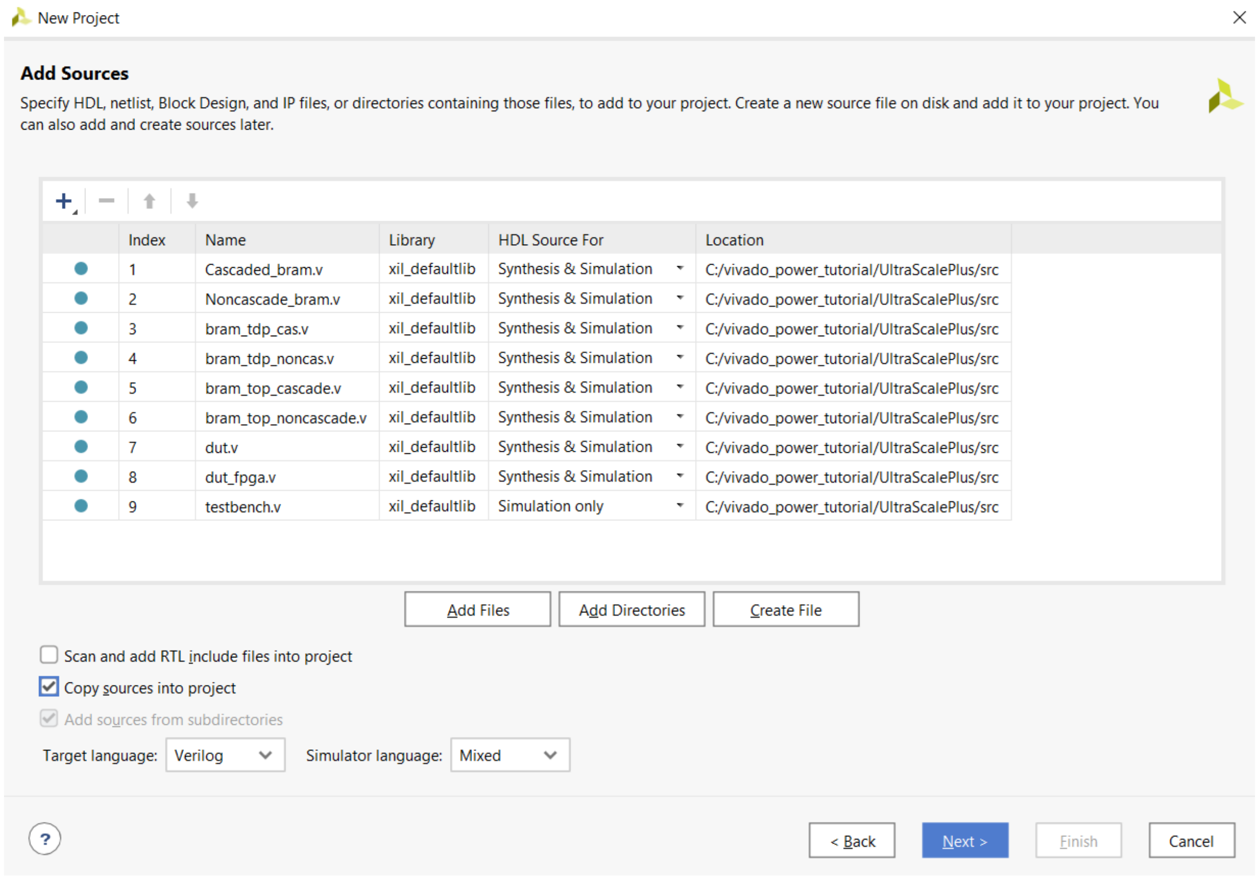
Project name: vivado_power_tutorial

Project location: C:/vivado_power_tutorial

[✓] Create project subdirectory

Project will be created at: C:/vivado_power_tutorial/vivado_power_tutorial

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**.



Add Sources

Specify HDL, netlist, Block Design, and IP files, or directories containing those files, to add to your project. Create a new source file on disk and add it to your project. You can also add and create sources later.

	Index	Name	Library	HDL Source For	Location
●	1	Cascaded_bram.v	xil_defaultlib	Synthesis & Simulation	C:/vivado_power_tutorial/UltraScalePlus/src
●	2	Noncascade_bram.v	xil_defaultlib	Synthesis & Simulation	C:/vivado_power_tutorial/UltraScalePlus/src
●	3	bram_tdp_cas.v	xil_defaultlib	Synthesis & Simulation	C:/vivado_power_tutorial/UltraScalePlus/src
●	4	bram_tdp_noncas.v	xil_defaultlib	Synthesis & Simulation	C:/vivado_power_tutorial/UltraScalePlus/src
●	5	bram_top_cascade.v	xil_defaultlib	Synthesis & Simulation	C:/vivado_power_tutorial/UltraScalePlus/src
●	6	bram_top_noncascade.v	xil_defaultlib	Synthesis & Simulation	C:/vivado_power_tutorial/UltraScalePlus/src
●	7	dut.v	xil_defaultlib	Synthesis & Simulation	C:/vivado_power_tutorial/UltraScalePlus/src
●	8	dut_fpga.v	xil_defaultlib	Synthesis & Simulation	C:/vivado_power_tutorial/UltraScalePlus/src
●	9	testbench.v	xil_defaultlib	Simulation only	C:/vivado_power_tutorial/UltraScalePlus/src

☐ Scan and add RTL include files into project
☑ Copy sources into project
☑ Add sources from subdirectories

Target language: Verilog   Simulator language: Mixed

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, you will find the `dut_fpga_zcu102.xdc` file below the `/src` folder.

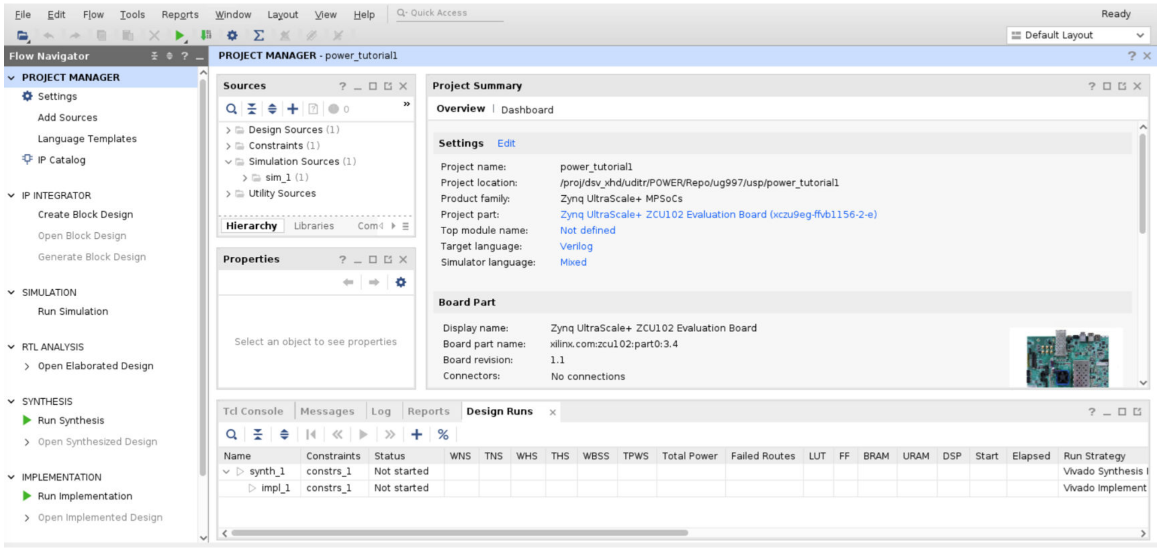
10. Click **Next** to continue.

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



Settings Edit

Project name:	power_tutorial1
Project location:	/proj/dsv_xhd/uditr/POWER/Repo/ug997/usp/power_tutorial1
Product family:	Zynq UltraScale+ MPSoCs
Project part:	Zynq UltraScale+ ZCU102 Evaluation Board (xczu9eg-ffvb1156-2-e)
Top module name:	Not defined
Target language:	Verilog
Simulator language:	Mixed

Board Part

Display name:	Zynq UltraScale+ ZCU102 Evaluation Board
Board part name:	xilinx.com:zcu102:part0:3.4
Board revision:	1.1
Connectors:	No connections

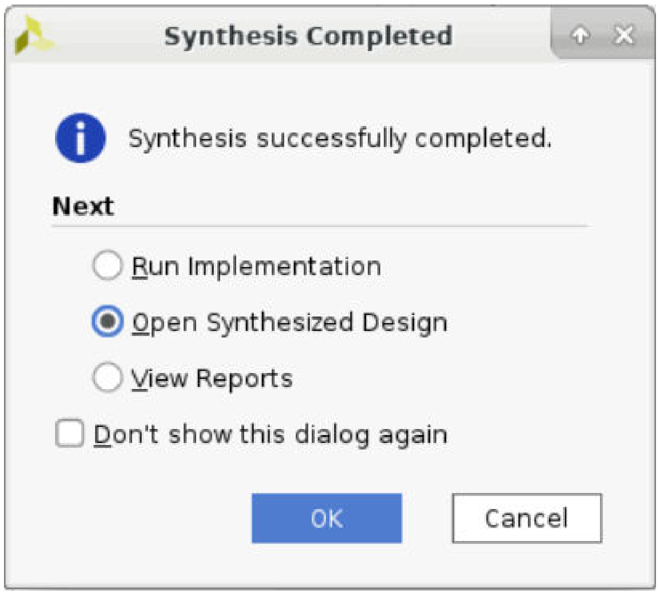
Name	Constraints	Status	WNS	TNS	WHS	THS	WBSS	TPWS	Total Power	Failed Routes	LUT	FF	BRAM	URAM	DSP	Start	Elapsed	Run Strategy
synth_1	constrs_1	Not started																Vivado Synthesis I
impl_1	constrs_1	Not started																Vivado Implement

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



Synthesis Completed

Synthesis successfully completed.

Next

( ) Run Implementation

[✓ Open Synthesized Design]

( ) View Reports

☐ Don't show this dialog again

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

Step 3: Report Power Setup

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

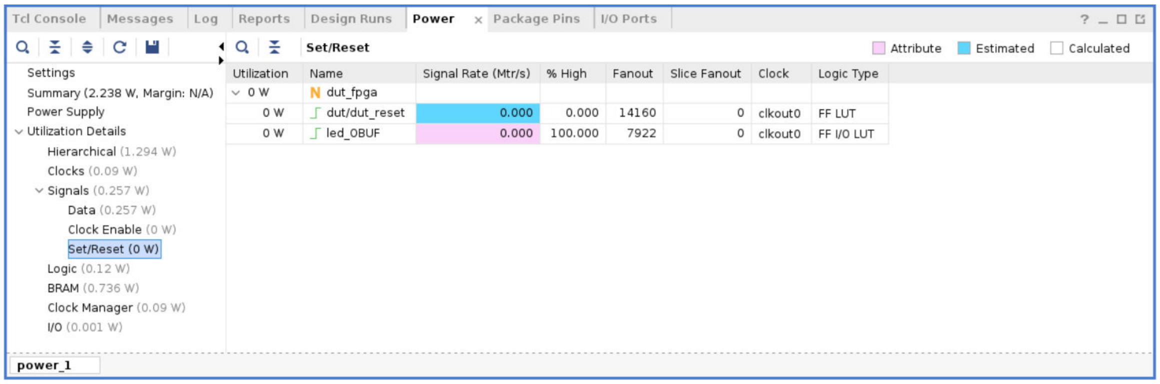
Step 4: Running Report Power

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

Step 5: Viewing the Power Properties

This step walks you through the process of getting 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.

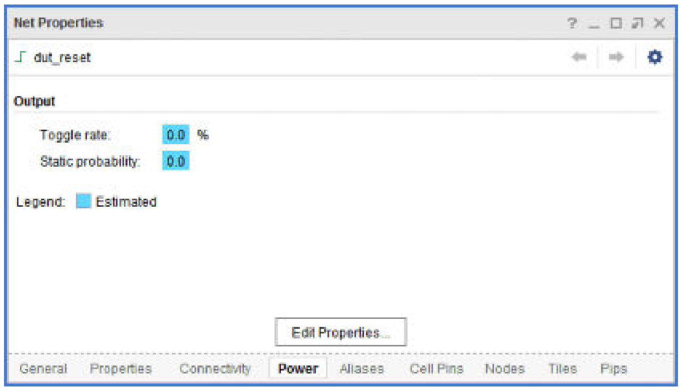


Settings
Summary (2.238 W, Margin: N/A)
Power Supply
Utilization Details
Hierarchical (1.294 W)
Clocks (0.09 W)
Signals (0.257 W)
Data (0.257 W)
Clock Enable (0 W)
Set/Reset (0 W)
Logic (0.12 W)
BRAM (0.736 W)
Clock Manager (0.09 W)
I/O (0.001 W)

Set/Reset — Attribute, Estimated, Calculated

Utilization	Name	Signal Rate (Mtr/s)	% High	Fanout	Slice Fanout	Clock	Logic Type
0 W	dut_fpga						
0 W	dut/dut_reset	0.000	0.000	14160	0	clkout0	FF LUT
0 W	led_OBUF	0.000	100.000	7922	0	clkout0	FF I/O LUT

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 get the power information for the first time.



dut_reset

Output

Toggle rate: 0.0 %
Static probability: 0.0

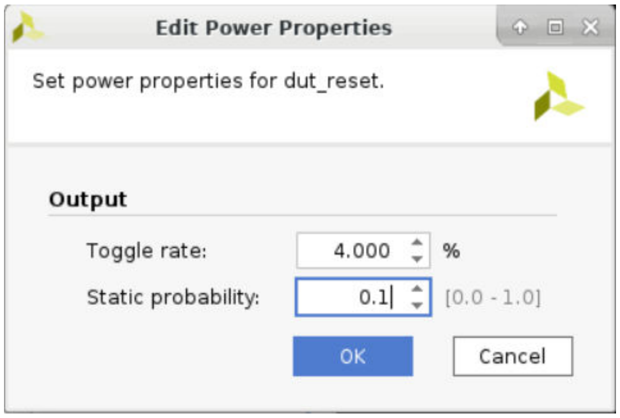
Legend: 🟦 Estimated

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.



Edit Power Properties

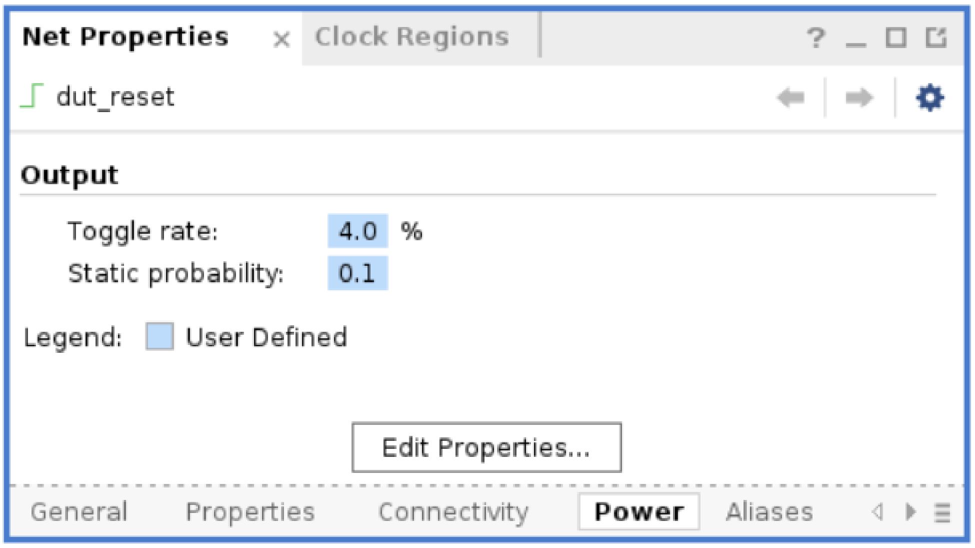
Set power properties for dut_reset.

Output

Toggle rate: 4.000 %

Static probability: 0.1 [0.0 - 1.0]

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.



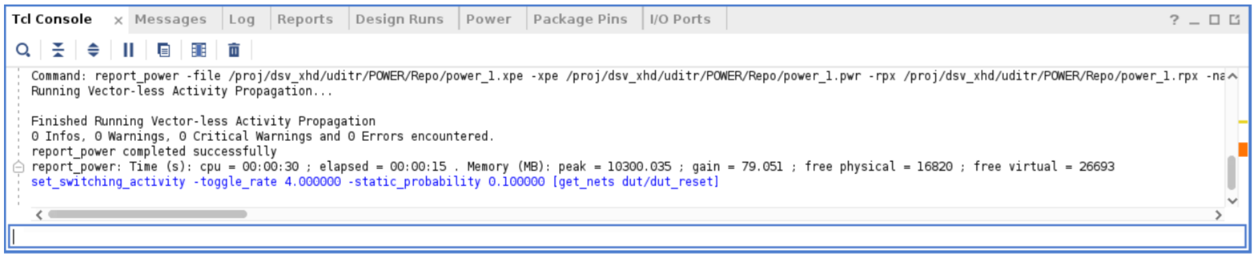
Output

Toggle rate: 4.0 %

Static probability: 0.1

Legend: 🟦 User Defined

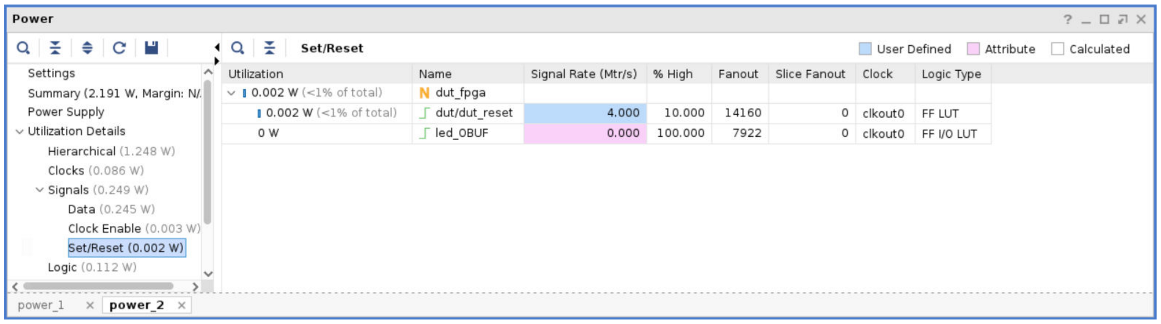
You can also observe the equivalent Tcl command that is executed in the Tcl Console.



```
Command: report_power -file /proj/dsv_xhd/uditr/POWER/Repo/power_1.xpe -xpe /proj/dsv_xhd/uditr/POWER/Repo/power_1.pwr -rpx /proj/dsv_xhd/uditr/POWER/Repo/power_1.rpx -na
Running Vector-less Activity Propagation...

Finished Running Vector-less Activity Propagation
0 Infos, 0 Warnings, 0 Critical Warnings and 0 Errors encountered.
report_power completed successfully
report_power: Time (s): cpu = 00:00:30 ; elapsed = 00:00:15 . Memory (MB): peak = 10300.035 ; gain = 79.051 ; free physical = 16820 ; free virtual = 26693
set_switching_activity -toggle_rate 4.000000 -static_probability 0.100000 [get_nets dut/dut_reset]
```

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. 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.



Utilization	Name	Signal Rate (Mtr/s)	% High	Fanout	Slice Fanout	Clock	Logic Type
0.002 W (<1% of total)	dut_fpga						
0.002 W (<1% of total)	dut/dut_reset	4.000	10.000	14160	0	clkout0	FF LUT
0 W	led_OBUF	0.000	100.000	7922	0	clkout0	FF I/O LUT

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 towards the 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]
```

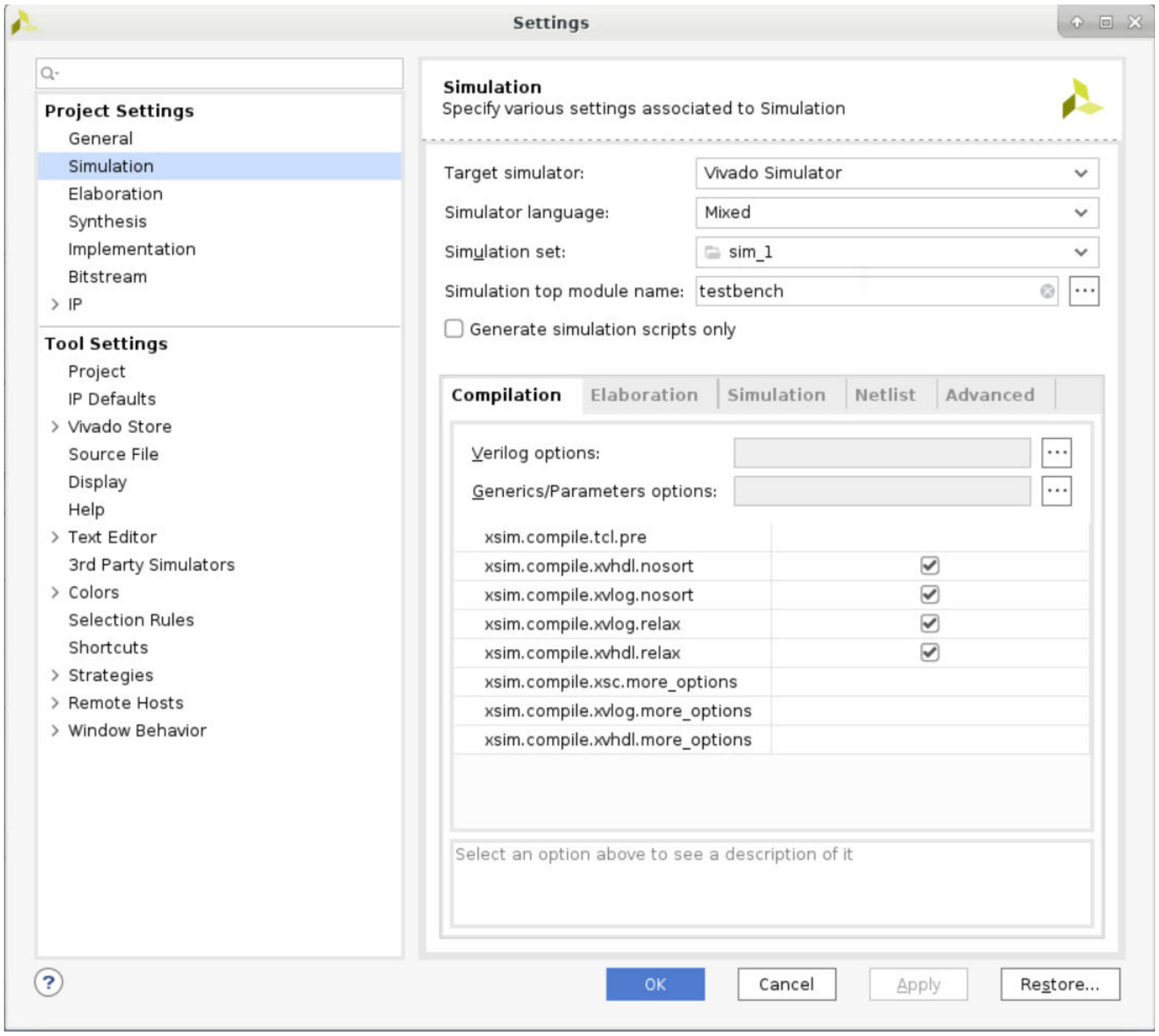
IMPORTANT! *Switching activity can also be specified in terms of toggle rate. Toggle rate is always associated with a clock. The primary ports can be associated with a specific clock using the* `set_input_delay` *and* `set_output_delay` *commands. If no clock association is found, Report Power will associate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



Simulation
Specify various settings associated to Simulation

Target simulator:	Vivado Simulator
Simulator language:	Mixed
Simulation set:	sim_1
Simulation top module name:	testbench

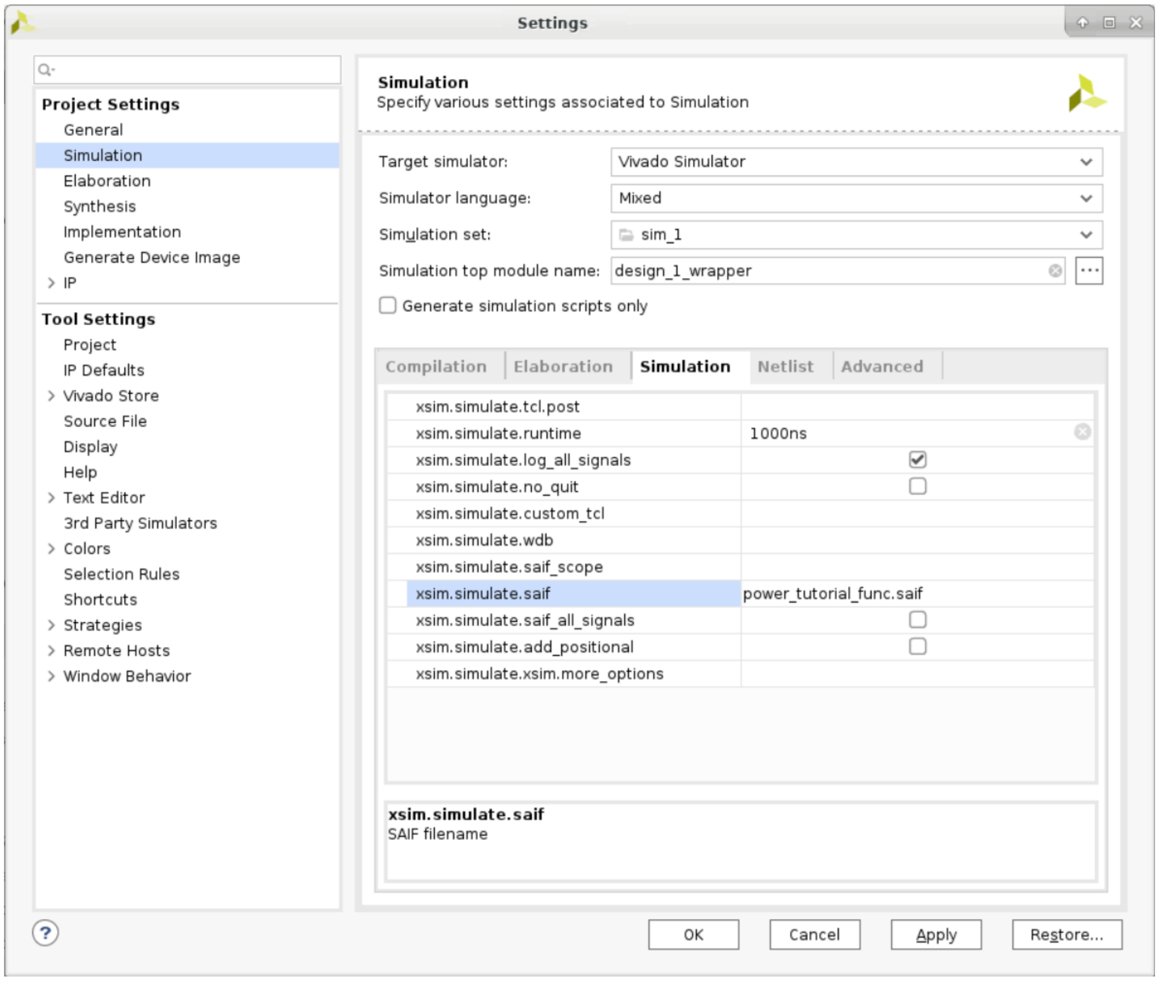
☐ Generate simulation scripts only

Verilog options:
Generics/Parameters options:

xsim.compile.tcl.pre	
xsim.compile.xvhdl.nosort	✓
xsim.compile.xvlog.nosort	✓
xsim.compile.xvlog.relax	✓
xsim.compile.xvhdl.relax	✓
xsim.compile.xsc.more_options	
xsim.compile.xvlog.more_options	
xsim.compile.xvhdl.more_options	

Select an option above to see a description of it

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**.


Settings

Simulation
Specify various settings associated to Simulation

Target simulator:	Vivado Simulator
Simulator language:	Mixed
Simulation set:	sim_1
Simulation top module name:	design_1_wrapper

☐ Generate simulation scripts only

Compilation | Elaboration | **Simulation** | Netlist | Advanced

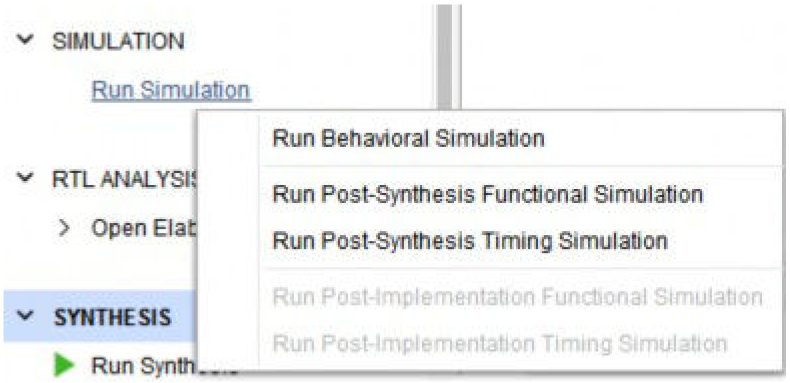
xsim.simulate.tcl.post	
xsim.simulate.runtime	1000ns
xsim.simulate.log_all_signals	☑
xsim.simulate.no_quit	☐
xsim.simulate.custom_tcl	
xsim.simulate.wdb	
xsim.simulate.saif_scope	
xsim.simulate.saif	power_tutorial_func.saif
xsim.simulate.saif_all_signals	☐
xsim.simulate.add_positional	☐
xsim.simulate.xsim.more_options	

xsim.simulate.saif
SAIF filename

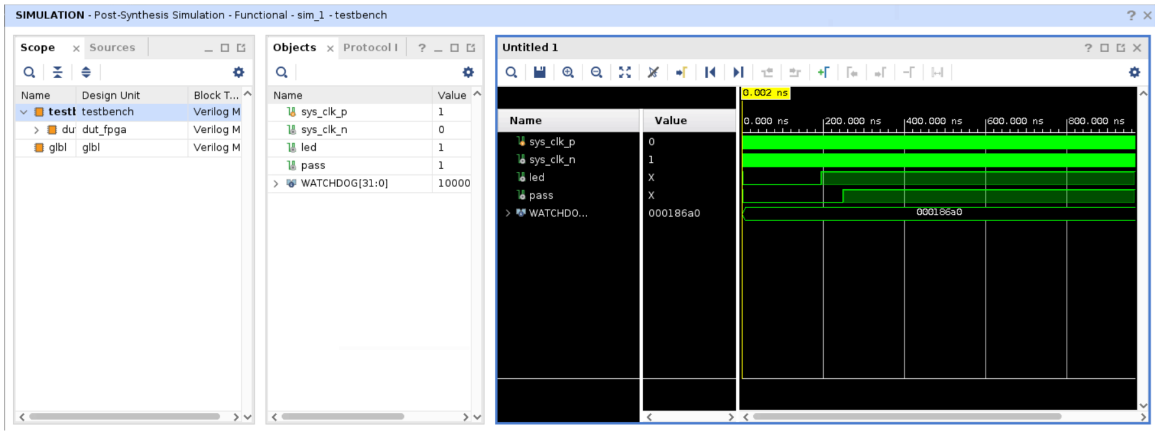
The simulation settings are now properly configured. You can launch the Vivado simulator to perform a post-synthesis functional simulation of the design.

***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**.


SIMULATION
Run Simulation
RTL ANALYSIS
Open Elab
SYNTHESIS
Run Synth
Run Behavioral Simulation
Run Post-Synthesis Functional Simulation
Run Post-Synthesis Timing Simulation
Run Post-Implementation Functional Simulation
Run Post-Implementation Timing 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.

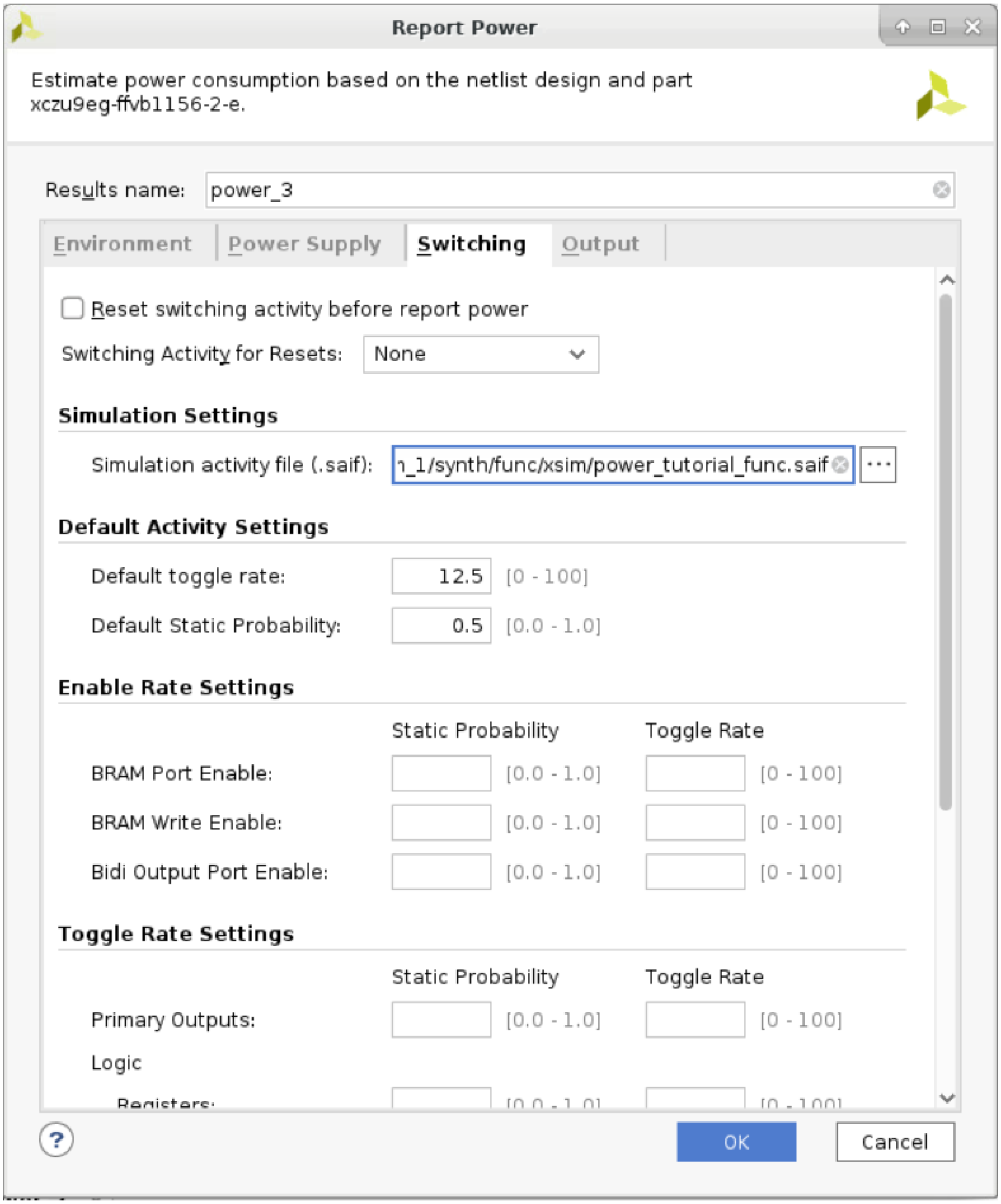
Step 8: Incorporating SAIF Data into Power Analysis

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 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.



Estimate power consumption based on the netlist design and part xczu9eg-ffvb1156-2-e.

Results name: power_3

☐ Reset switching activity before report power

Switching Activity for Resets: None

Simulation Settings

Simulation activity file (.saif): ı_1/synth/func/xsim/power_tutorial_func.saif

Default Activity Settings

Default toggle rate:	12.5	[0 - 100]
Default Static Probability:	0.5	[0.0 - 1.0]

Enable Rate Settings

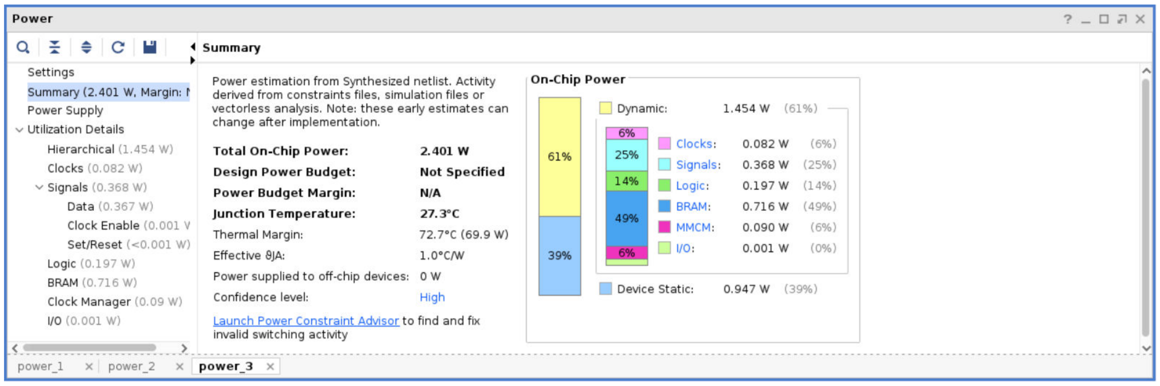
	Static Probability		Toggle Rate	
BRAM Port Enable:		[0.0 - 1.0]		[0 - 100]
BRAM Write Enable:		[0.0 - 1.0]		[0 - 100]
Bidi Output Port Enable:		[0.0 - 1.0]		[0 - 100]

Toggle Rate Settings

	Static Probability		Toggle Rate	
Primary Outputs:		[0.0 - 1.0]		[0 - 100]
Logic				
Registers:		[0.0 - 1.0]		[0 - 100]

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.


Power

Summary

- Settings
- Summary (2.401 W, Margin: N
- Power Supply
- Utilization Details
  - Hierarchical (1.454 W)
  - Clocks (0.082 W)
  - Signals (0.368 W)
    - Data (0.367 W)
    - Clock Enable (0.001 W
    - Set/Reset (<0.001 W)
  - Logic (0.197 W)
  - BRAM (0.716 W)
  - Clock Manager (0.09 W)
  - I/O (0.001 W)

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	**2.401 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**27.3°C**
Thermal Margin:	72.7°C (69.9 W)
Effective θJA:	1.0°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	High

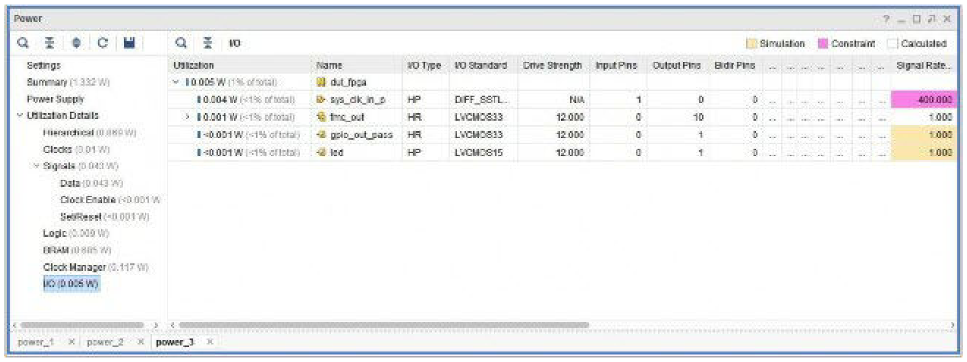
Launch Power Constraint Advisor to find and fix invalid switching activity

On-Chip Power

Dynamic:	1.454 W	(61%)
Clocks:	0.082 W	(6%)
Signals:	0.368 W	(25%)
Logic:	0.197 W	(14%)
BRAM:	0.716 W	(49%)
MMCM:	0.090 W	(6%)
I/O:	0.001 W	(0%)
Device Static:	0.947 W	(39%)

power_1 | power_2 | **power_3**

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


Power

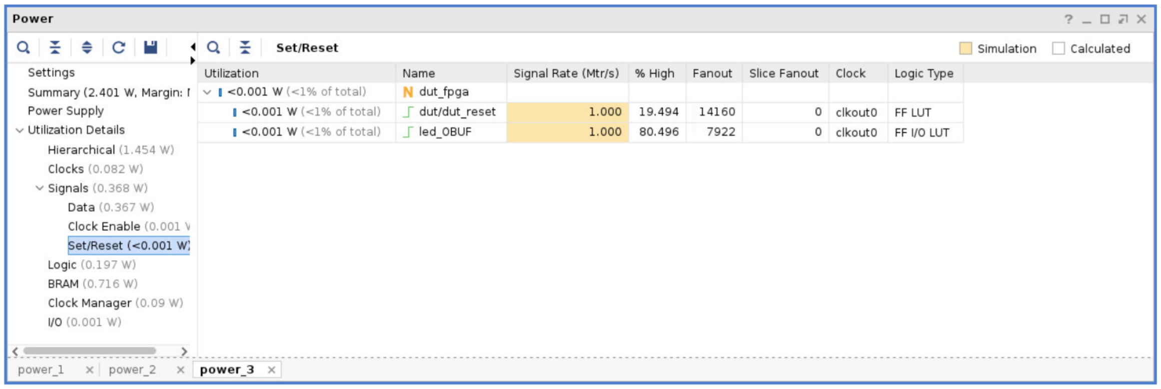
I/O — Simulation, Constraint, Calculated

- Settings
- Summary (1.332 W)
- Power Supply
- Utilization Details
  - Hierarchical (0.889 W)
  - Clocks (0.01 W)
  - Signals (0.043 W)
    - Data (0.043 W)
    - Clock Enable (<0.001 W
    - Set/Reset (<0.001 W)
  - Logic (0.009 W)
  - BRAM (0.805 W)
  - Clock Manager (0.117 W)
  - I/O (0.005 W)

Utilization	Name	I/O Type	I/O Standard	Drive Strength	Input Pins	Output Pins	Bidir Pins	Signal Rate..
0.005 W (1% of total)	dut_fpga							
0.004 W (<1% of total)	sys_clk_in_p	HP	DIFF_SSTL..	N/A	1	0	0	400.000
0.001 W (<1% of total)	fmc_out	HR	LVCMOS33	12.000	0	10	0	1.000
<0.001 W (<1% of total)	gpio_out_pass	HR	LVCMOS33	12.000	0	1	0	1.000
<0.001 W (<1% of total)	led	HP	LVCMOS15	12.000	0	1	0	1.000

power_1 | power_2 | **power_3**

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.


Power

Set/Reset — Simulation, Calculated

- Settings
- Summary (2.401 W, Margin: N
- Power Supply
- Utilization Details
  - Hierarchical (1.454 W)
  - Clocks (0.082 W)
  - Signals (0.368 W)
    - Data (0.367 W)
    - Clock Enable (0.001 W
    - Set/Reset (<0.001 W)
  - Logic (0.197 W)
  - BRAM (0.716 W)
  - Clock Manager (0.09 W)
  - I/O (0.001 W)

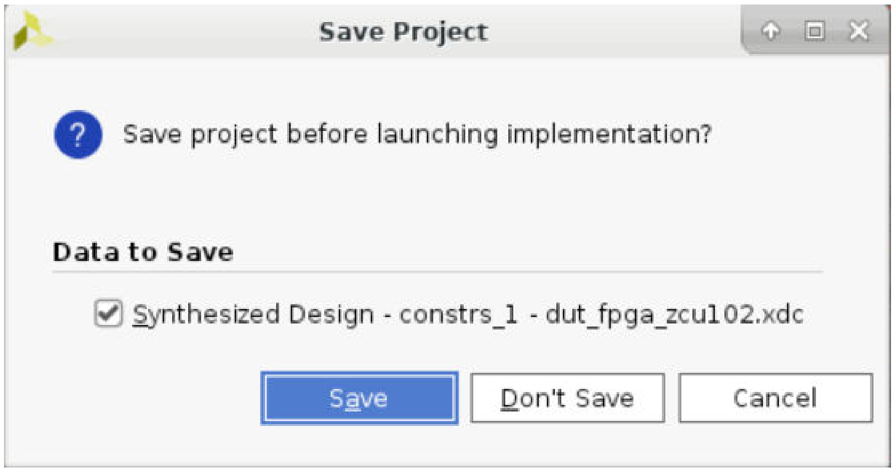
Utilization	Name	Signal Rate (Mtr/s)	% High	Fanout	Slice Fanout	Clock	Logic Type
<0.001 W (<1% of total)	dut_fpga						
<0.001 W (<1% of total)	dut/dut_reset	1.000	19.494	14160	0	clkout0	FF LUT
<0.001 W (<1% of total)	led_OBUF	1.000	80.496	7922	0	clkout0	FF I/O LUT

power_1 | power_2 | **power_3**

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. 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**.



Save Project

Save project before launching implementation?

Data to Save

☑ Synthesized Design - constrs_1 - dut_fpga_zcu102.xdc

Conclusion

In this lab, you have learned how to set the power analysis in the Vivado. In Lab 3, you will learn about the timing simulation and its effect on the power analysis.

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

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 Vivado® simulator and Questa Advanced Simulator. The lab will walk you through the steps to create a SAIF file, run timing simulation, and estimate power using the SAIF data.

***Note*:** This lab onwards, Xilinx® 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**.

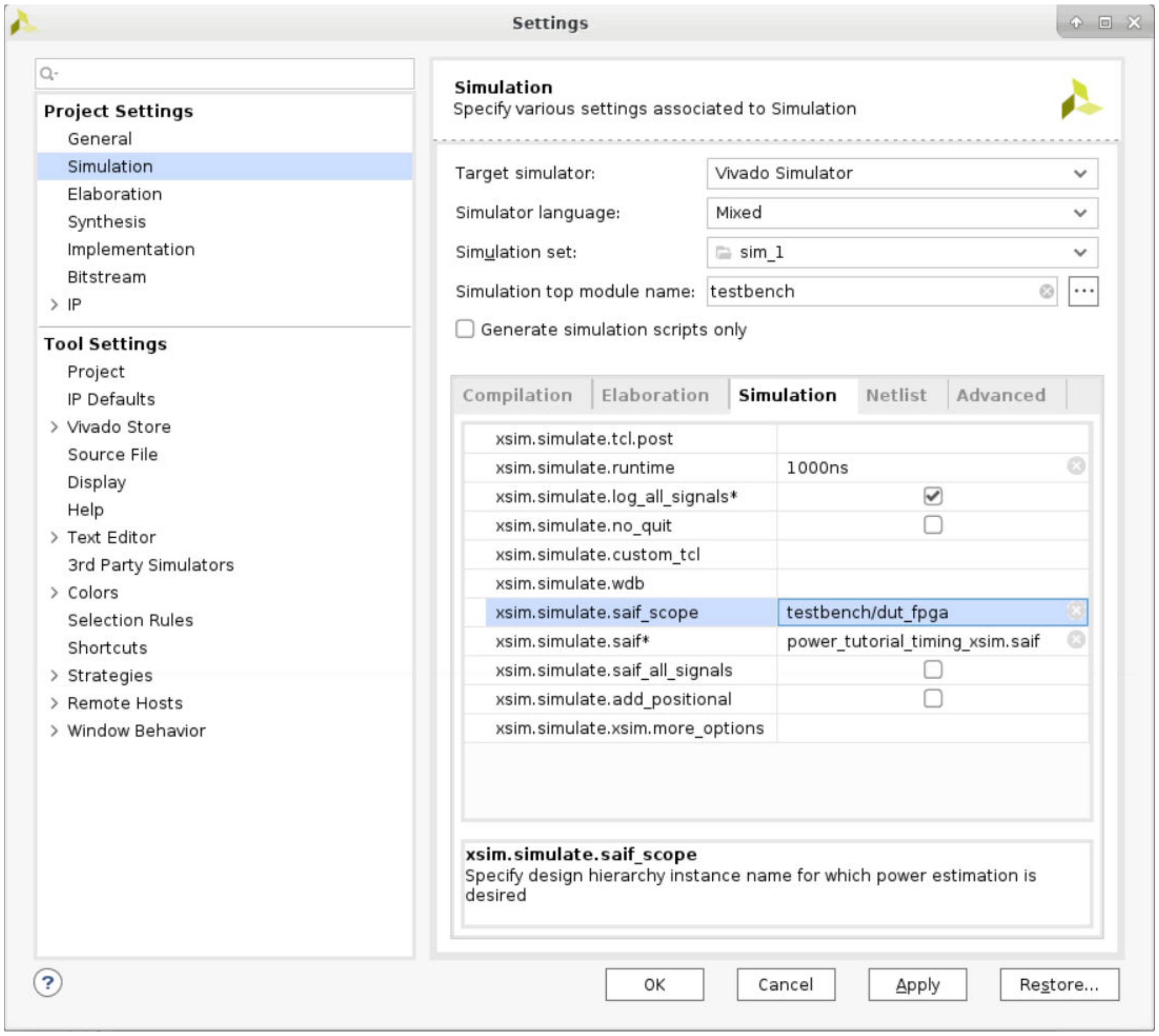
   Now you are ready to set up and launch the Vivado simulator to run post implementation timing simulation. You will set the timing simulation properties in Vivado IDE, then run the timing simulation.

2. In the Flow Navigator, click **Settings** and select **Simulation** to set the timing simulation properties. In the Settings dialog box, 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. Check **xsim.simulate.log_all_signals**.
8. Click **OK**.



Simulation
Specify various settings associated to Simulation

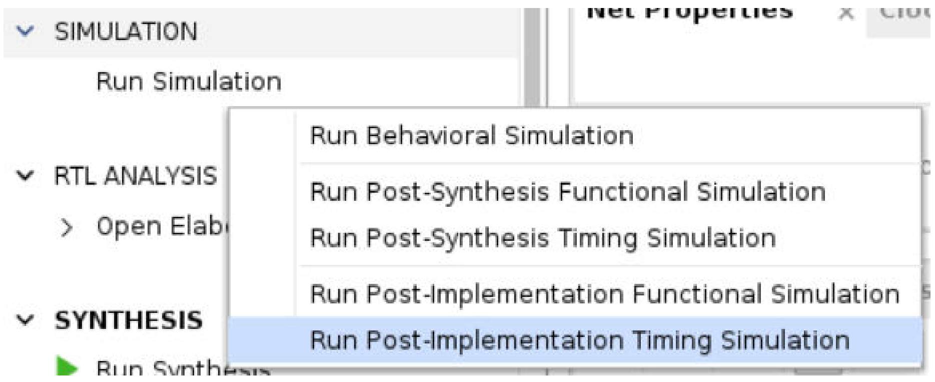
Target simulator: Vivado Simulator
Simulator language: Mixed
Simulation set: sim_1
Simulation top module name: testbench
☐ Generate simulation scripts only

xsim.simulate.tcl.post	
xsim.simulate.runtime	1000ns
xsim.simulate.log_all_signals*	☑
xsim.simulate.no_quit	☐
xsim.simulate.custom_tcl	
xsim.simulate.wdb	
xsim.simulate.saif_scope	testbench/dut_fpga
xsim.simulate.saif*	power_tutorial_timing_xsim.saif
xsim.simulate.saif_all_signals	☐
xsim.simulate.add_positional	☐
xsim.simulate.xsim.more_options	

xsim.simulate.saif_scope
Specify design hierarchy instance name for which power estimation is desired

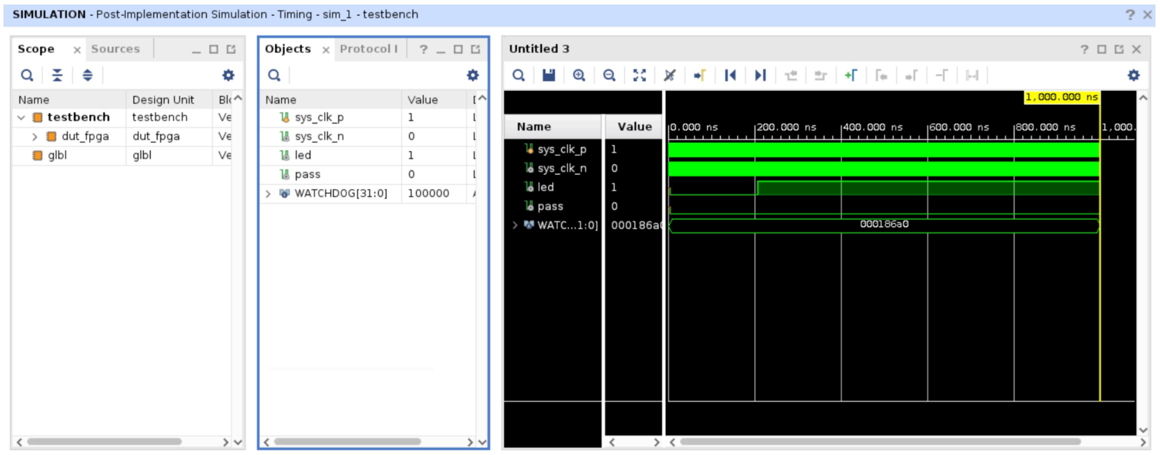
With the simulation settings properly configured, you can launch the Vivado simulator to perform a timing simulation of the post implemented design.

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


SIMULATION
Run Simulation
RTL ANALYSIS
Open Elab
SYNTHESIS
Run Synthesis
Net Properties
Run Behavioral Simulation
Run Post-Synthesis Functional Simulation
Run Post-Synthesis Timing Simulation
Run Post-Implementation Functional 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

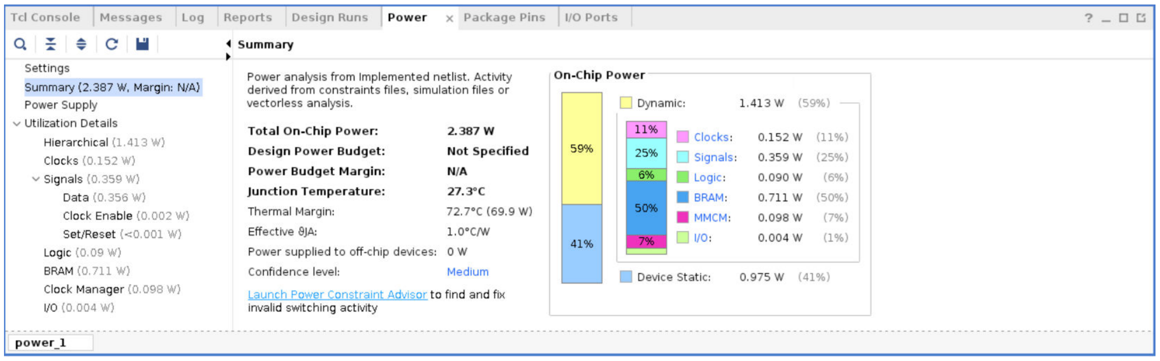
1. In the Flow Navigator, select **Open Implemented Design → Report Power** to open the Report Power dialog box.

   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.



Settings
Summary (2.387 W, Margin: N/A)
Power Supply
Utilization Details
- Hierarchical (1.413 W)
- Clocks (0.152 W)
- Signals (0.359 W)
  - Data (0.356 W)
  - Clock Enable (0.002 W)
  - Set/Reset (<0.001 W)
- Logic (0.09 W)
- BRAM (0.711 W)
- Clock Manager (0.098 W)
- I/O (0.004 W)

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**2.387 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**27.3°C**
Thermal Margin:	72.7°C (69.9 W)
Effective ϑJA:	1.0°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Medium

Launch Power Constraint Advisor to find and fix invalid switching activity

On-Chip Power

Dynamic:	1.413 W	(59%)
Clocks:	0.152 W	(11%)
Signals:	0.359 W	(25%)
Logic:	0.090 W	(6%)
BRAM:	0.711 W	(50%)
MMCM:	0.098 W	(7%)
I/O:	0.004 W	(1%)
Device Static:	0.975 W	(41%)

Vectorless analysis is done 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. We use this SAIF file to further guide the power analysis algorithm.

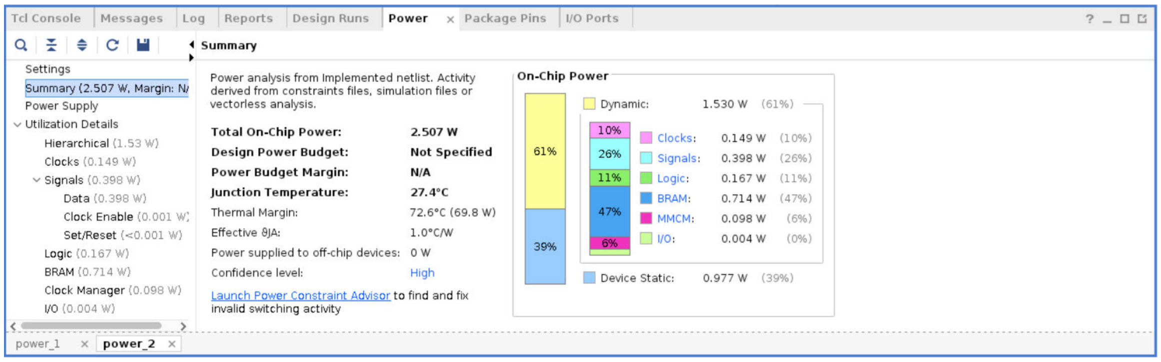
1. From the main menu, select **Reports → Report Power**.
2. In the Report Power dialog box, specify the SAIF file location in the Switching tab.

   The SAIF file, which was requested in the simulation settings prior to running timing simulation, should appear 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 you that the generated SAIF file is correct and matched with all design nets.



Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**2.507 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**27.4°C**
Thermal Margin:	72.6°C (69.8 W)
Effective θJA:	1.0°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	High

Launch Power Constraint Advisor to find and fix invalid switching activity

On-Chip Power

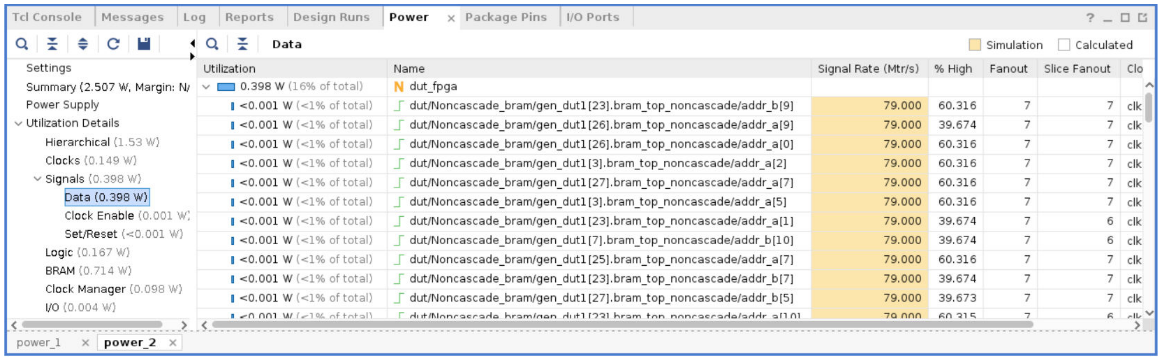
Dynamic:	1.530 W	(61%)
Clocks:	0.149 W	(10%)
Signals:	0.398 W	(26%)
Logic:	0.167 W	(11%)
BRAM:	0.714 W	(47%)
MMCM:	0.098 W	(6%)
I/O:	0.004 W	(0%)
Device Static:	0.977 W	(39%)

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 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.

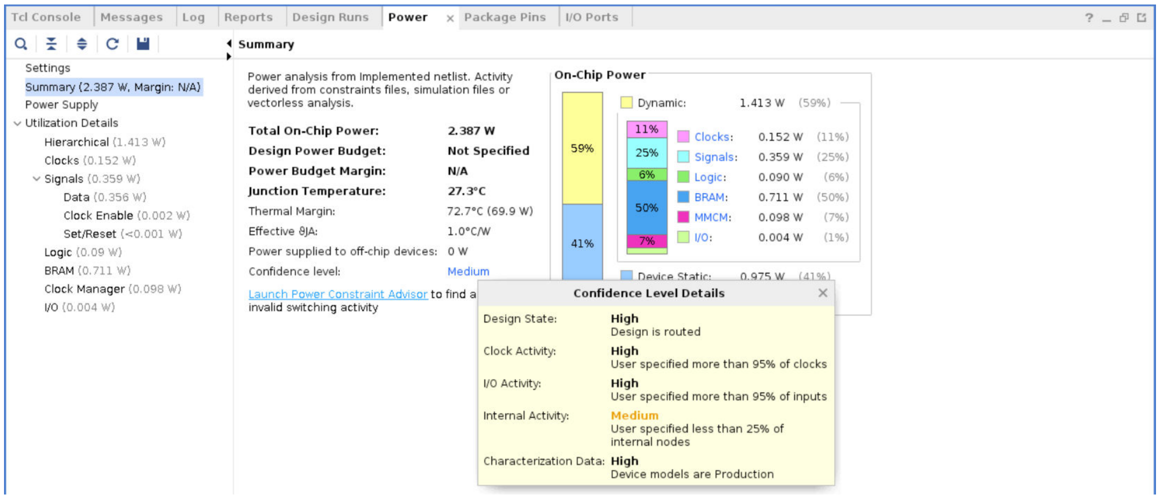


Utilization	Name	Signal Rate (Mtr/s)	% High	Fanout	Slice Fanout	Clo
0.398 W (16% of total)	dut_fpga					
<0.001 W (<1% of total)	dut/Noncascade_bram/gen_dut1[23].bram_top_noncascade/addr_b[9]	79.000	60.316	7	7	clk
<0.001 W (<1% of total)	dut/Noncascade_bram/gen_dut1[26].bram_top_noncascade/addr_a[9]	79.000	39.674	7	7	clk
<0.001 W (<1% of total)	dut/Noncascade_bram/gen_dut1[26].bram_top_noncascade/addr_a[0]	79.000	60.316	7	7	clk
<0.001 W (<1% of total)	dut/Noncascade_bram/gen_dut1[3].bram_top_noncascade/addr_a[2]	79.000	60.316	7	7	clk
<0.001 W (<1% of total)	dut/Noncascade_bram/gen_dut1[27].bram_top_noncascade/addr_a[7]	79.000	60.316	7	7	clk
<0.001 W (<1% of total)	dut/Noncascade_bram/gen_dut1[3].bram_top_noncascade/addr_a[5]	79.000	60.316	7	7	clk
<0.001 W (<1% of total)	dut/Noncascade_bram/gen_dut1[23].bram_top_noncascade/addr_a[1]	79.000	39.674	7	6	clk
<0.001 W (<1% of total)	dut/Noncascade_bram/gen_dut1[7].bram_top_noncascade/addr_b[10]	79.000	39.674	7	6	clk
<0.001 W (<1% of total)	dut/Noncascade_bram/gen_dut1[25].bram_top_noncascade/addr_a[7]	79.000	60.316	7	7	clk
<0.001 W (<1% of total)	dut/Noncascade_bram/gen_dut1[23].bram_top_noncascade/addr_b[7]	79.000	39.674	7	7	clk
<0.001 W (<1% of total)	dut/Noncascade_bram/gen_dut1[27].bram_top_noncascade/addr_b[5]	79.000	39.673	7	7	clk
<0.001 W (<1% of total)	dut/Noncascade_bram/gen_dut1[23].bram_top_noncascade/addr_a[10]	79.000	60.315	7	6	clk

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.

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



Settings
Summary (2.387 W, Margin: N/A)
Power Supply
Utilization Details
Hierarchical (1.413 W)
Clocks (0.152 W)
Signals (0.359 W)
Data (0.356 W)
Clock Enable (0.002 W)
Set/Reset (<0.001 W)
Logic (0.09 W)
BRAM (0.711 W)
Clock Manager (0.098 W)
I/O (0.004 W)

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**2.387 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**27.3°C**
Thermal Margin:	72.7°C (69.9 W)
Effective θJA:	1.0°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Medium

Launch Power Constraint Advisor to find a invalid switching activity

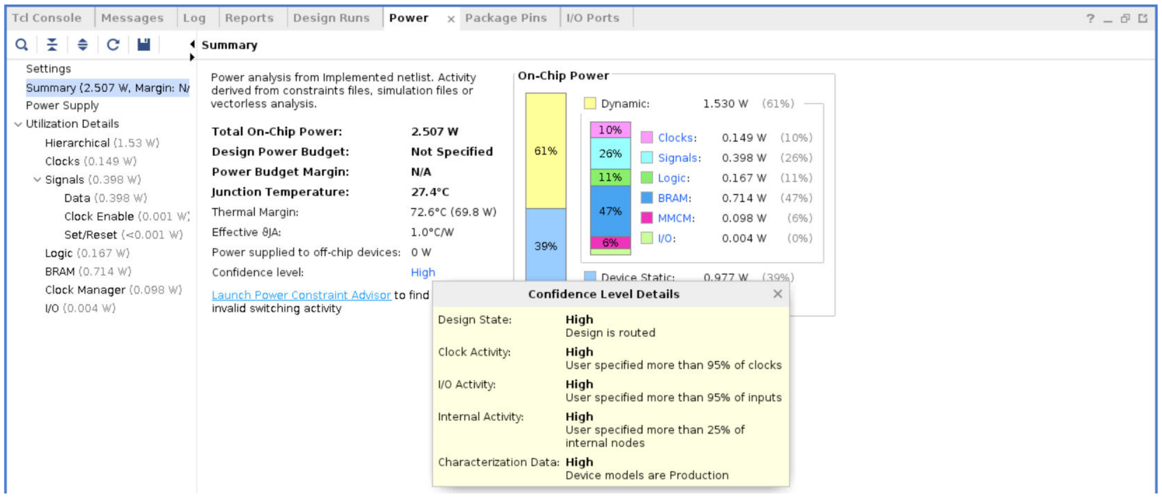
Dynamic:	1.413 W	(59%)
Clocks:	0.152 W	(11%)
Signals:	0.359 W	(25%)
Logic:	0.090 W	(6%)
BRAM:	0.711 W	(50%)
MMCM:	0.098 W	(7%)
I/O:	0.004 W	(1%)
Device Static:	0.975 W	(41%)

Confidence Level Details

Design State:	**High** Design is routed
Clock Activity:	**High** User specified more than 95% of clocks
I/O Activity:	**High** User specified more than 95% of inputs
Internal Activity:	**Medium** User specified less than 25% of internal nodes
Characterization Data:	**High** Device models are Production

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).

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



Settings
Summary (2.507 W, Margin: N/
Power Supply
Utilization Details
Hierarchical (1.53 W)
Clocks (0.149 W)
Signals (0.398 W)
Data (0.398 W)
Clock Enable (0.001 W)
Set/Reset (<0.001 W)
Logic (0.167 W)
BRAM (0.714 W)
Clock Manager (0.098 W)
I/O (0.004 W)

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**2.507 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**27.4°C**
Thermal Margin:	72.6°C (69.8 W)
Effective θJA:	1.0°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	High

Launch Power Constraint Advisor to find invalid switching activity

Dynamic:	1.530 W	(61%)
Clocks:	0.149 W	(10%)
Signals:	0.398 W	(26%)
Logic:	0.167 W	(11%)
BRAM:	0.714 W	(47%)
MMCM:	0.098 W	(6%)
I/O:	0.004 W	(0%)
Device Static:	0.977 W	(39%)

Confidence Level Details

Design State:	**High** Design is routed
Clock Activity:	**High** User specified more than 95% of clocks
I/O Activity:	**High** User specified more than 95% of inputs
Internal Activity:	**High** User specified more than 25% of internal nodes
Characterization Data:	**High** Device models are Production

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 SAIF data using the Questa Advanced Simulator.

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

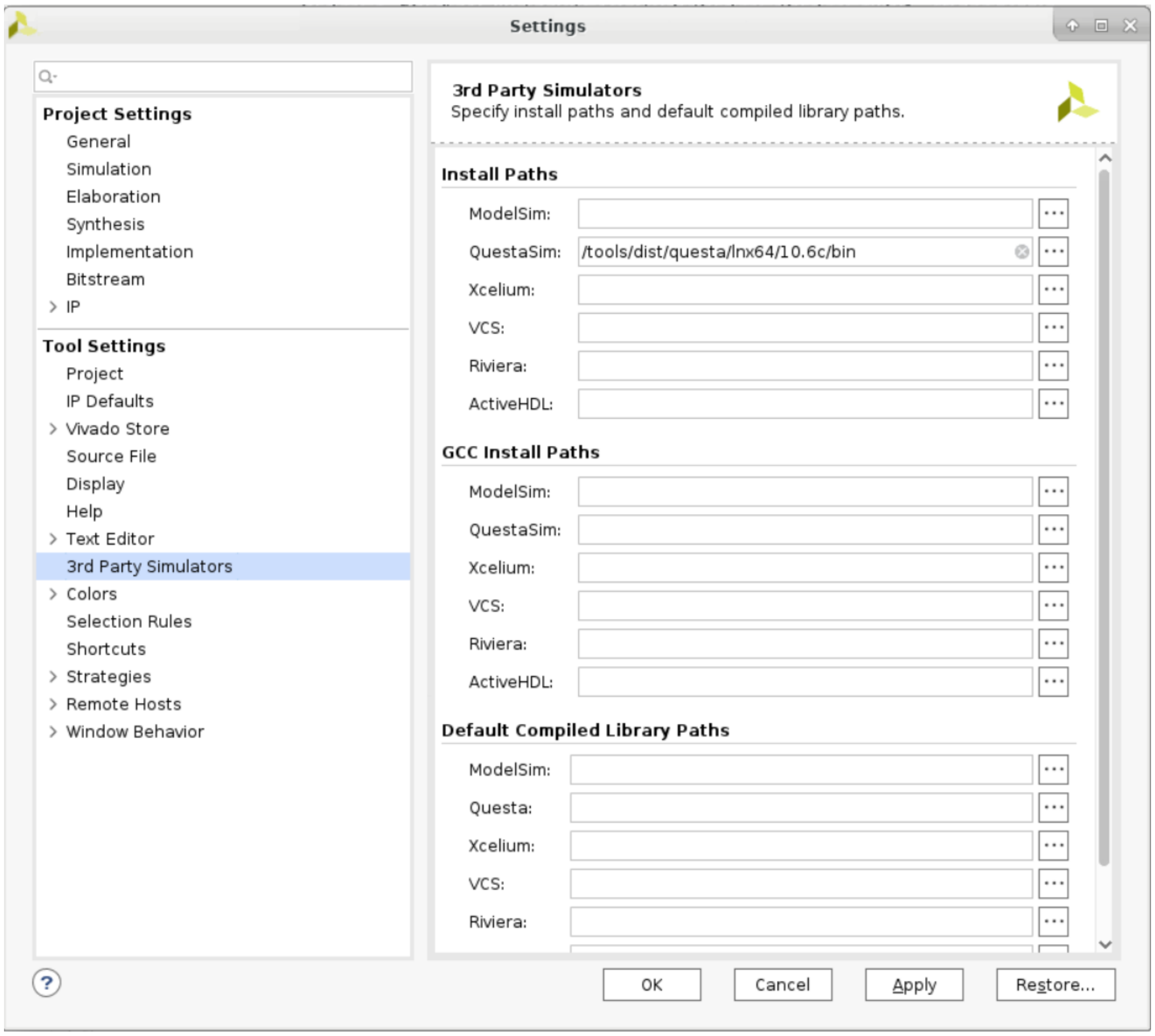
Manually set the path to ModelSim/Questa Advanced Simulator using the `$PATH` environment variable *or*

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



3rd Party Simulators
Specify install paths and default compiled library paths.

Install Paths

ModelSim:	
QuestaSim:	/tools/dist/questa/lnx64/10.6c/bin
Xcelium:	
VCS:	
Riviera:	
ActiveHDL:	

GCC Install Paths

ModelSim:	
QuestaSim:	
Xcelium:	
VCS:	
Riviera:	
ActiveHDL:	

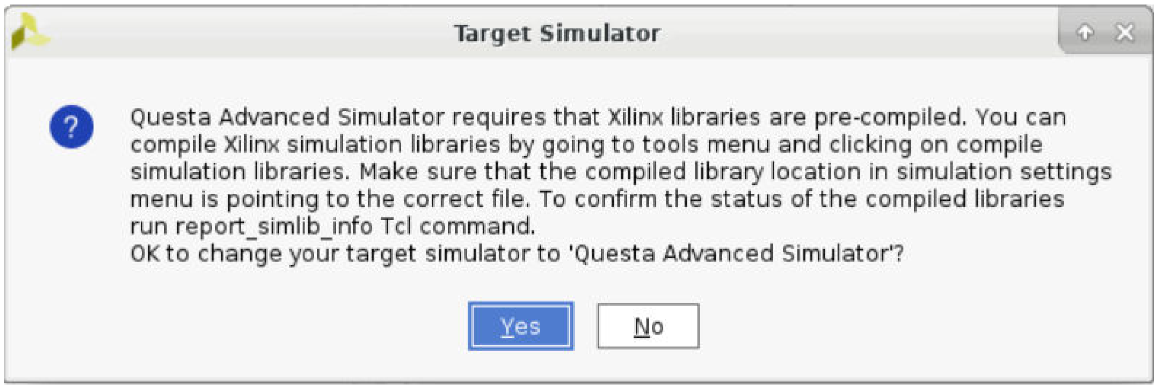
Default Compiled Library Paths

ModelSim:	
Questa:	
Xcelium:	
VCS:	
Riviera:	

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

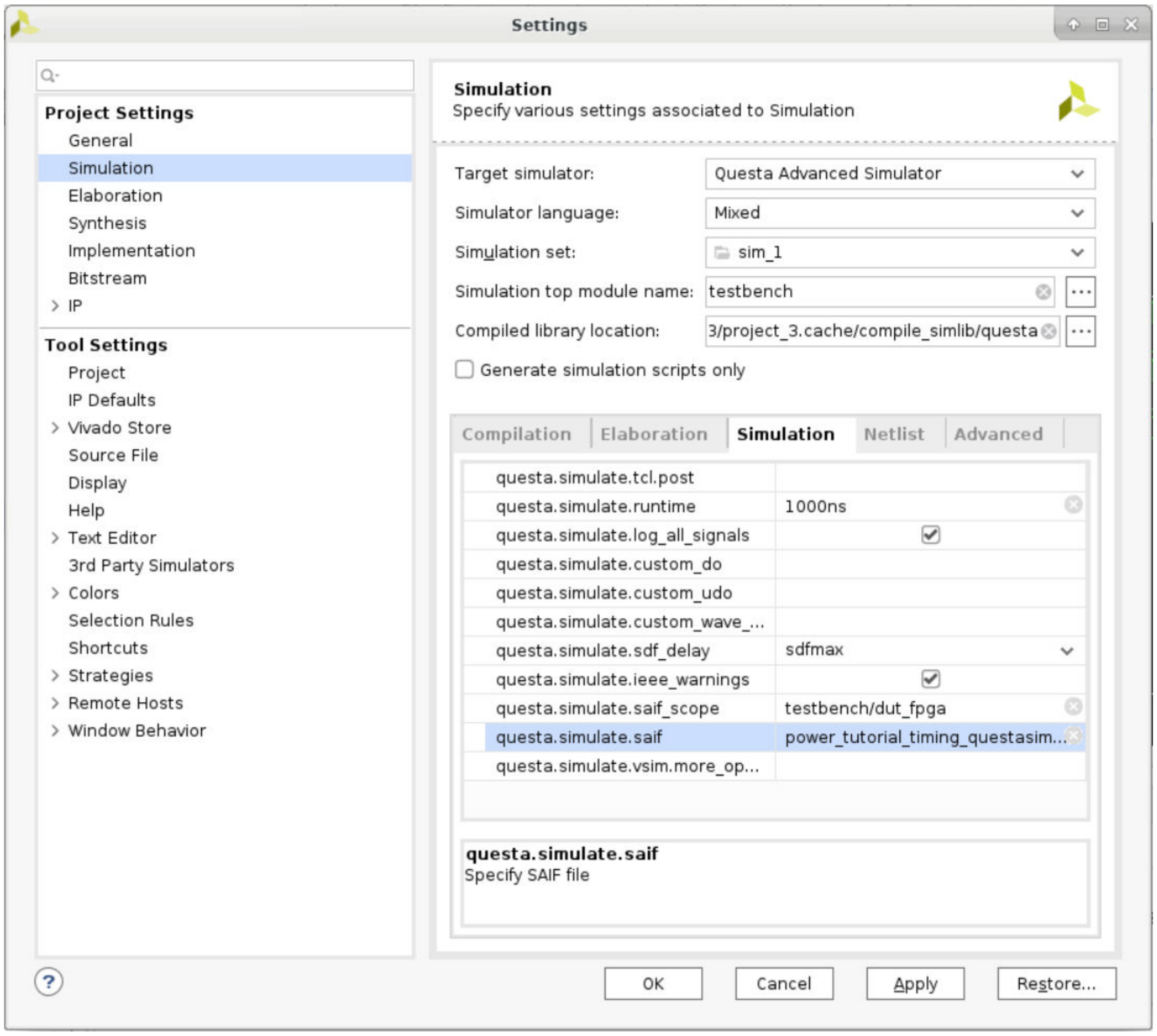
Now you are ready to set up and launch the Questa Advanced Simulator to run post-implementation timing simulation. You will set the timing simulation properties in the Vivado IDE, and run the timing simulation

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.


Target Simulator

Questa Advanced Simulator requires that Xilinx libraries are pre-compiled. You can compile Xilinx simulation libraries by going to tools menu and clicking on compile simulation libraries. Make sure that the compiled library location in simulation settings menu is pointing to the correct file. To confirm the status of the compiled libraries run report_simlib_info Tcl command.
OK 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. Note that the questa.simulate.runtime is 1000ns.



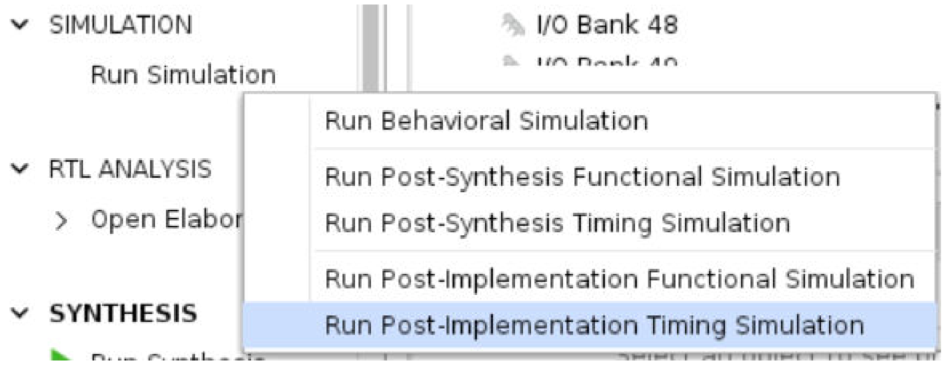
Target simulator:	Questa Advanced Simulator
Simulator language:	Mixed
Simulation set:	sim_1
Simulation top module name:	testbench
Compiled library location:	3/project_3.cache/compile_simlib/questa

☐ Generate simulation scripts only

questa.simulate.tcl.post	
questa.simulate.runtime	1000ns
questa.simulate.log_all_signals	✓
questa.simulate.custom_do	
questa.simulate.custom_udo	
questa.simulate.custom_wave_...	
questa.simulate.sdf_delay	sdfmax
questa.simulate.ieee_warnings	✓
questa.simulate.saif_scope	testbench/dut_fpga
questa.simulate.saif	power_tutorial_timing_questasim...
questa.simulate.vsim.more_op...	

questa.simulate.saif
Specify SAIF file

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**.


SIMULATION
Run Simulation
RTL ANALYSIS
Open Elabor
SYNTHESIS
I/O Bank 48
Run Behavioral Simulation
Run Post-Synthesis Functional Simulation
Run Post-Synthesis Timing Simulation
Run Post-Implementation Functional Simulation
Run Post-Implementation Timing Simulation

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

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
    ```

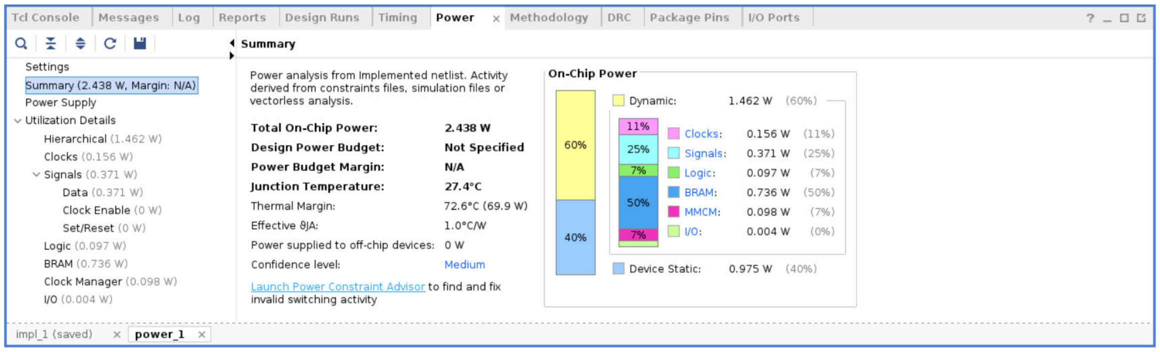
Step 2: Running Report Power in Vectorless Mode

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 that the total power for vectorless analysis runs with default switching rates.



Settings
Summary (2.438 W, Margin: N/A)
Power Supply
Utilization Details
Hierarchical (1.462 W)
Clocks (0.156 W)
Signals (0.371 W)
Data (0.371 W)
Clock Enable (0 W)
Set/Reset (0 W)
Logic (0.097 W)
BRAM (0.736 W)
Clock Manager (0.098 W)
I/O (0.004 W)

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**2.438 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**27.4°C**
Thermal Margin:	72.6°C (69.9 W)
Effective ϑJA:	1.0°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Medium

Launch Power Constraint Advisor to find and fix invalid switching activity

On-Chip Power

🟨	Dynamic:	1.462 W	(60%)
🟪	Clocks:	0.156 W	(11%)
🟦	Signals:	0.371 W	(25%)
🟩	Logic:	0.097 W	(7%)
🟦	BRAM:	0.736 W	(50%)
🟥	MMCM:	0.098 W	(7%)
🟩	I/O:	0.004 W	(0%)
🟦	Device Static:	0.975 W	(40%)

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.

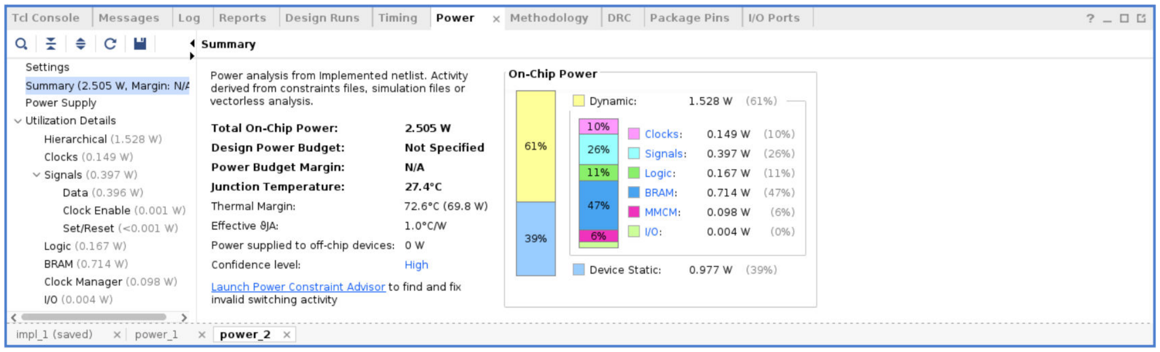
1. In the main menu bar, select **Reports → Report Power**.
2. In the Report Power dialog box, specify the SAIF file location in the Switching tab.

   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.



Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**2.505 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**27.4°C**
Thermal Margin:	72.6°C (69.8 W)
Effective θJA:	1.0°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	High

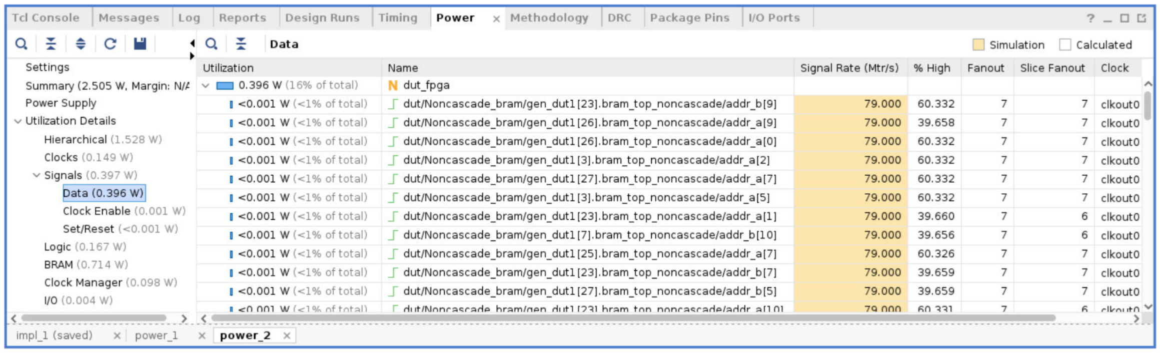
Launch Power Constraint Advisor to find and fix invalid switching activity

On-Chip Power

Dynamic:	1.528 W	(61%)	61%
Clocks:	0.149 W	(10%)	10%
Signals:	0.397 W	(26%)	26%
Logic:	0.167 W	(11%)	11%
BRAM:	0.714 W	(47%)	47%
MMCM:	0.098 W	(6%)	6%
I/O:	0.004 W	(0%)	
Device Static:	0.977 W	(39%)	39%

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.



- Settings
- Summary (2.505 W, Margin: N/A
- Power Supply
- Utilization Details
  - Hierarchical (1.528 W)
  - Clocks (0.149 W)
  - Signals (0.397 W)
    - Data (0.396 W)
    - Clock Enable (0.001 W)
    - Set/Reset (<0.001 W)
  - Logic (0.167 W)
  - BRAM (0.714 W)
  - Clock Manager (0.098 W)
  - I/O (0.004 W)

Utilization	Name	Signal Rate (Mtr/s)	% High	Fanout	Slice Fanout	Clock
0.396 W (16% of total)	dut_fpga					
<0.001 W (<1% of total)	dut/Noncascade_bram/gen_dut1[23].bram_top_noncascade/addr_b[9]	79.000	60.332	7	7	clkout0
<0.001 W (<1% of total)	dut/Noncascade_bram/gen_dut1[26].bram_top_noncascade/addr_a[9]	79.000	39.658	7	7	clkout0
<0.001 W (<1% of total)	dut/Noncascade_bram/gen_dut1[26].bram_top_noncascade/addr_a[0]	79.000	60.332	7	7	clkout0
<0.001 W (<1% of total)	dut/Noncascade_bram/gen_dut1[3].bram_top_noncascade/addr_a[2]	79.000	60.332	7	7	clkout0
<0.001 W (<1% of total)	dut/Noncascade_bram/gen_dut1[27].bram_top_noncascade/addr_a[7]	79.000	60.332	7	7	clkout0
<0.001 W (<1% of total)	dut/Noncascade_bram/gen_dut1[3].bram_top_noncascade/addr_a[5]	79.000	60.332	7	7	clkout0
<0.001 W (<1% of total)	dut/Noncascade_bram/gen_dut1[23].bram_top_noncascade/addr_a[1]	79.000	39.660	7	6	clkout0
<0.001 W (<1% of total)	dut/Noncascade_bram/gen_dut1[7].bram_top_noncascade/addr_b[10]	79.000	39.656	7	6	clkout0
<0.001 W (<1% of total)	dut/Noncascade_bram/gen_dut1[25].bram_top_noncascade/addr_a[7]	79.000	60.326	7	7	clkout0
<0.001 W (<1% of total)	dut/Noncascade_bram/gen_dut1[23].bram_top_noncascade/addr_b[7]	79.000	39.659	7	7	clkout0
<0.001 W (<1% of total)	dut/Noncascade_bram/gen_dut1[27].bram_top_noncascade/addr_b[5]	79.000	39.659	7	7	clkout0
<0.001 W (<1% of total)	dut/Noncascade_bram/gen_dut1[23].bram_top_noncascade/addr_a[10]	79.000	60.331	7	6	clkout0

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 Vivado® for UltraScale+™ 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.

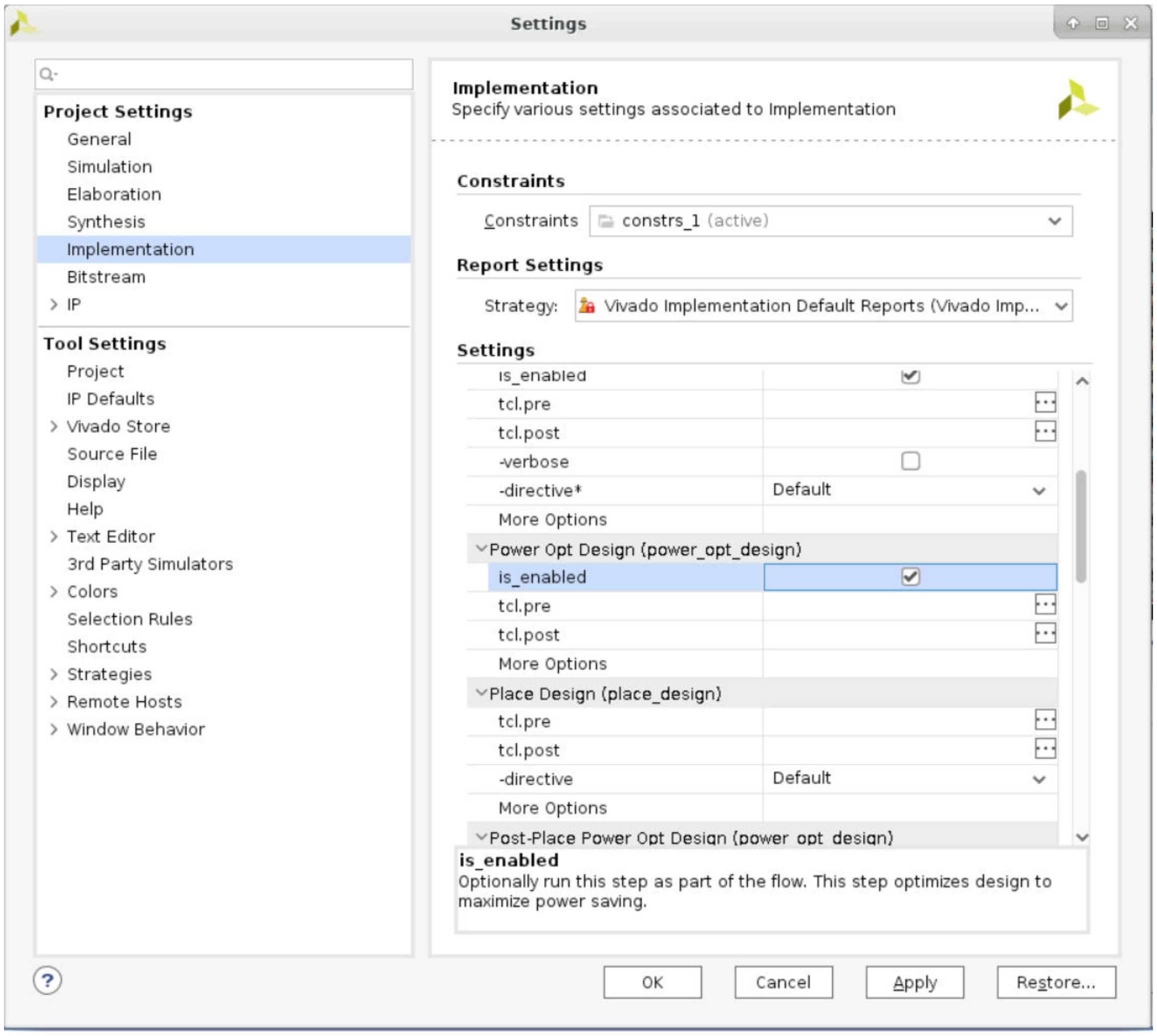
TIP: *When you run Implementation on your design, the Vivado tools may perform block RAM power optimizations by default during `opt_design`. These optimizations do not affect performance, and have little impact on area and compile time. In the previous Lab, the default block RAM power optimization was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



Implementation
Specify various settings associated to Implementation

Constraints
Constraints: constrs_1 (active)

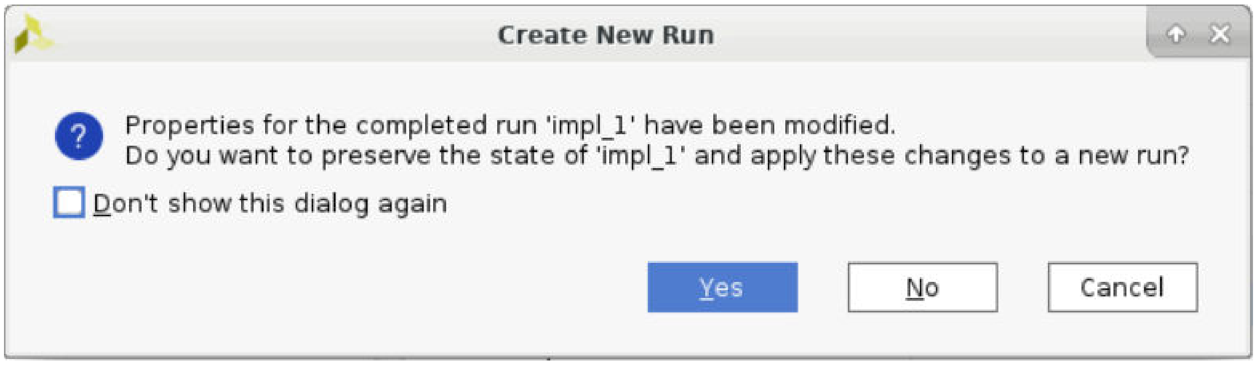
Report Settings
Strategy: Vivado Implementation Default Reports (Vivado Imp...

Settings

is_enabled	✓
tcl.pre	
tcl.post	
-verbose	☐
-directive*	Default
More Options	
Power Opt Design (power_opt_design)	
is_enabled	✓
tcl.pre	
tcl.post	
More Options	
Place Design (place_design)	
tcl.pre	
tcl.post	
-directive	Default
More Options	
Post-Place Power Opt Design (power_opt_design)	

is_enabled
Optionally run this step as part of the flow. This step optimizes design to maximize power saving.

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?.

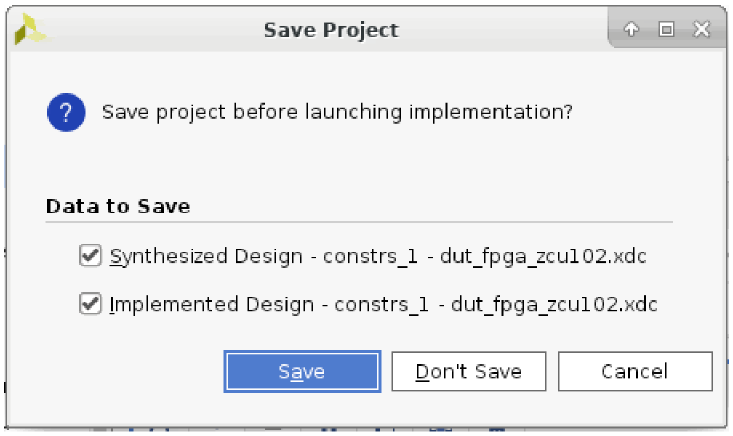


Properties for the completed run 'impl_1' have been modified.
Do you want to preserve the state of 'impl_1' and apply these changes to a new run?

☐ Don't show this dialog again

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.


Save Project

Save project before launching implementation?

Data to Save

☑ Synthesized Design - constrs_1 - dut_fpga_zcu102.xdc

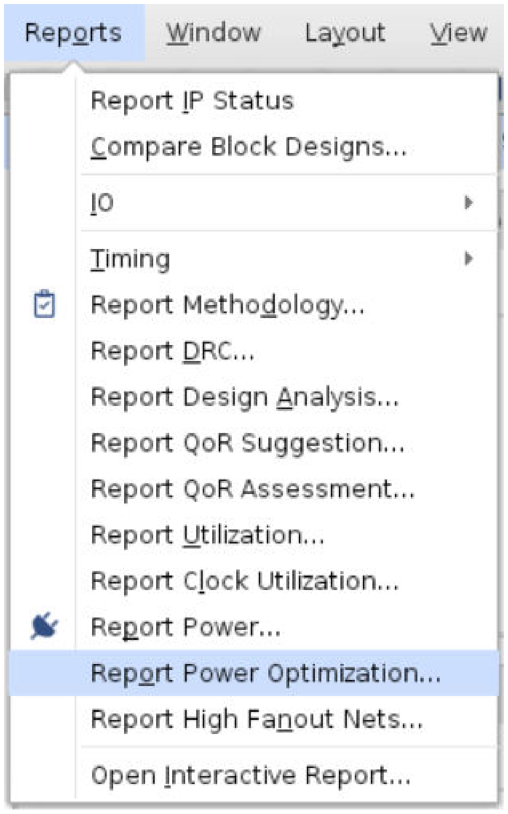
☑ Implemented Design - constrs_1 - dut_fpga_zcu102.xdc

You are running Implementation with Power Optimization turned on.

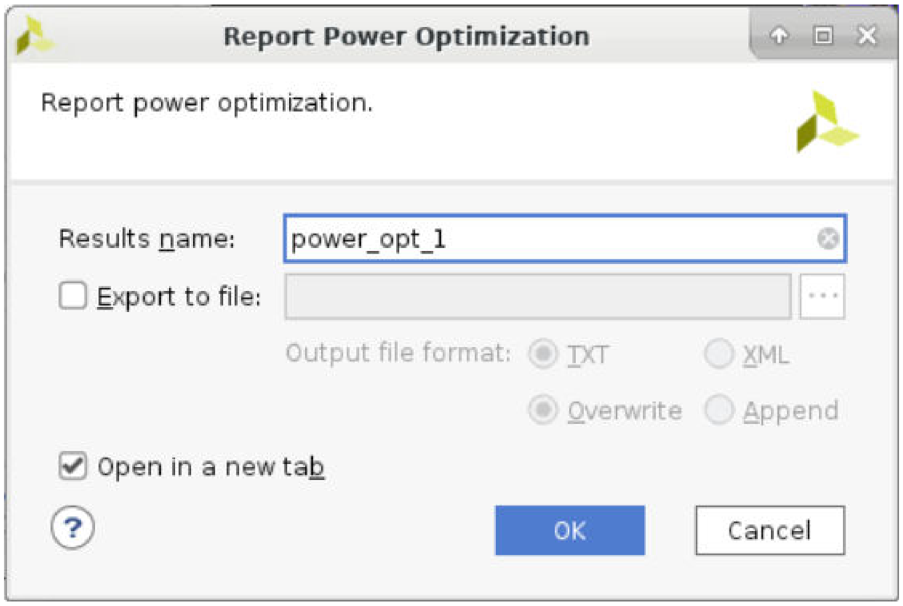
8. In the Implementation Completed dialog box, select **Open Implemented Design** and click **OK**. Click **Don't Save** when the Save Project window pops up to save both Synthesis and Implementation constraints.

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.



Report power optimization.

Results name: power_opt_1

[ ] Export to file:

[✓] Open in a new tab

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.

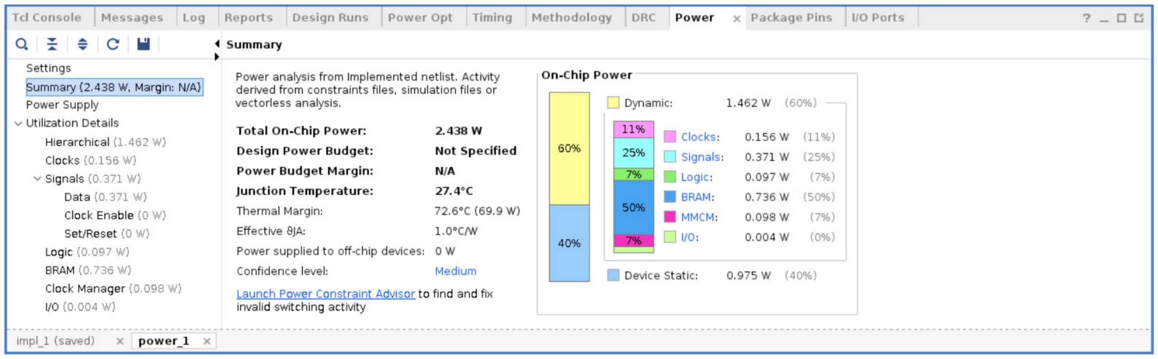
Step 3: Running report_power to Examine Power Savings

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.



Settings
Summary (2.438 W, Margin: N/A)
Power Supply
Utilization Details
Hierarchical (1.462 W)
Clocks (0.156 W)
Signals (0.371 W)
Data (0.371 W)
Clock Enable (0 W)
Set/Reset (0 W)
Logic (0.097 W)
BRAM (0.736 W)
Clock Manager (0.098 W)
I/O (0.004 W)

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**2.438 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**27.4°C**
Thermal Margin:	72.6°C (69.9 W)
Effective θJA:	1.0°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Medium

Launch Power Constraint Advisor to find and fix invalid switching activity

On-Chip Power

🟨	Dynamic:	1.462 W	(60%)
🟪	Clocks:	0.156 W	(11%)
🟦	Signals:	0.371 W	(25%)
🟩	Logic:	0.097 W	(7%)
🟦	BRAM:	0.736 W	(50%)
🟥	MMCM:	0.098 W	(7%)
🟩	I/O:	0.004 W	(0%)
🟦	Device Static:	0.975 W	(40%)

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:

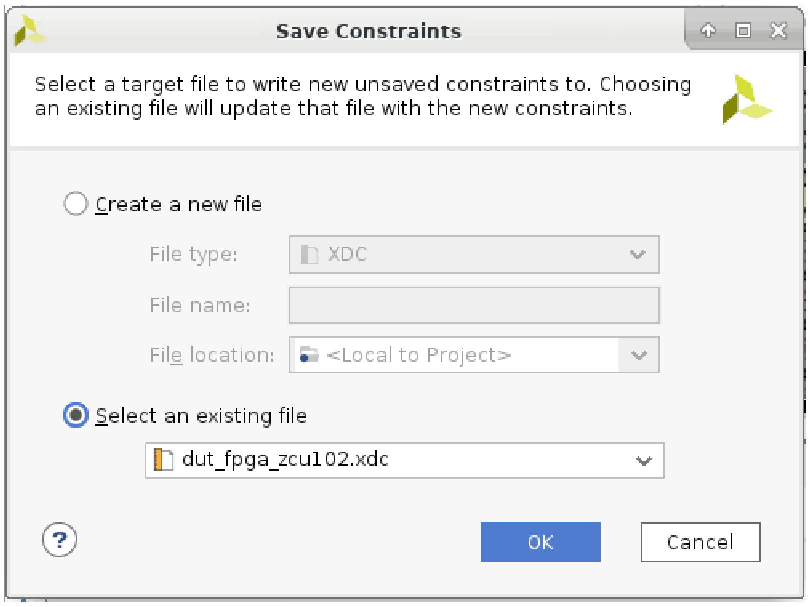
   ```
   set_power_opt -exclude_cells [get_cells dut/Cascaded_bram/
   gen_dut[0].bram_top_cascade/bram_cas/mem_reg_bram_0]
   ```

   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
```

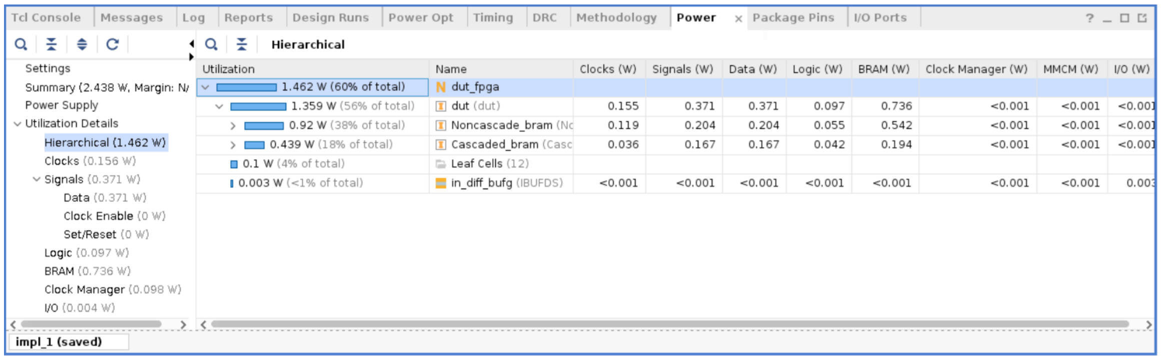
3. In the generated report power_opt_2, excluded block RAM will no longer be in the list of **Tool Gated BRAMs.**

***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 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. 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.



Settings
Summary (2.438 W, Margin: N/
Power Supply
Utilization Details
Hierarchical (1.462 W)
Clocks (0.156 W)
Signals (0.371 W)
Data (0.371 W)
Clock Enable (0 W)
Set/Reset (0 W)
Logic (0.097 W)
BRAM (0.736 W)
Clock Manager (0.098 W)
I/O (0.004 W)

Utilization	Name	Clocks (W)	Signals (W)	Data (W)	Logic (W)	BRAM (W)	Clock Manager (W)	MMCM (W)	I/O (W)
1.462 W (60% of total)	dut_fpga								
1.359 W (56% of total)	dut (dut)	0.155	0.371	0.371	0.097	0.736	<0.001	<0.001	<0.001
0.92 W (38% of total)	Noncascade_bram (Nc	0.119	0.204	0.204	0.055	0.542	<0.001	<0.001	<0.001
0.439 W (18% of total)	Cascaded_bram (Casc	0.036	0.167	0.167	0.042	0.194	<0.001	<0.001	<0.001
0.1 W (4% of total)	Leaf Cells (12)								
0.003 W (<1% of total)	in_diff_bufg (IBUFDS)	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	0.003

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

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)

Revision History

10/19/2022: Released with Vivado® Design Suite 2022.2 without changes from 2022.1.

Section	Revision Summary
06/15/2022 Version 2022.1	
General updates	Entire document

Please Read: Important Legal Notices

Copyright