

Vivado Design Suite Tutorial

Power Analysis and Optimization

UG997 (v2021.2) January 19, 2022





Revision History

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

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





Table of Contents

| Revision History2 |
|---|
| Lab 1: Power Analysis and Optimization Tutorial |
| Software Requirements |
| Hardware Requirements |
| Locating Tutorial Design Files |
| Lab 2: Running Power Analysis in the Vivado Tools |
| Introduction9 |
| Step 1: Creating a New Project9 |
| Step 2: Synthesizing the Design14 |
| Step 3: Setting Up the Report Power15 |
| Step 4: Running Report Power 19 |
| Step 5: Viewing the Power Properties 20 |
| Step 6: Editing Power Properties and Refining the Power Analysis |
| Step 7: Running Functional Simulation with SAIF Output23 |
| Step 8: Incorporating SAIF Data into Power Analysis |
| Step 9: Implementing the Design |
| Conclusion |
| Lab 3: Running Timing Simulation and Estimating Power |
| Introduction |
| Step 1: Configuring and Running the Timing Simulation using Vivado Simulator31 |
| Step 2: Running Report Power in Vectorless Mode |
| Step 3: Running Report Power with Vivado Simulator SAIF Data |
| Generating a SAIF File using Questa Advanced Simulator |
| Step 1: Configuring and Running Timing Simulation in Questa Advanced Simulator 37 |
| Step 2: Running Report Power in Vectorless Mode |
| Step 3: Running Report Power with Questa Advanced Simulator SAIF Data40 |
| Conclusion41 |



Lab 4: Measuring Hardware Power Using the KC705 Evaluation

| Board | 43 |
|--|----|
| Introduction | |
| Step 1: Generating a Bit File from the Implemented Design (Non-Power | |
| Optimization) | 43 |
| Step 2: Setting Up the KC705 Evaluation Board | 44 |
| Step 3: Setting Up the Fusion Digital Power Designer Software | 45 |
| Step 4: Programming the Bitstream | 46 |
| Step 5: Measuring the Hardware Power Rails | |
| Step 6: Estimating Vectorless Power with Junction Temperature | 51 |
| Conclusion | 53 |

Lab 5: Measuring Hardware Power Using the KCU105 Evaluation

| Board | 54 |
|---|----|
| Introduction | 54 |
| Step 1: Generating a Bit File from the Implemented Design | 54 |
| Step 2: Setting up the KCU105 Evaluation Board | 55 |
| Step 3: Configuring the Maxim Digital Power Tool Software | |
| Step 4: Programming the Bitstream | 57 |
| Step 5: Measuring the Hardware Power Rails | 61 |
| Step 6: Estimating the Vectorless Power with Junction Temperature | 63 |
| Conclusion | 65 |
| | |
| Lab 6: Performing Power Optimization | |
| Introduction | 66 |
| Step 1: Setting Up Options to Run Power Optimization | 66 |
| Step 2: Running report_power_opt to Examine User/Design Specific Power | |
| Optimizations | 68 |
| Step 3: Running report_power to Examine Power Savings | 70 |
| Step 4: Turning Off Optimizations on Specific Signals and Rerunning the | |
| Implementation | 71 |
| Step 5: Running report_power_opt to Examine Tool Optimizations Again | 72 |
| Step 6: Saving Power using UltraScale Block RAM in Cascaded Mode | 73 |
| Conclusion | 74 |
| | |
| Appendix A: Additional Resources and Legal Notices | |
| Xilinx Resources | |
| Documentation Navigator and Design Hubs | 76 |



| References76 | |
|--|--|
| Please Read: Important Legal Notices77 | |





I ab 1

Power Analysis and Optimization Tutorial

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



VIDEO: The Vivado Design Suite Quick Take Video: Power Estimation and Analysis Using Vivado shows bow 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 0 that affect power consumption in an FPGA, shows how the Vivado Design Suite helps to minimize power consumption in your design, and looks at some advanced control and best practices for getting the most out of Vivado power optimization.

Software Requirements

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

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

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

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



Hardware Requirements

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

Hardware Requirements for 7 Series Devices

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

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

Hardware Requirements for UltraScale Devices

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

Locating Tutorial Design Files

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

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

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

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

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



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

7 Series Tutorial Design Files

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

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

| Directories/Files | Description |
|-------------------------------|--|
| /src | Contains the design HDL and testbench for the functional simulation. |
| /src/dut_fpga.v | Top module for the design. |
| /src/bram_tdp.v | Other design blocks - synthesized module. |
| /src/bram_top.v /src/dut.v | |
| dut_fpga_kc705.xdc | Contains clocking and timing constraints for the design. |
| /src/testbench.v | Testbench for simulating the design. |

UltraScale Device Tutorial Design Files

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

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

| Directories/Files | Description |
|----------------------------|---|
| /src | Contains the design HDL and testbench for the simulation. |
| /src/dut_fpga.v | Top module for the design. |
| /src/dut.v | Other design blocks. |
| /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 | |
| dut_fpga_kcu105.xdc | Contains clocking and timing constraints for the design. |
| /src/testbench.v | Testbench for simulating the design. |

Table 1: Example table



Lab 2

Running Power Analysis in the Vivado Tools

Introduction

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

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

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

Step 1: Creating a New Project

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

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



On Linux, do the following.

- 1. Go to the directory where the lab materials are stored:
 - cd <Extract_Dir>/7_series (for 7 series devices) or
 - cd <Extract_Dir>/UltraScale (for UltraScale devices)
- 2. Launch the Vivado IDE: vivado

| A | Vivado 2021.1 | + _ 3 × |
|---|--|-----------|
| Eile Flow Tools Window Help Q- Quick Access | | |
| | | £ XILINX. |
| Quick Start Create Project > Open Project > Open Example Project > | Recent Projects Project_1 Project_1 Promovular/Desktopsproject_1 | |
| Tasks Manage IP > Open Hardware Manager > Vivado Store > | | |
| Learning Center | · | |
| Tcl Console | | ? _ 🗆 🖒 X |
| | |)° |

On Windows, do the following.

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

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

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

- 4. In the Getting Started page, click **Create New Project** to start the New Project wizard.
- 5. Click **Next** to continue to the next screen.



| A-New Project | | |
|-------------------------------------|---|--------|
| Project Name Enter a name for yo | your project and specify a directory where the project data files will be stored. | A |
| Project name: | power_tutorial1 | 0 |
| Project location: | C:Mvado_power_tutorial | 0 |
| Create project | ect subdirectory | |
| Project will be cre | created at: C:/vivado_power_tutorial/power_tutorial1 | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| ۲ | < Back Next > Einish | Cancel |

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



| +. | [::] | + + | | | | | |
|----|------------------|---|----------------|------------------------|----|---------------------------------------|--|
| | Index | Name | Library | HDL Source For | | Location | |
| 0 | 1 | bram_tdp.v | xil_defaultlib | Synthesis & Simulation | ٠ | C:/vivado_power_tutorial/7_series/src | |
| | 2 | bram_top.v | xil_defaultlib | Synthesis & Simulation | | C:/vivado_power_tutorial/7_series/src | |
| 0 | 3 | dut.v | xil_defaultlib | Synthesis & Simulation | * | C:Nivado_power_tutorial/7_series/src | |
| | 4 | dut_fpga.v | xil_defaultlib | Synthesis & Simulation | ٠ | C:/vivado_power_tutorial/7_series/src | |
| | 5 | testbench.v | xil_defaultlib | Simulation only | ٠ | C:/vivado_power_tutorial/7_series/src | |
| | opy <u>s</u> our | add RTL includ ces into project es from subdiri | | | IS | <u>C</u> reate File | |

- f. Verify that the files are added and **Copy sources into project** is checked. Click **Next**.
- 9. In the Add Constraints (optional) page, click Add Files and select dut_fpga_kc705.xdc in the file browser. In the directory structure, you will find the dut_fpga_kc705.xdc file below the /src folder.

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

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

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

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

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



| power_tutorial [C/vivedo_power_ | tatoriel/power_tatoriell/power_tatoriell.opr] - Viverlo 2017.3 | | |
|---|---|---|---|
| Tas Eq. Lion Tools R | Bridow Lagard Daw galp Qr Deack Access | | Ready |
| ■, + + = = = × | F Φ Σ K Ø Ø | | E Default Lavaul |
| Now Novagelor 5 0 7 - | PROJECT INARACER - power_judaral1 | | ? |
| · PROJECT NANAGER | Sources 2 - D IS X | Project Summary | 200) |
| O Swittings Add Bources | Q ± + 0 04 0 | Settings Eff | |
| Canguage Temptakes P IP Catalog IP Catalog IP INTEGRATOR Catalog Block Design Open Block Design | Consequence (1) Set dat, provide (1) Set dat, provide (1) Set dat, provide (1) Set data for (1) | Project name power Jancels 1 Project location Childrado, power Jancels 1 Project location Childrado, power Jancels 1 Project and Internet Internet 7/2015 Graudeon Platform (schild)(14)(14)(14)(14)(14)(14)(14)(14)(14)(14 | |
| Generale Stock Design | Properties P = 0 11 × | Board Part | |
| Run Struckson | er e | Display manie Kalifiko-7 XXC756 Evaluation Platform Ricard part name allino.com kc750 partiel 15 Constellars Resolutions parties URL www.ditin.com/sc700 Ricard sensities Kontec 7 XC756 Scaladion Platform | |
| > Open Surthasked Dasign | TelConsole Mexicages Log Reports Design Russ. × | | ? _ 0 (|
| MPLEVENTATION Run Implementation Open implementation | Q | THS TPAS TotalPower FaredReads LUT PF DRABE LIKAR DDP Stat Elegand | Shaagy Visado Sertheans Débaute (V Visado Implementation Debu |
| PROCRAM AND DEBUG | | | |
| > Open Hardevare Vanager | 8 | | |

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**.

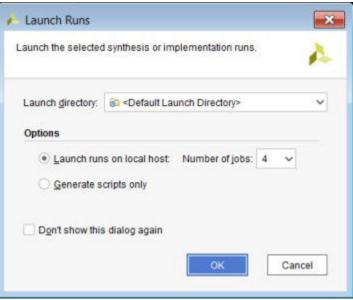
| 2- | Project | | 2 |
|---|---|------|---------------------------------------|
| Project Settings General | Specify various settings related to project | ts. | · · · · · · · · · · · · · · · · · · · |
| Simulation | Default Project Directory | | |
| Elaboration Synthesis Implementation Bitstream | Start in directory (C:/Users/venkal) | as) | |
| Fool Settings | Specify project directory: C:/pro | | |
| Project | Target Language | | |
| IP Defaults Source File Display WebTalk | Verilog VHDL | | |
| Help | Reopen last project on startup | | |
| Text Editor 3rd Party Simulators | Number of recent projects to list: | 10 🗘 | |
| > Colors | Number of recent directories to list | 15 🗘 | |
| Selection Rules Shortcuts Strategies > Window Behavior | Number of recent files to list | 10 ‡ | |
| _ | | | |

Now, the design is ready for synthesis.

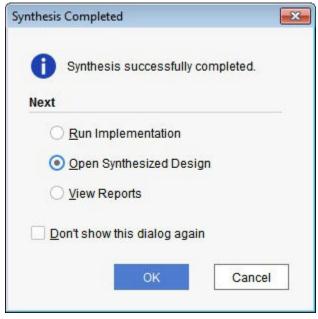


Step 2: Synthesizing the Design

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



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



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



Step 3: Setting Up the Report Power

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

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

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

| Res <u>u</u> lts name: | power_1 | | | ¢ |
|------------------------|-----------------|--------------------------|----------------|---|
| <u>Environment</u> | Power Supply | Switching Output | | |
| Device Setting | s | | | ^ |
| Temp grad | le: | extended 🗸 | | |
| Pro <u>c</u> ess: | | maximum 🗸 | | |
| Environment S | ettings | | | |
| Output Loa | d: | 0 🗘 | pF [0 - 10000] | |
| Unctio | n temperature: | 25.679 | °C | |
| Ambient te | mperature: | 25 🗘 | °C | |
| Effectiv | e ƏJ <u>A</u> : | 1.42 | °C/W [0 - 100] | |
| A <u>i</u> rflow: | | 250 🗸 | LFM | |
| <u>H</u> eat sink: | | medium (Medium Prof 🗸 | | |
| ₽SA: | | 2.4 🌲 | °C/W [0 - 100] | |
| Board sele | ction: | medium (10"x10") 🗸 | | |
| Number of | board layers: | 12to15 (12 to 15 Layer 🗸 | | |
| JB: | | 2.5 🗘 | °C/W [0 - 100] | |
| < Boord tom | a a ratura : | ne 🔺 | 1 CE 4001 | ~ |
| Legend | | | | |

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



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

| tesylts name: | power_1 | | | | | c |
|------------------|--------------|------------|------------|--------|--|---|
| Environment | Power Supply | <u>5</u> 1 | witching | Output | | |
| Settings | | | | | | |
| Vccint: | 1.000 ‡ | v | (0.970 - 1 | .030] | | â |
| Vccaux | 1.800 \$ | v | [1 710 - 1 | MICCIC | | |
| Vcco <u>3</u> 3: | 3.300 \$ | v | [3.000 - 3 | .450] | | |
| Vcco25: | 2.500 🗘 | v | (2.380 - 2 | .630) | | |
| Vcco <u>1</u> 8: | 1.800 🌲 | v | [1.710 - 1 | .900] | | |
| Vcco1 <u>5</u> : | 1.500 ‡ | v | [1.430 - 1 | 580] | | |
| Vcco135: | 1.350 🗘 | v | [1.300 - 1 | .400] | | |
| Vcco12 | 1.200 🌻 | v | [1.140 - 1 | .260J | | |
| Vccaux_io: | 1.800 🗘 | v | [1.710 - 1 | [068. | | |
| Vccbram: | 1.000 ‡ | v | [0.970 - 1 | .030] | | |
| MGTAVcc: | 1.000 🌲 | ۷ | [0.950 - 1 | .050] | | |
| MGTAVII: | 1.200 🗘 | v | [1,140 - 1 | .260] | | U |
| MGTVccaux | 1.800 🗘 | V | [1.710 - 1 | .890] | | |
| MGTZVccl: | 1.075 ‡ | v | [1.050 - 1 | .100] | | |
| Legend | + | | | | | × |

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

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

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



| timate power c | onsumption based | on the netli | st design and | d part xc7k325t | tg900-2. | |
|---------------------------------|------------------------|--------------|---------------|-----------------|-----------|--|
| Resylts name: | power_1 | | | | | |
| Environment | Power Supply | Switching | Output | | | |
| Simulation Se | ttings | | | | | |
| Simulation | activity file (.saif): | 1 | | | | |
| Default Activit | y Settings | | | | | |
| Default tog | gle rate: | 12.5 | [0 - 100] | | | |
| Default Sta | tic Probability: | 0.5 | [0.0 - 1.0] | | | |
| Enable Rate S | ettings | | | | | |
| | | Static Proba | bility | 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 S | ettings | | | | | |
| | | Static Proba | bility | Toggle Rate | | |
| Primary Ou | itputs: | | [0.0 - 1.0] | | [0 - 100] | |
| Logic | | | | | | |
| Register | s: | | [0.0 - 1.0] | | [0 - 100] | |
| Shift Reg | gisters: | | [0.0 - 1.0] | | [0 - 100] | |
| Distribut | ed RAMs: | | [0.0 - 1.0] | | [0 - 100] | |
| LUTs: | | | [0.0 - 1.0] | | [0 - 100] | |
| DSPs: | | | [0.0 - 1.0] | | [0 - 100] | |
| Block R/ | Ws: | | [0.0 - 1.0] | | [0 - 100] | |
| GTs | | | | | | |
| RX Data | | | [0.0 - 1.0] | | [0 - 100] | |
| TX Data: | | | [0.0 - 1.0] | | [0 - 100] | |
| Constrained | Clocks | | | | | |
| Clock | | | Period | | | |
| sys_clk_in_p | | | 5 ns | | | |
| clk_0 | | | 5 ns | | | |
| clkout0 | | | 10 ns | | | |

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



| stimate power co | onsum | ption base | d on the netlist | design and part | xc7k325tffg90(| 0-2. |
|------------------------|---------|------------|------------------|-----------------|----------------|------|
| Res <u>u</u> lts name: | powe | er_1 | | | | 0 |
| Environment | Powe | er Supply | Switching | Output | | |
| Output text file | 9 | power_1 | pwr | | | |
| Output XPE file | e: | power_1 | хре | | | 0 |
| Output RP | X file: | C:/proj/pr | ower_1.rpx | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

Legends in Report Power Tool

The following legends appear consistently in the Report Power tool:

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



Step 4: Running Report Power

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

This runs the report_power command.

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

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

| Power | | 7 _ 0 2 3 |
|---|---|-----------|
| a ¥ ≑ c ∎" | Summary | |
| Setings Summary (1.379 W) Power Supply V Ulication Details Hierarchical (0.918 W) Clocks (0.011 W) State (0.051 W) Date (0.051 W) Clock Reate (0 W) SetReate (0 W) SetReate (0 W) Logic (0.011 W) ERVM (0.714 W) Clock Manager (0.117 W) KD (0.004 W) | Power estimation from Synthesized netlist Adduly derived form constraints files, simulation files or vectories analysis. Note: these early estimates can change after implementation. Total On Chip Power: 1.379 W Junction Temperature: 27.4 °C, Thermail Margin: 57.6 °C (30.3 W) Effective 2UA: 1.3 °C.W Power supplied to off-chip devices: 0.W Confidence level: 8654 um Launch Power Constraint Addiagr to find and fix invalid switching activity | |

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

| Power | | | | | | | 7 |
|---|---------------|-------------|-----------|-------------|------------|--|-----------------------|
| Q 😤 单 C 🕍 " | Power Supply | | | | | | 🔚 Detault 🔛 Calculate |
| Sellings | Supply Source | Vollage (V) | Total (A) | Dynamic (A) | Static (A) | | |
| Summary (5.378 W) | Vocint | 1.000 | 1.041 | 0.745 | 0.295 | | |
| Power Supply | Voceux | 1.800 | 0.152 | 0.067 | 0.085 | | |
| Utilization Details | Voco33 | 3.300 | 0.042 | 0.008 | 0.042 | | |
| Hierarchical (0.918 W) | Voco25 | 2.500 | 0.000 | 0.000 | 0.000 | | |
| Clocks (0.011 W) | Vcco18 | 1.800 | 0.000 | 0.000 | 0.000 | | |
| Signata (0.061 W) | Voco15 | 1.500 | 0.000 | 0.000 | 0.000 | | |
| Data (0.061 V/) | Veco135 | 1.350 | 0.000 | 0.000 | 0.005 | | |
| Clock Enable (0 W) | Voco 12 | 1.200 | 0.000 | 0.000 | 0.000 | | |
| SeoReset (0 W) | Vecaux_io | 1.800 | 0.000 | 0.000 | 0.000 | | |
| Logic (0.011 W) | Vocbram | 1.000 | 0.065 | 0.054 | 0.013 | | |
| BRAW (0.714 W) | MGTAVec | 1.000 | 0.000 | 0.000 | 0.000 | | |
| Clock Manager (0.117 W) | MGTAVI | 1.200 | 0.000 | 0.000 | 0.000 | | |
| VO (0.004 W) | MGTVccaux | 1.800 | 0.000 | 0.000 | 0.000 | | |
| | Vocado | 1.800 | 0.030 | 0.000 | 0.030 | | |

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

| a 🛛 e c 📕 " | Q 📱 Hierarchical | | | | | | | | | |
|--|---|-------------------|------------|-------------|----------|-----------|----------|-------------------|----------|--------|
| Setings | Utilization | Name | Clocks (W) | Signals (W) | Data (W) | Logic (W) | BRAM (W) | Clock Manager (W) | MICH (W) | NO (W) |
| Summary (1.370 W) | E 0.918 W (67% of total) | agat_tub 🎉 | | | | | | | | |
| Power Supply | Compared 0.795 W (58% of total) | (tut) tub 📳 | 0.009 | 0.061 | 0.061 | 0.011 | 0.714 | <0.001 | <0.001 | <0.001 |
| Utilization Details | > 🔲 0.079 W (6% of total) | 🗐 gen_dutj0] br | 0.001 | 0.005 | 0.006 | 0.001 | 0.071 | =0.001 | <0.001 | -0.001 |
| Hierarchical (0.918 W) | > 0.079 W (0% of total) | 🗿 gen_dut(1).br | 0.001 | 0.005 | 0.006 | 0.001 | 0.071 | <0.001 | <0.001 | -0.001 |
| Clocks (0.011 W) | > 0.079 W (0% of total) | gen_dut(2).br | 0.001 | 0.005 | 0.006 | 0.001 | 0.071 | <0.001 | <0.001 | <0.001 |
| ✓ Signals (0.061 (0) Data (0.081 W) | > 0.079 W (5% of total) | i gen_dut(3).br | 0.001 | 0.005 | 0.006 | 0.001 | 0.071 | +0.001 | <0.001 | -0.001 |
| | > 0.079 W (0% of total) | I gen_dut[4].br | 0.001 | 0.006 | 0.006 | 0.001 | 0.071 | <0.001 | <0.001 | +0.001 |
| Clock Enable (0.97) | > 0.079 W (6% of total) | al gen_dut(6).br | 0.001 | 0.006 | 0.006 | 0.001 | 0.071 | +0.001 | +0.001 | +0.001 |
| SebReset (0 W) | > 0.079 W (6% of total) | 📓 gen_dul(8) br. | 0.001 | 0.006 | 0.006 | 0.001 | 0.071 | *0.001 | +0.001 | -0.001 |
| Logic (0.041 W) | > 0.079 W (5% of local) | gen_dul(?).br | 0.001 | 0.006 | 0,006 | 0.001 | 0.071 | +0.001 | +0,001 | -0.001 |
| BRAW (0.714 W) Clock Manager (0.117 W) | > 0.078 W (8% of total) | 🧾 gen_dut(ii) br | 0.001 | 0.006 | 0.006 | 0.001 | 0.071 | *0.001 | +0.001 | -0.001 |
| | > @ 0.079 W (8% of total) | 📓 gen_dul(9) br | 0.001 | 0,006 | 0.006 | 0.001 | 0.071 | +0.001 | +0.001 | +0.001 |
| UC (0.004 W) | \$ <0.001 W (<1% of fotal) | Lesf Cells (8) | | | | | | | | |
| | 0.124 W (9% of total) | E Leaf Cells (18) | | | | | | | | |



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

| Q 😤 🗢 C 📕 🎽 | Q 🔮 Clocks | | | | | | | | Constru | int [| Calculated |
|--|----------------------------|---------------|-----------------|--------|-------------------------|---------------|------------|-------|-------------|-------|------------|
| Settings | Utilization | Name | Frequency (MHz) | Buffer | Clock Buffer Enable (%) | Enable Signal | Bel Fanout | Sites | FanoutiSite | Type | |
| Summary (1.379 W) | • 8 0.011 W (1% of total) | 🔉 dut_fpga | | | | | | | | | |
| Power Supply | > 10.009 W (1% of total) | J cikout0 | 100.000 | NIA | NA | N/A | 842 | 611 | 1.378 | NIA | |
| Utilization Details | > 10.002 W (<1% of lotal) | _f_dk_0 | 200.000 | NK | NiA | NA | 2 | 2 | 1.000 | NØ. | |
| Hierarchical (0.91839) | > 1<0.001 W (<1% of total) | J sys_dk_in_p | 200.000 | NIA | 1404 | MA | 2 | 2 | 1.000 | NiA | |
| Clocks (0.011 W) | | | | | | | | | | | |
| ✓ Signals (0.061 W) | | | | | | | | | | | |
| Data (0.061 W) | | | | | | | | | | | |
| Clock Enable (0.W) | | | | | | | | | | | |
| | | | | | | | | | | | |
| SetReset (0 W) | | | | | | | | | | | |
| Logic (0.011 W) | | | | | | | | | | | |
| | | | | | | | | | | | |
| Logic (0.011 W) | | | | | | | | | | | |
| Logic (0.011 W) BRAN (0.714 W) | | | | | | | | | | | |
| Logic (0.011 W) BRAM (0.714 W) Clock Manager (0.117 W) | | | | | | | | | | | |

Step 5: Viewing the Power Properties

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

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

| Xower | | | | | | | | | | | ? |
|-------------------------|-------------|------------------|---------------------|---------|--------|--------------|---------|------------|-----------|-----------|------------|
| Q 🔮 🗢 C 📕 🎽 | QI | SetReset | | | | | | | Attribute | Estimated | Calculated |
| Settings | Utilization | Name | Signal Rate (Mtn/s) | % High | Fanout | Slice Fanout | Clock | Logic Type | | | |
| Summary (1.370 W) | ~ DW | 🙀 dut_fpga | | | | | | | | | |
| Power Supply | 0 W 0 | _f_dut/dut_reset | 0.000 | 0.000 | 530 | 0 | clkout0 | FFLUT | | | |
| Utilization Details | 0 W D | . Ied_OBUF | 0.000 | 100,000 | 3 | 0 | clkout0 | FEIOLUT | | | |
| Hierarchical (0.918 W) | | | | | | | | | | | |
| Clocks (0.011 W) | | | | | | | | | | | |
| ✓ Signals (0.061 W) | | | | | | | | | | | |
| Data (0.061 W) | | | | | | | | | | | |
| Clock Enable (0.W) | | | | | | | | | | | |
| SetReset (0 W) | | | | | | | | | | | |
| Logic (0.011 W) | | | | | | | | | | | |
| BRAN (0 714 W) | | | | | | | | | | | |
| Clock Manager (0.117 W) | | | | | | | | | | | |
| BO (8:004 W) | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |

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





| Legend: Edit Properties | Net Properties | | | ? _ D 7 X |
|--|---------------------|-------|-------|--------------|
| Toggle rate: 000 % Static probability: 000 Legend: Estimated | _f_dut_reset | | | + + 0 |
| Static probability: 0.0 | Output | | | |
| Legend: Estimated | Toggle rate: | 0.0 % | | |
| | Static probability: | 0.0 | | |
| Edit Dronartiae | | | | |
| Edit Proprintes | | | - | |

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

Step 6: Editing Power Properties and Refining the Power Analysis

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

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

| et power properties for de | ut_reset. | |
|----------------------------|-----------|---|
| Output | | |
| | | |
| Toggle rate: | 4.000 🗘 | % |

- 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.



| Net Properties | ? _ D .7 × |
|---|--|
| .F dut_reset | + - O |
| Output | |
| Toggle rate: 4.0 % Static probability: 0.1 | |
| | Edit Properties |
| General Properties Connectivi | Power Aliases Cell Pins Nodes Tiles Pips |

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

| cl Console x Micessajos Log Reports Design Runs Power | 5 – D D |
|--|---------|
| Q. 茶 ゆ 川 田 麗 首 | |
| Command: report power -file Cr/DorshNMm/power 1.pwr -xpc Cr/DorshNMm/power 1.apc -rpm Cr/peo//power 1.rpm -mame power 1 Running Vector-Leas Activity Propagation | ^ |
| Pinished Numming Vector-less Activity Propagation 0 Infos, 0 Warnings, 0 Critical Warnings and 0 Errors encountered. | |
| report_power completed successfully ret switching sciivity -topple rate 4.000000 -static probability 0.100000 [pet sets dis/dut reset] | |
| and have a state of the second state of the se | |
| • | 3 |
| Igpo a Tel command here | |

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

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

| a 🗄 🔹 C 🔛 | Q 🛨 SetReset | | | | | | | 1000 | User Defined | Attribute | Calculated |
|--------------------------|---------------------------|------------------|--------------------|---------|--------|--------------|----------|-----------|--------------|-----------|------------|
| Setings | Utilization | Name | Signal Rate (Mh/s) | % High | Fanout | Slice Fanout | Clock | LogicType | | | |
| Summary (1.358 W) | 4<0.001 W (<1% of total) | Si cut_tpga | | | | | | | | | |
| Power Supply | 1 <0.001 W (<1% of total) | _f_cutidut_reset | 4.000 | 10.000 | 630 | D | clicout0 | FFLUT | | | |
| Utilization Details | 0 W | J led_OBUF | 0.000 | 100.000 | 3 | 0 | (decard) | FFBOLUT | | | |
| Hierarchical (0.898 W) | | | | | | | | | | | |
| Clocks (0.011 W) | | | | | | | | | | | |
| Signals (0.065 W) | | | | | | | | | | | |
| Data (0.665 W) | | | | | | | | | | | |
| Clock Enable (0 W) | | | | | | | | | | | |
| SetReset (<0.001 W) | | | | | | | | | | | |
| Logic (0.01 \V) | | | | | | | | | | | |
| BRAM (0.7 W) | | | | | | | | | | | |
| Clock Manager (0, 117 W) | | | | | | | | | | | |
| BO (0.004 W) | | | | | | | | | | | |

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



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

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

 \bigotimes IN as

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

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

Step 7: Running Functional Simulation with SAIF Output

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

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



| Q | Simulation | | | | | | |
|--|-----------------|------------------|-----------------|---------|----------|---|----------|
| Project Settings General | Specify various | settings assoc | iated to Simula | tion | | | <u>^</u> |
| Simulation | Target simulat | or: | Vivado Simul | ator | | | ~ |
| Elaboration Synthesis | Simulator lang | uage: | Mixed | | | | ~ |
| Implementation | Simulation set | | 🚞 sim_1 | | | | ~ |
| Bitstream | Simulation top | module name: | testbench | | | | © |
| Tool Settings Project | 🗹 Clean up si | mulation files | | | | | |
| IP Defaults | Compilation | Elaboration | Simulation | Netlist | Advanced | | |
| Source File | xsim.elat | orate.snapshot | E . | | | | |
| Display WebTalk | xsim.elat | orate.debug_le | vel | | typical | | ~ |
| Help | xsim.elat | porate relax | | | | Z | |
| > Text Editor | xsim.elat | orate.mt_level | | | auto | | ~ |
| 3rd Party Simulators | xsim.elat | orate.load_glbl | | | | × | |
| > Colors | | orate.rangeche | | | | | |
| Selection Rules | | oorate.sdf_delay | | | sdfmax | | ~ |
| Shortcuts Strategies > Window Behavior | xsim.elat | oorate xelab.mo | re_options | | | | |
| | Select an opti | on above to see | a description (| of it | | | |
| | | | | | | | |

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



| imulation pecify various setting: arget simulator: imulator language: imulation set: | s assoc | iated to Simula Vivado Simul | | ***** | | 4 | | | | |
|--|---|---|---|---|--|---|--|--|--|--|
| imulator language: | | Vivado Simul | ator | | | | | | | |
| | | | | | | | | | | |
| imulation set: | | Mixed | | | | | | | | |
| | | 📾 sim_1 | | | | ~ | | | | |
| imulation top module | name; | testbench | | | | | | | | |
| Clean up simulation | n files | | | | | | | | | |
| Compilation Elabo | ration | Simulation | Netlist | Advanced | | | | | | |
| xsim.simulate.tcl | post | | | | | | | | | |
| xsim.simulate.rur | ntime | | | 1000ns | 0 | | | | | |
| xsim.simulate.log | _all_sig | gnals | | | | | | | | |
| xsim.simulate.cu | stom_to | 1 | | | | | | | | |
| xsim.simulate.wd | lb | | | | | | | | | |
| xsim.simulate.sa | if_scope | 9 | | | | | | | | |
| xsim.simulate.sa | if | | | power_tutorial | l_func.saif | | | | | |
| xsim.simulate.sa | if_all_si | gnals | | | | | | | | |
| xsim.simulate.xsi | m.more | _options | | | | | | | | |
| xsim.simulate.saif SAIF filename | | | | | | | | | | |
| | compilation Elabor xsim.simulate.tcl. xsim.simulate.tcl. xsim.simulate.log xsim.simulate.cu xsim.simulate.cu xsim.simulate.sa xsim.simulate.sa xsim.simulate.sa xsim.simulate.sa | xsim.simulate.tcl.post xsim.simulate.tcl.post xsim.simulate.log_all_si xsim.simulate.log_all_si xsim.simulate.custom_tc xsim.simulate.custom_tc xsim.simulate.saif_scope xsim.simulate.saif_scope xsim.simulate.saif_ xsim.simulate.saif_all_si xsim.simulate.saif_all_si | Elaboration Simulation xsim.simulate.tcl.post xsim.simulate.tcl.post xsim.simulate.tcl.pog_all_signals xsim.simulate.custom_tcl xsim.simulate.custom_tcl xsim.simulate.saif_scope xsim.simulate.saif_scope xsim.simulate.saif_scope xsim.simulate.saif_scope xsim.simulate.saif_scope xsim.simulate.saif_all_signals xsim.simulate.saif_scope | Compilation Elaboration Simulation Netlist xsim.simulate.tcl.post xsim.simulate.tcl.post xsim.simulate.tcl.post signals xsim.simulate.custom_tcl xsim.simulate.saif_scope xsim.simulate.saif_scope xsim.simulate.saif_all_signals xsim.simulate.saif_all_signals xsim.simulate.saif_all_signals xsim.simulate.saif_all_signals xsim.simulate.saif | Compilation Elaboration Simulation Netlist Advanced xsim.simulate.tcl.post | Compilation Elaboration Simulation Netlist Advanced xsim.simulate.tcl.post 1000ns xsim.simulate.tcl.pog_all_signals 2 xsim.simulate.log_all_signals 2 xsim.simulate.custom_tcl 2 xsim.simulate.saif_scope 2 xsim.simulate.saif_scope 2 xsim.simulate.saif_all_signals 3 xsim.simulate.saif_all_signals 3 xsim.simulate.saif_all_signals 3 xsim.simulate.saif_all_signals 3 xsim.simulate.saif_all_signals 3 | | | | |

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

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

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

| Run Simula | tion |
|----------------|---|
| | Run Behavioral Simulation |
| ✓ RTL ANALYSIS | Run Post-Synthesis Functional Simulation |
| > Open Elat | Run Post-Synthesis Timing Simulation |
| ✓ SYNTHESIS | Run Post-Implementation Functional Simulation |
| Run Synth. | Run Post-Implementation Timing Simulation |

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



| cope × Sources. | | - 0 | 23 | Objects | ? | × D O _ | Untitled 1 | | 2 🗆 0 |
|-----------------|-------------|-------------|----|---------------|--------|-----------|-------------|------------|----------------------|
| Q <u>∓</u> ¢ | | | ¢ | Q | | 0 | Q 🖬 @ @ 💥 | - H H | et et infini at inti |
| lame | Design Cont | Block Type | | Name | Value | Date T. C | | | 1,000.030 |
| testbench | testbench | Verilog N., | | The sys_clk_p | 0 | Logic | Name | Value | 0 ns 400 24 |
| > 🔒 dut_togs | dut_togs | Verilog M. | | Ti sys_dk_n | 1 | Logic | 5)5_6K_0 | | 0 ns |
| 🖬 qibi | pibi | Verilog M. | | bal 🧬 | 1 | Logic | W ass. ck.n | 1 | |
| | | | | Va pass | 1 | Logic | lik ted | 1 | |
| | | | | > N WATCHDD | 100000 | Array | Té pass | the second | |
| | | | | | | | | 000186a0 | (010)8650 |
| | | | | | | | | | |

After the simulation completes, click \mathbf{x} at the top right corner to close the simulation window.

Step 8: Incorporating SAIF Data into Power Analysis

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

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

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

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



| imate power ci | onsumption base | d on the neti | st design an | d part xc7k325 | ffg900-2. |
|-----------------------|------------------------|---------------|--------------|-----------------|------------|
| es <u>u</u> lts name: | power_3 | | | | |
| nvironment | Power Supply | Switching | Output | | |
| Reset swit | ching activity befor | re report pow | er | | |
| Switching Activ | ity for Resets: | None | ~ | | |
| Simulation Set | ttings | | | | |
| Simulation | activity file (.saif): | (/sim_1/sy | nähäuncipow | er_tutorial_tun | c.salf 🕢 … |
| Default Activity | y Settings | | | | |
| Default tog | gie rate | 12.5 | [0-100] | | |
| Default Sta | tic Probability. | 0.5 | [0.0 - 1.0] | | |
| Enable Rate Se | ettings | | | | |
| | | Static Prob | ability | Toggle Rate | |
| BRAM Port | Enable | | [0.0 - 1.0] | | [0 - 100] |
| BRAM Write | e Enable: | | [0.0 - 1.0] | | [0 - 100] |
| Bidi Output | Port Enable: | | [0.0 + 1.0] | | [0~100] |
| Toggle Rate Se | ettings | | | | |
| | | Static Prob | ability | Toggle Rate | |
| Primary Ou | tputs: | | [0 0 - 1 0] | | [0 - 100] |
| Logic | | | | | |
| Register | 5 | | 0.0 - 1.0] | | (0 - 100) |
| | uisters: | | [0.0 - 1.0] | | [0 - 100] |
| Shift Reg | | | | | |

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.

| Q Image of the system Summary Settings Power statistics files or constraints files or vectorises analysis. Nete faces any constraints files or vectorises analysis. Nete faces and vectorises analysis. Nete face and vectoris | 4.1 | | - | 1.1 | | 1 | 538 | | | | | | | | |
|--|-----|--|---|--|---|--------------------------|---|--|-----|--------------------|---|---|--|--|--|
| Summary (19.332.W) Power estimation from Synthesized indist. Korthy Control Supply Power estimation from Synthesized indist. Korthy Web (1969) Abit Mynemication. Power estimation from Synthesized indist. Korthy Web (1969) Abit Mynemication. Power estimation from Synthesized indist. Korthy Web (1969) Abit Mynemication. Power estimation from Synthesized indist. Korthy Web (1969) Abit Mynemication. Power estimation from Synthesized indist. Korthy Web (1969) Abit Mynemication. Power estimation from Synthesized indist. Korthy Web (1969) Abit Mynemication. Power estimation from Synthesized indist. Korthy Web (1969) Abit Mynemication. Power estimation from Synthesized indist. Korthy Web (1969) Abit Mynemication. Power estimation from Synthesized indist. Korthy Web (1969) Abit Mynemication. Power estimation from Synthesized indist. Korthy Web (1969) Abit Mynemication. Power estimation from Synthesized indist. Korthy Web (1960) Abit Mynemication. Power estimation from Synthesized indist. Korthy Bit Mynemication. Power estimatin from Synthy | Q | | ÷ | Ŧ | C | | Summary | | | | | | | | |
| | S F | Sum Pow Jtile I I I I I I I I I I I I I I I I I I I | mary er Sup atton Herar Clocks Signal Da Cl Sig Signal Clock Signal Clock | opiy Deta chica s (0.0 ls (0) ls (0) ats (0 lock E stiRe (0.0 c (0.0 c Mana | 45 4 (0.85 11 W) 0.43 W) 0.43 W 0.43 W 2045 W 2045 W 2045 W 2045 W 2045 W 2045 W 2045 W |) (<0.001 W 801 WI | derved from constants files, ar vectorists analysis. Nete free- change after implementation. Total Os-Chip Power: Juncikon Temperature: Therma Margin: Effective 3JA: Power supplied to off-chip device Confidence invel Lauroch Power Constant Adviso | nulation fires or early estimates can 1.332 W 27.4 °C 57.6 °C (30.3 W) 1.8 °C W 55.0 W High | 50% | Dyna 756 79% | Clocks: Signals Logic: BRAM: NMCM: NMCM: | 0.043 W 0.009 W 0.685 W 0.117 W 0.005 W | (5%) (1%) (79%) (14%) (0%) | | |



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

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



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

| QIOCH | Q ¥ 10 | | | | | | | 133 | Sim | latio | n 🖩 | Cor | strait | 1 | Calculated |
|--|---------------------------|------------------|---------|-------------|---------------|------------|-------------|------------|-----|-------|-----|-----|--------|---|--------------|
| Settings | Utilization | Name | 10 Type | VO Standard | Onve Strength | Input Pins | Output Pins | Bidir Pins | | - | | | - | | Signal Rate. |
| Summary (1.3.32 W) | > 10.005 W (1% of total) | 🗿 dut_fpga | | | | | | | | | | | | | |
| Power Supply | 0.004 W (<1% of total) | ₽ sys_dk_in_p | HP | DIFF_SSTL. | NA | 1 | 0 | 0 | la. | | | | 4 | | 400.00 |
| Vilization Details | > 10.001 W (<1% of total) | timc_out | HR | LVCM0833 | 12.000 | 0 | 10 | 0 | | - | | - | - | - | 1.000 |
| Hierarchical (0.889 W) | 1<0.001 W (<1% of total) | -2 gsic_out_pass | HR | LVCM0833 | 12.000 | 0 | 1 | 0 | 2 | - | | - | - | | 1.00 |
| Clocks (0.01 W) | <0.001 W (<1% of total) | -2 led | HP | LVCMD815 | 12.000 | 0 | 1 | 0 | | 1 | | 14 | | | 1.00 |
| Crock Enable (<0.001 W Selficest (<0.001 W) Logic (0.009 W) BRAM 0 Sits /// Clock Hanager (0.117 W) ICO (0.005 W) | | | | | | | | | | | | | | | |

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 | | | | | | | | | | ? 0 7 × |
|-------------------------|-----------------------------|-----------------|---------------------|--------|--------|--------------|---------|------------|------------|------------|
| Q X 0 C M | Q 🔮 SetiReset | | | | | | | | Simulation | Calculated |
| Settings | Utilization | Name | Signal Rate (Min/s) | % High | Fanout | Slice Fanout | Clock | Logic Type | | |
| Summary (1.332 W) | > 1 <0.001 W (<1% of total) | 🕅 dut_fpga | | | | | | | | |
| Power Supply | 1 40.001 W (41% of total) | . dut/dut_reset | 1.000 | 26.250 | 530 | 0 | cikout0 | FF LUT | | |
| ~ Utilization Details | 0 W | F led_OBUF | 1.000 | 73.740 | 3 | 0 | cikout0 | FF VO LUT | | |
| Hierarchical (0.859 W) | | | | | | | | | | |
| Clocks (0.01 VII) | | | | | | | | | | |
| Signals (0.043 W) | | | | | | | | | | |
| Data (0:043 W) | | | | | | | | | | |
| Clock Enable (<0.001 W | | | | | | | | | | |
| SelfReset (<0.001 W) | | | | | | | | | | |
| Logic (0.009 W) | | | | | | | | | | |
| BRAN (0.685 W) | | | | | | | | | | |
| Clock Manager (0.117 W) | | | | | | | | | | |
| NO (0.005.10) | | | | | | | | | | |
| | | | | | | | | | | |
| (| | | | | | | | | | |
| power_1 × power_2 × po | | | | | | | | | | |



Step 9: Implementing the Design

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

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

| 2- | Implementation | | | | |
|----------------------------|-------------------------------------|---------------------|--------------------------------|---------------|---------------|
| Project Settings | Specify various settings as | ssociated to Implen | nentation | | P |
| General | | | | | |
| Simulation Elaboration | Constraints | | | | |
| Synthesis | Default constraint set | 📾 constrs_1 (a | ctive) | | ~ |
| Implementation | Options | | | | |
| Bitstream | Opuons | | | | |
| > IP | Run report UltraFas | t methodology after | r routing | | |
| Fool Settings Project | Incremental compile: | | | | |
| IP Defaults Source File | Strategy: | 🙏 Vivado Implem | nentation Defaults | s* (Vivado lm | - |
| Display | Description: | | | | |
| WebTalk | VOpt Design (opt_des | iign) | | | |
| Help | is_enabled | | | | |
| > Text Editor | tcl.pre | | | | *** |
| 3rd Party Simulators | tcl.post | | | | *** |
| > Colors | -verbose | | | | |
| Selection Rules | -directive | | NoBramPower | Opt | in the second |
| Shortcuts | More Options | | Explore | | |
| Strategies | YPower Opt Design () | ower_opt_design) | ExploreArea ExploreSequent | alArea | |
| > Window Behavior | is_enabled | | AddRemap | | |
| | tcl.pre | | RuntimeOptimiz NoBramPowerO | | |
| | tcl.post | | ExploreWithRem | | |
| | More Options | | Default | | |
| | YPlace Design (place | _design) | | | ~ |
| | -directive Opt design directive. | | | | |
| > | | | | | |
| *) | | OK | Cancel | Apply | Restore |

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



| ementation? |
|--------------------|
| |
| dut_fpga_kc705.xdc |
| n't Save Cancel |
| |

Conclusion

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





Lab 3

Running Timing Simulation and Estimating Power

Introduction

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

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

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

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

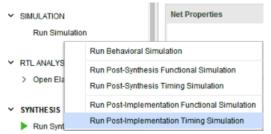
- 2. In the Flow Navigator, click **Settings** and select the **Simulation** to set the timing simulation properties. In the Settings dialog box, note that the following defaults are automatically set:
 - Simulation set: **sim_1**
 - Simulation top-module name: testbench
- 3. In the Elaboration tab, make sure that debug_level is set to **typical**, which is the default value.
- 4. In the Simulation tab, set the SAIF file name xsim.simulate.saif to **power_tutorial_timing_xsim.saif**.
- 5. Set the xsim.simulate.saif_scope to testbench/dut_fpga.
- 6. Observe that the simulation run time xsim.simulate.runtime is 1000ns.
- 7. Click OK.



| N 8. 55 R 2726 3 | | | | | |
|--|---|--|--|---|--|
| Specify various settings associ | iated to Simulation | | | <u>^</u> | |
| Target simulator: | Vivado Simulator | | * | | |
| Simulator language: | Mixed | | | ~ | |
| Simulation set: | 🚟 sim_1 | | | ~ | |
| Simulation top module name: | testbench | 0 | | | |
| Compilation Elaboration | Simulation Netlist | Advanced | | | |
| xsim.simulate.runtime | | 1000ns | | 6 | |
| xsim.simulate.log_all_sig | ignals* | | | | |
| xsim.simulate.custom_tc | 1 | | | | |
| xsim.simulate.wdb | | | | | |
| xsim.simulate.salf_scope | 9 | testbench/dut_fpga | | | |
| xsim,simulate.saif* | | power_tutorial_t | 0 | | |
| | | | | ~ | |
| xsim.simulate.runtime Specify simulation run time | | | | | |
| | Target simulator: Simulator language: Simulation set Simulation top module name: ✓ Clean up simulation files Compilation Elaboration xsim.simulate.runtime xsim.simulate.log_all_sig xsim.simulate.custom_toc xsim.simulate.salf_scope xsim.simulate.salf_scope xsim.simulate.salf_scope xsim.simulate.salf_scope xsim.simulate.salf_scope xsim.simulate.salf_scope xsim.simulate.salf_scope xsim.simulate.salf_scope | Target simulator: Vivado Simulator Simulator language: Mixed Simulation set: im sim_1 Simulation top module name: testbench Image: clean up simulation files Compilation Elaboration Simulation mulate.runtime xsim.simulate.log_all_signals* xsim.simulate.custom_tcl xsim.simulate.salf_scope xsim.simulate.salf_scope xsim.simulate.salf* xsim.simulate.salf* testeate xsim.simulate.salf* testeate xsim.simulate.runtime testeate xsim.simulate.salf* testeate | Target simulator Vivado Simulator Simulator language: Mixed Simulation set: iiii sim_1 Simulation top module name: testbench Image: Itestbench Image: Compliation files Compliation Elaboration Simulation Netlist Advanced xsim.simulate runtime xsim.simulate log_all_signals* xsim.simulate custom_tcl xsim.simulate.saif_scope testbench/dut_fig xsim.simulate.saif* power_tutorial_t interside sim_simulate.runtime | Target simulator: Vivado Simulator Simulator language: Mixed Simulation set: im sim_1 Simulation top module name: testbench Image: Clean up simulation files Compilation Elaboration Simulate.untime 1000ns xsim.simulate.untime 1000ns xsim.simulate.ustom_tcl xsim.simulate.ustom_tcl xsim.simulate.salf power_tutorial_timing_xsim salf uster size idea est interest xsim.simulate.salf* power_tutorial_timing_xsim salf | |

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

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



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

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



| Q | Design Unit | Block Type | ۰ | Q | | 0 | | the last to show the | and the second second second | |
|-------------|-------------|-------------|---|---------------|--------|-----------|---------------|----------------------|--|--------------|
| estbench | Design Unit | Phone Phone | | | | w | 0 8 6 6 3 | | the states of the second secon | 4 |
| | | DOCK Sype | | Name | Value | Data T. ^ | | | | 1,000.000 20 |
| | testbench | Verlog M. | | Va sys_ck_p | 0 | Logic | Name | Value | 15 mil 1600 mil | |
| > 🔒 duLipga | duLlpga | Venlog M. | | 14 sys_dk_n | 1 | Logic | la sys. dk. p | 0 | 0 m 800 m | A |
| 🛢 gibi | gibi | Verlog M. | | Tá led | 1 | Logic | la sys_clk_n | 1 | | |
| | | | | G pasa | 1 | Logic | le led | 10 | | |
| | | | | > 👒 WATCHDO . | 100000 | Array | le pasa | 15 | | |
| | | | | | | | | | | |

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** \rightarrow **Report Power** from the main menu.

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

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

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

| Q X O C " | Summary | | | | | | | |
|--|--|---|--------------------|--------|---|--|---|--|
| Settings Summary (1.545 W) Power Supply Vikitation Details Hierarchical (1.08 W) Clocks (0.022 W) Clocks (0.022 W) Data (0.223 W) Clock Enable (0 W) | Power analysis from Implemented r derived from constraints lice, simuli vectoriess analysis. Total On-Chip Power: Junction Temperature: Thermal Margin: Effective 3JA Power supplied to off-chip devices. | 1.545 W 27.7 °C 57.3 °C (30.1 W) 1.8 °C/W 0 W | On-Chip Pov 70% | 21% | Clocks: Signals: Logic: BRAM: MMCM: | 0.022 W 0.022 W 0.223 W 0.009 W 0.705 W 0.117 W | (2%) (21%) (1%) (65%) (11%) | |
| SetFRead (~0.001 W) Logic (0.008 W) BRAM (0.705 W) Clock Manager (0.117 W) I/O (0.004 W) | Confidence level: <u>Launch Power Constraint Advisor</u> to invalid switching adivity | Medium 9 find and fix | 30% | Device | Static: 0.4 | 0.004 W | (0%) %) | |

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

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



Step 3: Running Report Power with Vivado Simulator SAIF Data

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

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

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

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

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

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

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

| Q X + C * | Summary | | | |
|---|--|----------|--|--|
| Settings Summary (1,485 W) Power Supply Utilization Details Hierarchical (1,019 W) Clocks (0,021 W) Clocks (0,021 W) Data (0,182 W) Clock Enable (=0,00 | Power analysis from Implemented neillist. Act derived from constraints files, simulation files vectories analysis. Total On-Chip Power: 1.445 W Junction Temperature: 27.6 °C Thermal Margin: 57.4 °C Efficience 3.4k 1.8 °C W Power supplied to off-chip devices. 0.107 | (30.2 W) | Dynamic: 1.019 W (59%) Diff Clocks: 0.021 W (2%) Signals: 0.182 W (15%) Dynamic: Break Mincur: 0.182 W (15%) Mincur: 0.685 W (17%) Mincur: 0.117 W (12%) | |
| SetReset (=0.001 W, Logic (0.008 W) BRAM (0.665 W) Clock Manager (0.117 W, WO (0.005 W) 2 mpl 1 (saved) × power 1 | Launch Power Constraint Advisor to find and invalid switching advity | fx 31% | 1256 UO: 0.005 W (0%) Device Static: 0.466 W (01%) | |

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

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



| Q ¥ ≜ C » | Q 😤 Data | | | | Simulat | ion Calcu | dater |
|------------------------|--------------------------|---|---------------------|--------|---------|--------------|-------|
| Settings | Utilization | Name | Signal Rate (Mtr/s) | % High | Fanout | Silce Fanout | CI |
| Summary (1.485 W) | 182 W (12% of total) | 🗃 dut_fpga | | | | | |
| Power Supply | 1 0.001 W (<1% of total) | _f_dul/gen_dut(9).bram_top_inst/addr_a(1) | 73.000 | 63.935 | 34 | 34 | d |
| V Utilization Details | 0.001 W (<1% of total) | / dutigen_dut(9].bram_top_instladdr_b(11) | 73.000 | 36.049 | 34 | 34 | d |
| Hierarchical (1.019 W) | 10.001 W (<1% of lotal) | _f_dutigen_dut[1].bram_top_inst/bram_inst/mem_reg_2_0_i_2_n_0 | 74.000 | 36.001 | 16 | 16 | d |
| Clocks (0.021 W) | 1 0.001 W (<1% of total) | f dutigen_dut[9].bram_top_inst/bram_inst/mem_reg_0_0_i_3_n_0 | 73.000 | 36.011 | 16 | 16 | d |
| ~ Signals (0.182 W) | 1 0.001 W (<1% of total) | _f_dutigen_dut[1].bram_top_instladdr_b(16) | 73.000 | 63.934 | 37 | 35 | d |
| Data (0.182 W) | 10.001 W (<1% of lotal) | _f_dutigen_dut(9).bram_top_instladdr_b(3) | 73.009 | 36.049 | 34 | 34 | d |
| Clock Enable (<0.00 | 1 0.001 W (<1% of total) | f dutigen_dut[9].bram_top_instibram_instimem_reg_0_0_i_4_n_0 | 73.000 | 35.019 | 16 | 15 | d |
| Sel/Reset (<0.001 W | 0.001 W (<1% of total) | J dutigen_dut[8].bram_top_inst/bram_inst/mem_reg_2_0_i_1_n_0 | 72.000 | 36.000 | 16 | 16 | d |
| Logic (0.008 W) | 10.001 W (<1% of total) | ∫ dutigen_dut[2].bram_top_inst/addr_b[0] | 73.000 | 63.934 | 34 | 34 | d |
| BRAM (0.685 W) | 10.001 W (<1% of total) | J dutigen_dut[9].bram_top_inst/bram_inst/mem_reg_2_0_i_2_n_0 | 74.000 | 36.001 | 16 | 16 | d |
| Clock Manager (0.117 W | 1 0.001 W (<1% of total) | / dutigen_dut(9).bram_top_instladdr_b(0) | 73.000 | 63.934 | 34 | 34 | d |
| VO (0.005 W) | 1 0.001 W (+1% of lotal) | _f_dutigen_dut(2).bram_top_inst/addr_b(5) | 73.000 | 63.936 | 34 | 34 | d |
| | | | | | | | 1 |

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

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

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

| Power | | | | | | | ? _ |
|--|--|--|------------------|--|---|------------------------------|-----|
| Q ¥ ¢ C * | Summary | | | | | | |
| Summary (1.545 W) Power Supply V Utilization Details Hierarchical (1.08 W Clocks (0.022 W) Signals (0.223 W) Data (0.223 W) Clock Enable (C SetReset (<0) Logic (0.009 W) Launch Power C | Junction Temperature: 27.7 Thermal Margin: 57.3 | 545 W 545 W 7.7 °C 7.3 °C (30.1 W) 8 °C/W W | | On-Chip Power Dynamic: 1.080 W (70%) 21% Clocks: 0.022 W (2%) Signals: 0.223 W (21%) Explic: 0.009 W (1%) 55% BRAM: 0.705 W (65%) MMCM: 0.117 W (11%) UQ: 0.004 W (0%) | | | |
| | Launch Power Constraint Advisor to find an invalid switching activity | r Design State: Clock Activity: | Design is routed | | | | |
| impl_1 (saved) × powe | <u>r_1 ×</u> power_2 × | I/O Activity: Internal Activit Characterizati | r | High User specifie Medium User specifie nodes High | d more than 95% d less than 25% s are Productio | % of inputs 6 of internal | |

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

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



| | | | | | | | | ? _ |
|---|---|--|---|---|---|----------------|-----------|-----|
| Q ₹ ♦ | C | Summary | | | | | | |
| Settings Summary (1.485 W) Power Supply > Utilization Details Hierarchical (1.019 Clocks (0.021 W) > Signals (0.182 W) Data (0.182 W) Clock Enable (* | | Junction Temperature: 27.6 Thermal Margin: 57.4 Effective 3JA: 1.8 Power supplied to off-chip devices: 0.0 | files or 85 W 5 °C 4 °C (30.2 W) °C/W | 69% | (2%) (18%) (1%) (67%) (12%) | | | |
| Set/Reset (<0.0 Logic (0.008 W) BRAM (0.685 W) Clock Manager (0.1 WO (0.005 W) | Confidence level: Higt Launch Power Constraint Advisor to find | 12% (10) U 005 W | | | | | (0%) × | |
| | invalid switching activity | Design State: Clock Activity: | Hig De Hig | ph Isign is route ph | | | | |
| impl_1 (saved) X power | | r_1 × power_2 × | VO Activity: Internal Activity: | I/O Activity: High User specified more than 95% of inputs Internal Activity: High User specified more than 25% of interna nodes | | | | |
| | | | Characterizatio | | | 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 the SAIF data using Questa Advanced Simulator.

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

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

or

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

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

To create new compiled libraries:

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



| Q- | 3rd Party Simula | | a the second state with the | A |
|---|------------------|---|-----------------------------|----|
| Project Settings General | | ths and default compiled library paths. Default compiled w project creation. | o library path will be | ** |
| Simulation Elaboration | Install Paths | | | |
| Synthesis Implementation | ModelSim: | | | |
| Bitstream | QuestaSim: | G:/gensys/gensys/questa/10.5c/win32 | ··· | |
| > IP | IES: | | | |
| Tool Settings Project | VCS: | | | |
| IP Defaults | Riviera: | | | |
| Source File | ActiveHDL: | | | |
| Display WebTalk | Default Compiled | Library Paths | | |
| > Text Editor | ModelSim: | | ••• | |
| 3rd Party Simulators | Questa: | | | |
| Colors Selection Rules | IES: | | | |
| Shortcuts | VCS: | | | |
| Strategies > Window Behavior | Riviera: | | | |
| | ActiveHDL: | | | |

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 postimplementation 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.



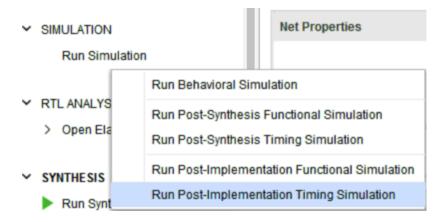
| Targ | et Simulator |
|------|---|
| ? | Questa Advanced Simulator requires that Xilinx libraries are pre-compiled. You can compile Xilinx simulation libraries by going to tools menu and clicking on compile simulation libraries. Make sure that the compiled library location in simulation settings menu 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'? |
| | Yes No |

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

| Q | Simulation | | | | | | |
|----------------------------------|--|------------------------|-----------|---------------|----------|------------------------------------|----|
| Project Settings General | Specify various settings assoc | iated to Simulation | | | | | ** |
| Simulation | Target simulator: | Questa Advanced | Simulat | or | | | ~ |
| Elaboration Synthesis | Simulator language: | Mixed sim_1 testbench | | | | | ~ |
| Implementation | Simulation set | | | | | | • |
| Bitstream | Simulation top module name: | | | | | | |
| Tool Settings | Clean up simulation files | | | | | | |
| Project IP Defaults | Compiled library location: | i/akasha/tmp/Tuto | orial/rea | d_salf/projec | ct_3/pro | ject_3.cache/compile_simlib/questa | 0 |
| Source File Display | Compilation Elaboration | Simulation No | otlist | Advanced | | | |
| WebTalk | questa.simulate.log_all_ | signals* | | | | 2 | ^ |
| Help | questa simulate custom | | | | | | |
| > Text Editor | questa.simulate.custom | _udo | | | | | |
| 3rd Party Simulators > Colors | questa.simulate.custom | wave_do | | | | | |
| Selection Rules | questa.simulate.sdf_dela | ay | | sdfma | sdfmax | | |
| Shortcuts | questa.simulate.leee_wa | amings | | | | × | |
| Strategies | questa.simulate.saif_sco | ppe* | | testbe | ench/du | t_fpga | |
| > Window Behavior | questa.simulate.salf* | | | power_ | tutorial | _timing_questasim_saif | |
| | questa simulate vsim mo | re_options | | | | | ~ |
| | questa.simulate.saif* Specify SAIF file | | | | | | |
| (s | | | | | | Cancel Apply | |

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





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

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

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

Step 2: Running Report Power in Vectorless Mode

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

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

Alternatively, select **Reports** \rightarrow **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.



| Q 🗄 🖨 | Summary | | | | | | |
|--|---|---|--------------------|---------------------|--|---|--|
| Settings Summary (1.485 W) Power Supply Utilization Details Hierarchical (1.019 Clocks (0.021 W) Signals (0.182 W) Data (0.182 W) Clock Enable (Set/Reset (<0 (Logic (0.008 W) BRAM (0.685 W) Clock Manager (0.1 VO (0.005 W) | Power analysis from Implemented derived from constraints files, simu vectorless analysis. Total On-Chip Power: Junction Temperature: Thermal Margin: Effective 3JA Power supplied to off-chip devices Conflidence level: Launch Power Constraint Advisor 1 invalid switching activity | 1.485 W 27.6 °C 57.4 °C (30.2 W) 1.8 °C/W : 0 W High | On-Chip Pox 69% | Dynam 18% 67% | Clocks: Signals: Logic: BRAM: MMCM: VO: | 0.021 W (69%) 0.021 W (2%) 0.182 W (18%) 0.008 W (1%) 0.685 W (67%) 0.117 W (12%) 0.005 W (0%) 166 W (31%) | |

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

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

- 1. In the main menu bar, select **Reports** \rightarrow **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.



| | | רקם_? |
|---|---|--|
| Q ₹ ♦ C × | Summary | |
| Settings Summary (1485 W) Power Supply Utilization Details Hierarchical (1.019 W) Clocks (0.021 W) > Signals (0.182 W) Data (0.182 W) Clock Enable (=0.00 | Power analysis from Implemented netlist. Achity derived from constraints files, simulation files or vectoriess analysis. Total On-Chip Power: 1.485 W Junction Temperature: 27.6 °C Thermal Margin: 57.4 °C (30.2 W) Effective 3JA 1.8 °C/W Power supplied to off-chip devices: 0.W | 67% BRAM: 0.685 W (67%) |
| SetReset (0 W) Logic (0.068 W) BRAM (0.685 W) Clock Manager (0 117 W | Confidence level: High Launch Power Constraint Advisor to find and fix invalid switching adwity | 31% Image: Characterized and Characterized a |

- 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.

| q ¥ ≑ C " | Q 🛨 Data | | | 1 | Simulatio | on 🗌 Calcula | ated |
|---|--------------------------|---|---------------------|--------|-----------|--------------|------|
| Settings | Utilization | Name | Signal Rate (Mtr/s) | % High | Fanout | Slice Fanout | ¢ |
| Summary (1.485 W) | 0.182 W (12% of total) | 😫 dut_fpga | | | | | |
| Power Supply V Utilization Details Hierarchical (1.019 W) | 0.001 W (<1% of total) | J dutigen_dut[9].bram_top_inst/addr_a[1] | 73.000 | 63.935 | 34 | 34 | ¢ |
| | 0.001 W (<1% of total) | J dut/gen_dut(9).bram_top_inst/addr_b(11) | 73.000 | 36.049 | 34 | 34 | ¢ |
| | 0.001 W (<1% of total) | J dut/gen_dut[1]bram_top_inst/bram_inst/mem_reg_2_0_i_2_n_0 | 74.000 | 36.001 | 16 | 16 | ¢ |
| Clocks (0.021 W) | 0.001 W (<1% of total) | dut/gen_dut(9).bram_top_inst/bram_inst/mem_reg_0_0_i_3_n_0 | 73.000 | 36.011 | 16 | 16 | ξ |
| Signals (0.182 W) | 10.001 W (<1% of lotal) | <pre>_ dut/gen_dut[1].bram_top_inst/addr_b[18]</pre> | 73.000 | 63.934 | 37 | 35 | c |
| Data (0.182 W) | 0.001 W (<1% of total) | J dut/gen_dut(9).bram_top_inst/addr_b(3) | 73.000 | 36.049 | 34 | 34 | |
| Clock Enable (<0.00 | 10.001 W (<1% of total) | f dut/gen_dut(9).bram_top_inst/bram_inst/mem_reg_0_0_i_4_n_0 | 73.000 | 36.019 | 16 | 16 | 4 |
| SetReset (0 W) Logic (0 008 W) | 10.001 W (<1% of total) | _f_dut/gen_dut(8) bram_top_inst/bram_inst/mem_reg_2_0_i_1_n_0 | 72.000 | 36.000 | 16 | 16 | |
| BRAM (0.685 W) | E 0.001 W (<1% of total) | <pre>f dutigen_dut[2] bram_top_inst/addr_b[0]</pre> | 73.000 | 63.934 | 34 | 34 | |
| Clock Manager (0.117 W | 10.001 W (=1% of total) | J dutigen_dut(9) bram_top_insbbram_insbimem_reg_2_0_i_2_n_0 | 74.000 | 35.001 | 16 | 16 | 1 |
| UD (0.005 W) | 0.001 W (<1% of total) | ∫ dut/gen_dut(9) bram_top_inst/addr_b(0) | 73.000 | 63.934 | 34 | 34 | 4 |
| 10 (0 000 H) | 0 001 W (<1% of total) | dut/gen_dut[2].bram_top_inst/addr_b[5] | 73.000 | 63.936 | 34 | 34 | 4 |
| | 0.001 W (<1% of total) | f dut/gen_dut[9].bram_top_inst/bram_inst/mem_reg_2_0_i_1_n_0 | 72.000 | 35.000 | 16 | 16 | 1 |
| | 0.001 W (<1% of total) | J dutigen_dut(9).bram_top_inst/addr_b(16) | 73.000 | 63.934 | 37 | 36 | 1 |
| | E 0.001 W (<1% of total) | ∫ dutigen_dut(8).bram_top_inst/bram_inst/mem_reg_0_0_i_4_n_0 | 73.000 | 35.019 | 16 | 16 | 1 |
| | 0.001 W (+1% of total) | <pre>_ dut/gen_dut(2) bram_top_instraddr_b[1]</pre> | 73.000 | 63.936 | 34 | 34 | |

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

Conclusion

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



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





Lab 4

Measuring Hardware Power Using the KC705 Evaluation Board

Introduction

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

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

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

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



| 1 Bitstream (| Generation successfull | y completed. |
|----------------------|------------------------|--------------|
| lext | | |
| 🔿 <u>V</u> iew Repo | rts | |
| Open Hard | lware Manager | |
| O <u>G</u> enerate M | lemory Configuration F | File |
| Don't show thi | s dialog again | |
| | ок | Cancel |

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

Step 2: Setting Up the KC705 Evaluation Board

IMPORTANT! This project is created for the KC705 Rev 1.0 Evaluation Board. The pin constraints are set based on this Evaluation Board. If you are using any other revisions, update the XDC file $dut_{fpga_kc705.xdc}$ with the correct pin constraints.

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





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

Step 3: Setting Up the Fusion Digital Power Designer Software

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

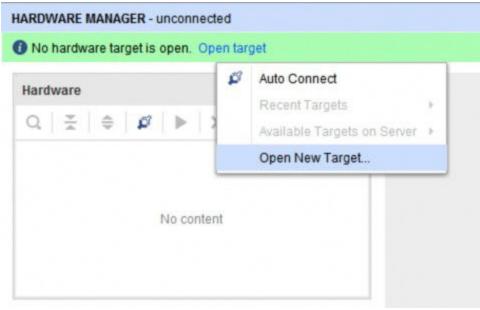
The software detects the USB adapter and brings up the GUI.



| Device Tools | Hep | UCD9248 @ Address 523 - Rail #1 |
|--|--|--|
| ifigure | Vout Config Sout, Vin, Temp Config Other Config Phase, Rail Config GP30 Config | |
| | Sequence On and Off Timing | UCD9246 @ 52d - Rad #1 |
| Auto write on rail or device change Discard Changes Econe RANI To Plash esto ne Flash to RAM war Reactor w Nortkae Ceory Rail/Design | Sub Sub <th>Woldsey Setgenere, Hangten, and Limits Vac: Mass: 1.600 € V Ouw Faulti 1.100 € V 15.0 € F Oww Faulti 1.100 € V 10.0 € % Norgin Hight 1.000 € V 5.0 € F Watt 1.000 € V 5.0 € F Watt 1.000 € V 5.0 € F Margin Hight 0.500 € V 5.0 € F Under Ware 0.500 € V -10.0 € F</th> | Woldsey Setgenere, Hangten, and Limits Vac: Mass: 1.600 € V Ouw Faulti 1.100 € V 15.0 € F Oww Faulti 1.100 € V 10.0 € % Norgin Hight 1.000 € V 5.0 € F Watt 1.000 € V 5.0 € F Watt 1.000 € V 5.0 € F Margin Hight 0.500 € V 5.0 € F Under Ware 0.500 € V -10.0 € F |
| : | Select Rail to Talt Baylow InLarger Window (?) | Under Wahr 0.500 ⊕ V -10.0 ⊕ % UV Response |
|) Soft Start) Soft Stop) Both Show call Abels | Dentce Raf # Raf hame Yout PG De PG DF On Delay Rate Off Delay Pall Dependencies (Prect | Over Current LV: 0.870 (1)/2 V -13.0 (1)/2 % 00 LV Response (v) Power Good Drc 0.900 (2)/2 V -10.0 (2)/2 % % |
| Show current device anly Show calls that thack | UCD3248 ⊕ 523 3 Rai#3 3.300 2.970 2.865 0.0 5.0 4.0 1.0 (N_0 (M_0 0.000), (N_0 CF) UCD3248 ⊕ 523 4 0345 4 2.540 1.00 1.400 1.660 0.0 5.0 3.0 1.0 100 (CF) ⊕ 6.000 , (N_0 CF) UCD3248 ⊕ 533 1 2.54 1 2.560 2.290 2.125 0.0 5.0 1.0 1.0 100 (CF) ⊕ 6.000 , (N_0 CF) | Power Good Off 0.850 ⊕ V -15.0 ⊕ % On/Off Confg: 0x00 ∨ (#ways Cenverting) |
| external Show external Vitadi sources | U LCD242 & £ 24 2 14 2 14 2 1500 1200 1277 0.0 5.0 5.0 1.0 10 10 10 10 20 20 00 1, 10 1 20 1 10 10 10 10 10 10 10 10 10 10 10 10 | Turn On Timing Turn Off Timing Turn On Delayi 0.0 ± ms Rise Time: 5.0 ⊕ ms |
| Configure | Tes & Hints PMBLs Log | |
| Design Monitor | TOPF HAX_WARE_LITET (0+65.87# 41] let us upper technologies the can benefic to power down the output without maching 12.7% of the subject voltage programmed at the time the current as is strandoff. A value of (0x747* Hereors hat there as in with and that the usive value value value of the technology for the | |
| Status | Niget with an interview of the second s | |

Step 4: Programming the Bitstream

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



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



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

When the hardware is detected successfully, the part information will be displayed in the Open New Hardware Target wizard.

5. Verify the part information, then click **Next** then click **Finish**.

| in the contract of the contrac | are Target | | | | | | | | |
|--|-----------------------------|--------------------------|---|--------------|------------------|-----------------|----------------|-------------|--|
| | | | able targets, then s select a different ta | | opriate JTAG clo | ck (TCK) freque | ncy. If you do | not see the | |
| lardware <u>T</u> arç | jets | | | | | | | | |
| Туре | Name | | JTAG Clock Free | quency | | | | | |
| xilinx_tcf | Digilent/210203 | 3337377A | 15000000 | ~ | | | | | |
| | | | | | | | | | |
| | | | Add | Xilinx Virtu | al Cable (XVC) |] | | | |
| łardware <u>D</u> evi | ces (for unknow | vn devices, | Add specify the Instru | | |] | | | |
| fardware <u>D</u> evi | ices (for unknow ID Code | /n devices, IR Length | specify the Instru | | |] | | | |
| | ID Code | | specify the Instru | | |] | | | |
| Name xc7k325t_ | ID Code | IR Length 6 | specify the Instru | | |] | | | |

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



| Hardware ? | _ 🗆 | c × |
|--|--------|--|
| Q ¥ ♦ Ø ▶ ≫ ■ | | 0 |
| Name | Status | |
| Iocalhost (1) | Conne | cted |
| ✓ ■ xilinx_tcf/Digilent/2102033373 | Open | |
| w xc7k325t_0 (1) | Proora | |
| XADC (System Monitor) | | Hardware Device Properties Ctrl+E |
| | | Program Device |
| | | Verify Device |
| | C | Refresh Device |
| | | Add Configuration Memory Device |
| | | and the second state of the second state of the second states |
| | | Boot from Configuration Memory Device |
| < | - | Boot from Configuration Memory Device |
| < | | Program BBR Key |
| Hardware Device Properties ? | | |
| | - | Program BBR Key |
| xc7k325t_0 | | Program BBR Key Clear BBR Key |
| xc7k325t_0 Name: xc7k325t_0 | | Program BBR Key Clear BBR Key Program eFUSE Registers |
| xc7k325t_0 Name: xc7k325t_0 Part xc7k325t | | Program BBR Key Clear BBR Key Program eFUSE Registers |
| xc7k325t_0 Name: xc7k325t_0 Part: xc7k325t ID code: 33651093 | | Program BBR Key Clear BBR Key Program eFUSE Registers |
| xc7k325t_0 Name: xc7k325t_0 Part xc7k325t ID code: 33651093 IR length: 6 | | Program BBR Key Clear BBR Key Program eFUSE Registers |
| xc7k325t_0 Name: xc7k325t_0 Part xc7k325t ID code: 33651093 IR length: 6 Status: Programmed | | Program BBR Key Clear BBR Key Program eFUSE Registers Export to Spreadsheet |
| xc7k325t_0 Name: xc7k325t_0 Part: xc7k325t ID code: 33651093 IR length: 6 | | Program BBR Key Clear BBR Key Program eFUSE Registers Export to Spreadsheet |
| xc7k325t_0 Name: xc7k325t_0 Part xc7k325t ID code: 33651093 IR length: 6 Status: Programmed | | Program BBR Key Clear BBR Key Program eFUSE Registers Export to Spreadsheet |

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



| | O MOC (scholes), () | |
|--|--|---------|
| l localitest (1) C | A CONTRACT OF A | 7 . 0 . |
| Ber wink, settingheiterstatusstatu, ito en en hatas (2011) P @ subject (System Menter) | Press RFC Press RFC | |
| stern Munitor Core Properties 🦷 🗧 🕳 | | -1995 |
| arne XADC landa: localhoat3123/cillio_jstDigilan8210 | | - 0.992 |
| | я | 0.99 |
| | | 3 999 |

Step 5: Measuring the Hardware Power Rails

- Pile D AC01245 @ Add ton 524 - Rol at Configure Curify Taut us term carite 1 other cu Phone that covine 3 conto can he UC24348 0 534 - Rod #1 10417 Voltage Setpoint, Plarges, and Lincks Auto vetter ani dence charge 3.00 1.600 💮 # 2.50 1.160 101 1 18.0 4 16 14E 17 1.100 33 4 10.1 (2) % Store 542172 Flash 14 2 1 1.050 2 1 1.400 HH N 0.950 (i) X 64 11 1 N HE WAR Capy 0.880 (1) v 15.5 E. N 0.870 🗄 v 0.1 1 1 1 Soft Stop (i) man 1 100 E 100 E 184日% 242 (0.52) ON ON B & 0001048-0-555-3 1,300 2,470 2,905 6.0 5.0 40 10 YOU CH & & MOLY, COL 0.450 () v -15.4 22 % 2.5 1.0 VOL (01 & 5.005 V, VOL OF 1010 0 0100 2.508 1.780 1.410 5.0 1071240 (\$ 531 2.508 2.290 2.025 44 8.0 1.0 14 YON ON # 5 000 7, 101 OF 0×00[~ UCDR248 @ 536 UCDR248 @ 53d 1.500 1.350 1.275 -5.0 8.0 1.0 YON ON @ 5.000 5.8 5.0 1.0 VINJIM & 5.000 V, VINJIM 13.2 (m) 44 0.0 [] == tumoth 1001246-@ 836 1.209 1.080 1.030 54 5.0 LO VEN ON BELOOF A 17H OFF 4.3 SOE Mal Pall Tena 1.0 0 00 0001248-0 546 3.005 1.000 1,200 1.0 2.0 10 YOL ON & 6.000 V. 101 OF Configu pe & Heits Design ION_DELAY 13 12 58 634 Monito State E 1000 08 rer Designer v1.8.325 [2013-05-07] UCD9248 Fansware v5.8.0.13400 @ Address 520 USB Adapter v1.0.14 (PEC, 400 kH de Reseal
- 1. In the Fusion Digital Power Designer, select a rail in the Configure view and click **Monitor**.

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



| Fre Device Loois | Halp | | | | UCD5248 @ Address 53d | -1al #1 |
|--|--|---|--------|-------|-----------------------|----------------|
| Ionitar | Readings - Half #1 | A Post - Ostpot Power | | | | |
| unyhtie Plotz: 90. In 91. Vinet. 192. Visit Visitati 1947. Visit Visitati 1947. Visit Visitati 1947. Visit Visitati 1947. Visitati 1947. Visitati 1944. Visit | Vite 12.300 V 316 V(4, A) Pain V(4, W) Miniski: V Vesto: 1.203 V Painto: V Delay Cycle: 1.00 W Delay Cycle: 1.00 S Paint Speect: - 00H | 1.40 1.40 1.20 1.01 0.40 0.40 | | | | Lauw |
| Soliti Health 200(3) Sol Mart & Post Liniteditors Show Natur Labols or Pote | Phone 41 L 20 A 51 °C Status Heusshers/Laws Vout 41: 0K Dout 41: 0K Train 41: 0K Train 41: 0K Train 41: 0K | e 0.00 0.00 excao 0009249 InternetTemperature | 09/42 | BER | Juže | |
| Ang Kater, KD (2) MMM () Stas Fo ling Tuel funkward | DPL: OK Mac OK LoperTrk #1. HT: OK SH5425TF Rel.Assorted Torr rol: Clear Lagge Trails | 30 45 | | | | ur ur |
|) Configure | Castrolius (058) © Hgh O Lav | 10 | | | | |
| Monitor | Operation - Rail #1 | | | | | and the second |
| Status | Constant of Ondersteining | - 0 - 04:30 | 179-40 | 10-00 | 10:30 | 105 |
| | | | | | | |

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

| | | | KC70 | 5 (UCD9248) | |
|--------|------|------------|---------|---------------|-----------------|
| device | rail | purpose | voltage | IOUT_CAL_GAIN | IOUT_CAL_OFFSET |
| | 1 | VCCINT | 1.0V | 0xEBDC | 0x8000 |
| 52 | 2 | VCCAUX | 1.8V | 0xEBDC | 0x8000 |
| 52 | 3 | VCC3V3 | 3.3V | 0xEBDC | 0x8000 |
| | 4 | VADJ | 2.5V | 0xEBDC | 0x8000 |
| | 1 | VCC2V5 | 2.5V | 0xEBDC | 0x8000 |
| - | 2 | VCC1V5 | 1.5V | 0xEBDC | 0x8000 |
| 53 | 3 | MGT_AVCC | 1.0V | 0xEBDC | 0x8000 |
| | 4 | MGT_AVTT | 1.2V | 0xEBDC | 0x8000 |
| | 1 | VCCAUX_IO | 1.8V | 0xEBDC | 0x8000 |
| | 2 | VCCBRAM | 1.0V | 0xEBDC | 0x8000 |
| 54 | 3 | MGT_VCCAUX | 1.8V | 0xEBDC | 0x8000 |
| | 4 | N/A | N/A | N/A | N/A |

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



Step 6: Estimating Vectorless Power with Junction Temperature

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

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

```
reset_switching_activity -all
```

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



| Res <u>u</u> lts name: | power_1 | | | | | | | 8 |
|----------------------------------|------------------|-------------|------------|-----|----------------|----------|--|---|
| Environment | Power Supply | Switching | Output | | | | | |
| Device Setting | js | | | | | | | |
| Temp grade: Pro <u>c</u> ess: | | extended v | | | | | | |
| | | maximum 🗸 | | ~ | | | | |
| Environment | Settings | | | | | | | |
| Output Loa | ad: | | 0 | * | pF [0 | - 10000] | | |
| Junctio | on temperature: | | 34 | * | °C | | | |
| Ambient te | mperature: | 25 🗘 | | | *C | | | |
| Effectiv | ve ðJ <u>A</u> : | 1.42 | | | *CAW [0 - 100] | | | |
| Airflow: | | 250 🗸 | | | LFM | | | |
| <u>H</u> eat sink: | | medium (M | edium Prof | ~ | | | | |
| 9SA: | | 2.4 🗘 | | 4 7 | *C/W [0 | - 100] | | |
| Board sele | ection: | medium (10 | | ~ | | | | |
| | f board layers: | 12to15 (121 | o 15 Layer | | | | | |
| alB: | | | 2.5 | | *CW [0 | | | |
| Board tem | perature: | | 25 | * * | *C [-5 | 5 - 100] | | |
| Legend | | | | | | | | |

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



| Q ₹ ♦ " | Summary | | | | | | | |
|--|---|------------------------|--------------------|---|--|--|--|--|
| Settings Summary (1.576 W) Power Supply > Utilization Details Hierarchical (1.01) Clocks (0.024 W) > Signals (0.232 W) Data (0.232 W) Clock Enable (| Power analysis from implemented r derived from constraints files, simul vectoriess analysis. Total On-Chip Power: Junction Temperature: Thermal Margin: Effective &JA: Power supplied to off-chip devices: | | On-Chip Pon 65% | Dynamic 1.017 W (85%) 23% Clocks: 0.024 W (2%) Signals: 0.232 W (23%) Logic: 0.016 W (2%) 61% BRAM: 0.624 W (61%) MMCM: 0.117 W (12%) | | | | |
| Set/Reset (0 W Logic (0.015 W) BRAM (0.624 W) Clock Manager (0: WO (0.004 W) | Confidence level: Launch Power Constraint Advisor to invalid switching activity | Medium find and fix | 35% | 1296 UO: 0.004 W (0%) Device Static: 0.559 W (35%) | | | | |

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

Conclusion

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

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



Lab 5

Measuring Hardware Power Using the KCU105 Evaluation Board

Introduction

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

Step 1: Generating a Bit File from the Implemented Design

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

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



| Bitstrea | am Generation Completed |
|------------|--|
| () Next | Bitstream Generation successfully completed. |
| (| <u>Vi</u> ew Reports |
| (| Open Hardware Manager Generate Memory Configuration File |
| | on't show this dialog again |
| | OK Cancel |

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

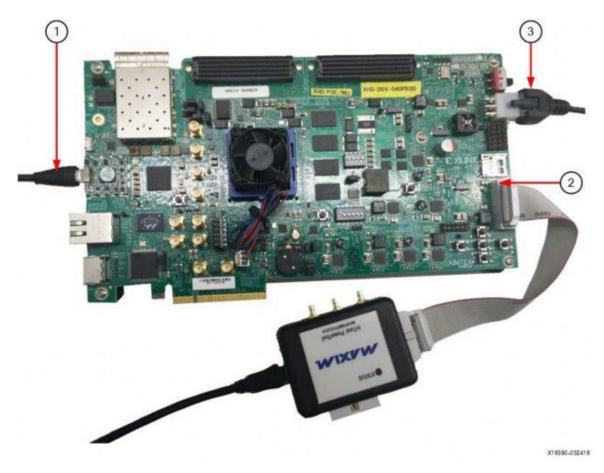
Step 2: Setting up the KCU105 Evaluation Board

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

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







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

Step 3: Configuring the Maxim Digital Power Tool Software

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



| Deshboard | Ov0A | 0x08 | (0x0F | (0e11 | 0e12 | 0614 | 0x15 | 0x16 | Oe16 | 0.16 | |
|---------------------------|----------|--------------|--------------|--------------|----------|--------------|--------------|--------------|--------------|----------|------------------------------------|
| 10000000 | MAX15301 | MAX15308AADO | MAX15303AA00 | MAX15303A400 | MAX15301 | MAX1530BAADO | MAX15303AA00 | MAX15303A400 | MAX15303A400 | MAX15301 | |
| SMBus Address | Ov04 | 80x0 | OvOf | (611 | Ox12 | 0x14 | Oct 5 | 0016 | 0x18 | 0x15 | Output Voltage Plots |
| Product | MAX15301 | MAX15303A400 | MAX15309A400 | MAX15303AA00 | MAX15301 | MAX15303AA00 | MAX15303A400 | MAX15303A400 | MAX15303AA00 | MAX15301 | 0x3A-3MAX16301 |
| Survere Revision | 4018 | 4196 | 4196 | 4196 | 4018 | 4196 | 4196 | 4196 | 4196 | 4018 | 6 0e18:MAX15363 |
| Use Haidware Enable | x | Z | ¥. | 2 | Ø | Z | Z | R | Ø | Z | |
| PMBus Enable | | | | 1.1 | 12 | 3 | | | | | e * |
| Input Voltage (V) | 11.422 | 11.459 | 11.469 | 11.484 | 51.406 | 11.484 | 11.484 | 11.484 | 11.469 | 11.406 | Voort (V) |
| Output Voltage (V) | 0.95 | 1,80 | 0.95 | 1.80 | 1.90 | 1.20 | 1.00 | 1,20 | 1.80 | 3.30 | · |
| Output Current (4) | 0.96 | 0.14 | 0.30 | 0.00 | 0.01 | 0.03 | 0.36 | -0.01 | 0.00 | 0.69 | |
| Switching Frequency (kHz) | 401 | 000 | 600 | 600 | 002 | 801 | 603 | 599 | 600 | 601 | 0 1280 |
| Internal Temperature (*C) | 46 | 49 | 43 | 49 | 45 | 47 | 45 | 42 | 46 | 45 | |
| External Temperature ("C) | 42 | 48 | 43 | 45 | 47 | 44 | 43 | 36 | 4b | 49 | |
| Duty Cycle (%) | 7.43 | 14.41 | 6.85 | 14.06 | 3158 | 8.97 | 7.68 | 8.96 | 1198 | 27.28 | System Sequencing |
| Bratikod | | | | | | | | | | | 11 |
| Power Good | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10 |
| Faults | | | | | | | | | | | E 11 |
| Clear Feults | Clear | Cicar | Clear | (Clear) | Clean | Cicar | Cloar | [Clear] | Clear | Cicer | |
| Vourt Set Point (V) | 11.950 | 1.800 🗘 | 0.990 拿 | 1.800 🗘 | 1.400 🗘 | 1.200 🗘 | 5.000 🗘 | 1.200 😨 | 1.800 | 1.300 🗘 | |
| On Delay Time (ms) | 4 56 | 11.00 | 7.82 | 19.75 | 24.84 🔹 | 24.88 | 10/09 🔹 | 15.22 | 19.75 | 1.00 | |
| Cin Risse Tarse (rrs) | 2.02 | 2.32 🚖 | 1.77 | 2.32 🚖 | 235 🔹 | 1.88 | 1.83 🤶 | 1.85 💼 | 2.34 | 2.13 | |
| Off Delay Time (ms) | 24.97 🚖 | 1437 😩 | 19.88 🚖 | 7.96 🛫 | 511 🗘 | 5.11 | 19,88 🚖 | 14.78 | 10.23 | a) 90.06 | Stæt låp |
| Olf Fall Time (ms) | 2.02 | 2.32 💠 | 1.77 🚖 | 232 🕂 | 235 💠 | 1.88 💠 | 1.83 🔹 | 1.85 🛧 | 234 | 2.13 | Search For Device: Save Contigurat |

Step 4: Programming the Bitstream

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

When the hardware is detected successfully, the part information will be displayed in the Open New Hardware Target dialog box.

4. Verify the part information, then click **Next** and **Finish**.





| elect a hardwar | are Target e target from the , decrease the fr | | | | propriate JTA | G clock (TCK) | frequency. If | you do not see th | ne j |
|-----------------------|--|----------------|-----------------|---------|-----------------------------------|---------------|---------------|-------------------|------|
| Hardware <u>T</u> arg | jets | | | | | | | | |
| Туре | Name | | JTAG Clock Fre | equency | | | | | |
| xilinx_tcf | Digilent/21025 | 1845034 | 15000000 | ~ | | | | | |
| | | | | | | | | | |
| 1.1.1 | ices (for unknov | | specify the Ins | | tual Cable (X) gister (IR) len | | | | |
| Name | ID Code | IR Length | specify the Ins | | | | | | |
| Name xcku040_(| ID Code | IR Length 6 | specify the Ins | | | | | | |

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





| | | _ 🗆 🗆 × | | | | | |
|---|---|--|--|--|--|--|--|
| | 8 | 0 | | | | | |
| ame | | Status | | | | | |
| localhost (1) | | Connected | | | | | |
| v description view of the second s | 0 | Open | | | | | |
| v (i) xcku040_0 (1) | | Not programm | | | | | |
| 😻 SysMon (System Moni | | Hardware Device Properties Ctrl+ | | | | | |
| | | Program Device | | | | | |
| | | Verify Device | | | | | |
| | | Refresh Device | | | | | |
| | | Add Configuration Memory Device | | | | | |
| | | Boot from Configuration Memory Device | | | | | |
| | | | | | | | |
| | | Program BBR Key | | | | | |
| rdware Device Properties | | Clear BBR Key Program eFUSE Registers | | | | | |
| xcku040_0 | | | | | | | |
| Name: xcku040_0 | | Export to Spreadsheet | | | | | |
| Part: xcku040 | | | | | | | |
| D code: 13822093 | | | | | | | |
| R length: 6 | | | | | | | |
| Status: Not programmed | | | | | | | |
| | | | | | | | |
| rogramming file | | | | | | | |
| Programming file: | | | | | | | |
| Probes file: | | | | | | | |

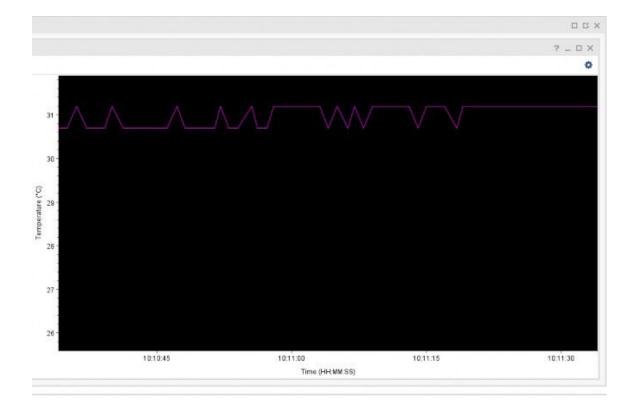
- 6. Select the bit file from the implementation runs directory of the project created in Lab 2 for the UltraScale[™] design (<project_dir>/power_tutorial2/ power_tutorial2.runs/impl_1/ dut_fpga.bit) and click Program.
- 7. After the program completes successfully, select **XADC (System Monitor)** in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



| Hardware | ? | _ 🗆 🖾 🗙 | da | shboard_2 | | | | |
|--|-------|-------------------|----------------|-----------------|-----|----------------|--|--|
| Q ¥ ♦ Ø ▶ 3 | > ■ | 0 | Π | SysMon (xcku040 | 0) | | | |
| Name ✓ ▮ localhost (1) ✓ ■ // xilinx_tcf/Digilent/2102518450 | | Status | | | | | | |
| | | Connected | hboard Options | + - C | | | | |
| | | Open | | Temp 31.2°C | | | | |
| v () xcku040_0 (1) | | Not programm | ooar | | | | | |
| 🦉 SysMon (System I | | tom Hanitar Car | | erties Ctrl+E | i l | | | |
| | Sys | stem Monitor Con | e Prop | cin+E | | | | |
| | C Ref | fresh System Mo | nitor | | | | | |
| | JTA | G Scan Rate | | • | | | | |
| | Da | Dashboard + | | | + | New Dashboard. | | |
| | Exp | oort to Spreadshe | et | | | dashboard_2 | | |
| | · | | | | _ | | | |
| System Monitor Core Properties | ? | _ 0 6 X | | | | | | |
| | | 11 11 22 | | | | | | |
| SysMon | + | • • • | | | | | | |

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





Step 5: Measuring the Hardware Power Rails

| Dashboard | .75) | | | | 6 9 1 |
|------------------------------------|--|-------------------------------|-----------------------------------|------------------------------------|--------|
| | 0x04 0x05 0x0F 0x0F 0x0F MAX15303AA00 | 0x11 MAX15303AA00 MAX15301 | 0x14 MAX15303AA00 MAX15303AA00 | 0x18 0x18 0x18 0x18 0x18 0x18 0x18 | |
| onfiguration Monitor | Faults Set PMBus Command Permissions Tools - | | | | |
| Basic Settings | | | Device States | | |
| | | Vout (V) 0.956 | | Output Voltage (V) | 0.950 |
| | Vout Margin | | | Input Voltage (V) | 11.875 |
| | Vout Margin | | | Output Current (A) | 3.18 |
| | Switching Frequency Set Por | | · 🖗 | Switching Frequency (kHz) | 400 |
| | Over Current I | | | Internal Temperature (*C) | 49 |
| | Power Coo | | | Esternal Temperature (*C) | 42 |
| | Power Good | 2 Off (V) 0.80 | | Duty Cycle (70) | 7.54 |
| Startup/Shutdown | limes | | | Status | 0 |
| | On Delay Tir | | |) | |
| | On Rise Tir | me (ms) 2.02 | ÷ | EXT_TEMP_CAL m | 1.0039 |
| | Off Delay Tir | | | EXT_TEMP_CAL-b | 0.00 |
| | OH Fall Tir | me (ms) 2.02 | | | |
| Operation Mode | | | | IDUT_CAL_GAIN (#D) | 0.550 |
| Margin High, Ignore Fau | | ct on laults | | IOUT_CAL_OFPSET (A) | 1,199 |
| Normal Margin Low, Ignore Fault | ta 💮 Margin Low, As | t on Faults | Advanced Configuration | | |
| Soft Stop | 💮 Tristate Stop | | | Advanced Configuration | |

tion. 1. **D**: : 1

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



| Dashboard | 0x0A MAX15301 | 0x08 MAX15303AA00 | 0x0F MAX15303A40 | 0x11 MAX15303AA00 | 0x12 MAX15301 | 0x14 M4x15303440 | 0v15 MAX15303AA0 | 0.10 MAX1 | 5303AA00 | 0v18 MAX153034400 | 0x18 MAX15301 | | |
|---------------------|-----------------------|----------------------|---------------------|----------------------|------------------|---------------------|---------------------|----------------------|----------|----------------------|--------------------|---------------|------------|
| Infiguration Monito | and the second second | Bus Command Per | missions Tools | | | | | | | | | | |
| avameters | | | | ddress 0x04 | | | | | | | | | |
| Configur | ation | | | Vout (V) | Min: 0.950 Ma | x: 0.950 Avg: 0.950 | | | | VIN (V): Min | : 11.875 Max: 11. | 875 Avg: 11.8 | 15 |
| | On/Off Config | 0v16 | 2.0 | - YOUT | ····· YOUT | FALLTLINET | | | - | VIN . | ····· VIN FAULT LI | WIT | |
| Vo | e Command (V) | 0.950 | | | | | | | 18 | | | | |
| | Vout Max (V) | 1.045 | 1.5 | | | | | | 12 | | | 600 (V000000) | |
| | lout Cal Gain | 0.55 | W 10 | | | | | MnIM | 8 | | | | |
| Parame | ters | | - | | | | | 5 | 6 | | | | |
| | Input Voltage (V) | 11.875 | 0.5 | | | | | | * | | | | |
| 0 | utput Voltage (V) | 0.950 | 0.0 | 900 | 102 | | 1000 | | 0 | 1000 | | | 333 |
| 0 | Natput Current (A) | 1.20 | | 900 | 910 Sar | 920 npie | 933 | 949 | 990 | 900 | 510 Sample | 920 | 333 |
| Switching | Frequency (kHz) | 401 | | lout (A): | Min: 3.203 Max | c: 3.344 Avg: 3.262 | | | Inte | emai Temperature | (C): Min: 48.563 | Max: 48.625 / | Wg: 48.608 |
| Internal | Temperature (°C) | 48 | | - XOUT | 1007 5 | AULTIUMIT | | | - | TEMPERATURE | TEMP FAU | LTUNT | |
| Esternal | Temperature (°C) | 43 | 43 | | | | | . 0 | 120 | | | | |
| | Duty Cycle (%) | 7.53 | | | | | | Internal temperature | 193 | | | | |
| Statu | •)- | | 15 IP00 27 | | | | | uber | 00 | | | | |
| | Fault | Warring | ₫ 20 - | | | | | alter | 60 | | | | |
| Vout Un | ider Voltage 🛛 🔘 | 0 | 10 | | | | | ferm | 40 | | | | |
| lout C | Iver Current 🛛 🔾 | 0 | | | | | | - 1 | 20 | | | | |
| Vin C | Ver Voltage 🛛 🔘 | 0 | 0 | 800 | 912 | 920 | 830 8 | 40 | 8 892 | 800 | 610 | 920 | 850 |
| Vin Un | ider Voltage 🛛 | 0 | | | San | nple | | | | | Sample | dense. | |
| Vout C | Ver Voltage 🛛 🔘 | 0 | | | | | 1 | top Plottin | 1 | | | | |

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

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

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

Table 2: Rail Information

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



Step 6: Estimating the Vectorless Power with Junction Temperature

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

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

```
reset_switching_activity -all
```

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



| Res <u>u</u> lts name: power_1 | 8 |
|--------------------------------|--------------------------|
| Environment Power Supply | Switching Output |
| Device Settings | |
| Temp grade: | extended 🗸 |
| Pro <u>c</u> ess: | maximum |
| Environment Settings | |
| Output Load: | 0 🗘 pF [0 - 10000] |
| ✓ Junction temperature: | 34 C |
| Ambient temperature: | 25 ‡ °C |
| Effective ϑJ <u>A</u> : | 1.42 °C/W [0 - 100] |
| A <u>i</u> rflow: | 250 🗸 LFM |
| <u>H</u> eat sink: | medium (Medium Prof 🛩 |
| J SA: | 2.4 🗘 °C/W [0 - 100] |
| Board selection: | medium (10"x10") 🗸 |
| Number of board layers: | 12to15 (12 to 15 Layer 🗸 |
| ୬ JB: | 2.5 🗘 °C/W [0 - 100] |
| Board temperature: | 25 🗘 °C [-55 - 100] |
| Legend | |
| User Defined 🗌 Calc | ulated 📃 Default |

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

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



| Settings Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorises analysis. On-Chip Power * Utilization Details Total On-Chip Power: 3.063 W * Utilization Details Total On-Chip Power: 3.063 W * Utilization Details Total On-Chip Power: 40.0 °C * Utilization Details Total On-Chip Power: 40.0 °C * Utilization Details Total On-Chip Power: 40.0 °C * Utilization Details Utilization Chip Power: 60% * Utilization Details Utilization Chip Power: 60% * Utilization Details Utilization Chip Power: 60% * Signals (0.385 W) Effective 3JA 1.4 °C/W * Dower supplied to off-chip devices: 0 W * Settreset (0 V Confidence level: Medium | • • • • » | | | |
|---|---|--|-------------|--|
| Summary (3.063 W) Power analysis from implemented netitist, Activity vectoriess analysis. On L hip Power > Utilization Details Total On-Chip Power: 3.063 W > Utilization Details Total On-Chip Power: 3.063 W > Unitication Details Total On-Chip Power: 3.063 W > Signals (0.385 W) Diffective 8JA 1.4 "C/W > Data (0.385 W) Effective 8JA 1.4 "C/W Power supplied to off-chip devices: 0.W SettReset(0 V) Confidence level: | Q ₹ ♦ C ″ | Summary | | |
| Utilization Defaults Total On-Chip Power: 3.063 W 60% 6% 6% 6% 1 Clocks: 0.114 W (8%) Clocks: 0.114 Within Temperature: 40.0 °C 60% 5% 21% 5% 5igmals: 0.385 W 21% 5% 5% 21% 5% 5% 21% 5% | | derived from constraints files, simulation files or | On-Chip Pov | |
| Hierarchical (185 Clocks (0.114 W) Total On-Chip Power: 3.063 W 60% 21% Clocks: 0.114 W (8%) Signata (0.385 W) Thermal Margin: 60.0 °C (39.4 W) 5% Signata: 0.385 W Logic: 0.089 W (5%) Data (0.385 W) Effective 3J.A 1.4 °C/W 60 °C (39.4 W) Effective 3J.A 1.4 °C/W 60% Effective 3J.A 1.12 W (5%) Clock Enable Power supplied to off-chip devices: 0 W IMMCM: 0.122 W (7%) SettReset(0 V) Confidence level: Medium 40% 7% 10C: 0.04 W (7%) | Power Supply | vectoriess analysis. | | Dynamic: 1.850 W (60%) |
| SetReset (0 V Confidence level: Medium 40% 7% UC: 0.024 V((%) | Clocks (0.114 W) ~ Signals (0.385 W) | Junction Temperature: 40.0 °C Thermal Margin: 60.0 °C (39.4 W) Effective 3JA: 1.4 °C/W | 60% | 21% Clocks: 0.114 W (6%) 5% Signats: 0.385 W (21%) Logic: 0.089 W (5%) |
| Logic (0.889 W) Launch Power Constraint Advisor to find and fix BRAM (1.132 W) Invalid switching activity Device Static: 1.213 W (40%) | Set/Reset (0 V Logic (0.089 W) | Confidence level: Medium Launch Power Constraint Advisor to find and fix | 40% | 7% UO: 0.004 W (0%) |

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

Conclusion

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

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



Lab 6

Performing Power Optimization

Introduction

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

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

Step 1: Setting Up Options to Run Power Optimization

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

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

• In the Power Opt Design settings, check the is_enabled box.

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



| Q | Implementation | | | |
|--|---|------------------------|---------------------------------------|------------|
| Project Settings | Specify various settings ass | sociated to Implement | ation | |
| General | ****** | | | |
| Simulation | Constraints | | | |
| Elaboration Synthesis | Default constraint set: | active |) | ~ |
| Implementation | Options | | | |
| Bitstream | Options | g== ; | | |
| > IP | is_enabled | | 1 | ^ |
| Tool Settings | tcl.pre | | | |
| Project | tcl.post | | | *** |
| IP Defaults | -verbose | | | |
| Source File | -directive* | | NoBramPowerOpt | ~ |
| Display | More Options | | | |
| WebTalk | Y Power Opt Design (po | wer_opt_design) | | |
| Help | is_enabled | | v | |
| > Text Editor | tcl.pre | | | |
| 3rd Party Simulato | tcl.post | | | |
| > Colors | More Options | | | |
| Selection Rules | | | | ~ |
| Shortcuts Strategies > Window Behavior | is_enabled Optionally run this step as | part of the flow. This | step optimizes design to maximize pow | er saving. |
| window Benavior 🧹 | | | | |

- 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?.

| I_1' have been | modified. ply these changes | to a new run? |
|----------------|--------------------------------|---|
| Yes | No | Cancel |
| | impl_1' and ap | N_1' have been modified. impl_1' and apply these changes <u>Y</u> es <u>N</u> o |

- 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.



| i Save Project |
|---|
| Save project before launching implementation? |
| Data to Save |
| Synthesized Design - constrs_1 - dut_fpga_kc705.xdc |
| Implemented Design - constrs_1 - dut_fpga_kc705.xdc |
| Save Don't Save Cancel |
| |

You are running Implementation with Power Optimization turned on.

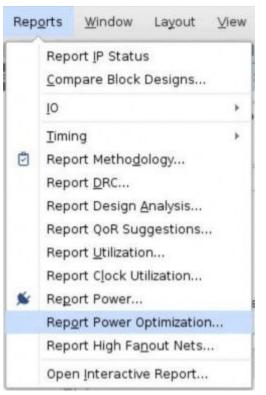
8. In the Implementation Completed dialog box, select **Open Implemented Design** and click **OK**. Click **Don't Save** when the Save Project window pops up to save both Synthesis and Implementation constraints.

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** \rightarrow **Report Power Optimization**.







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

| Report Power C | Optimization | |
|-----------------------|--------------------|---------|
| Report power optimi | zation. | 1 |
| Results <u>n</u> ame: | power_opt_1 | 0 |
| Export to file: | Output file format | т 🔿 хмі |
| 🗹 Open in a new | | |
| ? | ОК | Cancel |

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

```
report_power_opt -name power_opt_1
```



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

| Create | New Run | | | × |
|--------|--|-------------|----|-----------------|
| 2 | Properties for the completed Do you want to preserve the s on't show this dialog again | | | s to a new run? |
| | | <u>Y</u> es | No | Cancel |

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

Step 3: Running report_power to Examine Power Savings

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

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



| ower | | | ? _ 🗆 ?) |
|--|---|------------|--|
| Q <u>∓</u> ♦ C " | Summary | | |
| Settings Summary (1.48 W) Power Supply | Power analysis from implemented netlist. Activity derived from constraints files, simulation files or vectoriess analysis. | On-Chip Po | Dynamic: 1.017 W (59%) |
| Utilization Details Hierarchical (1.017 Clocks (0.024 W) Signals (0.232 W) Data (0.232 W Clock Enable (| Total On-Chip Power: 1.48 W Junction Temperature: 27.6 °C Thermal Margin: 57.4 °C (30.2 W) Effective & JA: 1.8 °C/W Power supplied to off-chip devices: 0 W | 69% | 23% Cłocka: 0.024 W (2%) Signala: 0.232 W (2%) Logic: 0.016 W (2%) BRAM: 0.524 W (51%) MMCM: 0.117 W (12%) |
| SetReset (0 V) Logic (0.016 W) BRAM (0.624 W) Clock Manager (0 | Confidence level: Medium Launch Power Constraint Advisor to find and fix invalid switching activity | 31% | 12% INO: 0.004 W (0%) Device Static: 0.463 W (31%) |

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

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

IMPORTANT! Power optimization works to minimize the impact on timing while maximizing power savings. However, in certain cases, if timing degrades after power optimization, you can identify and apply power optimizations only on non-timing critical clock domains or modules using the *set_power_opt* XDC command.

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

Assume that this block RAM is in the critical path:

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

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

1. In the Tcl Console, type this command:

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

This will prevent the tool from gating this block RAM.

- 2. From the Flow Navigator choose Run Implementation, which in turn reruns power_opt_design.
- 3. Click **Save** in the Save Project dialog box to save the synthesized design and implemented design constraints before launching implementation.



| 👃 Sav | e Project |
|---------|---|
| ? | Save project before launching implementation? |
| Data to |) Save |
| 9 | Synthesized Design - synth_1 - constrs_1 - dut_fpga_kc705.xdc |
| 5 | Implemented Design - impl_1 - constrs_1 - dut_fpga_kc705.xdc |
| | Implemented Design - impl_2 - constrs_1 - dut_fpga_kc705.xdc |
| | |
| | Save Don't Save Cancel |
| | |

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

| E Sav | e Constraints Conflict | × |
|-------|---|------|
| Chang | There are unsaved changes to constraints set 'constrs_1' in multiple editors. Please choose which changes to save. es to Save | |
| | Synthesized Design - synth_1 Implemented Design - impl_1 Implemented Design - impl_2 | |
| | OK Car | icel |

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** \rightarrow **Report Power Optimization**.
- 2. In the Report Power Optimization dialog box, type in the Results name as **power_opt_2**. Alternatively, execute this Tcl command in the Tcl Console:

report_power_opt -name power_opt_2



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

| Power Opt | | | | 2 - 0 2 3 |
|---|------------------------|---------|---|-----------|
| QIEC | Q. Tool Galed BRAMs | | | |
| General Information | Cell Name | CE Port | CE Net Name | |
| Summary | mem_reg_0_0 (RAME36E1) | ENARDEN | dutigen_dut(0) bram_top_inst/bram_inst/mem_reg_0_0_ENARDEN_cocolgate_en_sig_2 | |
| Recommendations | mem_reg_0_0 (RAME36E1) | ENEWREN | duligen_dul(0).bram_top_inst/bram_inst/mem_reg_0_0_ENEWREN_cooolgate_en_sig_11 | |
| Hierarchical Information | mem_reg_0_1 (RANB36E1) | ENARDEN | dutigen_dut(0) bram_top_inst/bram_inst/mem_reg_0_1_ENARDEN_cooolgate_en_s/g_3 | |
| ~ BRAMs | mem_reg_0_1 (RAME38E1) | ENDWREN | dutigen_dut(0) bram_top_inst/bram_inst/mem_reg_0_1_ENBWREN_cooodgate_en_sig_12 | |
| User Gated BRAMs | mem_reg_0_2 (RANB36E1) | ENARDEN | dutigen_dut(0j.bram_top_inst/bram_inst/mem_reg_0_2_ENARDEN_cooolgate_en_sig_4 | |
| Tool Gated BRAMs | mem_reg_0_2 (RANE36E1) | ENEWREN | dutigen_dutio] bram_top_inst/bram_inst/mem_reg_0_2_ENEWREN_coopolgate_en_sig_13 | |
| BRAM WRITE_MODE 0 | mem_reg_0_3 (RAME36E1) | ENARDEN | duligen_dul(0).bram_top_inst/bram_inst/mem_reg_0_3_ENARDEN_cooolgate_en_sig_5 | |
| SRLs | mem_reg_0_3 (RANB36E1) | ENBWREN | dutigen_dut(0) bram_top_inst/bram_inst/mem_reg_0_3_ENBWREN_cookolgate_en_sig_14 | |
| Slice Registers | mem_reg_0_4 (RANE36E1) | ENARDEN | dullgen_dul[0].bram_lop_inst/bram_inst/mem_reg_0_4_ENARDEN_cooolgate_en_sig_6 | |
| User Gated Slice Regis | mem_reg_0_4 (RAWB36E1) | ENEWREN | dutigen_dut(0) bram_top_inst/bram_inst/mem_reg_0_4_ENBWREN_cooolgate_en_sig_15 | |
| Tool Gated Slice Regist | mem_reg_0_5 (RAMB36E1) | ENARDEN | dubgen_dut(0) bram_top_inst/bram_inst/mem_reg_0_5_ENARDEN_cocolgale_en_sig_7 | |
| XPM URAMS | mem_reg_0_5 (RAWB36E1) | ENEWREN | dutigen_dut[0].bram_top_inst/bram_inst/mem_reg_0_5_ENEWREN_coopligate_en_sig_16 | |
| | mem_reg_0_6 (RANB36E1) | ENARDEN | dutigen_dut(0) bram_top_inst/bram_inst/mem_reg_0_6_ENARDEN_cocolgale_en_sig_8 | |
| 1 mart | mem reg 0 6 (RAME36E1) | ENEWREN | dutioen dutiol/bram top inst/bram inst/mem reg 0 6 ENEWREN coopligate en sto 17 | |

Note that this block RAM is no longer in the list of Tool Gated BRAMs: ${\tt dut}\,/$

gen_dut[0].bram_top_inst/bram_inst/mem_reg_0_0

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

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

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

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

5. Select Run Implementation in the Synthesis Completed dialog box and click OK.



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

| | ? _ 0 |
|--|---|
| Q X \$ C | Summary |
| Settings Bummary (2,348 W) Power Supply > Utilization Details Hierarchical (1.85 W) Clocks (0.114 W) > Signals (0.385 W) Data (0.385 W) Clock Enable (0.W) SetReset (0.W) Logic (0.095 W) BRAM (1.132 W) Clock Manager (0.125 W, WO (0.004 W) | Power analysis from implemented netlist. Addity derived from constraints files, simulation files or vectories analysis. On-Chip Power Total On Chip Power: 2.349 W Junction Temperature: 28.3 °C Thermal Margin. 71.7 °C (49.1 W) Effective 5JA: 1.4 °CW Power supplied to 6ff-chip devices: 0 W Condence levet Medium Launch Power Constraint Advisor to find and fix invalid switching advity Device Static 0.499 W (21%) |

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

| Power | | | | | | | | ? - 0 7 X |
|--|--|-------------------------------|------------|-------------|----------|-----------|----------|-----------------|
| Q ž ≑ C | Q 😤 Hierarchical | | | | | | | |
| Settings | Utilization | Name | Clocks (W) | Signals (W) | Data (W) | Logic (W) | ERAM (W) | Clock Manager (|
| Summary (2.352 W) | III 1.053 W (79% of total) | 🗿 dut_toga | | | | | | |
| Power Supply | I.719 W (73% of total) | 🗃 dut (dut) | 0.105 | 0.393 | 0.393 | 0.089 | 1.132 | ≺0.0 |
| Utilization Details | > 1.152 W (49% of total) | Roncascade_bram (Noncascade | 0.052 | 0.173 | 0.173 | 0.051 | 0.875 | <0.0 |
| Hierarchical (1 853 W) | > 0.567 W (24% of total) | Cascaded_bram (Cascaded_bram) | 0.052 | 0.22 | 0.22 | 0.038 | 0.257 | ~0.0 |
| Clocks (0.11 W) ~ Signals (0.383 W) Data (0.393 W) Clock Enable (0 W) SetReset (0 W) Logic (0.089 W) BRAM (1.132 W) Clock Manager (0.125 W) WO (0.004 W) | 0.131 W (5% of total) | 🖾 Leaf Cells (12) | | | | | | |
| | > 10.003 W (<1% of total) | 📴 in_diff_bufg (IBUFDS) | <0.001 | <0.001 | -0.001 | ~0.001 | <0.001 | <0.0 |
| | | | | | | | | |

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

Conclusion

In this tutorial, we have accomplished the following:

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



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





Appendix A

Additional Resources and Legal Notices

Xilinx Resources

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

Documentation Navigator and Design Hubs

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

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

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

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

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

References

These documents provide supplemental material useful with this guide:



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

Please Read: Important Legal Notices

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

AUTOMOTIVE APPLICATIONS DISCLAIMER

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



Copyright

© Copyright 2012-2021 Xilinx, Inc. Xilinx, the Xilinx logo, Alveo, Artix, Kintex, Spartan, Versal, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.

