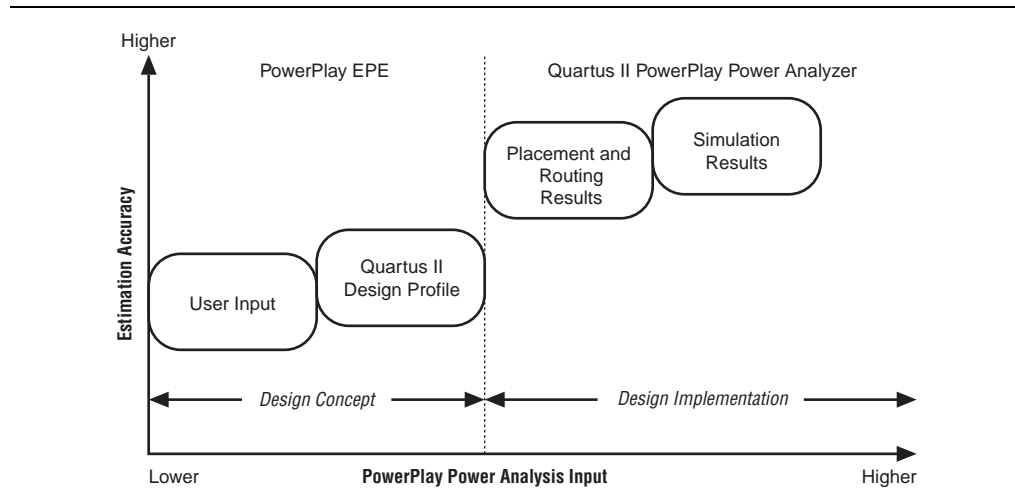


This chapter describes how to use the Altera® Quartus® II PowerPlay Power Analysis tools to accurately estimate device power consumption.

As designs grow larger and process technology continues to shrink, power becomes an increasingly important design consideration. When designing a PCB, the power consumed by a device must be accurately estimated to develop an appropriate power budget and to design the power supplies, voltage regulators, heat sink, and cooling system. As shown in [Figure 8–1](#), the PowerPlay Power Analysis tools provide the ability to estimate power consumption from early design concept through design implementation.

Figure 8–1. PowerPlay Power Analysis



For more information about the PowerPlay suite of power analysis and optimizations tools, refer to [About Power Estimation and Analysis](#) in Quartus II Help. For more information about acquiring the PowerPlay EPE spreadsheet, refer to [PowerPlay Early Power Estimators \(EPE\) and Power Analyzer](#) on the Altera website.

This chapter discusses the following topics:

- [“Types of Power Analyses”](#) on page 8–2
- [“Factors Affecting Power Consumption”](#) on page 8–2
- [“Creating PowerPlay EPE Spreadsheets”](#) on page 8–5
- [“PowerPlay Power Analyzer Flow”](#) on page 8–8
- [“Using Simulation Files in Modular Design Flows”](#) on page 8–11

© 2011 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera’s standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.



- “Using the PowerPlay Power Analyzer” on page 8–17
- “Conclusion” on page 8–25

Types of Power Analyses

Understanding the uses of power analysis and the factors affecting power consumption helps you to use the PowerPlay Power Analyzer effectively. Power analysis meets two significant planning requirements:

- **Thermal planning**—The cooling solution must be sufficient to dissipate the heat generated by the device. The computed junction temperature must fall within normal device specifications.
- **Power supply planning**—Power supplies must provide adequate current to support device operation.

The two types of analyses are closely related because much of the power supplied to the device is dissipated as heat from the device; however, in some situations, the two types of analyses are not identical. For example, if you are using terminated I/O standards, some of the power drawn from the power supply of the device dissipates in termination resistors rather than in the device.

Power analysis also addresses the activity of your design over time as a factor that impacts the power consumption of the device. Static power is the power consumed regardless of design activity. Dynamic power is the additional power consumed due to signal activity or toggling.



For power supply planning, you can use the PowerPlay EPE at the early stages of your design cycle, or use the PowerPlay Power Analyzer reports when your design is complete to get an estimate of your design power requirement.

Factors Affecting Power Consumption

This section describes the factors affecting power consumption. Understanding these factors allows you to use the PowerPlay Power Analyzer and interpret its results effectively.

Device Selection

Different device families have different power characteristics. Many parameters affect the device family power consumption, including choice of process technology, supply voltage, electrical design, and device architecture. For example, the Cyclone II device family architecture consumes less static power than the high-performance and full-featured Stratix II device family.

Power consumption also varies in a single device family. A larger device consumes more static power than a smaller device in the same family because of its larger transistor count. Dynamic power can also increase with device size in devices that employ global routing architectures, for example, the MAX device family. Cyclone, MAX II, and Stratix devices do not exhibit significantly increased dynamic power as device size increases.

The choice of device package also affects the ability of the device to dissipate heat. This choice can impact your cooling solution choice required to meet junction temperature constraints.

Process variation can affect power consumption. Process variation primarily impacts static power because sub-threshold leakage current varies exponentially with changes in transistor threshold voltage. As a result, it is critical to consult device specifications for static power and not rely on empirical observation. Process variation has a weak effect on dynamic power.

Environmental Conditions

Operating temperature primarily affects device static power consumption. Higher junction temperatures result in higher static power consumption. The device thermal power and cooling solution that you use must result in the device junction temperature remaining within the maximum operating range for the device. The main environmental parameters affecting junction temperature are the cooling solution and ambient temperature.

Airflow

Airflow is a measure of how quickly heated air is removed from the vicinity of the device and replaced by air at ambient temperature. Airflow can either be specified as “still air” when no fan is used, or as the linear feet per minute rating of the fan used in the system. Higher airflow decreases thermal resistance.

Heat Sink and Thermal Compound

A heat sink allows more efficient heat transfer from the device to the surrounding area because of its large surface area exposed to the air. The thermal compound that interfaces the heat sink to the device also influences the rate of heat dissipation. The case-to-ambient thermal resistance (θ_{CA}) parameter describes the cooling capacity of the heat sink and thermal compound employed at a given airflow. Larger heat sinks and more effective thermal compounds reduce θ_{CA} .

Junction Temperature

The junction temperature of a device is equal to:

$$T_{\text{Junction}} = T_{\text{Ambient}} + P_{\text{Thermal}} \cdot \theta_{JA}$$

in which θ_{JA} is the total thermal resistance from the device transistors to the environment, having units of degrees Celsius per watt. The value θ_{JA} is equal to the sum of the junction-to-case (package) thermal resistance (θ_{JC}) and the case-to-ambient thermal resistance (θ_{CA}) of your cooling solution.

Board Thermal Model

The thermal resistance of the path through the board is referred to as the junction-to-board thermal resistance (θ_{JB}), having units of degrees Celsius per watt. It is used in conjunction with the board temperature, as well as the top-of-chip θ_{JA} and ambient temperatures, to compute junction temperature.

Device Resource Usage

The number and types of device resources used greatly affects power consumption.

Number, Type, and Loading of I/O Pins

Output pins drive off-chip components, resulting in high-load capacitance that leads to a high-dynamic power per transition. Terminated I/O standards require external resistors that generally draw constant (static) power from the output pin.

Number and Type of Logic Elements, Multiplier Elements, and RAM Blocks

A design with more logic elements (LEs), multiplier elements, and memory blocks tends to consume more power than a design with fewer circuit elements. The operating mode of each circuit element also affects its power consumption. For example, a DSP block performing 18×18 multiplications and a DSP block performing multiply-accumulate operations consume different amounts of dynamic power because of different amounts of internal capacitance being charged on each transition. The operating mode of a circuit element also affects static power.

Number and Type of Global Signals

Global signal networks span large portions of the device and have high capacitance, resulting in significant dynamic power consumption. The type of global signal is important as well. For example, Stratix II devices support several kinds of global clock networks that span either the entire device or a specific portion of the device (a regional clock network covers a quarter of the device). Clock networks that span smaller regions have lower capacitance and tend to consume less power. The location of the logic array blocks (LABs) driven by the clock network can also have an impact because the Quartus II software automatically disables unused branches of a clock.

Signal Activities


The final important factor in estimating power consumption is the behavior of each signal in your design. The two vital statistics are the toggle rate and the static probability.

The toggle rate of a signal is the average number of times that the signal changes value per unit of time. The units for toggle rate are transitions per second and a transition is a change from 1 to 0, or 0 to 1.

The static probability of a signal is the fraction of time that the signal is logic 1 during the period of device operation that is being analyzed. Static probability ranges from 0 (always at ground) to 1 (always at logic-high).

Dynamic power increases linearly with the toggle rate as the capacitive load is charged more frequently for logic and routing. The Quartus II software models full rail-to-rail switching. For high toggle rates, especially on circuit output I/O pins, the circuit can transition before fully charging the downstream capacitance. The result is a slightly conservative prediction of power by the PowerPlay Power Analyzer.

The static power consumed by both routing and logic can sometimes be affected by the static probabilities of their input signals. This effect is due to state-dependent leakage and has a larger effect on smaller process geometries. The Quartus II software models this effect on devices at 90 nm (or smaller) if it is important to the power estimate. The static power also varies with the static probability of a logic 1 or 0 on the I/O pin when output I/O standards drive termination resistors.

-  To get accurate results from the power analysis, the signal activities for analysis must represent the actual operating behavior of your design. Inaccurate signal toggle rate data is the largest source of power estimation error.

Creating PowerPlay EPE Spreadsheets

You can use PowerPlay EPE spreadsheets to perform a preliminary thermal analysis and power consumption estimate for your design. You can either enter the data manually or use the tools in the Quartus II software to assist you with generating the device resources usage information for your design.

-  For more information about generating a PowerPlay EPE File in the Quartus II software, refer to *Performing an Early Power Estimate Using the PowerPlay Early Power Estimator* in Quartus II Help.

Figure 8-2 shows an example of the contents of a PowerPlay EPE File generated for a design that targets a Stratix III device.

Figure 8-2. Example of a PowerPlay EPE File

	A	B	C	D	E	F	G	H
1	EARLY_POWER_ESTIMATOR_FILE_FORMAT_VERSION		6					
2	QUARTUS_II_VERSION		9.0 Build 235 06/17/2009 SP 2 SJ Full Version					
3	PROJECT		fractal					
4	REVISION		fractal_extra					
5	PROJECT_FILE		C:/Powe_Lab_and_Test_designs/fractal.qpf					
6	TIME		Fri Aug 14 11:45:17 2009					
7	TIME_SECONDS		1250275517					
8	FAMILY		Stratix III					
9	DEVICE		EP3SE260					
10	PACKAGE		FBGA					
11	PART		EP3SE260H780C4					
12	POWER_USE_DEVICE_CHARACTERISTICS		TYPICAL					
13	POWER_AUTO_COMPUTE_TJ		ON					
14	POWER_TJ_VALUE		25					
15	POWER_USE_CUSTOM_COOLING_SOLUTION		OFF					
16	MIN_JUNCTION_TEMPERATURE		0					
17	MAX_JUNCTION_TEMPERATURE		85					
18	POWER_PRESET_COOLING_SOLUTION		23 mm heat sink with 200 Lfpm airflow					
19	POWER_BOARD_THERMAL_MODEL		None (Conservative)					
20	POWER_USE_TA_VALUE		25					
21	POWER_BOARD_TEMPERATURE		-1					
22	POWER_OJC_VALUE		0.1					
23	POWER_OCS_VALUE		0.1					
24	POWER_OSA_VALUE		1.8					
25	POWER_OJB_VALUE		-1					
26	VCCIO		1A		2.5 1B		0 1C	2.5 2C
27	VCCPD		1A		2.5 1B		0 1C	2.5 2C
28	RAIL_VOLTAGES		VCC		1.1 VCCPT		2.5 VCCA_PLL	2.5 VCC
29	HIGH_SPEED		NUM_HIGH_SPEED_M9K_block_TILES		35 NUM_M9K_block_TILES_USED		35	
30								
31								
32	BLOCK	M9K block	count		16 ram_mode	Simple Dual Port	ram_read_durir new	rar
33	BLOCK	M9K block	count		3 ram_mode	Simple Dual Port	ram_read_durir new	rar
34	BLOCK	M9K block	count		16 ram_mode	Simple Dual Port	ram_read_durir new	rar
35	BLOCK	Combinational cell	count		28 avg_toggle_rate		181153.408 avg_toggle_rate	0 av
36	BLOCK	Combinational cell	count		2967 avg_toggle_rate		7292423.461 avg_toggle_rate	0.147695 av
37	BLOCK	Combinational cell	count		47 avg_toggle_rate		639806.5769 avg_toggle_rate	0.044742 av
38	BLOCK	Clock enable block	count		1 avg_toggle_rate		0 avg_toggle_rate	0 av
39	BLOCK	Clock enable block	count		1 avg_toggle_rate		0 avg_toggle_rate	0 av
40	BLOCK	Clock enable block	count		1 avg_toggle_rate		28500000 avg_toggle_rate	1.993007 av
41	BLOCK	Clock enable block	count		1 avg_toggle_rate		28500000 avg_toggle_rate	2 av
42	BLOCK	Clock enable block	count		1 avg_toggle_rate		98750000 avg_toggle_rate	2 av
43	BLOCK	Register cell	count		2567 avg_toggle_rate		3790981.691 avg_toggle_rate	0.076779 av
44	BLOCK	Register cell	count		71 avg_toggle_rate		249731.7324 avg_toggle_rate	0.017464 av
45	BLOCK	MLAB cell	count		1 mlab_width		8 mlab_depth	4 av
46	BLOCK	I/O pad	count		16 avg_toggle_rate		442715.8125 avg_toggle_rate	0 av
47	BLOCK	I/O pad	count		26 avg_toggle_rate		982211.5385 avg_toggle_rate	0 av
48	BLOCK	I/O pad	count		1 avg_toggle_rate		100850000 avg_toggle_rate	2.042532 av
49	BLOCK	I/O pad	count		4 avg toggle rate		12500 avg toggle rate	0.000253 av

The PowerPlay EPE spreadsheet includes the Import Data macro that parses the information in the PowerPlay EPE File and transfers it into the spreadsheet. If you do not want to use the macro, you can manually transfer the data into the PowerPlay EPE spreadsheet.

For example, after importing the PowerPlay EPE File information into the PowerPlay EPE spreadsheet, you can add additional device resource information at any time. If the existing Quartus II project represents only a portion of your full design, you must enter the additional device resources used in the final design manually.

PowerPlay EPE File Generator Compilation Report

After successfully generating the PowerPlay EPE File, you can locate a PowerPlay EPE File Generator report under the **Compilation Report** section. This report contains different sections, such as Summary, Settings, Generated Files, Confidence Metric Details, and Signal Activities. For more information about the PowerPlay EPE File Generator report, refer to [“PowerPlay Power Analyzer Compilation Report” on page 8-21](#).

Table 8-1 lists the main differences between the PowerPlay EPE and the Quartus II PowerPlay Power Analyzer.

Table 8-1. Comparison of the PowerPlay EPE and Quartus II PowerPlay Power Analyzer

Characteristic	PowerPlay EPE	Quartus II PowerPlay Power Analyzer
Phase in the design cycle	Any time	Post-fit
Tool requirements	Spreadsheet program or the Quartus II software	The Quartus II software
Accuracy	Medium	Medium to very high
Data inputs	<ul style="list-style-type: none"> ■ Resource usage estimates ■ Clock requirements ■ Environmental conditions ■ Toggle rate 	<ul style="list-style-type: none"> ■ Post-fit design ■ Clock requirements ■ Signal activity defaults ■ Environmental conditions ■ Register transfer level (RTL) simulation results (optional) ■ Post-fit simulation results (optional) ■ Signal activities per node or entity (optional)
Data outputs ⁽¹⁾	<ul style="list-style-type: none"> ■ Total thermal power dissipation ■ Thermal static power ■ Thermal dynamic power ■ Off-chip power dissipation ■ Current drawn from voltage supplies 	<ul style="list-style-type: none"> ■ Total thermal power ■ Thermal static power ■ Thermal dynamic power ■ Thermal I/O power ■ Thermal power by design hierarchy ■ Thermal power by block type ■ Thermal power dissipation by clock domain ■ Off-chip (non-thermal) power dissipation ■ Device supply currents

Notes to Table 8-1:

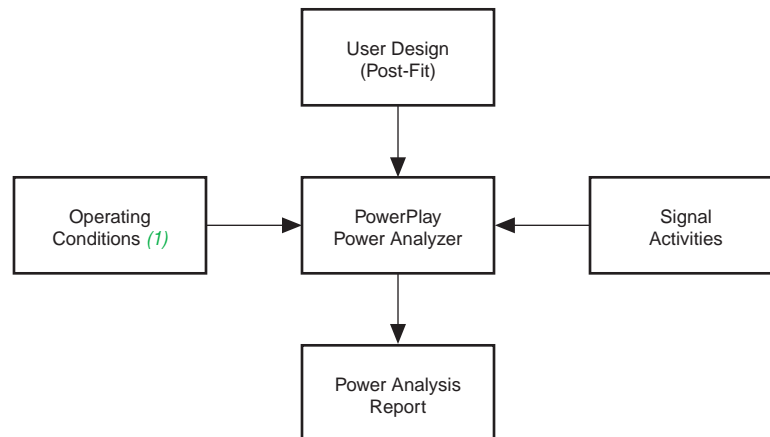
(1) PowerPlay EPE and PowerPlay Power Analyzer outputs vary by device family. For more information, refer to the [device-specific EPE User Guide](#) and [PowerPlay Power Analyzer Reports](#) in Quartus II Help.

The result of the PowerPlay Power Analyzer is only an estimation of power. Altera does not recommend using the result as a specification. The purpose of the estimation is to help you to establish a guide for the power budget of your design. Altera recommends measuring the actual power on the board. You must measure the total dynamic current of your design during device operation because the estimate is design dependent and depends on many variable factors, including input vector quantity, quality, and exact loading conditions of a PCB design. Static power consumption must not be based on empirical observation. The values reported by the PowerPlay Power Analyzer or data sheet must be used because the tested devices might not exhibit worst-case behavior.

PowerPlay Power Analyzer Flow

The PowerPlay Power Analyzer supports accurate power estimations by allowing you to specify all the important design factors affecting power consumption. [Figure 8-3](#) shows the high-level PowerPlay Power Analyzer flow.

Figure 8-3. PowerPlay Power Analyzer High-Level Flow



Note to Figure 8-3:

(1) Operating condition specifications are available only for some device families. For more information, refer to [Performing Power Analysis with the PowerPlay Power Analyzer](#) in Quartus II Help.

The PowerPlay Power Analyzer requires your design to be synthesized and fitted to the target device. You must specify the electrical standard used by each I/O cell and the capacitive load on each I/O standard in your design to obtain accurate I/O power estimates.

Operating Settings and Conditions

You can specify device power characteristics, operating voltage conditions, and operating temperature conditions for power analysis in the Quartus II software.

On the **Operating Settings and Conditions** page of the **Settings** dialog box, you can specify whether the device has typical power consumption characteristics or maximum power consumption characteristics.

- ❓ For more information, refer to [Operating Setting and Conditions Page \(Settings Dialog Box\)](#) in Quartus II Help.

On the **Voltage** page of the **Settings** dialog box, you can view the operating voltage conditions for each power rail in the device, and specify supply voltages for power rails with selectable supply voltages.

- ❓ For more information, refer to [Voltage Page \(Settings Dialog Box\)](#) in Quartus II Help.

On the **Temperature** page of the **Settings** dialog box, you can specify the thermal operating conditions of the device.

- ❓ For more information, refer to *Temperature Page (Settings Dialog Box)* in Quartus II Help.

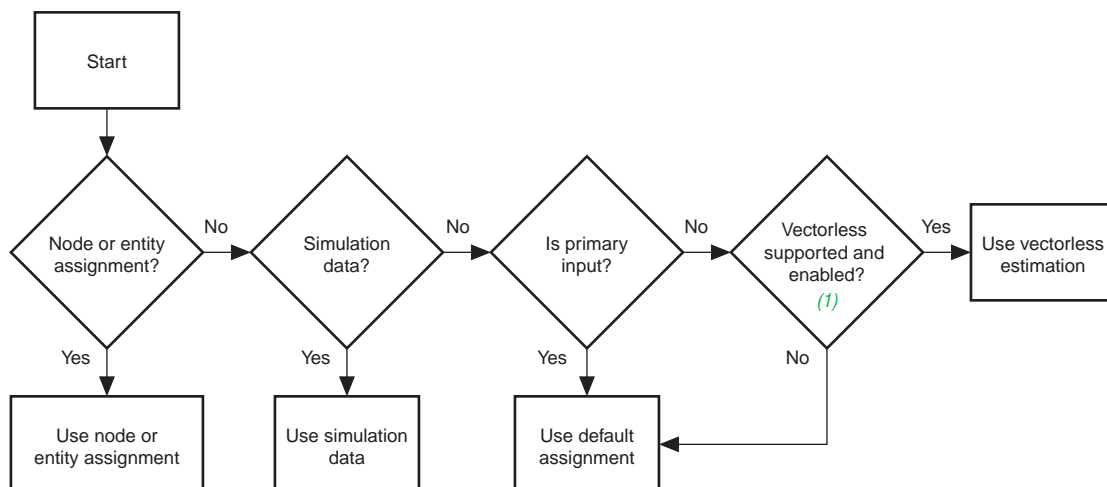
Signal Activities Data Sources

The PowerPlay Power Analyzer provides a flexible framework for specifying signal activities. It reflects the importance of using representative signal-activity data during power analysis. You can use the following sources to provide information about signal activity:

- Simulation results
- User-entered node, entity, and clock assignments
- User-entered default toggle rate assignment
- Vectorless estimation

The PowerPlay Power Analyzer allows you to mix and match the signal-activity data sources on a signal-by-signal basis. Figure 8-4 shows the priority scheme. The following sections describe the data sources.

Figure 8-4. Signal-Activity Data Source Priority Scheme



Note to Figure 8-4:

- (1) Vectorless estimation is available only for some device families. For more information, refer to *Performing Power Analysis with the PowerPlay Power Analyzer*.

Simulation Results

The PowerPlay Power Analyzer directly reads the waveforms generated by a design simulation. The static probability and toggle rate for each signal are calculated from the simulation waveform. Power analysis is most accurate when you use representative input stimuli to generate simulations.

The PowerPlay Power Analyzer reads results generated by the following simulators:

- ModelSim®
- ModelSim-Altera
- QuestaSim

- Active-HDL
- NCSim
- VCS
- VCS MX
- Riviera-PRO

Signal activity and static probability information derive from a Verilog Value Change Dump File (**.vcd**). For more information, refer to [“Signal Activities” on page 8-4](#).

For third-party simulators, use the **Quartus II EDA Tool Settings for Simulation** to specify a **Generate Value Change Dump** file script. These scripts instruct the third-party simulators to generate a **.vcd** that encodes the simulated waveforms. The Quartus II PowerPlay Power Analyzer reads this file directly to derive the toggle rate and static probability data for each signal.

Third-party EDA simulators, other than those listed, can generate a **.vcd** that can then be used with the PowerPlay Power Analyzer. For those simulators, you must manually create a simulation script to generate the appropriate **.vcd**.



You can use a **.vcd** created for power analysis to optimize your design for power during fitting by utilizing the appropriate settings in the **PowerPlay power optimization** list, available in the **Fitter Settings** page of the **Settings** dialog box.



For more information about power optimization, refer to the [Power Optimization](#) chapter in volume 2 of the *Quartus II Handbook*.

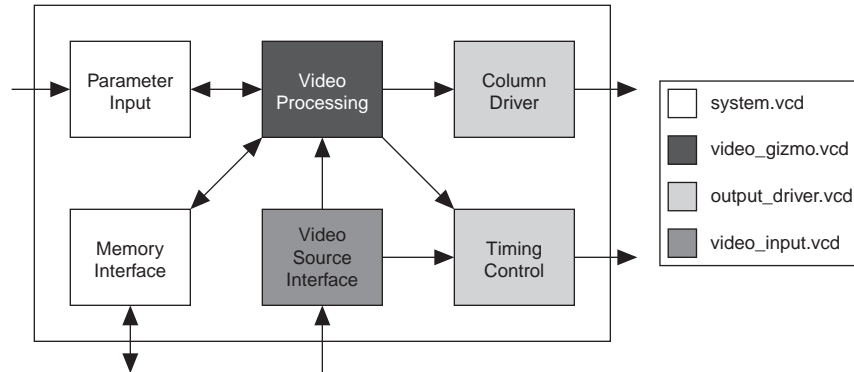


For more information about how to create a **.vcd** in other third-party EDA simulation tools, refer to [Section I. Simulation](#) in volume 3 of the *Quartus II Handbook*.

Using Simulation Files in Modular Design Flows

A common design practice is to create modular or hierarchical designs in which you develop each design entity separately, and then instantiate it in a higher-level entity, forming a complete design. You can perform simulation on a complete design or on each modular design for verification. The PowerPlay Power Analyzer supports modular design flows when reading the signal activities generated from these simulation files. An example of a modular design flow is shown in [Figure 8-5](#).

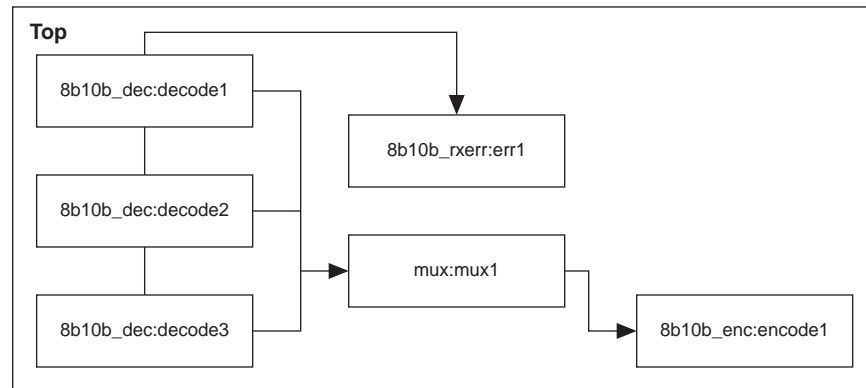
Figure 8-5. Modular Simulation Flow



When specifying a simulation file, an associated design entity name is given, such that the signal activities derived from the simulation file (**.vcd**) are imported into the PowerPlay Power Analyzer for that particular design entity. The PowerPlay Power Analyzer also supports the specification of multiple **.vcd** for power analysis, with each having an associated design entity name to allow the integration of partial design simulations into a complete design power analysis. When specifying multiple **.vcd** for your design, it is possible that more than one simulation file contains signal-activity information for the same signal. When you apply multiple **.vcd** to the same design entity, the signal activity used in the power analysis is the equal-weight arithmetic average of each **.vcd**. When you apply multiple simulation files to design entities at different levels in your design hierarchy, the signal activity in the power analysis derives from the simulation file that applies to the most specific design entity.

Figure 8-6 shows an example of a hierarchical design. The top-level module of your design, called **Top**, consists of three 8b/10b decoders, followed by a multiplexer. The output of the multiplexer is then encoded again before being the output from your design. There is also an error-handling module that handles any 8b/10b decoding errors. The top contains the top-level entity of your design and any logic not defined as part of another module. The design file for the top-level module might be just a wrapper for the hierarchical entities below it, or it might contain its own logic. The following usage scenarios show common ways that you can simulate your design and import .vcd into the PowerPlay Power Analyzer.

Figure 8-6. Example Hierarchical Design



Complete Design Simulation

You can simulate the entire design top, generating a .vcd from a third-party simulator. The .vcd can then be imported (specifying entity top) into the PowerPlay Power Analyzer. The resulting power analysis uses all the signal activities information from the generated .vcd, including those that apply to submodules, such as decode [1-3], err1, mux1, and encode1.

Modular Design Simulation

You can independently simulate submodules of the design top, and then import all the resulting .vcd into the PowerPlay Power Analyzer. For example, you can simulate the 8b10b_dec independent of the entire design, as well as multiplexer, 8b10b_rxerr, and 8b10b_enc. You can then import the .vcd generated from each simulation by specifying the appropriate instance name. For example, if the files produced by the simulations are 8b10b_dec.vcd, 8b10b_enc.vcd, 8b10b_rxerr.vcd, and mux.vcd, the import specifications in Table 8-2 are used.

Table 8-2. Import Specifications (Part 1 of 2)

File Name	Entity
8b10b_dec.vcd	Top 8b10b_dec:decode1
8b10b_dec.vcd	Top 8b10b_dec:decode2
8b10b_dec.vcd	Top 8b10b_dec:decode3
8b10b_rxerr.vcd	Top 8b10b_rxerr:err1

Table 8-2. Import Specifications (Part 2 of 2)

File Name	Entity
8b10b_enc.vcd	Top 8b10b_enc:encode1
mux.vcd	Top mux:mux1

The resulting power analysis applies the simulation vectors found in each file to the assigned entity. Simulation provides signal activities for the pins and for the outputs of functional blocks. If the inputs to an entity instance are input pins for the entire design, the simulation file associated with that instance does not provide signal activities for the inputs of that instance. For example, an input to an entity such as mux1 has its signal activity specified at the output of one of the decode entities.

Multiple Simulations on the Same Entity

You can perform multiple simulations of an entire design or specific modules of a design. For example, in the process of verifying the design top, you can have three different simulation testbenches: one for normal operation and two for corner cases. Each of these simulations produces a separate .vcd. In this case, apply the different .vcd names to the same top-level entity, shown in [Table 8-3](#).

Table 8-3. Multiple Simulation File Names and Entities

File Name	Entity
normal.vcd	Top
corner1.vcd	Top
corner2.vcd	Top

The resulting power analysis uses an arithmetic average that the signal activities calculated from each simulation file to obtain the final signal activities used. If a signal err_out has a toggle rate of zero toggles per second in **normal.vcd**, 50 toggles per second in **corner1.vcd**, and 70 toggles per second in **corner2.vcd**, the final toggle rate in the power analysis is 40 toggles per second.

Overlapping Simulations

You can perform a simulation on the entire design top and more exhaustive simulations on a submodule, such as 8b10b_rxerr. [Table 8-4](#) shows the import specification for overlapping simulations.

Table 8-4. Overlapping Simulation Import Specifications

File Name	Entity
full_design.vcd	Top
error_cases.vcd	Top 8b10b_rxerr:err1

In this case, signal activities from **error_cases.vcd** are used for all of the nodes in the generated .vcd and signal activities from **full_design.vcd** are used for only those nodes that do not overlap with nodes in **error_cases.vcd**. In general, the more specific hierarchy (the most bottom-level module) derives signal activities for overlapping nodes.

Partial Simulations

You can perform a simulation in which the entire simulation time is not applicable to signal-activity calculation. For example, run a simulation for 10,000 clock cycles and reset the chip for the first 2,000 clock cycles. If the signal-activity calculation is performed over all 10,000 cycles, the toggle rates are only 80% of their steady state value (because the chip is in reset for the first 20% of the simulation). In this case, you must specify the useful parts of the `.vcd` for power analysis. The **Limit VCD Period** option enables you to specify a start and end time to be used when performing signal-activity calculations.

Node Name Matching Considerations

Node name mismatches happen when you have `.vcd` applied to entities other than the top-level entity. In a modular design flow, the gate-level simulation files created in different Quartus II projects may not match their node names with the current Quartus II project.

For example, if you have a file named `8b10b_enc.vcd`, which was generated in a separate project called `8b10b_enc` and is simulating the 8b10b encoder, and you import that `.vcd` into another project called `Top`, you might encounter name mismatches when applying the `.vcd` to the `8b10b_enc` module in the `Top` project. This mismatch happens because all the combinational nodes in the `8b10b_enc.vcd` might be named differently in the `Top` project.

You can avoid name mismatching with only RTL simulation data, in which register names do not change, or with an incremental compilation flow that preserves node names in conjunction with a gate-level simulation.



To ensure the best accuracy, Altera recommends using an incremental compilation flow to preserve the node names of your design.





For more information about the incremental compilation flow, refer to the *Quartus II Incremental Compilation for Hierarchical and Team-Based Design* chapter in volume 1 of the *Quartus II Handbook*.

Glitch Filtering

The PowerPlay Power Analyzer defines a glitch as two signal transitions so closely spaced in time that the pulse or glitch occurs faster than the logic and routing circuitry can respond. The output of a transport delay model simulator contains glitches for some signals. The logic and routing structures of the device form a low-pass filter that filters out glitches that are tens to hundreds of picoseconds long, depending on the device family.

Some third-party simulators use different models than the transport delay model as default model. Different models cause differences in signal activity and power estimation. The inertial delay model, which is the ModelSim default model, filters out more glitches than the transport delay model and usually yields a lower power estimate.

-  Altera recommends using the transport simulation model when using the Quartus II software glitch filtering support with third-party simulators. Simulation glitch filtering has little effect if you use the inertial simulation model.
-  For more information about how to set the simulation model type for your specific simulator, refer to Quartus II Help.

Glitch filtering in a simulator can also filter a glitch on one LE (or other circuit element) output from propagating to downstream circuit elements to ensure that the glitch does not affect simulated results. It prevents a glitch on one signal from producing non-physical glitches on all downstream logic, which can result in a signal toggle rate and a power estimate that are too high. Circuit elements in which every input transition produces an output transition, including multipliers and logic cells configured to implement XOR functions, are especially prone to glitches. Therefore, circuits with such functions can have power estimates that are too high when you do not use glitch filtering.

Altera recommends using the glitch filtering feature to obtain the most accurate power estimates. For .vcd, the PowerPlay Power Analyzer flows support two levels of glitch filtering, both of which are recommended for power estimation.

In the first level of glitch filtering, glitches are filtered during simulation. To enable this level of glitch filtering in the Quartus II software for supported third-party simulators, follow these steps:


1. On the Assignments menu, click **Settings**. The **Settings** dialog box appears.
2. In the **Category** list, select **Simulation** under **EDA Tool Settings**. The **Simulation** page appears.
3. Select the **Tool name** to use for the simulation.
4. Turn on **Enable glitch filtering**.

The second level of glitch filtering occurs while the PowerPlay Power Analyzer is reading the .vcd generated by a third-party simulator. To enable this level of glitch filtering, follow these steps:

On the Assignments menu, click **Settings**. The **Settings** dialog box appears.

1. In the **Category** list, select **PowerPlay Power Analyzer Settings**. The **PowerPlay Power Analyzer Settings** page appears.
2. Under **Input File(s)**, turn on **Perform glitch filtering on VCD files**.


The .vcd file reader performs complementary filtering to the filtering performed during simulation and is often not as effective. While the .vcd file reader can remove glitches on logic blocks, it has no way of determining how downstream logic and routing are affected by a given glitch, and may eliminate the impact of the glitch completely. Filtering the glitches during simulation avoids switching downstream routing and logic automatically.

 When running simulation for design verification (rather than to produce input to the PowerPlay Power Analyzer), Altera recommends turning off the glitch filtering option to produce the most rigorous and conservative simulation from a functionality viewpoint. When performing simulation to produce input for the PowerPlay Power Analyzer, Altera recommends turning on the glitch filtering to produce the most accurate power estimates.

Node and Entity Assignments

You can assign specific toggle rates and static probabilities to individual nodes and entities in the design. These assignments have the highest priority, overriding data from all other signal-activity sources.


You must use the Assignment Editor or Tcl commands to create the **Power Toggle Rate** and **Power Static Probability** assignments. You can specify the power toggle rate as an absolute toggle rate in transitions using the **Power Toggle Rate** assignment or you can use the **Power Toggle Rate Percentage** assignment to specify a toggle rate relative to the clock domain of the assigned node for a more specific assignment made in terms of hierarchy level.

 If the **Power Toggle Rate Percentage** assignment is used, and the given node does not have a clock domain, a warning is issued and the assignment is ignored.

 For more information about how to use the Assignment Editor in the Quartus II software, refer to the *Constraining Designs* chapter in volume 2 of the *Quartus II Handbook*.

Assigning specific toggle rates and static probabilities to individual nodes and entities is appropriate for signals in which you have specific knowledge of the signal or entity being analyzed. For example, if you know that a 100 MHz data bus or memory output produces data that is essentially random (uncorrelated in time), you can directly enter a 0.5 static probability and a toggle rate of 50 million transitions per second.

Bidirectional I/O pins are treated specially. The combinational input port and the output pad for a given pin share the same name. However, those ports might not share the same signal activities. For the purpose of reading signal-activity assignments, the PowerPlay Power Analyzer creates a distinct name `<node_name~output>` when the bidirectional signal is configured as an output and `<node_name~result>` when the signal is configured as an input. For example, if a design has a bidirectional pin named MYPIN, assignments for the combinational input use the name MYPIN~result, and the assignments for the output pad use the name MYPIN~output.

 When creating the logic assignment in the Assignment Editor, you will not find the MYPIN~result and MYPIN~output node names in the Node Finder. Therefore, to create the logic assignment, you must manually enter the two differentiating node names to create the specific assignment for the input and output port of the bidirectional pin.

Timing Assignments to Clock Nodes

For clock nodes, the PowerPlay Power Analyzer uses the timing requirements to derive the toggle rate when neither simulation data nor user-entered signal-activity data is available. f_{MAX} requirements specify full cycles per second, but each cycle represents a rising transition and a falling transition. For example, a clock f_{MAX} requirement of 100 MHz corresponds to 200 million transitions per second.

Default Toggle Rate Assignment

You can specify a default toggle rate for primary inputs and all other nodes in the design. The default toggle rate is used when no other method has specified the signal-activity data.

The toggle rate is specified in absolute terms (transitions per second) or as a fraction of the clock rate in effect for each particular node. The toggle rate for a given clock derives from the timing settings for the clock. For example, if a clock is specified with an f_{MAX} constraint of 100 MHz and a default relative toggle rate of 20%, nodes in this clock domain transition in 20% of the clock periods, or 20 million transitions occur per second. In some cases, the PowerPlay Power Analyzer cannot determine the clock domain for a given node because there is either no clock domain for the node or it is ambiguous. In these cases, the PowerPlay Power Analyzer substitutes and reports a toggle rate of zero.

Vectorless Estimation

For some device families, the PowerPlay Power Analyzer automatically derives estimates for signal activity on nodes with no simulation or user-entered signal-activity data. Vectorless estimation statistically estimates the signal activity of a node based on the signal activities of all nodes feeding that node, and on the actual logic function implemented by the node. Vectorless estimation cannot derive signal activities for primary inputs. Vectorless estimation is generally accurate for combinational nodes, but not for registered nodes. Therefore, simulation data for at least the registered nodes and I/O nodes is required for accuracy.

- ② For more information, refer to *Performing Power Analysis with the PowerPlay Power Analyzer* in Quartus II Help.

The **PowerPlay Power Analyzer Settings** dialog box lets you disable vectorless estimation. When enabled, vectorless estimation takes priority over default toggle rates. Vectorless estimation does not override clock assignments.

Using the PowerPlay Power Analyzer

For all the flows that use the PowerPlay Power Analyzer, synthesize your design first and then fit it to the target device. You must either provide timing assignments for all the clocks in the design or use a simulation-based flow to generate activity data. The I/O standard used on each device input or output and the capacitive load on each output must be specified in the design.

- ② For more information about using the PowerPlay Power Analyzer, refer to *Performing Power Analysis with the PowerPlay Power Analyzer* in Quartus II Help.

Common Analysis Flows

You can use the analysis flows in this section with the PowerPlay Power Analyzer. However, vectorless activity estimation is only available for some device families.

Signal Activities from Full Post-Fit Netlist (Timing) Simulation

This flow provides the most accuracy because all node activities reflect actual design behavior, provided that supplied input vectors are representative of typical design operation. Results are better if the simulation filters glitches. The disadvantage of this method is that the simulation time is long.

Signal Activities from Full Post-Fit Netlist (Zero Delay) Simulation

The zero delay simulation flow is used with designs for which signal activities from a full post-fit netlist (timing) simulation are not available. Zero delay simulation is as accurate as timing simulation in 95% of designs with no glitches.



If your design has glitches, power may be underestimated. Altera recommends using the signal activities from a full post-fit netlist (timing) simulation to achieve accurate power estimation of your design.

The following designs might exhibit glitches:

- Designs with many XOR gates (for example, an encryption core)
- Designs with arithmetic blocks without input and output registers (DSPs and carry chains)

For more information about creating zero delay simulation signal activities, refer to [“Generating a .vcd from Full Post-Fit Netlist \(Zero Delay\) Simulation”](#) on page 8–20.

Signal Activities from RTL (Functional) Simulation, Supplemented by Vectorless Estimation

In this flow, simulation provides toggle rates and static probabilities for all pins and registers in the design. Vectorless estimation fills in the values for all the combinational nodes between pins and registers, giving good results. This flow usually provides a compilation time benefit to the user in the third-party RTL simulator.



RTL simulation may not provide signal activities for all registers in the post-fitting netlist because some register names might be lost during synthesis. For example, synthesis might automatically transform state machines and counters, thus changing the names of registers in those structures.

Signal Activities from Vectorless Estimation and User-Supplied Input Pin Activities


This flow provides a low level of accuracy, because vectorless estimation for registers is not entirely accurate.

Signal Activities from User Defaults Only

This flow provides the lowest degree of accuracy.


Generating a .vcd

In previous versions of the Quartus II software, you could use either the Quartus II simulator or an EDA simulator to perform your simulation. The Quartus II software no longer supports a built-in simulator, and you must use EDA simulators to perform simulation. Use the .vcd as the input to the PowerPlay Power Analyzer to estimate power for your design.


 For more information about the supported third-party simulators, refer to [“Simulation Results”](#) on page 8-9.

To create a .vcd for your design, follow these steps:

1. On the Assignments menu, click **Settings**. The **Settings** dialog box appears.
2. In the **Category** list, under **EDA Tool Settings**, click **Simulation**. The **Simulation** page appears.
3. In the **Tool name** list, select your preferred EDA simulator.
4. In the **Format for output netlist** list, select **Verilog HDL**, or **SystemVerilog HDL**, or **VHDL**.
5. Turn on **Generate Value Change Dump (VCD) file script**.


 This turns on the **Map illegal HDL characters** and **Enable glitch filtering** options. The **Map illegal HDL characters** option ensures that all signals have legal names and that signal toggle rates are available later in the PowerPlay Power Analyzer.

6. By turning on **Enable glitch filtering**, glitch filtering logic is the output when you generate an EDA netlist for simulation. This option is available regardless of whether or not you want to generate the .vcd scripts. For more information about glitch filtering, refer to [“Glitch Filtering”](#) on page 8-14.

 When performing simulation using ModelSim, the **+nospecify** option for the vsim command disables the **specify path delays and timing checks** option in ModelSim. By enabling glitch filtering on the **Simulation** page, the simulation models include specified path delays. Thus, ModelSim might fail to simulate a design if glitch filtering is enabled, and the **+nospecify** option is specified. Altera recommends removing the **+nospecify** option from the ModelSim vsim command to ensure accurate simulation for power estimation.

7. Click **Script Settings**. The **Script Settings** dialog box appears.

Select which signals must be output to the .vcd. With **All signals** selected, the generated script instructs the third-party simulator to write all connected output signals to the .vcd. With **All signals except combinational lcell outputs** selected, the generated script tells the third-party simulator to write all connected output signals to the .vcd, except logic cell combinational outputs.

 The file can become extremely large if you write all output signals to the file because its size depends on the number of output signals being monitored and the number of transitions that occur.

8. Click **OK**.
9. Type a name for your testbench in the **Design instance name** box.
10. Compile your design with the Quartus II software and generate the necessary EDA netlist and script that instructs the third-party simulator to generate a **.vcd**.



For more information about the NativeLink feature, refer to *Section I. Simulation* in volume 3 of the *Quartus II Handbook*.

11. Perform a simulation with the third-party EDA simulation tool. Call the generated script in the simulation tool before running the simulation. The simulation tool generates the **.vcd** and places it in the project directory.

Generating a .vcd from ModelSim Software

To successfully produce a **.vcd** with the ModelSim software, follow these steps:

1. In the Quartus II software, on the Assignments menu, click **Settings**. The **Settings** dialog box appears.
2. In the **Category** list, under **EDA Tool Settings**, click **Simulation**. The **Simulation** page appears.
3. In the **Tool name** list, select your preferred EDA simulator.
4. In the **Format for output netlist** list, select **Verilog HDL**, or **SystemVerilog HDL**, or **VHDL**.
5. Turn on **Generate Value Change Dump (VCD) file script**.
6. To generate the **.vcd**, perform a full compilation.
7. In the ModelSim software, compile the files necessary for simulation.
8. Load your design by clicking **Start Simulation** on the Tools menu, or use the `vsim` command.
9. Use the **.vcd** script created in step 6 using the following command:

```
source <design>_dump_all_vcd_nodes.tcl
```
10. Run the simulation (for example, `run 2000ns` or `run -all`).
11. Quit the simulation using the `quit -sim` command, if required.
12. Exit the ModelSim software. If you do not exit the software, the ModelSim software might end the writing process of the **.vcd** improperly, resulting in a corrupt **.vcd**.

Generating a .vcd from Full Post-Fit Netlist (Zero Delay) Simulation

To successfully generate a **.vcd** from the full post-fit Netlist (zero delay) simulation, follow these steps:

1. Compile your design in the Quartus II software to generate the Netlist `<project_name>.vo`.
2. In `<project_name>.vo`, search for the include statement for `<project_name>.sdo`, comment the statement out, and save the file.

3. Generate a **.vcd** for power estimation by performing the steps in “[Generating a .vcd](#)” on page 8–19.



Altera recommends using the Standard Delay Format Output File (**.sdo**) for gate-level timing simulation. The **.sdo** contains the delay information of each architecture primitive and routing element specific to your design; however, you must exclude the **.sdo** for zero delay simulation.



For more information about how to create a **.vcd** in other third-party EDA simulation tools, refer to [Section I. Simulation](#) in volume 3 of the *Quartus II Handbook*.

Running the PowerPlay Power Analyzer Using the Quartus II GUI

To run the PowerPlay Power Analyzer using the Quartus II GUI, refer to [Performing Power Analysis with the PowerPlay Power Analyzer](#) in Quartus II Help.

PowerPlay Power Analyzer Compilation Report

The PowerPlay Power Analyzer section of the Compilation Report consists of the following sections.

Summary

This section of the report shows the estimated total thermal power consumption of your design. This includes dynamic, static, and I/O thermal power consumption. The I/O thermal power consumption is the total I/O power contributed by both the V_{CCIO} power supplies and some portion of the V_{CCINT} . The report also includes a confidence metric that reflects the overall quality of the data sources for the signal activities. For example, a **Low** power estimation confidence value reflects that you have provided insufficient toggle rate data, or most of the signal-activity information used for power estimation is from default or vectorless estimation settings. For more information about the input data, refer to the PowerPlay Power Analyzer Confidence Metric report.

Settings

This section of the report shows the PowerPlay Power Analyzer settings information of your design, including the default input toggle rates, operating conditions, and other relevant setting information.

Simulation Files Read

This section of the report lists the simulation output file (**.vcd**) used for power estimation. This section also includes the file ID, file type, entity, VCD start time, VCD end time, the unknown percentage, and the toggle percentage. The unknown percentage indicates the portion of the design module that is not exercised by the simulation vectors.

Operating Conditions Used

This section of the report shows device characteristics, voltages, temperature, and cooling solution, if any, that were used during the power estimation. This section also shows the entered junction temperature or auto-computed junction temperature that was used during the power analysis.

Thermal Power Dissipated by Block

This section of the report shows estimated thermal dynamic power and thermal static power consumption categorized by atoms. This information provides you with estimated power consumption for each atom in your design.

Thermal Power Dissipation by Block Type (Device Resource Type)

This section of the report shows the estimated thermal dynamic power and thermal static power consumption categorized by block types. This information is further categorized by estimated dynamic and static power that was used, as well as providing an average toggle rate by block type. Thermal power is the power dissipated as heat from the FPGA device.

Thermal Power Dissipation by Hierarchy

This section of the report shows estimated thermal dynamic power and thermal static power consumption categorized by design hierarchy. This information is further categorized by the dynamic and static power that was used by the blocks and routing in that hierarchy. This information is very useful when locating problem modules in your design.

Core Dynamic Thermal Power Dissipation by Clock Domain

This section of the report shows the estimated total core dynamic power dissipation by each clock domain, which provides designs with estimated power consumption for each clock domain in the design. If the clock frequency for a domain is unspecified by a constraint, the clock frequency is listed as “unspecified.” For all the combinational logic, the clock domain is listed as no clock with zero MHz.

Current Drawn from Voltage Supplies

This section of the report lists the current that was drawn from each voltage supply. The V_{CCIO} voltage supply is further categorized by I/O bank and by voltage. The minimum safe power supply size (current supply ability) is also listed for each supply voltage.

Transceiver-based devices have multiple voltage supplies, which are V_{CCH} , V_{CCT} , V_{CCR} , V_{CCA} , and V_{CCP} . The report also shows the static and dynamic current (in mA) drawn from each voltage supply. Total static and dynamic power consumed by the transceivers on all voltage supplies is listed under the “Thermal Power Dissipation by Block Type” report section, which contains a row that starts with “GXB Transceiver.”

The I/O thermal power dissipation, which is listed on the summary page, does not correlate directly to the power drawn from the V_{CCIO} voltage supply listed in this report. This is because the I/O thermal power dissipation value also includes portions of the V_{CCINT} power, such as the I/O element (IOE) registers, which are modeled as I/O power, but do not draw from the V_{CCIO} supply.

Confidence Metric Details

The confidence metric indicates the quality of the signal toggle rate data used to compute a power estimate. The confidence metric is low if the signal toggle rate data comes from sources that are considered poor predictors of real signal toggle rates in the device during an operation. Toggle rate data that comes from simulation, user-entered assignments on specific signals, or entities are considered reliable. Toggle rate data from default toggle rates (for example, 12.5% of the clock period) or vectorless estimation are considered relatively inaccurate. This section gives an overall confidence rating in the toggle rate data, from low to high. This section also summarizes how many pins, registers, and combinational nodes obtained their toggle rates from each of simulation, user entry, vectorless estimation, or default toggle rate estimations. This detailed information helps you understand how to increase the confidence metric, letting you determine your own confidence in the toggle rate data.

Signal Activities

This section lists toggle rates and static probabilities assumed by power analysis for all signals with fan-out and pins. The signal type is provided (pin, registered, or combinational), as well as the data source for the toggle rate and static probability. By default, all signal activities are reported, but can be turned off by turning off the **Write signal activities to report file** option on the **PowerPlay Power Analyzer Settings** page.



Altera recommends turning off the **Write signal activities to report file** option for a large design because of the large number of signals present. You can use the Assignment Editor to specify that activities for individual nodes or entities are reported by assigning an on value to those nodes for the **Power Report Signal Activities** assignment.

Messages

This section lists the messages generated by the Quartus II software during the analysis.


Specific Rules for Reporting

In a Stratix GX device, the XGM II state machine block is always used together with GXB transceivers, so its power is lumped into the power for the transceivers. Therefore, the power for the XGM II state machine block is reported as zero Watts.

Scripting Support

You can run procedures and create settings described in this chapter in a Tcl script. You can also run some procedures at a command prompt. For more information about scripting command options, refer to the Quartus II Command-Line and Tcl API Help browser. To run the Help browser, type the following command at the command prompt:

```
quartus_sh --qhelp
```

 For more information about Tcl scripting, refer to the *Tcl Scripting* chapter in volume 2 of the *Quartus II Handbook* and *API Functions for Tcl* in Quartus II Help. For more information about all settings and constraints in the Quartus II software, refer to the *Quartus II Settings File Reference Manual*. For more information about command-line scripting, refer to the *Command-Line Scripting* chapter in volume 2 of the *Quartus II Handbook*.

Running the PowerPlay Power Analyzer from the Command-Line

The separate executable used to run the PowerPlay Power Analyzer is `quartus_pow`. For a complete listing of all command-line options supported by `quartus_pow`, type the following command at a system command prompt:

```
quartus_pow --help or quartus_sh --qhelp ←
```

The following is an example of using the `quartus_pow` executable with project **sample.qpf**:

- To instruct the PowerPlay Power Analyzer to generate a PowerPlay EPE File, type the following command at a system command prompt:

```
quartus_pow sample --output_epe=sample.csv ←
```

- To instruct the PowerPlay Power Analyzer to generate a PowerPlay EPE File without performing the power estimate, type the following command at a system command prompt:

```
quartus_pow sample --output_epe=sample.csv --estimate_power=off ←
```

- To instruct the PowerPlay Power Analyzer to use a **.vcd** as input (**sample.vcd**), type the following command at a system command prompt:

```
quartus_pow sample --input_vcd=sample.vcd ←
```

- To instruct the PowerPlay Power Analyzer to use two **.vcd** files as input files (**sample1.vcd** and **sample2.vcd**), perform glitch filtering on the **.vcd** and use a default input I/O toggle rate of 10,000 transitions per second, type the following command at a system command prompt:

```
quartus_pow sample --input_vcd=sample1.vcd --input_vcd=sample2.vcd \  
--vcd_filter_glitches=on --\  
default_input_io_toggle_rate=10000transitions/s ←
```

- To instruct the PowerPlay Power Analyzer to not use any input file, a default input I/O toggle rate of 60%, no vectorless estimation, and a default toggle rate of 20% on all remaining signals, type the following command at a system command prompt:

```
quartus_pow sample --no_input_file --\  
default_input_io_toggle_rate=60% \  
--use_vectorless_estimation=off --default_toggle_rate=20% ←
```



There are no command-line options to specify the information found on the **PowerPlay Power Analyzer Settings Operating Conditions** page. You can use the Quartus II GUI to specify these options.

The `quartus_pow` executable creates a report file, `<revision name>.pow.rpt`. You can locate the report file in the main project directory. The report file contains the same information in “PowerPlay Power Analyzer Compilation Report” on page 8-21.

Conclusion



PowerPlay power analysis tools are designed for accurate estimation of power consumption from early design concept through design implementation. You can use the PowerPlay EPE to estimate power consumption during the design concept stage. You can use the PowerPlay Power Analyzer tool to refine power estimations during design implementation. The PowerPlay Power Analyzer produces detailed reports that you can use to optimize designs for lower power consumption and verify that the design is in your power budget.

Document Revision History

Table 8-5 shows the revision history for this chapter.

Table 8-5. Document Revision History

Date	Version	Changes
November 2011	10.1.1	<ul style="list-style-type: none"> Template update. Minor editorial updates.
December 2010	10.1.0	<ul style="list-style-type: none"> Added links to Quartus II Help, removed redundant material. Moved “Creating PowerPlay EPE Spreadsheets” to page 8-5. Minor edits.
July 2010	10.0.0	<ul style="list-style-type: none"> Removed references to the Quartus II Simulator. Updated Table 8-1 on page 8-7, Table 8-2 on page 8-12, and Table 8-3 on page 8-13. Updated Figure 8-3 on page 8-8, Figure 8-4 on page 8-9, and Figure 8-5 on page 8-11.
November 2009	9.1.0	<ul style="list-style-type: none"> Updated “Creating PowerPlay EPE Spreadsheets” on page 8-5 and “Simulation Results” on page 8-9. Added “Signal Activities from Full Post-Fit Netlist (Zero Delay) Simulation” on page 8-18 and “Generating a .vcd from Full Post-Fit Netlist (Zero Delay) Simulation” on page 8-20. Minor changes to “Generating a .vcd from ModelSim Software” on page 8-20. Updated Figure 8-2 on page 8-6 and Figure 11-8 on page 11-24.
March 2009	9.0.0	<ul style="list-style-type: none"> This chapter was chapter 11 in version 8.1. Removed Figures 11-10, 11-11, 11-13, 11-14, and 11-17 from 8.1 version.
November 2008	8.1.0	<ul style="list-style-type: none"> Updated for the Quartus II software version 8.1. Replaced Figure 11-3. Replaced Figure 11-14.
May 2008	8.0.0	<ul style="list-style-type: none"> Updated Figure 11-5. Updated “Types of Power Analyses” on page 11-5. Updated “Operating Conditions” on page 11-9. Updated “PowerPlay Power Analyzer Compilation Report” on page 11-31. Updated “Current Drawn from Voltage Supplies” on page 11-32.

-  For previous versions of the *Quartus II Handbook*, refer to the [Quartus II Handbook Archive](#).
-  Take an [online survey](#) to provide feedback about this handbook chapter.