Power Optimization in FPGA Designs

Mouzam Khan

Altera Corporation

mkhan@altera.com

CP-PWROPT-1.0

ABSTRACT

IC designers today are facing continuous challenges in balancing design performance and power consumption. This task is becoming more critical as designs grow larger and more complex and process geometries shrink to 90-nm and below. FPGAs currently available provide performance and features that designers want, but suffer due to higher power consumption requirements. This growing need for maximizing performance while minimizing power consumption requires an increasingly efficient power optimization without sacrificing performance.

The two primary sources of power consumption in FPGAs are:

- Dynamic power dissipation during charging and discharging of internal capacitances in the logic array and interconnect networks of an active device
- Static power dissipation due to leaking currents during device standby

This paper will present power optimization techniques that reduce the dynamic power of the design without affecting performance. Accurate toggle rate data information on each signal of the design is most important when optimizing power requirements in the design. Using simulation results is the most accurate way to generate signal activities representative of design operating behavior. Also discussed will be VCS, an EDA simulator from Synopsys that assists in obtaining accurate design toggle rates and utilizing this information to further optimize design power, as well as an investigation of power estimation accuracy and the different components used for good power estimation.

Table of Contents

1 Introduction

IC designers today are facing continuous challenges in balancing design performance and power consumption. This task is becoming more critical as designs grow larger and more complex and process geometries shrink to 90-nm and below. Designers must meet these challenges by finding the right formula for optimizing power without sacrificing performance. FPGAs provide the performance and features designers need, but suffer due to higher power consumption.

Designing for low power in programmable logic is important to the overall design equation for different end markets. The communications industry is a more mature FPGA market in this respect, with programmable logic used heavily in routers, switches, base stations and storage servers. These newer generation products demand larger and higher-performing FPGAs. Engineers are continuing to push the performance envelope by migrating more functionality into FPGAs, replacing ASICs and ASSPs, while demanding higher system performance. Although the newest 90-nm-generation FPGAs consume less power than previous 130-nm products, FPGAs still consume a large portion of total system power. Engineers today design into tighter and more enclosed spaces, making it difficult to improve airflow and install proper size heat sinks. Thus thermal management, and consequently power management, continues to be an important topic in high-end FPGA design.

2 Designing for Low Power

Designing for the low-power marketplace is not a trivial task. Engineers use numerous techniques to reduce power in FPGA designs. Various types of FPGAs, different design methodologies, numerous intellectual property (IP) cores, assorted system design methods and diverse software algorithms and power tools all contribute to power used in a design.

The two primary sources of power consumption in FPGAs are:

- Static power dissipation due to leaking currents during device standby. Using finer semiconductor process geometries (specifically the 90-nm geometry) has caused an increase in static power consumption in FPGAs. As transistor size shrinks and lower voltages are utilized, a greater sub-threshold leakage current occurs in the transistor channel when the transistor is in the off state. Consequently, static power consumption rises when using the 90-nm process.

- Dynamic power dissipation during charging and discharging of internal capacitances in the logic array and interconnect networks of an active device. Dynamic power is affected in two ways by process scaling. First, the use of smaller feature sizes and lower voltages significantly reduces dynamic power consumption. However, higher device operating frequencies are possible in 90-nm technology, and, since dynamic power increases with operating frequency, designs that make full use of the speed of 90-nm technology will see less reduction in dynamic power.

As a result of the expected blanket increase in power consumption for smaller process geometries, semiconductor manufacturers use various techniques to optimize power

consumption, both in the dynamic and static domains. This paper discusses some of these techniques in more detail.

3 Architecture

It is important to choose the right design architecture. FPGA vendors employ various design techniques to reduce static and dynamic power consumption while balancing cost and performance. For example, Altera's latest 90-nm FPGA families, Stratix® II and Cyclone™ II, have varying gate and diffusion lengths to optimize transistors needed for speed, versus transistors not requiring speed. This adjustment involves determining what the power-performance tradeoff will be and setting the threshold voltage accordingly for each functional transistor group. Process changes such as low-k dielectrics improve performance as well as a decrease in parasitic capacitance. Using a low-k dielectric technique affects dynamic power consumption (internal and I/O transistor power consumption) by approximately 10 percent for Altera Stratix II devices. Internal and I/O transistor power consumption is a major component of total power consumption. Therefore, a 10 percent power reduction of these components is significant.

Other architectural changes, such as increased local interconnect for logic cells enable designers to implement larger functions without using more power-consuming global routing. Stratix II FPGAs take architectural changes further by introducing a 6-input look-up table (LUT), which minimizes the number of interconnects used and provides a more efficient use of the device logic resources. This results in faster performance and lower power consumption.

FPGAs increasingly incorporate more dedicated circuitry and general reconfigurable logic. The most advanced programmable logic devices (PLDs) have dedicated multipliers, DSP blocks, variable size RAM blocks and flash memory. These dedicated circuits are designed from the ground up and build many of these functions much more efficiently. Strategically designing with these blocks saves general logic resources, increases overall speed and reduces power consumption. For example, building large shift registers from RAM-based first-in first-out (FIFO) buffers instead of building the shift registers from the LE registers. Also improving logic efficiency allows engineers to fit designs into smaller devices that consume less static power.

Strategically using these dedicated blocks saves general logic resources and increases overall speed, with the potential to also reduce power. For example, the Stratix II device family allows efficient targeting of small, medium and large memories with the TriMatrix™ memory architecture. TriMatrix memory is composed of three sizes of embedded RAM blocks, 512-bit M512 blocks, 4-Kbit M4K blocks, and 512-Kbit M-RAM blocks, each can be configured to support a wide range of features. M512 memory blocks are useful for implementing small FIFO buffers, DSP and clock domain transfer applications. The M4K memory blocks are used to implement buffers for a wide variety of applications, including processor code storage, large look-up table implementation and large memory applications. M-RAM blocks are useful in applications requiring a large volume of data to be stored on-chip. Effective utilization of these memory blocks has a significant impact on power reduction in the design. For example, using a large memory block for a small memory function will consume more power as compared to

using a small memory block. Power savings of up to 45 percent can be achieved in Stratix II devices by effectively utilizing these dedicated blocks.

4 Software

Synthesis tools take advantage of target FPGA architecture by automatically utilizing the specialized dedicated blocks and intelligently mapping functions to general logic, which can significantly reduce dynamic power. Designers can choose from a variety of synthesis tool vendors such as Synopsys and others. Depending on the design, engineers can further reduce power by using area-driven synthesis settings versus timing-driven synthesis. Area-driven synthesis will reduce the amount of logic used in a design resulting in less switching. This is due to fewer logic levels, which translates to reduced dynamic power in the design.

Place-and-route tools are equally important, although once a particular FPGA is selected; the designer must use that FPGA vendor's placement and routing tools. Interconnect can potentially consume a large amount of power, thus FPGA vendors provide options to place and route their designs for power optimization. For example, Altera® Quartus® II software provides power-driven compilation that takes place at synthesis and fitter levels. Power-driven synthesis changes the synthesis netlist to optimize the design for power. Power-driven synthesis settings perform memory optimization and power-aware logic mapping during synthesis. The power-driven fitter performs place-and-route optimization during fitting to fully optimize the design for power. The fitter applies an extra effort to minimize power even after timing requirements have been met by effectively moving the logic closer during placement to localize high-toggling nets and using routes with low capacitance.

Accurate toggle-rate data information on each design signal is important for optimizing design power during place-and route. Place-and-route tools from Altera utilize this information when guiding the fitter for design power optimization based on design signal activity. The most accurate signal activity provides the best power optimization during fitting. Signal activities from full, post-fit netlist (timing) simulation provide the highest accuracy since all node activities reflect actual design behavior, providing supplied input vectors are representative of typical design operation. In order to utilize the signal activities information from post-fit simulation designers must compile the design using the default settings (Normal fitting) in order to generate a signal activity file for the design and then recompile the design using the power-driven fitting that utilizes the design signal activities information to further optimize the design for power as shown in the Figure 1. This makes the design flow abit more time consuming but effective for design power optimization.

VCS, an EDA simulator from Synopsys, can be used to generate a value change dump (.vcd) file to encode simulated waveforms. Quartus II power-driven fitter reads signal activity from a VCD file and uses it to guide the fitter in optimizing the design for power.

The recommended design flow to fully optimize a design for power during compilation using Quartus II software is shown in Figure 1. This flow utilizes the power-driven synthesis and power-driven fitter options. On average, core dynamic power is reduced by 16 percent with the extra-effort synthesis and extra-effort fitting settings, as compared to off settings in both

synthesis and fitter options for power-driven compilation. Power-driven synthesis with the extra-effort option is most effective on designs that utilize memory blocks, as it reduces memory power by shutting down memory blocks not being accessed. It automatically selects the best memory configuration for memory implementation and provides optimal power saving by determining the number of memory blocks, decoder, and multiplexer circuits needed. This level of memory optimization requires extra logic that may reduce design performance. The normal effort synthesis option performs the same memory optimization functions providing there is no effect on design performance.

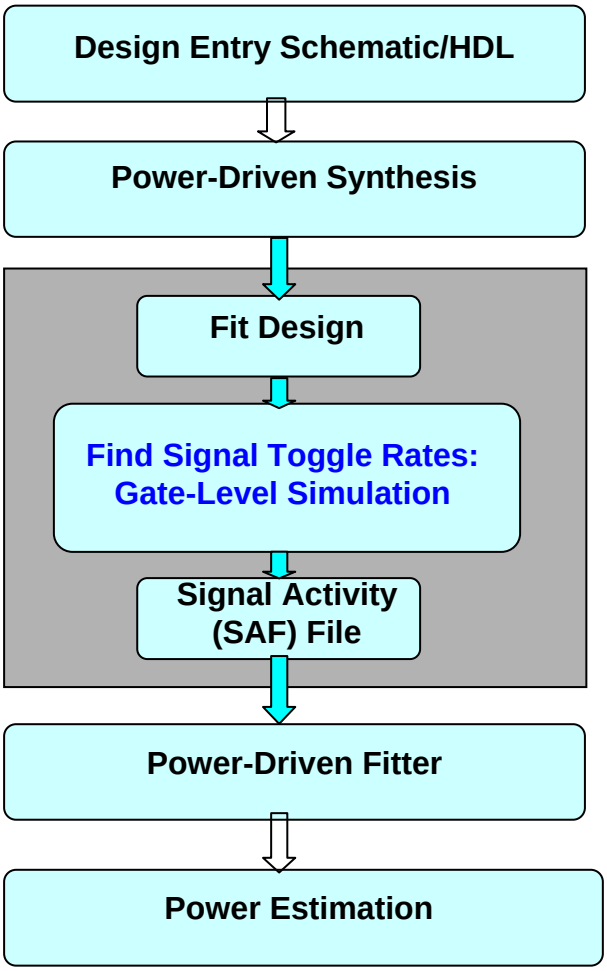

Figure 1. Quartus II Power-Driven Compilation Flow

5 Intellectual Property Cores

IP cores offered by different vendors are also important in the quest for low power. Choosing the correct core depends on the design application. This is important for efficient design if space versus performance or power is the primary design goal. Some vendors offer IP cores with multiple variations, such as Altera's Nios® II processor cores that are offered in fast, economy and standard versions.

Core	Description
Nios II/f (fast)	Optimized for maximum performance
Nios II/e (economy)	Optimized for minimum logic usage
Nios II/s (standard)	Offers a balance between performance and size

Table 1. Altera's Nios II Processor Family Members

Depending on the design, one core may consume significantly less power than another. Nios II/e (economy) consumes less power as it is optimized for minimum logic that translates into fewer switching activities. How IP cores are used in the design also have an effect on the design power consumption. For example, if a processor is making multiple memory calls to the same portion of memory, using a Nios II/f processor with an on-board cache can save power, as opposed to requiring the processor to access data from off-chip memory.

6 Design Methodology

Minimizing the use of dynamic power in a design by optimizing design algorithms reduces unnecessary and inadvertent switching. One example is memory blocks in FPGA devices that represent a large fraction of typical core dynamic power. Memory blocks are unlike most other blocks in the device. Most of their power is tied to clock rate and is insensitive to the toggle rate on data and address lines. When a memory block is clocked, there is a sequence of timed events that occur within the block that execute a read or write command. The circuitry controlled by the clock consumes the same amount of power regardless if the address or data has changed from one cycle to the next. Thus, the toggle rate of input data and the address bus have no impact on memory power consumption. Reducing memory power consumption is a reduction of memory clocking events. This can be achieved by using the clock enable signal on the memory ports. Using the clock enable signal enables memory only when necessary and shuts it down for the remaining time, thus reducing the overall memory power consumption.

Clocks represent a significant portion of dynamic power consumption due to their high switching activity and long paths. Reducing the switching on clock networks also saves a significant amount of power. Clock routing power is automatically optimized by most FPGAs software tools, which only enables those portions of the clock network that are required to feed downstream registers. Power can be further reduced by gating clocks when they are not needed. Most FPGAs are partitioned into sections where independent global clocks can be fed. If a design only uses a section of the logic intermittently, the design can turn off the clock feeding that section to save power. When a clock is turned off, the corresponding clock domain is shut down and becomes functionally inactive. Gated clocks can be a powerful technique to reduce power consumption in technologies such as ASICs. It is possible to build clock gating logic in FPGAs, but this approach is not recommended due to the difficulty in generating a glitch-free clock using LEs (logic elements). Altera's Stratix II and Cyclone II devices use clock control blocks that include an enable signal. A clock control block is a clock buffer allowing dynamic enabling or disabling of the clock network and dynamic switching between multiple sources to drive the clock network. The dynamic clock enable feature provides internal logic control of the clock network. When a clock network is powered down, all logic fed by that clock network does not toggle, thereby reducing the overall power consumption of the device.

Designs with many glitches consume more power due to faster switching activity. Glitching takes place when unnecessary temporary logic switching occurs at the output of combinatorial logic prior to the expected output becoming stable. These glitches propagate to subsequent logic and create unnecessary switching activity, increasing power consumption. Circuits with many XOR functions, such as arithmetic circuits or cyclic redundancy check (CRC) circuits, tend to have many glitches if there are several levels of combinational logic between registers.

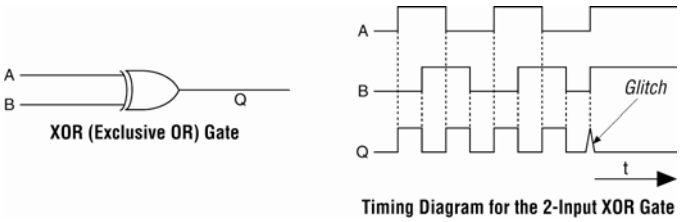

Figure 2. XOR Gate showing Glitch at the Output

Reducing inadvertent glitching of logic within a glitch-prone design significantly reduces dynamic power. One method of reducing this effect is retiming a design to balance the delay between timing critical paths and non-critical paths. Designers can perform this task on their own, but there are software tools that will accomplish the task much more efficiently. For example, certain software can move the placement of registers across combinatorial logic to balance timing. A second method is introducing pipelining to reduce the combinatorial logic depth. Pipelining reduces design glitches by inserting flipflops into long combinational paths. Flipflops do not allow glitches to propagate through combinational paths. Therefore, a pipelined circuit tends to have less glitching.

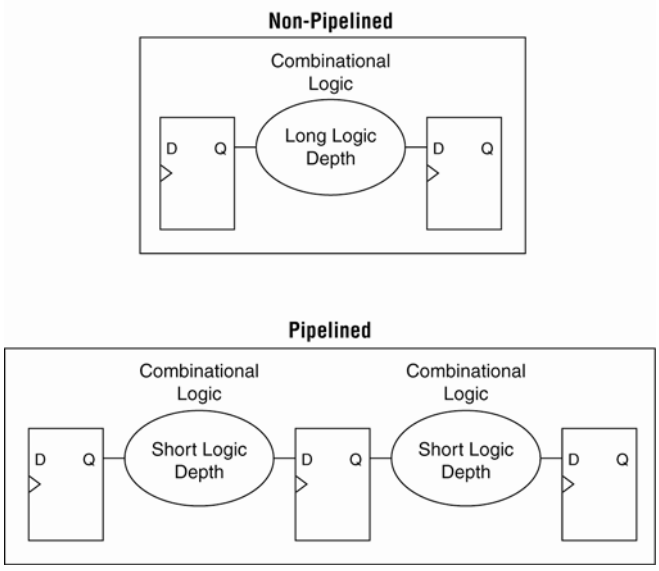

Figure 3. Pipelining Example

Pipelining has an additional benefit of allowing higher clock speed operations, although it does increase the latency of a circuit (in terms of the number of clock cycles to a first result). If there are not many glitches in the design, the disadvantage is that pipelining may increase power consumption by adding unnecessary registers. Pipelining can also increase resource utilization. For glitch-prone designs, pipelining can reduce dynamic power consumption by as much as 31 percent in some Altera devices.

7 System I/O

When considering I/O standards, lower voltage and non-terminated standards typically yield lower power ratings. Any reduction in voltage has a quadratic effect on power. Static power is significant for terminated standards. When the I/O buffer drives a high signal, the I/O supply voltage delivers power consumed in the external termination resistors. When driving a low signal, the chip dissipates power from the external termination voltage. Use the lowest drive strength I/O setting that meets the speed and waveform requirements to minimize I/O power when using terminated standards. Differential I/O standards such as LVDS (Low Voltage Differential Signaling) with a low switching voltage (typically 350 mV) provide lower power consumption, better noise margins, smaller electromagnetic interference and overall better performance.

8 Power Analysis Tools

Having accurate power estimation tools that are quick and easy to use allows designers to realistically hit power budget numbers and incrementally improve designs efficiently. Without realistic data, both in early power estimation tools and data sheets, the design phase will be seriously hindered. Obtaining early estimation tools in spreadsheet form allows designers to gather early estimates of power requirements before starting the design phase. As the design progresses, designers can load placed-and-routed designs into power estimator programs and receive more accurate power consumption estimates. The best tools allow simulation files to be integrated seamlessly into the power tools, acquiring an accurate representation of switching power. If a simulation has not been done, the power analysis tools will intelligently estimate toggling within the design. Altera's Quartus II PowerPlay Power Analyzer tool utilizes design signal activity information and other important design factors affecting power consumption to accurately estimate design power as shown in Figure 4.

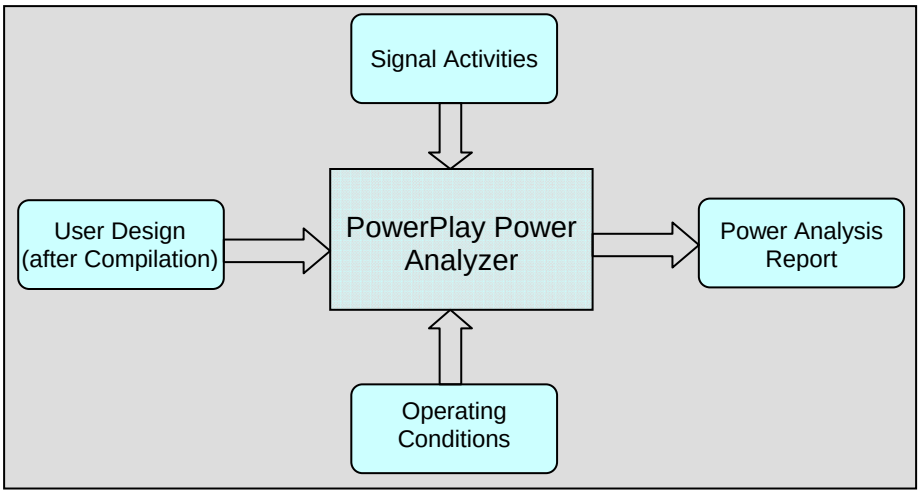

Figure 4. Quartus II PowerPlay Power Analyzer High-Level Flow

PowerPlay Power Analyzer requires the design to be fully compiled in order to extract the target device as well as place and route information of the device. When combined with signal activity information and operating conditions of the device comprehensive power consumption reports are produced. These reports facilitate both thermal and power supply planning requirements. Additionally, these reports also pinpoint which device structures and even design hierarchy blocks are dissipating the most thermal power, thus enabling design decisions that reduce power consumption. This provides very high quality power estimates and benchmarking data to be obtained using Quartus II PowerPlay Power Analyzer with an estimated accuracy within 20 percent of device measurements.

9 Conclusion

Power consumption of FPGAs is highly dependent upon device design. No single methodology represents the perfect solution for low-power design. Designing for lower power involves trade-offs. Designers must carefully weigh performance, ease-of-use, cost, density and power when conducting FPGA design. These design criteria must be evaluated both individually as well as together to achieve efficient power optimization without sacrificing performance.

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
http://www.altera.com