Hybrid Techniques Reduce Dynamic Power Consumption

July 2010

Author

Cary Chin
Synopsys, Inc.

Introduction

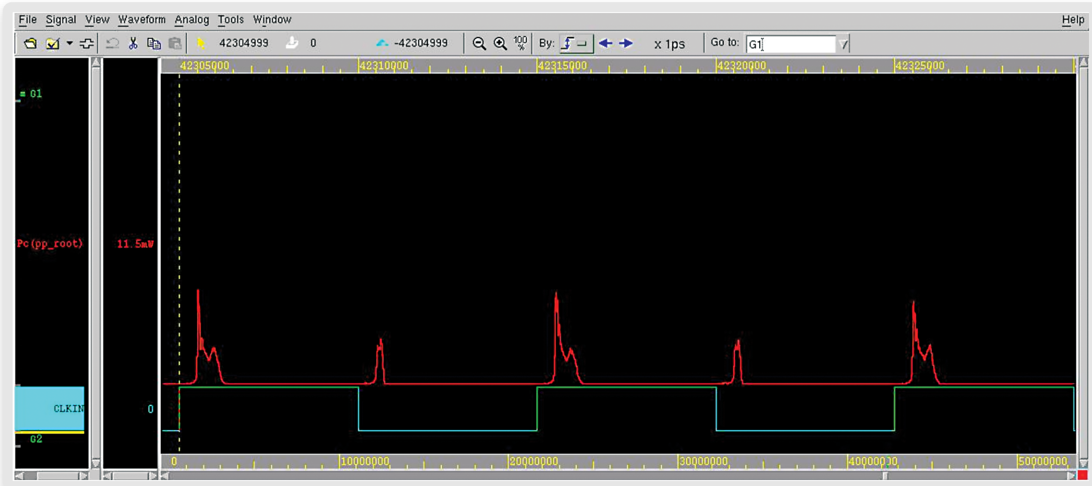
With the predominance of mobile devices, rising energy costs, and an awareness of green practices, power consumption has become a major concern for design engineers. When power consumption is analyzed, it breaks down into two main components: static or leakage power, which occurs naturally when components are idle and powered on; and dynamic power, which is the power consumed when components are switching. While both static and dynamic power remain important targets for power reduction, this paper will focus on only dynamic power, and explore ways to improve that metric.

A Secondary Concern in Traditional Design

For decades, designers have used synthesis-based design rules, which yielded synchronous clock-based architectures. With synthesis-based methodologies, all logic transitions are dictated by a master clock, which is distributed via a clock network. If the longest logic path exceeds the cycle time of the master clock, then either the clock has to be slowed down, or the long path has to be sped up. To optimize this trade-off, designers use static timing analysis tools to compute the longest paths, and then optimize logic until they achieve their target frequencies.

This is still the basic methodology on which today's designs are built. It works well, optimizes for speed at every technology node, and is extremely reliable. However, its main drawback in today's design arena is that it treats area and power as second and third order considerations. But with unprecedented consumer demands for smaller, faster, feature-laden devices, designers can no longer regard area and power as minor design goals. Both of these metrics must now be scrutinized and improved to satisfy today's design requirements.

When we analyze synchronous design with dynamic power in mind, we can identify two main causes for excessive consumption. One of these is clock distribution, which ensures that all logic is synchronized to a master clock. Clock skew across the entire clock structure has a direct impact on chip performance, so it is imperative that clock delays and skew be minimized. While this architecture allows designers to push frequencies (and therefore performance) ever higher, it can consume anywhere from 20% to 60% of a chip's power, and require a large amount of real estate. In short, this methodology is effective but expensive, particularly for power.



Synchronous design also implies that all transitions in the circuit "bunch up" at the clock edges. At every active clock edge, for example, all transitions at the beginning of a logic path are launched synchronously, and then slowly die down as conditions are satisfied along the combinatorial logic chain. Graphically, this phenomenon could be represented as a sawtooth wave. Most of the dynamic power is consumed at the launching clock edge until, at the end of the path, almost no power is being consumed. If the timing is fully optimized, the last transition should die down just before the next clock is launched.

Though this methodology works well for timing, the maximum amount of power needed at any given time would be significantly lower, as much as 30% to 50% lower, if the transitions were better distributed across the cycle and the sawtooth wave were smoothed out. This is because the power lines wouldn't need to handle the power surge of so many elements switching at the beginning of each clock cycle, resulting in smaller area and less power dissipation.

The simple, but impractical, resolution to this problem would be to generate custom logic with appropriate delays to achieve the desired function at each output, and could be tuned to distribute power nicely. This would also be very fast, similar to the performance of a printer dedicated to a single computer as opposed to one connected to a network with multiple users. However, such an architecture would obviously require an enormous amount of area, logic, and therefore power, to function properly. Obviously, there must be better ways to meet timing, performance, power and area goals, while still adhering to the tested and efficient methodology of synthesis-based design.

Methods to Improve Power Efficiency

Clock gating is the most commonly used technique to achieve better dynamic power performance in a synchronous design environment. While gating clocks is in itself a violation of traditional synchronous design rules, tools have exploited this "loophole" with great success for many years. With clock gating techniques, a flip-flop won't receive a clock signal unless the data input to that flip-flop is changing, avoiding the power that is dissipated internal to the flip-flop cell, and any portions of the clock tree that are gated as well. Today's tools heavily exploit clock gating to achieve significant dynamic power reduction.

But that's just the beginning of what can be done. In any complex cone of logic, many transitions are blocked as the paths travel further down the cone. These cannot be suppressed through clock gating. Some of these are candidates for "data gating," where logic that transitions without impacting the circuit's output is identified. Common examples might be operator inputs that are not used during particular times, or memory read addresses that are changing during the cycle. Quieting these types of transitions significantly reduces the number of power-consuming transitions in the circuit, with minimal impact on timing. The circuit simply becomes more power efficient.

DesignWare minPower Components: Minimizing Power Consumption

To further improve efficiency, designers must use innovative techniques at the architectural level. Synopsys' DesignWare® minPower Components, for example, are designed to operate with the lowest possible power in many different environments. Architecturally, these components are implemented using the most power-efficient configurations. They are designed to implement their respective functions while minimizing spurious transitions that use power without producing results. By allowing Design Compiler® Ultra (DC Ultra) to refine architectural selections in the implementation process where the local timing and switching activity can be considered, minPower components can provide an optimal architectural solution for all operating scenarios.

Further, analysis shows that datapath circuits often consume large amounts of dynamic power due to their large circuit size and high switching activities. They are especially impactful because they are often a major portion of the circuits that must be powered on for an extended time. To lower the power consumption of these circuits, the DesignWare minPower Components include a collection of datapath architectures with transition probability costing. When using DesignWare minPower Components IP within a DC Ultra flow, it enables the datapath generator to calculate the power effect of each architecture and encoding decision. This enables DC Ultra to generate datapath structures that can suppress switching activities, limit glitch generation and propagations, and utilize a higher percentage of low leakage cells.

The DesignWare minPower Components are also equipped with power models that take advantage of transition probabilities. When supplied with actual switching activities, either user-defined or from simulation vectors in the Switching Activity Interchange Format (SAIF), the DesignWare minPower Components can configure optimal datapath architectures according to the switching profile. It will reorder the datapath tree and change the operand encoding in an optimized way to suppress the switching and glitches to reduce power.

These high-level optimization technique allows designers to capture power saving opportunities based on the unique characteristics of each individual design. The datapath architectures are also designed to be datapath gating friendly, enabling designers to turn off switching in the entire datapath block when data are invalid. The datapath architectures are designed with structures that can be configured with embedded gating logic, eliminating the requirements for external isolation gates, which often degrade the timing characteristics.

In addition, utilizing DesignWare minPower Components is transparent to existing low power flows. Clock gating and low power synthesis, as well as utilization of multiple threshold voltage cells for leakage minimization, can all still be applied as well, and result in even more power efficient designs.

Summary

Even with existing techniques in place and new ones constantly evolving, dynamic power remains a big concern for low power design. Ultimately, any of a chip's useful functions are achieved through logic transitions, and these transitions require power. However, power efficiency is of such vital importance to today's complex chips that designers must constantly strive to find new ways to optimize for maximum benefit with minimum power. Traditional synchronous design methodology hasn't run out of gas yet, but it's increasingly clear that a hybrid solution incorporating non-synchronous ideas, data sensitivity, and higher level approaches is a good compromise, and one that must be fully investigated.

SYNOPSYS®
Predictable Success Synopsys, Inc. • 700 East Middlefield Road • Mountain View, CA 94043 • www.synopsys.com