

The Power of FPGA Architectures

The present and future of low-power FPGA design.

by Tim Tuan
Staff Research Engineer
Xilinx, Inc.
tim.tuan@xilinx.com

Steve Trimberger
Distinguished Engineer
Xilinx, Inc.
steve.trimberger@xilinx.com

Reducing power consumption in FPGAs delivers numerous benefits such as better reliability, lower cooling cost, simpler power supply and delivery, and longer battery life in portable systems. Designing for low power consumption without compromising performance requires a power-efficient FPGA architecture and good design practices to leverage the architectural features.

In this article, we'll present a brief overview of FPGA power consumption, current low-power features, user choices that impact power, and look at recent low-

power research for insight into future trends on power-efficient FPGAs.

Power Components

FPGA power consumption has two components: dynamic power and static power. Dynamic power is dissipated when signals charge capacitive nodes. These capacitive nodes may be inside logic blocks, routing wires in the interconnect fabric, external package pins, or board-level traces driven by chip outputs. Total FPGA dynamic power is the combined power from charging all capacitive nodes.

Static power, on the other hand, has nothing to do with the activity of the circuit. It is dissipated either as transistor leakage current or as bias current. Total static power is the combined total of each transistor's leakage power and all bias currents in the FPGA. As transistor dimensions shrink, dynamic power improves because it depends on the side of driven

capacitances. However, static power increases because smaller transistors leak more. As a result, static power is becoming an increasingly large fraction of the overall power consumption of integrated circuits.

Power consumption is highly dependent on supply voltage and temperature, as shown in Figure 1. Reducing the FPGA supply voltage results in a quadratic reduction in dynamic power and an exponential reduction in leakage power. Increasing temperature results in an exponential increase in leakage power. For example, an increase from 85 °C to 100 °C increases leakage power by 25%.

Power Breakdown

Let's examine the breakdown of total FPGA power to understand where most of the power is consumed. FPGA power is design-dependent; namely, it depends on the part family, clock frequency, toggle rate, and resource utilization. In this analysis, we'll use the Xilinx® Spartan™-3

XC3S1000 FPGA as an example and assume that clock frequency is 100 MHz, the toggle rate is 12.5%, and resource utilization is typical as determined by benchmarking many real designs.

Figure 2 shows the breakdown of the XC3S1000 FPGA in terms of active power and standby power. Active power is reported as the power of an active design at high temperature, which comprises both dynamic and static power. Standby power is the power of an idle design,

Configuration and clock circuitries consume nearly half of total standby power, largely because of bias currents. Therefore, total chip power reduction must come from multiple solutions that address all major power-consuming parts.

Designing for Low Power

Many power-driven design techniques are employed in the design of an FPGA. Because configuration memory cells can make up a third of the transistors in an

multiplier built from FPGA fabric. As manufacturing variations can lead to large spreads in leakage current, Spartan-3L FPGAs screen for low-leakage parts to effectively provide a part with 60% less core leakage power.

Beyond what is designed into the FPGA, many design choices also affect FPGA power consumption. Let's examine some of these choices.

Power Estimation

A key step in low power design is power estimation. Although the most accurate way to determine FPGA power consumption is through hardware measurements, estimation can help identify high-power modules and is useful for power budgeting early in the design cycle.

Several tools are available to aid power estimation: Xilinx Power Estimator (XPE) and Xilinx Power Analyzer (XPA). XPE gives quick power estimates through a simple user interface. Its estimation is based on high-level statistics such as the number of logic cells, number of block RAMs, and average switching activity. XPA gives detailed power estimates based on simulated switching activity and exact utilization statistics generated post-place and route. Choosing the right tool depends on what design information is available and what level of accuracy is needed.

As shown in Figure 1, some external factors have exponential effects on power consumption; slight changes in environment can cause large changes in estimated power. Exact accuracy is difficult to achieve through power estimation tools, but they nevertheless provide excellent guidance for power optimization by identifying high-power blocks.

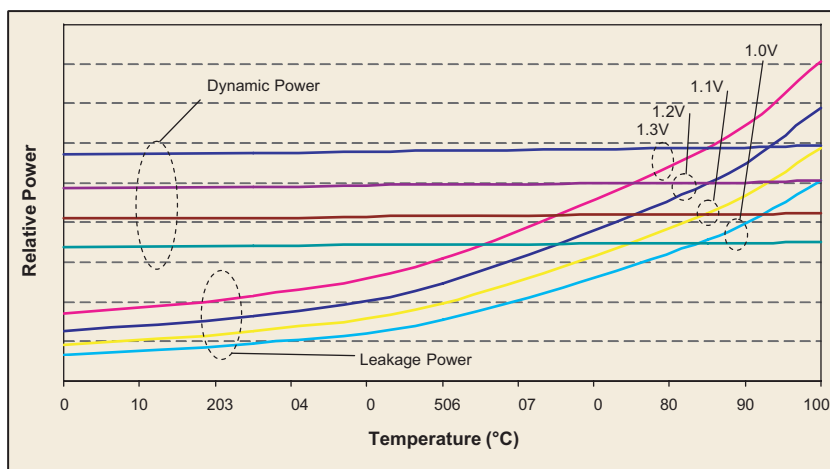


Figure 1 – Power dependency on voltage and temperature

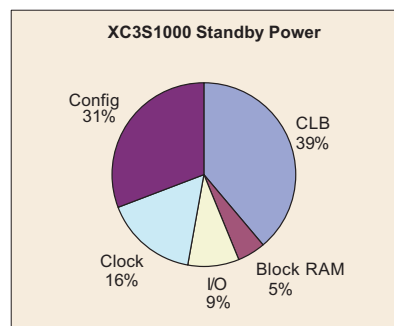
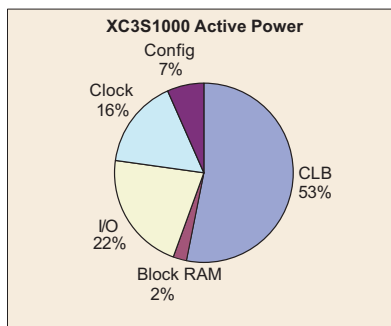


Figure 2 – Breakdown of a Spartan-3 XC3S1000 FPGA's typical power consumption

comprising static power at nominal temperatures. Not surprisingly, CLBs make up the largest component in both active power and standby power, but other blocks contribute significant power as well. I/Os and clock circuitries make up a third of total active power, which would be even higher if you used high-powered I/O standards.

FPGA, Xilinx uses a low-leakage “midox” transistor in Virtex™ families to reduce leakage in memory cells. Transistors with longer channel length and higher thresholds are also used throughout to reduce static power. Dynamic power consumption is addressed with low capacitance circuits and custom blocks. A multiplier in a DSP block consumes less than 20% of the power

Voltage and Temperature Control

As shown in Figure 1, both lower voltage and lower temperature can lower leakage current significantly. Just a 5% lower supply voltage can lead to 10% lower power. Supply voltages can be easily adjusted by changing the power supply configuration. Current FPGAs do not support wide voltage adjustment; the recommended voltage range is typically +/- 5%. Junction temper-

ature can be reduced by using cooling solutions such as heat sinks and airflow. A 20 °C reduction in temperature can lead to a more than 25% reduction in leakage power. Lower temperature also exponentially improves chip reliability. Studies have shown that a 20 °C reduction can lead to a 10x improvement in overall chip life.

Suspend and Hibernate Modes

Spartan-3A FPGAs provide two low-power idle states. In suspend mode, the circuitry on the VCCAUX power supply is disabled to reduce leakage power and eliminate bias currents, reducing static power consumption by more than 40%. Chip configuration and circuit state are retained. Exiting from suspend mode is triggered by asserting an awake pin. The process takes less than 1 ms.

The hibernate mode in Spartan-3A devices allows all power regulators to be switched off to achieve zero power. To restart, the part must be re-powered and reconfigured, which takes tens of milliseconds. While powered off, all I/Os are in a high impedance state. If any I/Os need to be actively driven during hibernate mode, the corresponding I/O bank must remain powered, consuming a small amount of standby power.

I/O Standard Choices

Different I/O standards have considerably different levels of power consumption. You can achieve significant reduction by choosing a lower power I/O standard at the expense of speed or logic utilization. For example, LVDS is a high power consumer, with 3 mA per input pair and 9 mA per output pair. Thus, from a power perspective, LVDS should only be used when it is required by system specifications or when the highest performance is needed.

A lower power, higher performance alternative to LVDS is HSTL or SSTL, but these still consume 3 mA per input. When possible, we recommend LVCMOS inputs instead. Lastly, DCI standards are high power consumers. When connecting to memory devices such as RLDRAM, consider using ODT on the memory and LVDCI on the FPGA to save power.

Embedded Blocks

Using embedded blocks instead of programmable fabric can save a substantial amount of power. Xilinx FPGAs have a number of embedded blocks such as the PowerPC™ hard-core processor, DSP slices, ChipSync™ technology, embedded Ethernet MAC(s), FIFOs, and SRL16. Embedded blocks are custom-designed; hence they are smaller and have less switching capacitance than programmable logic. These blocks have between 5x and 12x lower power than equivalent programmable logic implementations. Using embedded blocks can lead to static power reduction if the design becomes smaller and fits into a smaller part. A potential pitfall is that very simple functions may not be more efficiently implemented using large embedded blocks. This can be easily avoided by checking both implementations using XPE.

Clock Generators

Power considerations in clock generation can save power. Digital clock managers are widely used to generate clocks with different frequencies or phases. However, DCMs consume a non-trivial amount of power off VCCAUX; therefore, their use should be limited when possible. A single DCM can often generate multiple clocks by using multiple outputs such as CLK2X, CLKDV, and CLKFX. This is a lower power solution than using multiple DCMs for the same function.

Block RAM Construction

Multiple block RAMs are often combined to create a single large RAM. How this is done can have significant power implications. The timing-driven method is to access all RAMs in parallel. For example, a 2k x 36 RAM would be constructed out of four 2k x 9 RAMs. The access time of this larger RAM is the same as that of a single block RAM; however, the power consumption per access is that of four block RAMs.

A low-power solution is to construct the same 2k x 36b RAM out of four 512 x 36b RAMs. Each access would be pre-decoded to select one of the four block RAMs to access. Although the access time is

increased because of pre-decoding, the power-per-access of the larger RAM is approximately the same as that of a single block RAM.

Low-Power Research

In recent years, Xilinx Research Labs has studied various techniques for low-power FPGA design. These research items are not currently implemented in commercial FPGAs, but they do offer an insight into what FPGAs might look like in the future.

Voltage Scaling

As mentioned earlier, reducing supply voltage is one of the most effective ways to reduce power, and the associated performance degradation is acceptable to many designs that do not need peak performance. However, current FPGAs operate in a small range, and one of the limitations is found in some voltage-sensitive circuits.

At Xilinx Research Labs, we redesigned the CLB circuits to operate at a much lower voltage, enabling a substantial trade-off of performance headroom for lower power. For example, for a 90-nm process, a 200-mV reduction could bring a 40% power reduction at the expense of 25% peak performance; a 400-mV reduction could bring a 70% power reduction at the expense of 55% peak performance.

Fine-Grain Power Switching

One of the unique overheads of programmable logic design is that not all on-chip resources are used in any given design. However, they remain powered and contribute to total power in the form of leakage power. Block-level power switching can shut off power to individual unused blocks. Each block is coupled to the power supply through a power switch. When the switch is closed, the block is functional. When the switch is open, the block is effectively disconnected from the power supply, and so consumes 50-100x less leakage power. The granularity of the power switches may be as small as individual CLBs and block RAMs. In our design, these power switches can be programmable through configuration bitstreams or controlled by the user directly or through an access port. Benchmarking real

designs shows that fine-grain power switching can reduce leakage power by 30%.

Deep-Sleep Mode

One of the key requirements in portable electronics is to consume little or no power when the device is idle. In Spartan-3A FPGAs, you can achieve this by entering hibernate mode, which requires external control, is slow to wake up, and cannot restore the FPGA state. In our design, we dynamically control the fine-grain power switches described previously to power down all internal blocks while leaving the configuration and circuit state storage elements powered. The resulting state is a deep-sleep mode where leakage power is 1%-2% of nominal, the FPGA state is saved, and exiting this mode takes microseconds.

Heterogeneous Fabric

The maximum clock frequency of a circuit is determined by the delays of its timing-critical paths. Non-critical paths can be slower without affecting total chip performance. In large systems, a few blocks may be speed-critical, such as the data path in a processor, while other blocks may be non-critical, such as caches.

Today, FPGAs are homogeneous in terms of power and speed; every CLB has the same power and speed characteristics. Power can be saved in a heterogeneous architecture, where some blocks are low power (and slower), by implementing non-critical blocks in the low power blocks. Doing so does not impact overall chip performance, as the timing-critical blocks are not compromised.

One method of creating a heterogeneous architecture is to distribute two core power rails, a high-voltage rail (VDDH) and a low-voltage rail (VDDL). Each part of the FPGA selects from one or the other using embedded power switches and correspondingly takes on high-speed or low-power characteristics. Voltage selection is

complete once you have a detailed timing of the design, so only the non-critical blocks should operate from VDDL.

Another method of creating a heterogeneous architecture is to divide the FPGA into different regions that are pre-

will justify the added design complexity. Of course, FPGA users never see the different voltages of their internal signals.

Figure 3 depicts an FPGA architecture with some of the preceding concepts. The programmable fabric is heterogeneous,

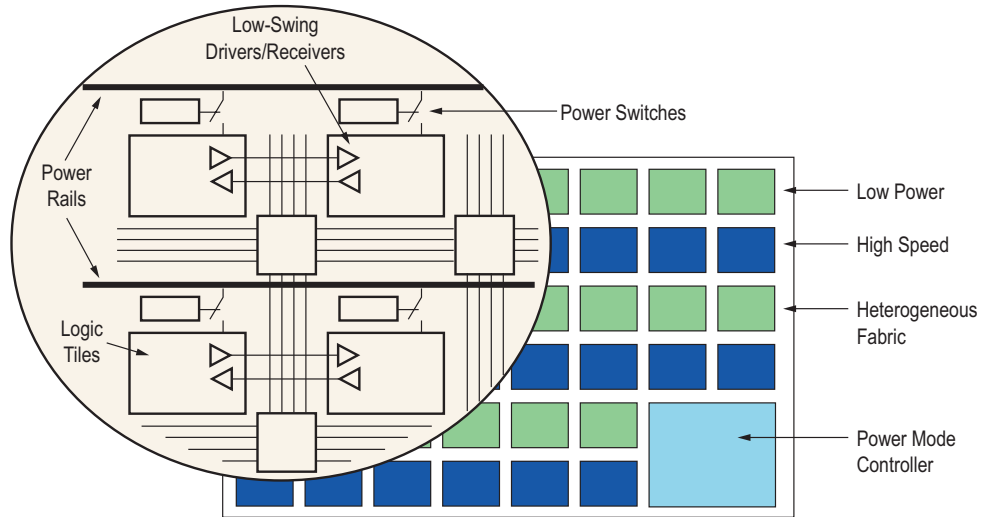


Figure 3 – Concept architecture with various solutions for power reduction

fabricated to be high speed and low power. Regions can be implemented with different supply voltages, different thresholds, or through a number of other design trade-offs. To avoid performance degradation, design tools must map timing-critical parts of the design into the high-speed regions and non-critical parts into the low power regions.

Low-Swing Signaling

As FPGAs increase in capacity, more and more power is consumed in on-chip programmable interconnect. An effective way to reduce this communication power is to use low-swing signaling, where the voltage swing on the wire is much lower than the supply voltage. Low-swing signaling is commonly found today in scenarios where communication takes place over high-capacitance wires such as buses or off-chip links. Low-swing drivers and receivers are more complex than CMOS buffers, so they consume more silicon area. However, as on-chip interconnect grows to be a larger portion of overall power, the power benefits of low-swing signaling

comprising both high-speed and low-power regions. An on-chip power-mode controller manages various power-down modes, be it deep sleep, suspend, or hibernate. Within the fabric, each logic tile can be powered off using dedicated power switches. Communication through the routing fabric goes through low-swing drivers and receivers to reduce interconnect power.

Conclusion

Beyond the power optimizations currently used in the design of modern FPGAs, a number of user design decisions can yield significant power benefits. Looking forward, more aggressive architectural solutions are available to contain power consumption in future technology generations and enable new FPGA applications.

In addition to the solutions described in this paper, Xilinx is also engaged in various software power optimization research and development work. These efforts are described in the article, “Optimizing FPGA Power with ISE Design Tools,” also in this issue of the *Xcell Journal*.