Power

Suddenly, We Care

For years it was like a slogan. "FPGAs are nice, but they're power hogs." For the customers that kept the lights on, however, buying thousands of FPGAs for backplane-based, rack-mounted equipment with monster power supplies and plenty of cooling, considerations like performance, I/O, and density far outweighed power as a design-in concern. If a new FPGA family offered a 50% performance increase or doubled the LUT count over the previous generation, damn the heatsinks and full-speed ahead. Designers rolled FPGAs in with reckless abandon.

Today, however, forces are conspiring to bring power concerns off of the back burner and into the forefront of FPGA design consideration. As the day approaches when firing up your FPGA-based network router threatens to cause brown-out conditions in neighboring counties, even the traditional FPGA consumer may think it's time to cool things down a bit. With high-end devices today based on 90nm processes, dynamic power consumption per gate has continued to drop, but it is largely offset by increased density. Additionally, static or leakage current as a percentage of total power consumption is on the rise due to smaller geometries. The net result is a generation of devices that require careful attention to power, from their initial design through their final application in FPGA-based systems.

"Even the fellows in wireless base stations and wired-access markets are feeling the power problem," says Anil Telikepalli – Marketing Manager for Virtex Solutions at Xilinx. "When infrastructure equipment is outside in sun, like the green boxes you see alongside the roads, temperatures in the chassis can reach 50-60 degrees C, so junction temperatures can easily rise to 85 degrees. At that temperature, static power goes up significantly, generating even more heat."

Power concerns are not limited to high-end devices, either. The new generation of low-cost FPGAs like Xilinx's Spartan-3, Altera's Cyclone-II, Lattice's EC/ECP/XP, Actel's ProASIC-3, and QuickLogic's Eclipse II families are all aimed at high-volume, low-cost applications. The target systems for these devices often run on batteries or have limited cooling capability or restricted power supplies, which makes lower power operation an absolute necessity.

Specifically targeting handheld, battery-powered applications, QuickLogic's Eclipse II holds the title as the lowest-power full-FPGA device on the market. "Everything has a power budget, but we're focusing on applications where the FPGA is a significant chunk of that budget," says Brian Faith, Sr. Director of Product Marketing at QuickLogic. "Our focus is on mobile platforms and on connecting embedded processors to embedded I/O systems like wireless LAN. ASSP developers come out with PC-based chipsets, but embedded applications developers want to connect to those same chipsets. We help them bridge that gap. Our devices become a negligible part of the power picture when the wireless LAN is active, and they enable power management to turn off clocks and components when the LAN is offline. We can even turn off other devices in the system and can de-couple I/O from the processor using our bridge."

In planning the power budget for your FPGA design, there are three types of power to ponder: startup power, dynamic power, and static power. The first of these, startup power, has two components that warrant consideration. As VDD ramps up to the correct voltage, the unknown state of SRAM cells on an SRAM-based FPGA can cause a current spike known as inrush current. This inrush current has been drastically improved over the past few generations of FPGAs by careful attention to power-up sequencing, but still it can rise to a level that warrants designer attention. The second startup hit in SRAM FPGAs is the increased current draw during the configuration process as the routing and look-up table (LUT) configurations are read from memory into the device.

Non-volatile devices such as flash and antifuse FPGAs offered by Actel and QuickLogic have negligible startup spikes because they do not require re-configuration at power-up. For many SRAM-based FPGA devices, however, peak current draw is reached during the startup phase and may determine or limit your power supply selection. "Our devices essentially have only static and dynamic power components," says Mike Holmlund, Product Marketing Manager at Actel. "Because they're statically configured and live at power-up, non-volatile devices don't have inrush and configuration current issues like SRAM-based FPGAs. We've seen a lot of interest from customers developing battery-powered applications because our technology is a good fit. Also, because of the static configuration, the standby current in a non-volatile FPGA is significantly lower than in a similar-density SRAM-based device."

Once your FPGA is powered up, configured, and ready to run, you're left with two flavors of power consumption, dynamic and static. Dynamic power is the energy that is consumed by your device when it is doing useful work. The culprit behind dynamic power consumption is the momentary dash of current through the transistors in a logic gate as both are temporarily turned on at the same time. Dynamic power, then, depends only on capacitance and the number of these events that happen in a given interval of time on your chip. Of course the number of events depends in turn on gate count, frequency, and toggle rate.

Dynamic power is the component over which you, as the designer, have the most direct control. Dynamic power is completely design-specific, so architectural changes that alter toggle rates, logic utilization, or operating frequency have a direct impact on dynamic power. When you study design techniques for reducing power or look at tools for low-power design, the target is almost always dynamic power consumption.

The arch-villain in the world of FPGA power consumption is static power. In examining static power, we are able to see a graphic and unfortunate demonstration of the remarkable leverage of positive feedback loops on functions that already have exponentially bad tendencies. Static power, as its name implies, is based on the current that is drawn by your FPGA when it is powered up, configured, and doing nothing. Deep within your FPGA, very thin oxide layers leak current even when transistors are not switching. As these oxide layers get thinner (as in when moving to smaller process geometries), the leakage increases. As the number of transistors on a device goes up (as in when moving to smaller process geometries), the leakage increases. When the junction temperature rises (as happens as a cumulative side-effect of moving to smaller process geometries), the leakage increases exponentially.

Now the real fun begins! With increased leakage, more heat is generated, raising junction temperatures even more, which moves us farther down the exponential curve of increased leakage current.

At the 90nm process node, FPGA companies were deeply concerned about the implications of leakage current, the possibility of thermal runaway, and the practicality of large FPGA devices based on that technology. As a result, bringing up new FPGA families on 90nm has been perhaps the most challenging task faced by programmable logic companies to date. Everything from the fundamental architecture of the FPGA to exotic process techniques was considered, and complex compromises were weighed between performance, functionality, yield, cost, and power.

Xilinx adopted a triple-oxide approach with their Virtex-4 family. In previous generations, they had employed two different oxide thicknesses, one in the core logic for fast switching, and a thicker one in the I/O area where increased drive strength was required. With Virtex-4, they added a third thickness for transistors involved in the routing and configuration circuitry. The thicker oxide reduces leakage, and these transistors did not require rapid toggle rates, as they generally remain in a constant state from configuration on.

Altera's big leap in moving to 90nm included a complete overhaul of their fundamental logic element. By dropping the traditional 4-input LUT in favor of a 7-input variable adaptive logic module, they decreased logic granularity and reduced the routing-related power overhead for most designs. In addition, they adopted process changes to reduce static power such as low-K dielectric, and longer transistors with increased Vt for non-performance-critical paths.

In their low-cost Spartan-3 and Cyclone II device families, Xilinx and Altera were less aggressive in their design changes for low power. The smaller, less expensive devices didn't require the same level of optimization as the flagship high-density device families, and power reduction took a back seat to cost optimization for high-volume markets. Late last year, however, Xilinx introduced a more power-efficient version of its 90nm Spartan-3 line called Spartan-3L. Spartan-3L has lower quiescent current than standard Spartan-3 devices, and it offers two power-saving modes for taking the device off-line during periods of inactivity.

If you're selecting an FPGA with an eye on power, it pays to think carefully about the details of your design constraints, and not just jump on the option with the smallest overall number. For example, is your design running on batteries? Do you have a current restriction such as that imposed by internal USB power? Will your device have a relatively small duty cycle that can benefit from hibernation, power, or clock gating? Do you have limited cooling capability, or are you concerned about junction temperatures? Can you trade off operating frequency for lower dynamic power consumption? Each of these answers might take you in a different direction in selecting the appropriate FPGA for your application and in optimizing your design for power budget management.

If your design has a short duty cycle, for example, Lattice's XP family offers fast reconfiguration of the SRAM-based FPGA fabric from on-chip flash memory. This means you

can power down the device during standby operation, and quickly and easily power it back up again when it needs to go active. "XP powers up in less than a millisecond, so it can swap out faster," Says Gordon Hands, Strategic Marketing Manager at Lattice Semiconductor. "You could easily implement a power saving design that does, say, a 10% duty cycle. Regular SRAM devices can take hundreds of milliseconds to reconfigure, so it's not practical to put them on standby for very short periods."

On the tool side, every FPGA vendor offers software that will help you estimate power consumption. Most of these offerings can generate very early estimates based on a spreadsheet-like system. You supply estimates of design parameters like logic, memory, and I/O utilization, clock frequencies, toggle rates, and operating temperatures. The tool will come back with an estimate of power consumption for those conditions. While these early estimates are the least accurate, they are ironically the most useful, as you can use them as the basis for deciding what micro-architecture will work best, and which parts of your logic design should be in or out of your FPGA.

"We added significant capability with the introduction of our PowerPlay tools in Quartus 4.2," says Chris Balough, Director of Software and Tools Marketing at Altera. "Particularly useful with PowerPlay is the notion that power estimation can contribute to architecture closure as well as timing and power closure. A systems designer creating a new consumer product on a short time-to-market window may have trouble hitting an overall power budget, so PowerPlay can help him understand very early what an FPGA can do. It allows a 'what-if' analysis on variables like clock frequency and cooling options, even when the design is at the 'back of a napkin' level."

Once you've generated synthesizable RTL code and have vectors that exercise your design in a realistic way, you can get much more accurate power estimates, but then it's often too late for many of the big changes that could affect power consumption. The more accurate estimates are primarily useful for checking your work at the end to be sure you got the results you expected.

Beyond power estimation there is power optimization. While power optimization tools have been available for ASIC design for a while, automated power reduction is still a relatively immature science for FPGA. In ASIC, there are a fair number of readily-applied techniques such as clock and power gating, buffer sizing, and voltage scaling that can be employed by automated power optimization tools. In FPGA, however, the options are somewhat limited by the underlying architecture, and the opportunity for improvement is reduced by the dominating presence of static power related to configuration circuitry. Nevertheless, since power is becoming a hot topic in FPGA design, there is serious research underway into techniques and tools for putting programmable logic on a power diet.

"We are strong believers that you need to do architectural development at the same time as software tool development," says Professor Jason Cong, Professor and Co-Director of the VLSI CAD Laboratory at UCLA and Chief Technologist for Magma Design Automation. "We first did a quantitative evaluation of power dissipation in current FPGAs with our fpgaEva-LP

tool. In that effort we wanted to measure both static and dynamic power in FPGA interconnect and logic. Once we understood the power profile of existing architectures, we were able to look at potential architecture changes to reduce power consumption." The team evaluated the impact of architectural changes such as changing the size and grouping of basic logic cells such as LUTs, adding architectural support for clock and power gating, and supporting multiple VDD voltage levels for different parts of FPGA designs with different performance requirements.

Beyond these architecture changes, the team is looking into design-specific improvements that can be done, particularly starting at a behavioral level of abstraction. "It requires the whole tool chain to address the power problem effectively," continues Cong. Once the FPGA architectures are changed to better support power reduction and the tools are in place to take advantage of those changes, Cong estimates that a 2x-5x improvement in power can be achieved.

With the hot competition underway to cool down your design, expect significant progress in power optimization over the next few years. Now that FPGA companies are simultaneously grappling with the challenges of 90nm and even 65nm process geometries and dealing with customers moving FPGAs into new lower-power applications, significant development effort is being poured into both architectural and design tool solutions to the power problem.

Kevin Morris, FPGA and Programmable Logic Journal

April 26, 2005