Power Optimization Design Techniques for FPGA

Sonal B. Prajapati
Vadodara Institute of Engineering, EC, department, Vadodara, India

***Abstract*- A variety of factors – from the micro to the macro, from conserving battery life to lessening global warming – has pushed power conservation rapidly up the list of system designers' concerns. Engineers have ranked power consumption first in recent surveys on key design priorities or as a close second next to performance, density, and cost. FPGAs present unique challenges when it comes to power consumption. Armed with a good understanding of these challenges and new technology, techniques, and tools to meet them, system designers can realize the advantages of an FPGA-based portable system deployment. This is increasingly crucial as FPGAs are depended on more and more to provide flexibility and fast time to market in an expanding universe of applications.**

I. INTRODUCTION

Assessing a given FPGA architecture's suitability for power-sensitive applications today warrants an in-depth examination of the power equation. We can do this by examining FPGA power characteristics and their effects before diving into optimization tools and possible design solutions, which include, among others, partitioning, clock and power gating, and voltage.

Depending upon the type of FPGA technology chosen, many different factors make up power consumption: static, dynamic, power-up (or inrush), configuration, and various low-power modes.

Static and dynamic power are familiar concerns for all IC designers. Leakage current in several forms dominates static power: sub threshold leakage, junction leakage, gate-induced drain leakage, and gate leakage. Dynamic power refers to power consumed during device operationand correlates with such factors as used functional resources (logic blocks, clock trees, embedded RAM, PLLs, and the like), loads and resistive terminations on I/Os, clock frequencies, data patterns and their arrival dynamics, signal activity or toggle rates, and signalstatic probabilities.

II.PROPOSED METHODS

Empowering low-power design

Reducing the supply voltage (VDD) is an effective technique for reducing both dynamic and static power. Dynamic power has a quadratic dependency on supply voltage, while both sub-threshold leakage and gate leakage exhibit exponential dependencies on the supply voltage. However, reducing supply voltage also negatively affects circuit performance. A well-known technique to reap the benefits of voltage scaling without the performance penalty is the use of dual-VDD. The active blocks in the design operate on the normal VDD (or VDDH), while inactive blocks operate on a second supply rail with a lower voltage (or VDDL). While dual- VDD ICs have been successfully used in low-power ASICs and custom ICs [12], no commercial FPGA today uses multiple VDD for power reduction.The difficulty of designing a dual-VDD FPGA is that the optimal VDD assignment changes from one design to another.

Despite the efforts of SRAM-based FPGA vendors to reduce power, these higher power components remain in the market, significantly increasing the overall system power consumption, especially when several FPGAs populate a single board or use power from a common supply on different boards. The impact is greater for systems with frequent on/off cycles. which must be considered when estimating battery life. Thus, when sizing power supplies or selecting batteries for SRAM-based programmable devices, system designers must account for configuration and inrush power dissipation. True flash FPGAs arenonvolatile, do not exhibit inrush or configuration currents, and have lower overall static power, thus making the design task easier and significantly lowering power.(Figure1)

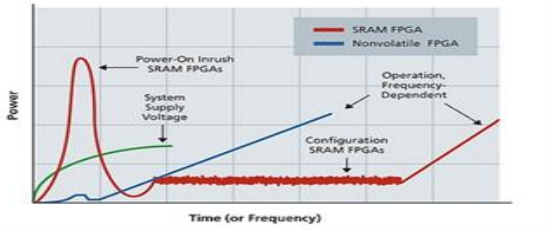


Fig 1: Volatile SRAM FPGA vs. Nonvolatile True Flash FPGA Current

B. Size matters

In the IC design world, minimizing die size is a constant concern, for cost and numerous other reasons. Power can now be added to that list. The smaller the die, the smaller the static power consumption. Choosing the smallest possible die that meets the functional and other demands of the application will make it easier to meet power consumption objectives.

It is also important to choose an FPGA that makes it possible to optimize the use of resources such as RAMs, PLLs, and I/O technologies. FPGA architecture selection should include consideration of any low-power FPGA modes and other power-saving capabilities of dynamic resources such as PLLs, RC oscillators, and I/O banks. For example, given that lower reference voltages save power over the life of a system, choosing an I/O offering that supports both 1.2 V LVCMOS and/or 1.5 V LVCMOS standards makes it possible to utilize higher voltage I/Os if necessary.

C. Watch the clock

The dynamic power consumed by an FPGA is due largely to the charging and discharging activities of capacitive elements, such as logic resources and the interconnecting fabric (Figure 2).

By considering each of the functions in the dynamic power equation, you can lower power consumption. In the clock domain, for instance, you can decide which parts of the design need a fast clock or a slower clock. Switching speed is one component of the dynamic power equation. Logic that is being driven by a fast clock will be switching more frequently than logic that's being driven by a slow clock. The designer knows which portions of the logic require a fast clock and which can be run at a slower speed and can therefore partition clocks according to the functions they control, conserving power.

Dynamic power can also vary widely as a function of placement and routing. For example, as two connected functional instances are placed closer together, the length of the route between the instances may be shortened, which in turn can reduce the capacitive loading of the net and lead to a reduction in power. Today's FPGA development software typically can support Power Driven Layout to automatically accomplish this. Depending on the number of clocks and nets in the design, 25 percent or more overall dynamic power reduction can be obtained.

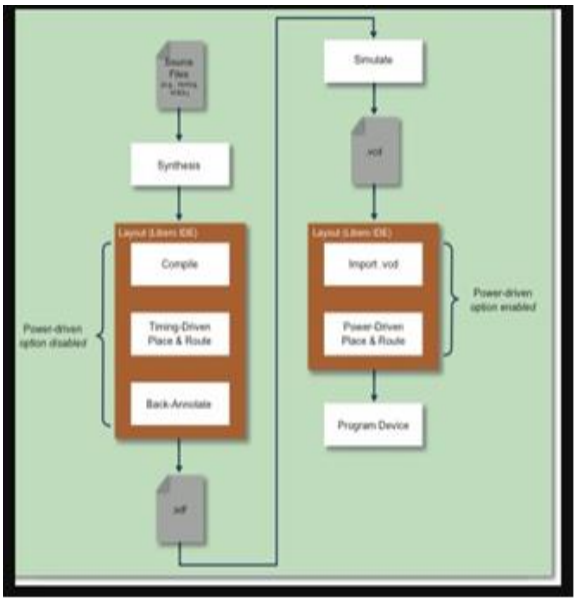
D. Architectural refinements

At the architectural level, it's beneficial to examine the clocking scheme of the design to seek out ways to employ clock gating for the clock tree. If a design is powered up but not clocking a portion of the system, you can reduce dynamic power by not connecting the clock tree to that portion of the design. For example, if a clock serves one function that is needed and another that isn't, a logic signal can be implemented to selectively control what functions are clocked and when, depending on the state of a control variable. Employing logic for clock gating may introduce clock skew however, which must be managed

E. Other power-saving design techniques

Selective power-down simply refers to shutting the power down to certain portions of a chip, or to certain chips on a board. Implement a multisupply strategy in which the power grid of some blocks is decorrelated from others in order to allow for selective shut-down. Power-down or sleep modes within the FPGA architecture can also be deployed to selectively power down blocks when not in use.

Macro optimization can also yield power savings. Some logic elements are offered in multiple versions optimized for high performance, high density, or low power. High-performance macros tend to consume more power than other versions, so power can be saved by deploying high-performance macros only when they are required. For example, a fast adder consumes more power than a slower ripple adder. Examination of the differences between them might reveal that the ripple carry adder consumes about one-tenth of the dynamic power compared to the fast adder. Depending on the speed required for a design and targeted functions, the low-power option might be perfectly adequate. This applies for almost any type of macro, including multipliers, FIFOs, and RAMs.



Time multiplexing and minimum I/O count design partitioning are techniques that can help switch an I/O bank on or off. Minimizing the different types of I/O technologies, ensuring that the right I/O technology is used, and reducing the I/O drive strengths and slew rates are also helpful.

Dynamic voltage scaling is another power-saving design technique. Power scales proportionally with the square of voltage, so reducing supply voltage can significantly impact power efficiency. If system requirements demand more performance than low-voltage I/Os can deliver, utilizing low-voltage I/Os on non-performance-critical pins and higher voltage I/Os for critical signals offers an excellent alternative. Some low-power FPGAs on the market today fully operate from a single 1.2 volt supply for the core and I/Os.

REFERENCES

[1] J E. Ahmed and J. Rose. The effect of LUT and cluster size on deep-submicron FPGA performance and density. In ACM International Symposium on FPGA, February 2000.USA:

[2] Altera. White paper, "Stratix II vs. Virtex-4 Power Comparison & Estimation Accuracy White Paper

[3] R. K. Brayton. Understanding SPFDs: A new method for specifying flexibility. In International Workshop on Logic Synthesis, 1997.

[4] J. K. Author, "Title of paper," in Abbreviated Name of Conf., City of Conf., Abbrev. State (if given), year, pp.

[5] J J. Cong, Y. Fan, G. Han, and Z. Zhang. Application-specific instruction generation for configurable processor architectures. In International Symposium on Field-Programmable Gate Arrays, February 2004

[6] J. A. Davis, V. K. De, and J. Meindl. A stochastic wire-length distribution for gigascale integration (GSI)—Part I: derivation and validation. 45(3):580–589, March 1998.

[7] R. J. Francis, et al. Chortle-crf: fast technology mapping for lookup tablebased FPGAs. In Design Automation Conference, 1991.

[8] J. Hwang, F. Chiang, and T. Hwang. A re-engineering approach to low power FPGA design using SPFD. In Design Automation Conference, 1998.

[9] M. Inuani and J. Saul. Re-synthesis in technology mapping for heterogeneous FPGAs. In International Conference on Computer Design, 1998

[10] T. Kong. A novel net weighting algorithm for timing-driven placement. In IEEE/ACM International Conference on Computer-Aided Design, pages 172–176, 2002.

[11] E. Lin and S. Wilton. Macrocell architectures for product term embedded memory arrays. Field Programmable Logic Applications, pages 48–58, August 2001.

[12] A. Marquardt, V. Betz, and J. Rose. Using cluster-based logic blocks and timing-driven packing to improve FPGA speed and density. In ACM/SIGDA International Symposium on Field Programmable Gate Arrays, pages 37–46, 1999