

EXPERIMENTS IN LOW POWER FPGA DESIGN

G. SUTTER and E. BOEMO

‡ *School of Engineering, Universidad Autónoma de Madrid, Madrid, Spain*
 {gustavo.sutter, eduardo.boemo}@uam.es

Abstract— This paper summarizes the utility of some low-power design (LPD) methods based on architectural and implementation modifications, for FPGA based systems. Power consumption is becoming one of the mayor design trade-off in today electronic. In this work, the contribution of spurious transitions to the overall consumption is evidenced and main strategies for its reduction are analyzed. Empirical results are present in order to show the effectiveness of pipelining and sequentialization as low-power design methodologies. The possibilities of power management techniques are explained and quantified. Algorithm level and Finite State Machines alternatives are also discussed and measured

Keywords— Low Power techniques, FPGA design, design methods.

I. INTRODUCTION

This work explores several end-user low power design methods. Power consumption is one of the major design tradeoff in current FPGAs (Shang *et al*, 2002; Mohanty and Prasanna, 2000; Sutter, 2005), the power dissipated can reach values of several watts. According to Shang *et al* (2002), a normal design has an average consumption of 1.5 μ W/ MHz/Slice in a Virtex II. Thus, a modest design of 8000 slices (2000 CLBs) running at 100 MHz can consume 1.2 W. However, this type of rule-of-thumb metrics do not take into account details like the logic depth, or the amount of glitches. For example, a 32-bits non-restoring divider with 576-slices and 32-LUTs logic depth exhibits a figure of 610 μ W/MHz/Slice in a Virtex. Dividing at 5 MHz can take more than 1.75 W (Sutter *et al*, 2004a).

FPGA users can only optimize the dynamic power component. That is, the part of the power that depends on the value of the capacitance, effective switching frequency, and power supply voltage of each circuit node. Setting aside V_{DD} manipulations, the power consumption can be modified by varying: the topology (that influences al the variables); the data (that vary effective frequency); and finally, the interconnection network, which affect both the capacitance and the effective frequency of each node.

A large fraction of the FPGA power consumption is caused by glitches. For example, a simulation of a fully combination 32-bit shift & add multiplier shows that glitches represent more than 80% of the activity. Glitches can be reduced in several ways:

- *Path equalization* (Wong, 1992; Boemo *et al*, 1995a): Equalize of all the delays inside each path of the circuit. The idea also leads to the wave pipeline technique. The main drawback is that it must be done manually on FPGA.
- *Dense LUT partitioning* (Boemo *et al*, 1995b): A dense technology mapping allows the designer to eliminate net count, and path unbalances. An intensive use of the LUT capability can lead to wiring congestion.
- *Pipelining* (Lemnios and Gabriel, 1994; Chandrakasan *et al*, 1992; Noll, 1992): Glitches can be blocked by pipeline registers. The snow-ball effect of glitches is thus neutralized. The latency of the circuit is increased.
- *Asynchronous barriers*: A line of latches can be introduced to stop glitches. There are controlled by asynchronous signal whose delay is matched with the longest delay of the path. Asynchronous delays depend on temperature, power supply voltage, and fabrication technology.
- *Registering output pads* (Sutter, 2005): Glitches at the output pad increase power by a double-effect: higher power-supply voltage at the pad rings; and second, higher off-chip capacitances to be driven. And extra effect is the increase of latency.

Although glitches strongly increase datapath power, other sources of dynamic consumption must be taken into account. Current FPGA models (Q1 2006) include up to 200K user flip-flops that can be commutating following the primary system clock. This lead to an important increment of the clock power or the energy per cycle involved in the synchronization of the circuit. Finally, off-chip power, the fraction dissipated at output pads (where the capacitances are several times larger than those for conventional microelectronics) can not be neglected.

The knowledge of the relationship between these components for a given FPGA technology is fundamental: It allows the effectiveness of any particular power reduction method to be determined *a priori*. A method to measure the power components of FPGA systems is based on the decomposition of the total power in four components (Todorovich *et al*, 2000):

- *Dynamic Power*: To calculate it, the total power is measured and then the static, off-chip and synchronization power is deducted.
- *Static power*: The chip is configured but neither stimulus nor clocking is applied. The pull-up resis-

tors and other external elements that require the FPGAs remain connected.

- *Off-chip power*: For the older families, the circuit is measured twice. First, during normal operation. Second, by disabling the tri-state output buffers. Thus, the off-chip component can be approximated to the difference between the two results. In addition, the use of the tri-state buffers in low-power design is also useful to separate the results from a particular PCB. From Virtex, as the power supply for the core is separated, just this line is measured.
- *Synchronization power*: A constant data (for example, all bit zeroed) is inputted to the circuit, meanwhile the clock signal is applied. Thus, only the clock tree has activity. It is important to note that FPGAs use multiplexers to emulate the effect of a clock enable. As a consequence, the use of the clock enable pin of a CLB does not interrupt the clocking of the flip-flops.

Other techniques to measure the power consumption are summarized in Mengibar *et al* (1999), Garcia (2000) and Rius *et al* (2003). Interesting studies in power breakdown are in Kusse and Rabaey (1998) and Garcia (2000) for XC4K family. Shang *et al* (2002) and Poon *et al* (2002) present similar analyses for Virtex II devices. Main results shows that interconnection power is dominant (50-70%), followed by logic and synchronization power with around 15-20% each. Finally the off-chip power is around 10-15% for typical designs.

An important effort has been done in LPD techniques applied to full-custom and cell-based integrated circuits. However, papers and theses about low-power in FPGA are recently emerging¹. In this line, the aim of this work is to detect test-and-true techniques at design level.

The results shown here embrace several of the Xilinx FPGA series. In all cases, the circuits have been constructed and measured. For XC4K families, all the measurements were done using XC4010EPC84-4, XC4005EPC84-3, or XC4003EPC84-3 samples. Input vectors were generated using another FPGA. Circuits were described in VHDL, and synthesized using the FPGA Express and the Xilinx Foundation tools (Synopsis, 2000; Xilinx, 2000a). Random vectors were utilized to stimulate the circuit.

In Virtex, the main experiments were developed using an XCV800hq240-6 and an XCV50hq240-6 chip samples mounted in Xilinx prototype board AFX PQ240-100. For Virtex II experiments a XC2V1500 FG676-6 is utilized. The circuits were described in VHDL instantiating low level primitives such as LUTs, *muxcy*, *xorcy* (Xilinx, 2003a) when necessary. Xilinx ISE 6.1 tool (Xilinx, 2003b) and XST (Xilinx, 2003c) for synthesis were utilized. A common pin assignment, the preservation of the hierarchy, speed optimization,

and timing constrains were fixed during the experiments. Chip measurements were done using three different sequences: a) random vectors (*avg_tog*); b) a sequence with a high transition probability (*max_tog*) and finally, c) a sequence with low activity (*min_tog*). The test vectors were inputted using a pattern generator (Tektronix, 2001).

For all the devices, the output, each pad supported the load of the logic analyzer probes (Tektronix, 2002). Area-delay information was extracted from Xilinx tools.

This paper is organized as follow. Section 2 shows the influence of pipeline as LPD technique. Section 3 compares architectural options. Section 4 describes power management techniques in FPGA. Section 5 shows some results related to finite state machine, meanwhile section 6 analyze some results at algorithm level. Finally, in section 7 general tips for low power are presented.

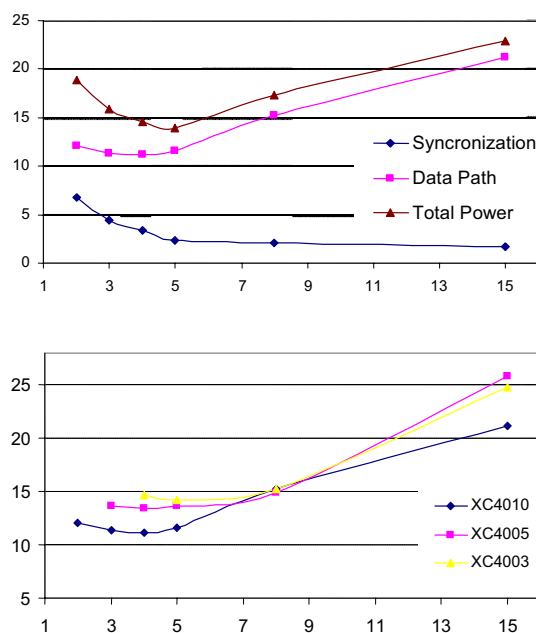


Fig. 1. Power consumption in mW/MHz as function of logic depth in XC4K families for 8 bits Hatamian & Cash multipliers. a. Power breakdown in XC4010. b. Consumption in different XC4K devices.

II. POWER REDUCTION THROUGH PIPELINE

Pipelining, a popular way to speed up circuits also allows power consumption to be reduced (Lemnios and Gabriel, 1994; Chandrakasan *et al*, 1992). Its usefulness is based on a marginal effect of the intermediate pipeline registers: the obstruction of the propagation of spurious (asynchronous) transitions. Pipelining also affects power consumption by the modification of datapath wiring loads: global lines (which usually broadcast the input data into the array) are split into a subset of lightly loaded lines, reducing the overall capacity (Boemo *et al*, 1995a).

The array multiplier proposed by Hatamian and Cash (1986) was selected as benchmark circuit for the

¹ The search of "low-power" AND "FPGA" returns 225 papers in the IEEE Explorer database, and over 651K links in Google. (april06)

XC4K family. This topology presents several benefits considering the objectives of the experiments. First, its high regularity makes straightforward the pipelining; and second, a large set of reconvergent paths exists, a feature that contributes to the production of glitches. Figure 1 shows dynamic power consumption as a function of logic depth (LD, measured in LUT).

For Virtex devices, 32-bits shift & add multiplier and several 32-bits dividers were implemented. Figure 2.a shows dynamic power consumption versus logic depth (LD, measured in LUT) for multiplier implementations, instead Figure 2.b present the same relation for a 32-bits SRT radix-2 divider. The different patterns have similar shape: it decreases practically linearly with the reduction of LD. It stands out the low influence of the synchronization power.

As more pipeline stages are added, less glitches are produced, and the power is lowered. This reduction in the activity makes less important the architecture selected. Thus, same experiment for others recurrence dividers (including restoring, non-restoring and SRT radix-4, -8 and -16) have similar consumption shape (Sutter *et al.*, 2004a). In dividers, a maximum pipeline architecture (LD = 1) saves up to 93 % of the dynamic power consumption respect to the fully combinational architecture (LD = 32). That is, combinational architectures consume more than twelve times more than the fully pipelined version.

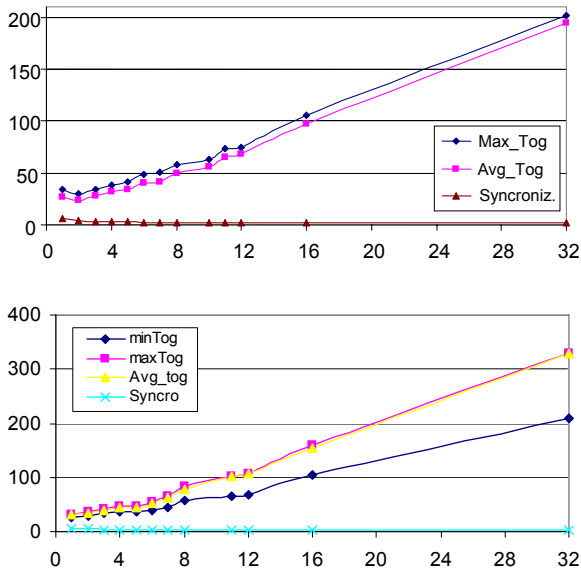


Fig. 2. Power consumption in mW/MHz as function of logic depth in Virtex devices. a. 32-bit shift & add multiplier; b. An SRT radix-2 32-bit divider.

Virtex results show a lower influence of synchronization power than XC4K. Thus, the optimum LD is lower in the first technologies (between 1-2 LUT) instead of 3-5 LUT in the second. Pipelining in FPGA shows a low impact in area due to the embedded registers distributed into the slices, and the SRL characteristics of LUT.

III. ARCHITECTURAL STRUCTURES

In previous sections, the importance of spurious activity was evidenced. Thus, a natural question is: What about iterative implementations? To reduce area, to use an iterative architecture is a common technique. The general architecture is composed of a state machine (FSM) that controls a data-path. Commonly the data-path power is low because the logic depth is minimal, but synchronization power grows due to intermediate register and the FSM consumption. Several circuits have been analyzed in this paper.

Example 1: Modular multipliers in XC4K

Modular multipliers are the central operation in many cryptographic systems. Three different algorithms have been analyzed: multiply and reduce (m_r), shift & add (s_a) and Montgomery (mont). Implementation details are described in (Deschamps and Sutter, 2002). Table 3 shows implementation results for 8-bit fully combinational modular multipliers, and fully sequential implementations. The power reduction in sequential implementations differs between algorithms, but is around the half. Area is reduced and total delay increases in a factor of two. Results are shown in table 1.

Table 1. Area, Delay and Power consumption for different 8 bit Modular Multipliers.

Algorithm	Array Impl.			Fully Sequential					
	Energy (nJ)	Area (CLBs)	Delay (ns)	Total	Syncro	Data Path	Area (CLBs)	Area (FF)	Delay (ns)
m_r	96,1	85	186	71,5	46,8	24,7	57	67	320
s_a	186,4	157	201	104,8	52,5	52,3	33	37	465
mont.	92,7	102	167	38,6	27,2	11,1	34	31	249

Example 2: 32 bit dividers in Virtex

Results for two 32-bit division algorithms are exhibited. The algorithms covered are: non-restoring (nr), and SRT radix 2 (srt). Details of the divider implementations are described in (Sutter *et al.*, 2004b), meanwhile a deeper power analysis is presented in (Sutter *et al.*, 2004a).

The circuits are sequentialized with different granularities G . For example, $G = 1$ indicates a fully iterative circuit. The circuit calculates, at each clock, G bits. Then, a total of p/G cycles are used to complete the operation, where p refers to precision.

Figure 3.a shows the average energy for an operation, in nJ, for 32-bit width divider. The synchronization and data-path components are also individually displayed. The synchronization power decreases as G grows, mainly because of smaller cycles. In the opposite, the data-path consumption grows with G , mainly because the glitches increase. Optimum G value seems to be 4.

An important point is that the value of G , rather than a particular algorithms, is the key to reduce the power

figure. In SRT radix-2, $G=4$ save 51% energy with respect to $G=1$. The energy savings with respect to the fully combinational implementations are: 85% as regards SRT radix-2, 89% as regards non-restoring division.

Figure 3.b shows ATP figure for the 32-bit dividers. The array implementations have the lowest latency, but as the cost of a great area and excessive power dissipation. Pipeline offers the best throughput, with a relatively low increment in area with respect to array implementations and a good power figure, but the initial latency could be prohibitive for some applications. Finally, sequential implementations have the smaller area, a delay less than twice the one of arrays, but have a good power figure.

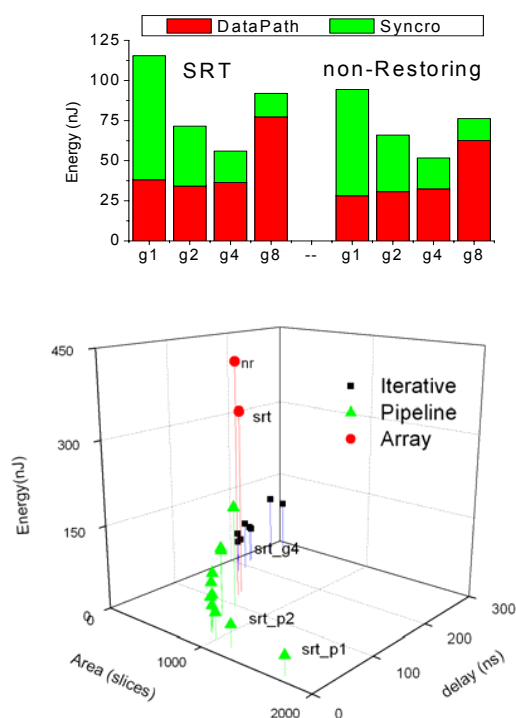


Fig. 3. a. Dynamic power consumption breakdown for sequential divider implementations. b. Area-Time-Power for sequential, array, and pipeline implementations.

IV. FINITE STATE MACHINES

Main idea in the design of low-power FSMs is minimize Hamming distance of the most probable state transitions. However, this solution usually increases the required logic to decode the next state. Then, a tradeoff between switching reduction and extra capacitance exists. Interesting contribution in low power FSM are Wu *et al* (2000); Tsui *et al* (1994a); Benini and De Micheli (1995); Nöth and Kolla (1999) and Tsui *et al* (1994b). The research line described above was targeted to gate arrays or cell-based integrated circuits. FPGA manufacturers and synthesis tools use One-Hot as default state encoding (Xilinx, 2000b; Synopsys, 1999). This assignment allows the designer to create state machine that are more efficient for FPGA architectures in terms

of area and logic depth (i.e. speed). FPGAs are plenty of registers but the LUTs are limited to few bits wide. One Hot increases the flip flop usage (one per state) and decreases the width of combinatorial logic. In addition, the Hamming distance of One Hot encoding is always two in spite of the machine size.

In Sutter *et al*, (2002a) the end user alternatives in encoding are studied using dense encoding (binary and minimum decode logic) and sparse encoding (one-hot and two-hot). The main conclusions are that in small state machines (up to 8 states), area, speed and power is minimized using binary state encoding. On the contrary, One Hot state encoding is better for large machines (over 16 states). A comparison between 26 test circuits shows important differences in power consumption. Depending on the state encoding, up to 57% of power saving can be obtained.

Other idea for low-power FSMs is the use of power management. That is, to shutdown the blocks of hardware in these periods where they are not producing useful data. Shutdown can be fulfilled in three ways: by turning off the power supply, by disabling the clock signal, or finally by “freezing” (blocking) the input data. Several works were published for standart cell (Benini *et al*, 1995, 1996, 1998; Chow *al*, 1996; Monteiro and Oliveira, 1998). Based on these previous ideas, Sutter *et al* (2002b) adapted or modified them to suit well with LUT-based FPGAs. The hardware overhead associated with the decomposition technique makes this method neither effective for FSMs with small numbers of states (under 10) nor applicable for circuits whose decomposition has a highly transition probability between submachines. However, for large machines, an improvement in power consumption up to 46% can be obtained.

V. POWER MANAGEMENT TECHNIQUES IN FPGA

In order to eliminate the activity in an idle part of a circuit, several alternatives exist. The most traditional technique is clock gating (Benini *et al*, 1996; Shelar *et al*, 2000), but it must be avoided in FPGA technology (Xilinx, 2003d). Gated clock can cause the flip-flop to clock at wrong times. In addition, in all Xilinx families, the flip-flops have the usual mux-based built-in clock-enable (CE) to implement this feature. From a power consumption point of view, the clock tree continues consuming power.

In Virtex II, II Pro, and Spartan-3, BUFGMUX is a multiplexed global clock buffer that can select between two inputs without glitches. This allows constructing circuits that work with different clocks. If one of the inputs of BUFGMUX is tied to 0 (or 1) it is transformed in a Global Clock MUX Buffer with Clock Enable (BUFGCE).

Another way to disable de combinational path is blocking the inputs. It can be carried out in several ways. The straightforward method is to utilize the CE of the normal FFs. But other alternatives exist: latches, ANDs gates, and OR gates.

In order to quantify the different disabling alternatives, a circuit with two big combinatorial blocks and a final multiplexer was implemented (figure 4.a). The selection logic block commutes, each eight clock cycles. Then, the different disabling techniques were applied to the circuit. Figure 4.b shows the chip enable (CE) alternative. Table 3 shows power improvement and delay penalty for the different techniques. The area overhead in FPGA of different alternatives is also very small. CE and gated clock seem to be the more effective disabling techniques.

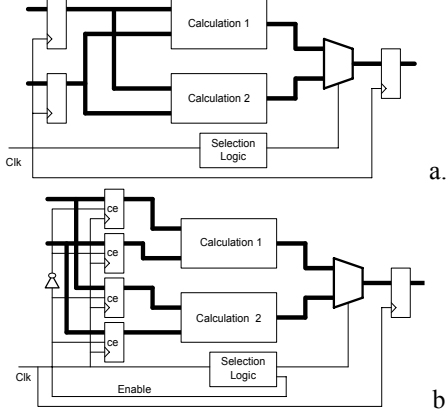


Fig. 4. a. Architecture to measure the impact of disabling. b. Modified architecture using CE of distributed flip-flops.

Circuito	Dynamic power reduction		Delay penalty (ns)	
	Virtex	Virtex II	Virtex	Virtex II
normal	0%	0%	0.0	0.0
ce	39%	38%	2.9	0.1
latch	34%	29%	4.7	1.6
gate_clk	-	34%	-	1.9
ands	37%	32%	2.5	1.0
ors	17%	21%	2.5	1.0

Table 2. Results for different disabling techniques.

VI. ALGORITHM LEVEL ALTERNATIVES

One of the most straightforward ways to reduce power is analyzing different algorithm for the same problem. In order to measure the influence we analyze the problems of section 4. a) modular multiplier in XC4K family, and b) division algorithm in Virtex and Virtex II. Main results show that the power consumption can be reduced in a factor of two only selecting the best algorithm.

Results for eight different algorithms that implement 32-bit division algorithms are exhibited in figure 5. The algorithms covered are: restoring (rest), non-restoring (nr), SRT radix 2, 4, 8 and 16 (srt_2, srt_r4, srt_r8, srt_r16), and finally an SRT implementation with carry save (srt_cs). More details of the divider implementations in Sutter *et al* (2004b).

The area-time-power shape of figure 5, reveal some results: The algorithm level it is one of the easiest way to reduce consumption. The Results shows that the

power consumption can be reduced in a factor of two only selecting the best algorithm. But, not the speed, nor the area, is sufficient to determine what algorithm will consume less power. There are a more subtle characteristics to be taken into account such as the ability to produce or not glitches by the algorithm.

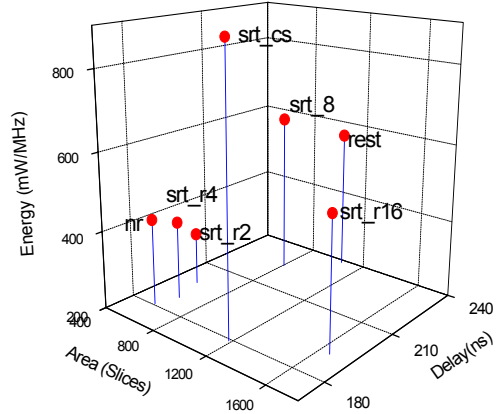


Fig. 5. Area-Time-Power for 32-bit dividers for the *avg_tog* sequence.

VII. GENERAL TIPS AND CONCLUSIONS

In this paper some of the most powerful end-user alternatives to design low power designs are presented. Due to the SRAM based reprogrammable interconnection, the FPGA is plenty of glitches. Some important rules are:

- At higher level of abstraction more power saving opportunities exists. Algorithmic and system levels are the most straightforward place to obtain power reduction.
- The reduction of logic depth is essential in order to obtain a power aware system. The exploration of pipeline (for application with regular data flow), and sequential architectures are useful to mitigate this problem.
- When designing FSM for low power: for big machines (more than 16 states) use one-hot, for smaller than 8 states use a binary based codification.

Additionally, some obvious tips are:

- Avoiding power waste: design systems should reach performance requirements, rather than exceed. The speed versus data width trade off must be analyzed.
- In such application, where part of the circuit is idle for a relative long period, disabling the clock or the data input is an interesting option.
- Register always the last stage before the pads. Glitches in the last stages produce activity at the PCB level, where the capacitances are much higher than internal. Furthermore, registering must be done, when possible, near the logic that produces the data. Therefore, instead of using IOB flip-flop, the internal FF must be employed, because there are nearer to the data, and additionally use lower voltage.

REFERENCES

- Benini, L. and G. De Micheli. "State Assignment for Low Power Dissipation". *IEEE Journ. of Solid State Circuits*, **30**, 258-268 (1995).
- Benini, L., P. Siegel and G. De Micheli. "Automatic synthesis of low-power gated-clock finite-state machines". *IEEE Trans.onCAD of IC*, **15**, 630-643 (1996).
- Benini, L., G. De Micheli and F. Vermeulen, "Finite-state machine partitioning for low power". *IEEE International Symposium on Circuits and Systems (ISCAS '98)*, Monterey, California, **2**, 5-8 (1998).
- Boemo, E., S. López, G. González and J. Meneses, "On the usefulness of pipelining and wave pipelining as low-power design technique", *Proc. PATMOS Conference*, (1995a).
- Boemo, E., G. Gonzalez de Rivera, S. Lopez-Buedo and J. Meneses, "Some Notes on Power Management on FPGAs", LNCS, Springer-Verlag, **975**, 149-157 (1995b).
- Chandrakasan, A., S. Sheng and R. Brodersen, "Low-Power CMOS Digital Design", *IEEE Journal of Solid-State Circuits*, **27**, 473-484. (1992)
- Chow, S., Y-C. Ho, and T. Hwang. "Low Power Realization of Finite State Machines Decomposition Approach". *ACM Trans on Design Aut. Elec. Systems*, 315-340, (1996).
- Deschamps, J-P. and G. Sutter, "FPGA Implementation of Modular Multipliers". *Proc. XVII Conference on Design of Circuits and Integrated Systems DCIS* (2002).
- García, A., "Power consumption and optimization in field programmable gate arrays," *Ph.D. thesis, Ecole Nationale Supérieure des Télécommunications* (2000).
- Hatamian, M. and G. Cash, "A 70-MHz 8-bit x 8 bit Parallel Pipelined Multiplier in 2.5-um CMOS", *IEEE Journal of Solid-State Circuits*, **21**, 505-513 (1986).
- Kusse, E., and J. Rabaey, "Low-energy embedded FPGA structures," *Int. Symp. On Low Power Electronics & Design*, 155-160 (1998).
- Lemnios, Z. and K. Gabriel, "Low-Power Electronic", *IEEE Design & Test of Computers*, 8-13 (1994).
- Mengibar, L., M. García, D. Martín, and L. Entrena, "Experiments in FPGA Characterization for Low-power Design", *Proc. DCIS'99 conf.*, Palma de Mallorca, (1999).
- Mohanty, S. and Prasanna, V., "Energy Efficient Application Design using FPGAs", *FPGA and Structured Asic Journal*, (2000).
- Monteiro J., A. Oliviera, "Finite State Machine Decomposition for Low Power", *Proceedings 35th Design Automation Conference*, San Francisco, 758-763. (1998).
- Noll, T.G., "Pushing the Performance Limits due to Power Dissipation of future ASICs", *Int. Symposium on Circuits and Systems*, IEEE Press, 1652-1655. (1992).
- Nöth, W. and R. Kolla. "Spanning Tree Based State Encoding for Low Power Dissipation". *Proc. of Date99 conference*, Munich, Germany, 168-174 (1999).
- Poon, K., A. Yan, and S. J. E. Wilton. A Flexible power model for FPGAs. *Lecture Notes in Computer Science*, **2438**, 312-321 (2002).
- Rius Vazquez, J., E. Boemo, A. Pedro Palanca, S. Manich Bou and R. Rodriguez Montañes, "Measuring Power and Energy of CMOS Circuits: A Comparative Analysis", *Proceedings DCIS (XVIII Conf. on Design of Circuits and Integrated Systems)*, Ciudad Real, 89-94 (2003).
- Shang, L., A. Kaviani and K. Bathala, "Dynamic Power Consumption in Virtex™-II FPGA Family", *Proc FPGA'02 conference*, Monterey, California, USA, 157-164 (2002).
- Shelar, R., H.Narayanan, M.Desai, "Orthogonal Partitioning and Gated Clock Architecture for Low Power Realization of FSMs", *IEEE Int ASIC/SOC conference*, 266-270. (2000).
- Sutter, G., E. Todorovich, S. Lopez-Buedo and E. Boemo, "Low-Power FSMs in FPGA: Encoding Alternatives", *Lecture Notes in Computer Science*, Springer-Verlag, **2451**, 363-370. (2002a).
- Sutter, G., E. Todorovich, S. Lopez-Buedo and E. Boemo, "FSM Decomposition for Low Power in FPGA", *Lecture Notes in Computer Science*, Springer-Verlag, **2438**, 350-359. (2002b)
- Sutter, G., G. Bioul, J-P. Deschamps and E.Boemo "Power Aware Dividers in FPGA", *Lecture Notes in Computer Science*, Springer-Verlag, **3254**, 574-584. (2004a).
- Sutter, G., G. Bioul and J-P. Deschamps, "Comparative Study of SRT-Dividers in FPGA", *Lecture Notes in Computer Science*, **3203**, (2004b).
- Sutter, G. "Aportes a la Reducción de Consumo en FPGAs", *Ph.D. Thesis, School of Engineering Universidad Autónoma de Madrid* (2005).
- Synopsys, Inc. "FPGA Compiler II / FPGA Express VHDL" Reference Manual, Version 1999.05 (1999).
- Synopsis, inc. "FPGA Express home page"; http://www.synopsys.com/products/fpga/fpga_express.htm (2000).
- Tektronix inc, TLA7PG2 Pattern Generator Module User Manual. www.tektronix.com. (2001).
- Tektronix inc, TLA 700 Series Logic Analyzer User Manual. www.tektronix.com (2002).
- Todorovich, E., G. Sutter, N. Acosta, E. Boemo and S. López-Buedo, "End-user low-power alternatives at topological and physical levels. Some examples on FPGAs", *XV Conf. on Design of Circuits and Integrated Systems (DCIS 2000)*, Le Corum, Montpellier, France, (2000).
- Tsui, C., M. Pedram and A. Despain, "Exact and Approximate Methods for Calculating Signal and Transition Probabilities in FSMs", *31st Design Automation Conf.*, 18-23, (1994a).
- Tsui, C., M. Pedram, C. Chen and A. Despain, "Low Power State Assignment Targeting Two- and Multi-level Logic Implementations", *Proc. of ACM/IEEE Internat. Conf. of Computer-Aided Design*, 82-87, (1994b).
- Wong, D., "Techniques for Designing High-Performance Digital Circuits Using Wave Pipelining", *Technical report No. CLS-TR-92-508*. Stanford University (1992).
- Wu, X., M. Pedram and L. Wang, "Multi-code state assignment for low power design", *IEEE Proc. - Circuits, Devices and Systems*, **147**, 271-275 (2000).
- Xilinx Inc, Xilinx Foundation Tools F3.1i, www.xilinx.com/support/library.htm (2000a).
- Xilinx inc, Xilinx software manual, Synthesis and Simulation Design Guide: Encoding State (2000b).
- Xilinx Inc, Libraries Guide for ISE 6.1 available at www.xilinx.com (2003a).
- Xilinx Inc, Xilinx ISE 6 Software Manuals, available at www.xilinx.com (2003b).
- Xilinx Inc, XST User Guide version 4.0, available at www.xilinx.com (2003c).
- Xilinx inc, "Data Feedback and Clock Enable", Development system design guide; Chapter 2: Design Flow (2003d).

Received: April 14, 2006.

Accepted: September 8, 2006.

Recommended by Special Issue Editor Hilda Larrondo.