Design of Power Aware FPGA-based Systems

Gustavo Sutter Elias Todorovich and Eduardo Boemo

School of Engineering, Universidad Autónoma de Madrid
{gustavo.sutter,elias.todorovich, eduardo.boemo}@uam.es

Abstract. Power consumption is one of the mayor design trade-off in today electronic. This paper explores the utility of some end-user low-power design (LPD) methods based on architectural and implementation modifications, for FPGA based systems. 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 also explained and quantified.

1 Introduction

Power consumption in current FPGA is significant: it can reach values of several watts. According to [1], a normal design has an average consumption of 1.5 μW/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 μW/MHz/Slice in a Virtex. Dividing at 5 MHz can take more than 1.75 W [2].

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 all 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: A list of techniques is summarized in Table 1.

Although glitches strongly increase datapath power, other sources of dynamic consumption must be taken into account. Current FPGA models (Q3 2004) include up to 90K user flip-flops that can be commuting 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-based

systems is described in Table 2. Other techniques are summarized in [3], [4] and [5]. Interesting studies in power breakdown are in [6] and [4] for XC4K family. [1] and [7] 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.

Technique	Idea	Comment	Ref.
Path equalization	Equalization of all the delays inside each path of the circuit. The idea also leads to the wave pipeline technique.	It must be done manually on FPGA.	[8] [9]
Dense LUT partitioning	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.	[10]
Pipelining	Glitches can be blocked by pipeline registers. The snow-ball effect of glitches is thus neutralized.	The latency of the circuit is increased.	[11] [12] [13]
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	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.	More latency.	[14]

Table 1. Techniques to reduce glitch-power in FPGAs.

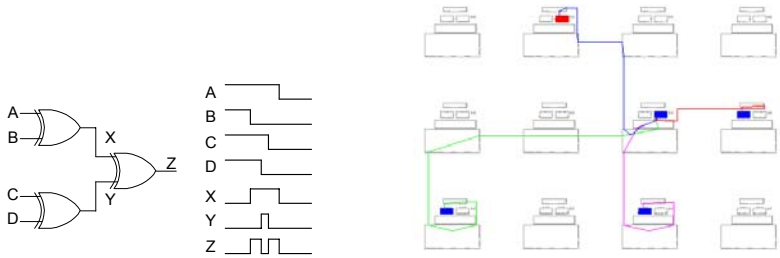

Fig. 1. a. Glitches in combinational circuits depend on wiring delay unbalances and logic depth. b. Routing in FPGA is intrinsically unbalanced.

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 emerging[1]. In this line, the aim of this work is to detect test-and-true techniques at design level.

This paper is organized as follow. Section 2 describes the experimental setup. Section 3 shows the influence of pipeline as LPD technique. Section 4 compares architectural options. Section 5 describes power management techniques in FPGA. Section 6 shows some

[1] The search of "low-power" AND "FPGA" returns 138 papers in the IEEE Explorer database and over 65K links in Google.

results related to finite state machine. Finally, in section 7 general tips for low power are presented.

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 resistors 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. 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.

Table 2. Determination of power component in arithmetic circuits

2 Experimental setup

This paper summarizes more than ten years of research in LPD in FPGAs. It embraces most of the Xilinx FPGA series. In all cases, the circuits have been constructed and measured. For XC4K families, all the measurements were done using XC4010EPC84-4C, XC4005EPC84-3C, or XC4003EPC84-3C samples. Input vectors were generated using another FPGA. Circuits were described in VHDL, and synthesized using the FPGA Express [15] and the Xilinx Foundation tools [16]. 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. The circuits were described in VHDL instantiating low level primitives such as LUTs, *muxcy*, *xorcy* [17] when necessary. Xilinx ISE 6.1 tool [18] and XST [19] 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 [20].

For all the devices, the output, each pad supported the load of the logic analyzer probes [21]. Area-delay information was extracted from Xilinx tools.

3 Power Reductions through Pipeline

Pipelining, a popular way to speed up circuits also allows power consumption to be reduced [11] [12]. 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 [9]

The array multiplier proposed by M. Hatamian and G. Cash [22] was selected as benchmark circuit for the 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 2 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 3.a shows dynamic power consumption versus logic depth (LD, measured in LUT) for multiplier implementations, instead Figure 3.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.

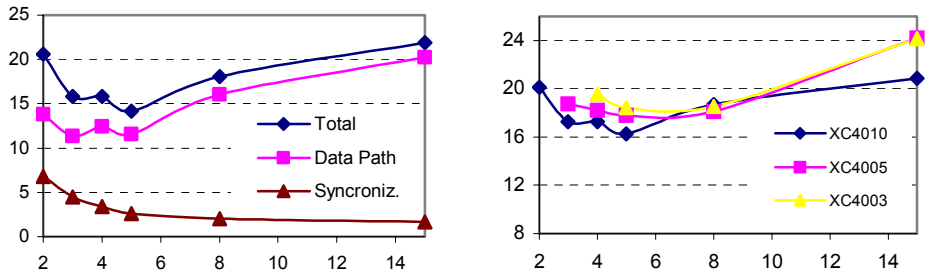

Fig. 2. Power consumption in mW/MHz as function of logic depth in XC4K families for 8 bits Hatamian & Cash multipliers. a. Power breakdown in XC4010. b. Consumption in different devices.

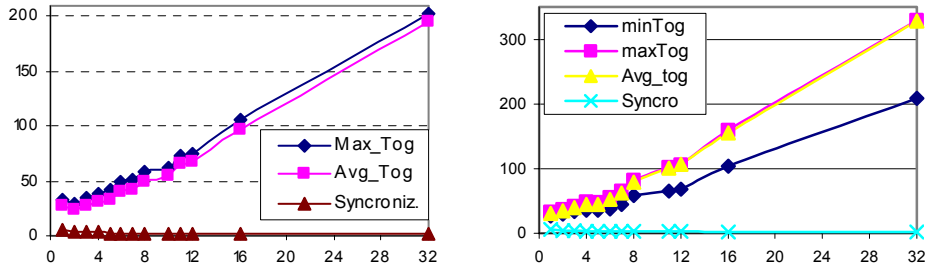

Fig. 3. Power consumption in mW/MHz as function of logic depth in Virtex devices. a. 32-bit shift & add multiplier; b. An SRT radix-2 32-bit divider.

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 [2]. 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.

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. Figure 4 shows normalized area-delay-power respect the logic depth.

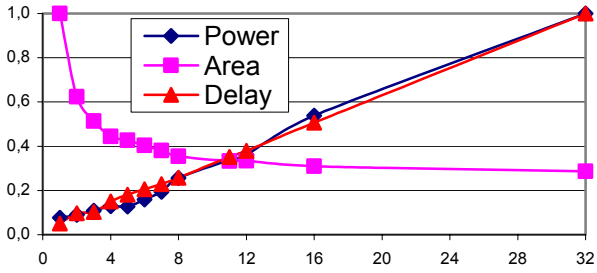

Fig. 4. Normalized Area, delay and power for a non-restoring divider respect to logic depth

4 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 [23]. 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.

Algorithm	Array Implementation			Fully Sequential					
	Energy	Area	Delay	Energy (nJ)			Area		Delay
	nJ	CLBs	ns	Total	Synchro	Data Path	CLBs	FF	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

Table 3. Area, Delay and Power consumption for different 8 bit Modular Multipliers.

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 [24], meanwhile a deeper power analysis is presented in [2].

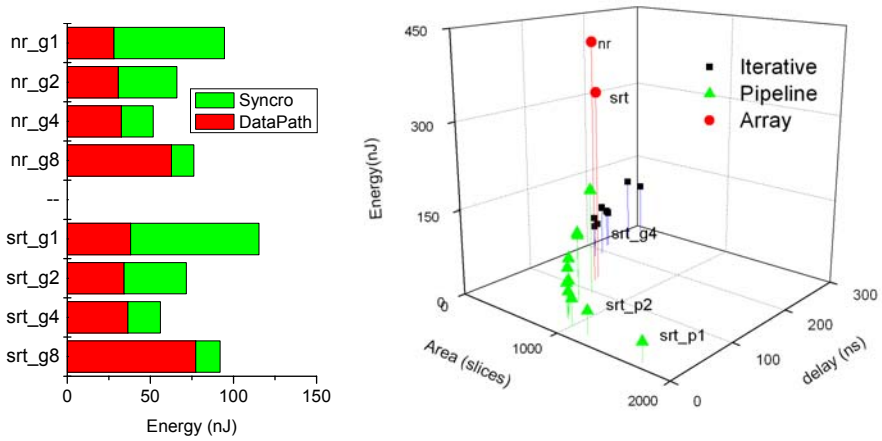

Fig. 5. a. Dynamic power consumption breakdown for sequential divider implementations. b. Area-Time-Power for sequential, array, and pipeline implementations.

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 5.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 that 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 5.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.

5 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 [25][26] , but it must be avoided in FPGA technology [27]. Gated clock can cause the flip-flop to clock at wrong times (figure 6.a). In addition, in all Xilinx families, the flip-flops have the usual mux-based built-in clock-

enable (CE) to implement this feature (figure 6.b). 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 6.c). The selection logic block commutes, each eight clock cycles. Then, the different disabling techniques were applied to the circuit. Figure 6.d 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.

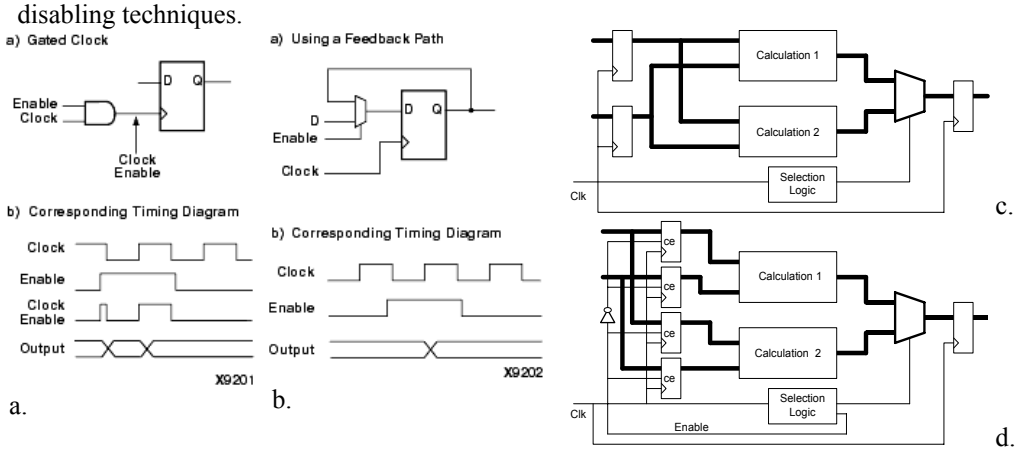

Fig. 6. a. Not recommended gated clock. b. using CE. c. Architecture to measure the impact of disabling. d. Modified architecture using CE of distributed flip-flops.

	Dynamic power reduction		Delay penalty (ns)	
Circuito	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 3. Results for different disabling techniques.

6 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 [28-32]

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 [33][34]. 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 [35] 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 [25][30][36][37][38]. Based on these previous ideas, [39] 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.

7 General Tips and Conclusions

In this paper same 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. 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.

- 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.
- 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.
- At higher level of abstraction more power saving opportunities exists. Algorithmic and system levels are the most straightforward place to obtain power reduction.
- 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. There-

fore, instead of using IOB flip-flop, the internal FF must be employed, because there are nearer to the data, and additionally use lower voltage.

Acknowledgement: This work is supported by Project 07T/0052/2003-3 of the *Consejería de Educación de la Comunidad de Madrid*, Spain

References

1. Li Shang, Alireza Kaviani, Kusuma Bathala, "Dynamic Power Consumption in Virtex™-II FPGA Family", FPGA'02, Monterey, California, USA. February 2002.
2. G. Sutter, G.Bioul, J-P. Deschamps, and E.Boemo, "Power Aware Dividers in FPGA", Fourteenth International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS 2004). Santorini, Greece. September, 2004.
3. L. Mengíbar, M. García, D. Martín, and L. Entrena, "Experiments in FPGA Characterization for Low-power Design", Proc. DCIS'99, Palma de Mallorca, 1999.
4. A. Garcia, "Power consumption and optimization in field programmable gate arrays," Ph.D. thesis, Ecole Nationale Supérieure des Télécommunications, 2000.
5. J. Rius Vazquez, 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)*, pp. 89-94, Ciudad Real, Nov 2003.
6. E. A. Kusse, and J. Rabaey, "Low-energy embedded FPGA structures," Int. Symp. On Low Power Electronics & Design, pp.155-160, Aug. 1998.
7. K. Poon, A. Yan, and S. J. E. Wilton. A Flexible power model for FPGAs. In Int. Conf. on Field-Programmable Logic and Applications, pages 312–321, La Grande Motte, France, 2002.
8. D. Wong, "Techniques for Designing High-Performance Digital Circuits Using Wave Pipelining", Technical report No. CLS-TR-92-508. Stanford University, february 1992.
9. E. Boemo, S. López, G. González and J. Meneses, "On the usefulness of pipelining and wave pipelining as low-power design technique", Proc. PATMOS Conf, 1995.
10. E. Boemo, G. Gonzalez de Rivera, S.Lopez-Buedo and J. Meneses, "Some Notes on Power Management on FPGAs", LNCS, No.975, pp.149-157. Berlin: Springer-Verlag 1995.
11. Z. Lemnios y K. Gabriel, "Low-Power Electronic", IEEE Design & Test of Computers, pp. 8-13, winter 1994.
12. A. Chandrakasan, S. Sheng y R. Brodersen, "Low-Power CMOS Digital Design", IEEE Journal of Solid-State Circuits, Vol.27, Nº4, pp.473-484. April 1992
13. T.G. Noll, "Pushing the Performance Limits due to Power Dissipation of future ASICs", Int. Symposium on Circuits and Systems, pp.1652-1655. IEEE Press, 1992.
14. E. Boemo, S. López-Buedo, G. Sutter, E. Todorov. "Entrenamiento Intensivo: FPGA Xilinx-VHDL-ISE", EPS-Univ. Autónoma de Madrid, Junio 2004.
15. FPGA Express home page. Synopsis, inc.; http://www.synopsys.com/products/ fpga/fpga_express.htm
16. Xilinx Foundation Tools F3.1i, www.xilinx.com/ support/library.htm
17. Xilinx Inc, Libraries Guide for ISE 6.1 available at www.xilinx.com, 2003.
18. Xilinx Inc, Xilinx ISE 6 Software Manuals, available at www.xilinx.com, 2003.
19. Xilinx Inc, XST User Guide 4.0, available at www.xilinx.com, June 2003.
20. Tektronix inc, TLA7PG2 Pattern Generator Module User Manual. www.tektronix.com.
21. Tektronix inc, TLA 700 Series Logic Analyzer User Manual. www.tektronix.com.
22. M. Hatamian 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*, August 1986.
23. J-P. Deschamps and G. Sutter , "FPGA Implementation of Modular Multipliers". XVII Conference on Design of Circuits and Integrated Systems (DCIS 2002). Santander, Nov, 2002

24. G. Sutter, G.Bioul, and J-P. Deschamps, "Comparative Study of SRT-Dividers in FPGA", 14th International Conference on Field Programmable Logic and Applications (FPL'04), Antwerp, Belgium, October, 2004.
25. L.Benini P.Siegel and G.De Micheli. Automatic synthesis of low-power gated-clock finite-state machines. *IEEE Trans.on CAD of IC, vol.15, Issue6*, pp. 630– 643, June 1996.
26. R.Shelar, H.Narayanan, M.Desai, Orthogonal Partitioning and Gated Clock Architecture for Low Power Realization of FSMs, *IEEE Int ASIC/SOC conf*, Sep 2000, pp. 266-270.
27. Xilinx inc, "Data Feedback and Clock Enable", Development system design guide; Chapter 2: Design Flow, 2003
28. X. Wu, M. Pedram, and L. Wang, Multi-code state assignment for low power design, *IEEE Proceedings - Circuits, Devices and Systems, Vol. 147, No. 5*, *pp. 271-275*, Oct. 2000.
29 C-Y Tsui, M. Pedram, A. M. Despain, Exact and Approximate Methods for Calculating Signal and Transition Probabilities in FSMs, *31st Design Aut. Conf.*, pp. 18-23, 1994.
30. L.Benini and G. De Micheli. State Assignment for Low Power Dissipation. *IEEE Journ. of Solid State Circuits*, Vol. 30, No. 3, pp. 258-268, March 1995.
31 Winfried Nöth and Reiner Kolla. Spanning Tree Based State Encoding for Low Power Dissipation. *In Proc of Date99, pp 168-174,* Munich, Germany, March 1999.
32. Chi-Ying Tsui, Massoud Pedram, Chih-Ang Chen, and Alvin Despain, Low Power State Assignment Targeting Two- and Multi-level Logic Implementations, Proceedings of ACM/IEEE International Conf. of Computer-Aided Design, pp. 82-87, November 1994.
33. Xilinx software manual, Synthesis and Simulation Design Guide: Encoding State. *Xilinx inc*, 2000
34. FPGA Compiler II / FPGA Express VHDL Reference Manual, Version 1999.05, Synopsys, Inc.,May 1999
35 G. Sutter, E. Todorovich, S. Lopez-Buedo, and E. Boemo, "Low-Power FSMs in FPGA: Encoding Alternatives", 12th Power and Timing Modeling, Optimization and Simulation Conference. (Patmos 2002), Sevilla - Spain, September 2002.
36. L. Benini, G. De Micheli, and F. Vermeulen, Finite-state machine partitioning for low power. *In Proc. IEEE Int'l Symposium on Circuits and Systems (ISCAS '98), volume 2, pages 5-8*, Monterey, California, May31-June3, 1998.
37. S-H.Chow, Y-C.Ho, and T.Hwang. Low Power Realization of Finite State Machines Decomposition Approach. *ACM Trans on Design Aut. Elec. Systems*, 315-340, July 1996.
38. J. Monteiro, A. Oliviera, Finite State Machine Decomposition for Low Power, *Proceedings 35th Design Automation Conference* , San Franscisco, 1998, pp. 758-763.
39. G. Sutter, E. Todorvich, S. Lopez- Buedo, and E. Boemo, "FSM Decomposition for Low Power in FPGA", Proc of the 12th Field Programmable Logic and Application Conference (FPL 2002), Montpellier - France. September, 2002