Some Notes on Power Management on FPGA-based Systems

Eduardo I. Boemo, Guillermo González de Rivera*,
Sergio López-Buedo, and Juan M. Meneses*

E.T.S.I. Telecomunicación, Universidad Politécnica de Madrid,
28040 Ciudad Universitaria. Madrid - España.

Current address: E.T.S. Informática, Universidad Autónoma de Madrid,
Ctra. De Colmenar Km.15, 28049 Madrid.
http://www.ii.uam.es

Abstract. Although the energy required to perform a logic operation has continuously dropped at least by ten orders of magnitude since early vacuum-tube electronics [1], the increasing clock frequency and gate density of the current integrated circuits has appended power consumption to traditional design trade-offs. This paper explore the usefullness of some low-power design methods based on architectural and implementation modifications, for FPGA-based electronic systems. The contribution of spurious transitions to the overal consumption is evidenced and main strategies for its reduction are analized. The efectiveness of pipelining and partitioning inprovements as low-power design methodologies are quantified by case-studies based on array multipliers. Moreover, a methodology suitable for FPGAs power analysis is presented.

1 Introduction

The general advantages of power consumption reduction are well-known: it allows expensive packaging to be avoided, the chip life operation to be increased, cooling to be simplifyed and the autonomy of battery powered systems to be extended (or their weight to be reduced). Even in fast prototyping, excessive consumption can be inconvenient: CMOS delays increase 0,3 % per ºC [2], as well as synchronous circuits can exhibit current peaks so they affect apparently independent variables like PCB features. Analogous to throughput or area occupation, the reduction of consumption can be achieved at any level of hierarchy; however, in this paper attention will be focused on architectural-implementation transformations, availables to FPGA end-users. Additionally, these approaches are not aggressive and can be applied in conjunction with any other strategy.

The main consumption in CMOS technology correspond to dynamic power: the energy per clock cycle involved in the charge/discharge of all circuit node capacitances. This power component can be modelled by:

$$P \quad \sum_{all\ nodes} c_n\ f_n\ V_{DD}^2 \qquad (1)$$

where f_n is the effective frequency of each circuit node (usually different from the system clock), C_n is the output capacitance of each node, and V_{DD} is the power supply voltage (Eq.(1) assumes that all capacitances reach V_{DD} after the loading period). Thus, setting aside V_{DD} manipulations, the power consumption can be modified by varying: the topology (that influences all the variables); the data (that vary f_n); and finally, the interconnection network, which affect C_n, but also f_n. However, the estimation or control of the effective frequency f_n of each node is difficult due to the appearance of glitches. Although the glitches do not produce errors in a well-designed synchronous systems, they can significantly increase the circuit activity.

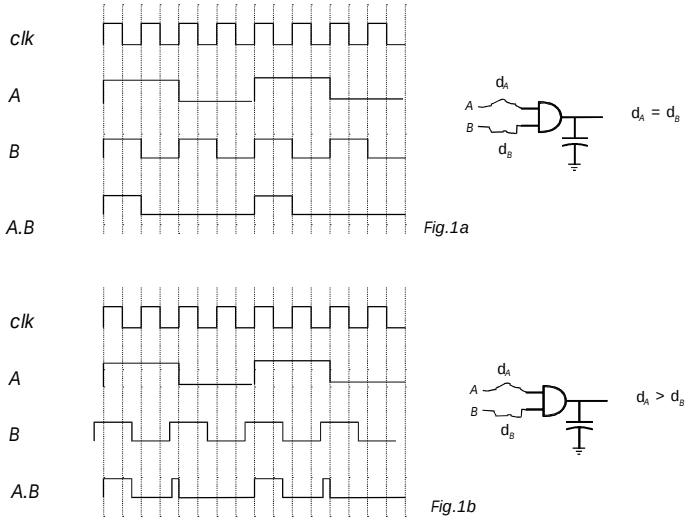

Fig.1: The effect of net unbalance on node activity.

The effect of glitches is illustrated on Fig.1: in the above graph the delay nets are equalized and the spurious activity level is zero; however if an unbalance between the paths exists (Fig.1b), glitches appear and power consumption increases. Depending on the circuit topology, these spurious transitions can progress across the following stages, producing an avalanche effect on power consumption. For example, combinational array multipliers with automatic placement-routing utilized in this work gave rise to arround 25 to 40 intermediate values before reaching the correct result, meanwhile a manually

path-equalized version of the same circuit just exhibed 5 to 8 intermediate values.

FPGA user has three ways to diminish glitches: pipelining, partitioning improvements and path delay equalization. Pipelining, a popular way to speed up circuits also allows power consumption to be reduced [3]-[4]. 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. The second way to disminish spurious activity is to pack critical parts into look-up tables (LUTs) by using a manual partitioning process: thus glitches, wiring and nodes can be reduced. Finally, path delay equalization also reduces spurious, as well as can conduce to wave pipelines or maximum-rate circuits [5]; the application of this technique on FPGAs is analyzed in [6].

In the next section, a methodology suitable for the analysis of power consumption on FPGAs is presented. In section 3, the effect of spurious transitions on datapath power and the efectiveness of pipelining and partitioning improvements is quantified by a set of case-studies; additionally, the magnitude of off-chip and clocking consumption is evaluated.

2 Power Budget on FPGAs

Dynamic consumption on FPGAs can be separated into three parts: datapath, synchronization, and off-chip power. The first component corresponds to the combinational blocks and associated interconnection power; the second part is the consumption by registers, clock lines, and buffers; and finally, off-chip power, is the fraction dissipated in the circuit output pads (where the capacitances are several times larger than those for conventional microelectronics). 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*. For example: partitioning improvements and "cold scheduling" [7] would be superfluous if datapath power is relatively small; Gray Code counters for addressing external circuits would be useful only if heavily loaded buses exist; self-timed synchronization, wave pipelining, DET registering [8], or stoppable clocks strategies would be effective provided synchronization power is dominant.

Datapath power measurements require the definition of a test vector set. Random sequences allow average consumption to be determined (and thus battery life operation to be predicted). Meanwhile special vector sequences that try to maximize the toggle of the circuit nodes allows the peak of dissipation to be deduced (and thereby stablishing the power supply requierements or testing off-chip power characteristics). In this work,

circuits have been tested using a 2^{16} pseudo-random data as well as a reduced set of 16 data, which toggle near the 93 % of the outputs in each cycle, rather than the 56% toggled by the random sequence. Because current pipelined circuits can run faster than affordable pattern generators, both test sequences were produced by another FPGA, a XC3120-3, allowing a low-cost high-speed pattern generator to be obtained.

In several circuits synchronization power consumption can be determined by clocking the circuit while maintaining the input data constant. Thus, there is no activity neither in the datapath nor in the I/O pads; then, the measured power can be assigned to registers, clock lines and buffers. Finally, although off-chip power is strongly application-dependent, for a given FPGA-based system it can be easily determined simply by measuring chip power twice: first in normal operation, and then activating the 3-state output pad option. The difference between the values, allows the designer to diagnose if off-chip power reduction is necessary.

The average value of the power components can be indirectly determinated by measuring average FPGA input current; or subtracting average system power values measured with and without clocking the FPGA chip under test (all the circuits should include registered I/O). Both methods require the voltage on the FPGA chip to be held constant.

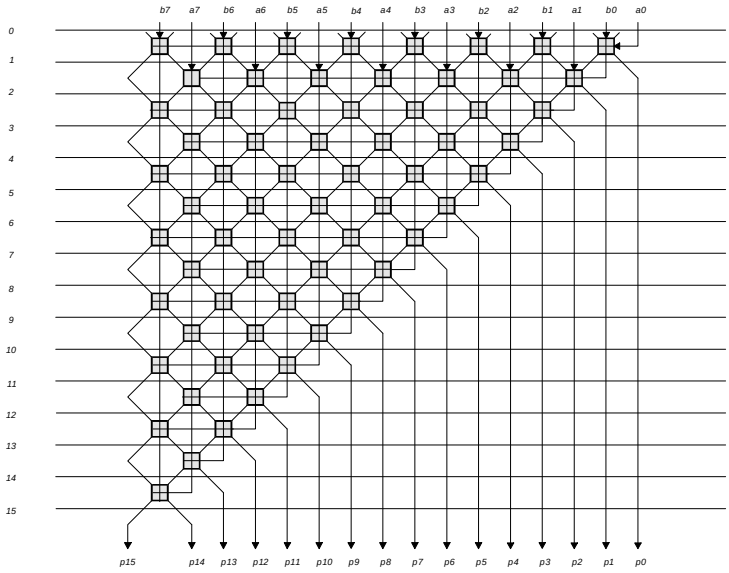

Fig.2: 8-bit Guild Multiplier. Equitemporal lines.

3 Experimental Results

The relationship between pipelining, partitioning and power consumption has been quantified by a set of 8-bit Guild pipelined array multipliers [9] implemented using a XC3090PC84-100 and a XC4005PC84-6 Xilinx FPGAs. Each array family includes versions pipelined with five different granularities
of elementary processors (EP) between successive register banks [10]: β=1 (all EP I/O registered),β=2 (data registered in even lines of Fig.2),β=4 (data registered in lines 0, 4, 8, ...),β=8 (data just are registered in lines 0, 8 and 15), and finally β=15, a combinational array with registered I/O. In order to assess the effect of efficient LUT utilization, two versions for each circuit has been constructed for the XC3090: a non-optimized (default APR) implementation; and another corresponding to a manual partitioning optimization. Additionally, some full manual high-optimized prototypes has been also developed.

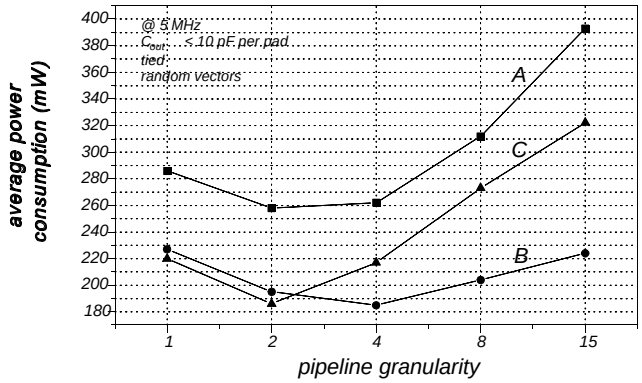

Fig.3: 8-bit Guild Array. Average power consumption vs granularity. XC3090 default PPR (curve A) and optimized partitioning (curve B), and XC4005 default PPR (curve C)

3.1 Pipelining as a Low-Power Strategy: Fig.3 shows the average power consumption of three pipelined array sets versus pipeline granularity (measured at 5 MHz, a frequency at wich all prototypes can be compared). The off-chip power quota has been maintained as low as possible in order to avoid masking datapath power effects; thus, each pad supports just the 10 pF (max.) logic analyzer probe load. Despite the hardware overhead, fine grain pipelines not only ran faster than combinational versions (β=15), but also exhibed lower consumption if operated at the same frequency. In all cases the minimum power dissipated corresponded to logic depth from two to four LUTs between registers. Thus, pipelining allows the designer to trade power consumption for additional logic blocks and latency. For example, for default implementation conditions, the consumption

of a 5 MHz β=15 multiplier can be reduced by 33 % (XC3090) or by 58 % (XC4005) if it is β=4 pipelined. In both cases the number of registers required would increase from 32 to 104, and the latency from one to four clock cycles.

3.2 Power Reduction via Partitioning Improvements: This strategy not only diminishes CLB occupation and speed up circuits but also reduces power consumption. It can be evaluated comparing curve A and B on Fig.3. Note that, for each β, both versions have the same synchronization and off-chip power; thus, the difference corresponds exclusively to datapath power consumption. Note that the benefits of partitioning improvements increase with the logic depth.

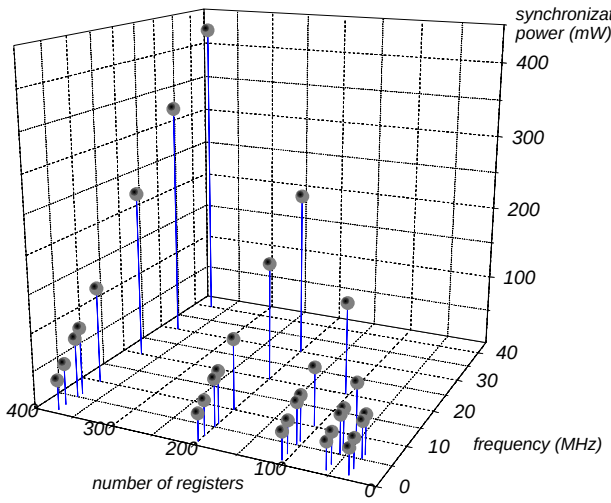

Fig.4: Synchronization power vs frequency and number of registers.

3.3 Synchronization Power: This component can be modeled by measuring multipliers with different numbers of registers. In Fig.4 the value of the power consumption has been plotted, versus frequency and number of register. Thus, the following model has been derived for 3090PC84-100 synchronization power (tied option) as a function of the number of registers (NR) and frequency:

$$Synch.\ Power\ (\ mW\ )\quad (\ 2{,}7\quad 0{,}019\ NR\ )\ frequency\ (MHz)\quad 31\ mW$$

Note that the term of 31 mW includes overall chip consumption at low frequency. This model has been successfully utilized to predict the results of other tied arrays with

different topologies and numbers of registers[1]. Error for different partitioning and placement have been estimated near 7% (circuits that make use of both CLB registers exhibed less sinchronization power). Instead of the number of register, a similar model can be developed using the number of *.k* pins connected to the clock line.

Although for high frequency operation, the synchronization power fraction is not excessive, it can not be reduced by using the CLB clock enable facility. Thus, the application of techniques like stopable clocks to FPGA-based systems, must block the clock signal at chip input pin in order to be effective. Finally, in Fig.5 an example is shown of the three components of total power for a full manual PPR, 8-bits-240CLBs-70MHz,β=1, pipelined Guild array multiplier on a 3090PC84-100.

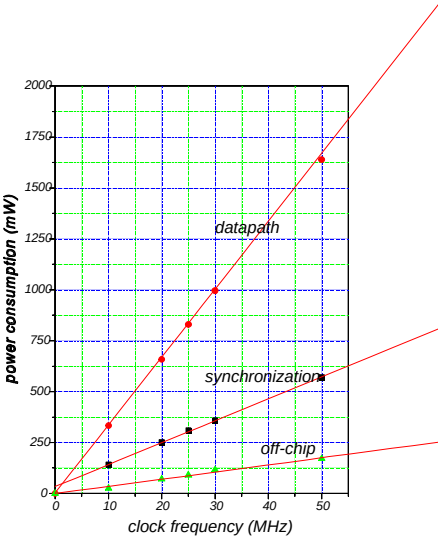

Fig.5: Power components. High-optimized 8-bit β=1 pipelined array multiplier

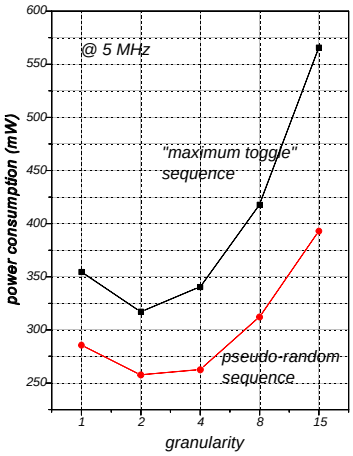

Fig.6: Data-dependence of power consumption. Non-optimized part.

3.4 Data-dependence of Power Consumption: In Fig.6 the power consumption is shown for the same family of circuits when they process different input data. Note that the sequence of sixteen vectors for maximum output toggle produces a significant increase in power consumption, even for the small off-chip capacitance values. However, this effect can be utilized in a reverse mode: for a particular application, a subset of data

[1] All the experiments presented in this paper have been repeated using a Hatamian array [11], providing similar results.

that minimizes toggle output would be effective to disminish datapth power consumption. In another application, a similar idea is utilized in [7], where the internal activity is reduced by minimizing it at the circuit inputs.

3.5 Correlation between Occupation, Bandwidth and Power Consumption: In spite of architectural trade-offs, from an implementational point of view, occupation, bandwidth and power consumption can be improved simultaneously. In Fig.7 is plotted power consumption @ 5 MHz versus minimun clock period of a set β=15 multipliers implemented on a XC3090PC84-100 using the "aprloop" facility. Although there are some exceptions, the faster circuit runs, the smaller is the power consumption for a given frequency.

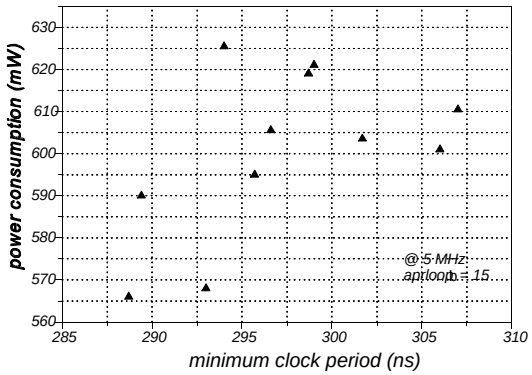

Fig.7: Aprloop results from a power consumption perspective.

5 Conclusions

The effect of pipelining as a low-power design technique on FPGAs has been quantified; a model for synchronization power has been proposed; and a general methodology to characterize others applications or FPGAs has been presented. The results show that pipelining can produce a reduction of power consumption by about 25% - 40 %, and nearly 15% - 45 % can be achieved simply by improving partitioning. It can be stated as a rule of thumb, that circuits than run faster, use less CLBs and dissipate less power. The common origin of these improvements is the reduction of the interconnection network influence.

From a research or educational point of view, it has been demonstrated that RAM-based FPGAs exhibit important advantages over other technologies in terms of power analysis; their layout editors combined with the changeable structure of logic blocks allow circuit modifications like: inserting/deleting registers without altering the routing; modifying the

routing without affecting the logic or placement; isolating any block from the system clock; confining critical parts to LUTs; using positive or negative edge registers; disconnecting the outputs pads, etc. Additionally, the fast design cycle and reprogramability of this technology allows prototypes to be builded and measured without significant cost. However, the FPGA net information based on delays rather than node capacitances make modelling of the consumption difficult.

Acknowledges

This work has been supported by the CICYT of Spain under contract TIC92-0083. The authors wish to thank Seamus McQuaid for his constructive comments.

References

1. R. Keyes, "Miniaturization of electronics and its limits", *IBM J. of Res. Develop.* Vol.32, nº1. January 1988.
2. Xilinx, Inc., *Technical Conference and Seminar Series*, 1995.
3. Z. Lemnios y K. Gabriel, "Low-Power Electronic", *IEEE Design & Test of Computers*, pp. 8-13, winter 1994.
4. 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
5 D. Wong, *"Techniques for Designing High-Performance Digital Circuits Using Wave Pipelining"*, Technical report No. CLS-TR-92-508. Stanford University, february 1992.
6. E. Boemo, S. López, G. González and J. Meneses, "On the usefulness of pipelining and wave pipelining as low-power design technique", Proc. 1995 PATMOS Conf. (in press).
7. C. Su, C. Tsui y A. Despain, "Low Power Architecture Design and Compilation Techniques for High-Performance Processors", *Proc. Sprint COMPCON 94*, pp.489-498. IEEE Press, 1994.
8. R. Hossain, L. Wronski y A. Albicki, "Low Power Desing Using Double Edge Triggered Flip-Flops", IEEE Trans. on VLSI Systems, Vol.2, Nº2, pp.261-265. June 1994.
9. H.H. Guild, "Fully Iterative Fast Array for Binary Multiplication and Addition", *Electronic Letters*, pp.263, Vol.5, Nº12, June 1969.
10. C. Hauck, C. Bamji and J. Allen, "The Systematic Exploration of Pipelined Array Multiplier Performance", *Proceeding ICASSP 85*, pp.1461-1464. New York: IEEE Press, 1985.
11. 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.

This work was published in Lecture Notes in Computer Science, No.975, pp.149-157. Berlin: Springer-Verlag 1995.