Title
Minimizing Leakage Energy in FPGAs Using Intentional Post-Silicon Device Aging

Author
Wei, Sheng

Publication Date
2013

Peer reviewed|Thesis/dissertation

University of California

Los Angeles

Minimizing Leakage Energy in FPGAs Using Intentional Post-Silicon Device Aging

A thesis submitted in partial satisfaction

of the requirements for the degree

Master of Science in Computer Science

by

Sheng Wei

2013

Abstract of the Thesis

Minimizing Leakage Energy in FPGAs Using Intentional Post-Silicon Device Aging

by

Sheng Wei

Master of Science in Computer Science

University of California, Los Angeles, 2013

Professor Miodrag Potkonjak, Chair

The presence of process variation (PV) in deep submicron technologies has become a major concern for energy optimization attempts on FPGAs. We develop a negative bias temperature instability (NBTI) aging-based post-silicon leakage energy optimization scheme that stresses the components that are not used or are off the critical paths to reduce the total leakage energy consumption. Furthermore, we obtain the input vectors for aging by formulating the aging objectives into a satisfiability (SAT) problem. We synthesize the low leakage energy designs on Xilinx Spartan6 FPGA and evaluate the leakage energy savings on a set of ITC99 and Opencores benchmarks.

The thesis of Sheng Wei is approved.

Jason Cong

Milos Ercegovac

Miodrag Potkonjak, Committee Chair

University of California, Los Angeles

2013

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

As the rapid growth of FPGA applications in various domains, especially in portable digital applications and remote sensing, energy efficient FPGA design has drawn a great deal of attention in the community [7][15][28][22][33].

The existing energy optimization strategies conduct pre-silicon voltage tuning in order to reduce the energy consumption [37][28][15]. However, these approaches cannot provide optimal solution considering the fact that there is process variation (PV) [13] during the IC manufacturing period. As PV completely varies the IC key properties, such as delay and power, from their nominal specifications, one may argue that the approaches proposed in [37] and [15] during the pre-silicion stage are no longer optimal after manufacturing due to the impact of process variation.

In order to consider and compensate for the impact of process variation on energy efficient FPGA design, we conduct energy optimization during the post-silicon stage after the IC is manufactured [55]. Our key observation is that threshold voltage plays an important role in the leakage energy consumption, as shown in Equation (1.1) [31]:

$$P_{leakage} = 2 \cdot n \cdot \mu \cdot C_{ox} \cdot \frac{W}{L} \cdot (\frac{kT}{q})^2 \cdot D \cdot V_{dd} \cdot e^{\frac{\sigma \cdot V_{dd} - V_{th}}{n \cdot (kT/q)}} \tag{1.1}$$

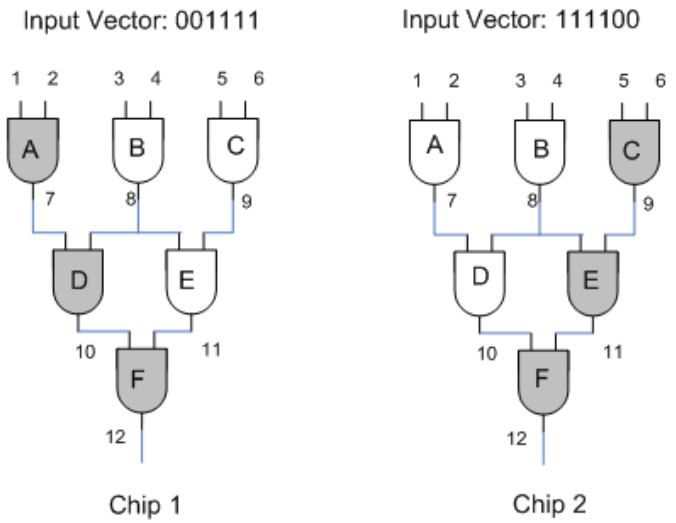
where L is effective channel length, V_{th} is threshold voltage, W is gate width, V_{dd}

is supply voltage, n is subthreshold slope, μ is mobility, C_{ox} is oxide capacitance, D is clock period, $\phi_t = kT/q$ is thermal voltage, and σ is drain induced barrier lowering (DIBL) factor.

It is important to note from Equation (1.1) that the leakage energy of a transistor or a logic gate decreases exponentially with the increase of the threshold voltage. Also, threshold voltage is the main parameter affected by device aging, such as negative bias temperature instability (NBTI) [47]. Therefore, one can control the threshold voltages by conducting device aging during the post-silicon stage. Based on these observations, we develop a post-silicon device aging-based approach to reduce the total leakage energy consumption on a given FPGA design. Our approach leverages NBTI aging to intentionally stress the FPGA, so that the threshold voltages of selected transistors are increased and the leakage energy is decreased exponentially.

However, we face two issues that may impact the effectiveness and applicability of the proposed approach on FPGAs. Firstly, it is well known that NBTI mainly affects the threshold voltage of the PMOS transistors while having little impact on the NMOS transistors [47]. Furthermore, NMOS transistors are major components of the FPGA, such as in the multiplexers that implement the lookup tables (LUTs). Therefore, it is important to find a post-silicon leakage reduction approach for NMOS transistors as well. Secondly, aging impacts the speed of the design and may introduce performance degradations to the target FPGA.

For the NMOS transistors, we select input vectors for the unused NMOS transistors so that their leakage energy can be minimized. Consequently, the leakage energy of the PMOS transistors dominates the total leakage consumption of the FPGA, which can be reduced by NBTI aging. Note that our input vector selection is different from the traditional input vector control (IVC) methods such as



Chip	Gate	Vth	Before Aging		After Aging		on Critical Path?
			Delay	Leakage Power	Delay	Leakage Power	
1	A	0.27	3.46	0.07	3.46	0.07	Y
	B	0.22	3.38	0.28	3.80	0.09	N
	C	0.19	3.26	0.57	3.44	0.21	N
	D	0.33	4.93	0.01	4.93	0.01	Y
	E	0.20	3.46	0.39	3.66	0.13	N
	F	0.28	5.49	0.04	5.49	0.04	Y
	Sum	-	13.88	***1.36***	14.30	***0.55***	-
2	A	0.26	2.95	0.10	3.19	0.02	N
	B	0.24	2.55	0.21	2.73	0.06	N
	C	0.33	4.47	0.01	4.47	0.01	Y
	D	0.28	3.11	0.07	3.38	0.01	N
	E	0.25	4.38	0.10	4.38	0.10	Y
	F	0.30	3.92	0.03	3.92	0.03	Y
	Sum	-	12.77	***0.52***	12.77	***0.23***	-

Figure 1.1: Example of aging-based leakage energy reduction.

[43] and [1], because we take into consideration of the impact of process variation, and our goal is not to minimize the total leakage energy but to expose the leakage energy of one group of transistors (PMOS transistors) while minimizing the other group (NMOS transistors).

For the delay degradation problem, we selectively age only the unused components of the FPGA, or the components that are off the critical path, so that aging does not impact the critical path delay of the entire design. To achieve this goal, we leverage satisfiability (SAT) to determine the aging input vectors. In this way, we can avoid impacting the performance while still obtaining total leakage energy savings. Also, one key observation that supports this approach is that there is a very small percentage of gates that would appear on critical path, especially in large designs, leaving a large room for aging and reducing leakage energy for.

Figure 1.1 shows a motivating example of our aging-based energy reduction scheme on two chips from the same 6-gate design. The nominal design value of threashold voltage is 0.25V. However, due to process variation, the threshold voltage values after manufacturing vary greatly from the nominal specification, as shown in the table. In this case, no matter how effective the pre-silicon optimization is, the PV would completely change the IC properties and invalidate any pre-silicon optimization efforts. Also, PV exhibits different impacts on different chips, which results in different critical paths on chip 1 and chip 2, e.g., the critical path is A-D-F on chip 1 and C-E-F on chip 2 even though they are of the same design. This phenomenon makes it difficult to keep track of the performance impact in the post-silicon stage.

In our aging-based energy optimization scheme, we intentionally age the PMOS transistors that are off the critical path (non-CP gates) by using NBTI,

e.g., gates B, C, and E on chip 1 and gates A, B, and D on chip 2, to reduce the leakage energy of the circuit while maintaining the original performance. To achieve this goal, we find the input vectors that stress only the non-CP gates using a SAT solver. For example, for chip 1, we find input vectors that set gates B, C, and E to signal 1 (stressed) and gates A, D, and F to signal 0 (unstressed). Consequently, the aging process reduces the leakage energy of non-CP gates exponentially without impacting the delays of the CP gates. Furthermore, we leverage a SAT-based approach to find the input vectors for aging, which will be discussed in details in Section V. In this example, the SAT solver provides us with the solutions of input vector 001111 and 111100 for chip 1 and chip 2, respectively. With the intentional aging, we observe that the total leakage energy is reduced by around 2X in both cases.

Although there exist other post-silicon approaches that adjust threshold voltages at runtime to reduce leakage energy consumption, such as adaptive body biasing (ABB) [33][16][9], our device aging-based approach distinguishes from and outperforms them by requiring no hardware instrumentation and providing higher granularities of voltage control. Firstly, the ABB-based approaches requires additional hardware, such as body bias circuitry [45], to be embedded in the target design, in order to adapt and apply the bias voltage at runtime; Our device aging-based approach does not introduce any hardware instrumentation and, therefore, induces zero area overhead. Secondly, the ABB-based approach assigns only a limited number of different bias voltages to the target circuit and, consequently, many gates must share the same body bias voltage despite of the fact that they require different optimal threshold adjustments due to process variation; however, our device aging-based approach provides a fine-grained granularity (i.e., at gate level) in terms of voltage control by stressing different transistors for different periods of time.

Originally, we started this work on application-specific integrated circuit (ASIC), where the applicability of the aging approach is more obvious, and the limitations compared to FPGAs are much less. However, FPGAs have the unique feature that, in most cases, a large number of the logic blocks are not used after mapping the specific application. This provides us with a much larger room for leakage energy reduction than in the ASIC case, where all gates are used by the application. With the rapid growth of FPGA applications, especially application-specific FPGAs [17], we see a potentially good use case for our approach.

Our technical contributions include the following:

- the first use of intentional device aging to compensate for process variation and ensure low leakage energy FPGA design;
- a complete design flow for low leakage energy FPGAs via intentional device aging of non-critical components; and
- the use of SAT to determine aging input vectors.

CHAPTER 2

Related Work

In this chapter, we survey the directly related research low power design. We start with process variation in FPGAs and techniques for reducing its negative impacts. Next, after covering enabling technologies for our new approach, we discuss most popular low power FPGA and IC design approaches.

2.1 Process Variation

Process variation has been firmly established as one of main design and manufacturing issues [34]. Its relevance is well illustrated by the observation that FPGA is proposed as the most suitable platform for PV characterization for a pertinent silicon process [29].

Several research groups used sensors and side channels measurements to characterize several FPGA chips generations [46][5] , such as the techniques for improving timing yields under PV [21], reserving resource in routing channels [40], and power characterizations [35]. For example, University of Virginia researchers demonstrated excellent match between thermal profiles obtained using thermal simulation and FPGA sensor measurements. Several authors presented techniques for improving manufacturing and timing yields under PV [21]. Rubin and DeHon have introduced a simple but effective strategy and a set of algorithms that reserve resource in routing channels to greatly increase FPGA yield [40].

Nowroz and Reda proposed use of two step procedure for power characterization of FPGAs [35]. Zick and Hayes introduces an approach for tracking changes in dynamic power by tracking corresponding changes in the frequencies of nearby ring oscillators [59]. Techniques for optimized implementation on FPGA platforms in circuits subject to PV have attracted a great deal of research and development attention [11]. AIST FPGA group has proposed an approach that eliminates need for FPGA characterization for optimizing delay in presence of process variation by creating multiple structures with identical functionality [32]. Sivaswamy et al. have developed process variation-aware FPGA router that improves yield or reduces delay [42].

2.2 Low Power FPGA Design

FPGA has been making great strides in technology and performances against application-specific integrated circuit (ASIC). For example, while typical ASIC nowadays is realized in 90 nm technology modern FPGA are realized in 32, 28, and even 22 nm. It is well known that FPGAs have significantly higher energy budgets than ASICs often by an order of magnitude or even higher [60]. This is particularly true for leakage energy component that keeps increasing as the feature size keeps being reduced.

Li et al. [27] developed a mixed-level power model and an accurate power analysis framework for FPGAs. Several research groups developed techniques for low energy FPGA implementation using dual supply and dual threshold voltages and analyzed their benefits [8][15][28]. Also, several research teams investigated benefits of dynamic voltage scaling strategies for reducing power and/or energy in FPGA-based systems [12][38]. Energy reduction in FPGA system through reduction of glitching has also received significant attention [22].

Bijansky and Aziz [4] advocated a FPGA architecture with a dual voltage supply to improve the yield at an increased level of implementation complexity, wherein the supply voltage for individual configurable logic blocks (CLBs) can be assigned after fabrication. They showed an order of magnitude yield improvement at the cost of increased implementation complexity. Nabaa et al. [33] proposed use of adaptive body bias through the use of the new FPGA architecture to reduce the leakage energy by 3 times. An EFPL FPGA group has developed techniques for power reduction and compensation of process variations in FPGA chips [19]. Srinivasan et al. proposed use of input vector control technique in unused CLBs for reduction of leakage energy [43].

To the best of our knowledge the new approach is the first technique that employs device aging for reduction of leakage energy. It has an advantage over the existing gate sizing and dual supply/threshold voltages in that it exploits precise information about benefits and limitations due to slow-down of each specific gate.

2.3 Adaptive Body Biasing

Adaptive body biasing has been widely adopted as an efficient post-silicon approach in the research efforts of leakage energy reduction and performance optimization [33][16][9]. For example, Nabaa et al. proposed use of ABB through the use of the new FPGA architecture that includes an additional characterizer circuit to reduce the leakage energy by 3 times [33]. Gregg et al. [16] proposed using ABB to compensate for the process variation and improve delay and leakage. Chen et al. [9] compare the effectiveness of adaptive supply voltage (ASV) and ABB.

Furthermore, similar with the pre-silicon dual V_{th} approach, researchers have

proposed multiple body bias values in the target circuit, each drives a subset of the gates. For example, Xu et al. [56] cluster the gates at a finer-grained level and apply multiple ABB values to control the leakage energy consumption.

However, due to the impact of process variation, every gate varies from its nominal specifications in a different way and, consequently, there does not exist a single or small number of body bias values that can compensate for the PV of all gates. Also, the implementation of ABB in the target design requires additional body bias circuitry, which significantly increases the area overhead. More importantly, although the bias voltages can be adapted at runtime, the placement of body bias circuitry has to be determined at the design time, i.e., which components of the target circuit should be grouped in one cluster and applied the same bias voltage. This makes it difficult for the approach to accommodate the systematic and random impacts caused by process variation.

We for the first time leverage device aging as the primary post-silicon method to reduce the leakage energy consumption. By leveraging aging instead of ABB for leakage energy reduction, we are able to not only adjust the threshold voltage at the gate level, but also introduce no additional hardware to the target design.

CHAPTER 3

Preliminaries

In this chapter, we introduce the system models that we employ in this paper, in addition to the leakage power model in Equation (1.1). By analyzing the models, we show that leakage energy decreases exponentially with the increase of threshold voltage, while the delay degradation is close to linear. This observation serves as the foundation of our aging-based leakage energy optimization approach.

3.1 Process Variation

Process variation in IC manufacturing is the deviation of IC parameter values from nominal specifications [10]. It causes major variations in gate-level physical properties such as L_{eff} and V_{th}, which are two major sources of PV. For example, due to the impact of PV, the actual L_{eff} of a manufactured gate can be expressed by Equation (3.1), where L_{nom} is the nominal design value of the effective length, and ΔL is the variation in the manufacturing process.

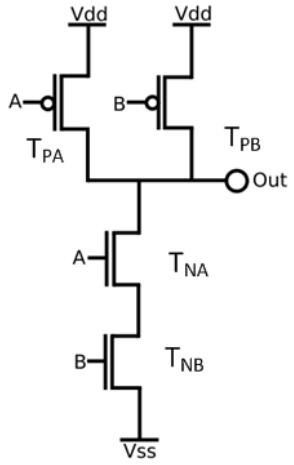
$$L_{eff} = L_{nom} + \Delta L \tag{3.1}$$

In particular, we use the PV model from Asenov et al. [2] and the quad-tree spatial correlation model from Cline et al. [13] for modeling the ΔL.

3.2 Energy Model

As discussed earlier, Equation (1.1) is the gate-level leakage energy model [31], which indicates that the leakage energy of a transistor decreases exponentially with the increase of threshold voltage. This enables us to significantly reduce the leakage energy by stressing (i.e., setting to the *Open* state) the target transistors and thus increasing their threshold voltage.

Figure 3.1 shows the leakage energy of a NAND gate that is composed of two PMOS and two NMOS transistors. The transistors are either in the *Open* (O) or *Close* (C) states depending on the inputs (A and B) of the NAND gate.



A	B	Out	T_{PA}	T_{PB}	T_{NA}	T_{NB}	Leakage (nA)
0	0	1	C	C	O	O	37.84
0	1	1	C	O	O	C	95.17
1	0	1	O	C	C	O	100.3
1	1	0	O	O	C	C	454.5

Figure 3.1: Leakage current of PMOS and NMOS transistors in a NAND gate. "C" and "O" represent *Close* and *Open* states of the transistor, respectively.

There are other FPGA components that can be used for further leakage re-

duction. For example, one can try to reduce leakage of configuration bits. We do not conduct these degrees of freedom because usually configuration bits are not on the critical paths and implemented using high threshold voltage transistors that have exponentially lower leakage.

The gate-level switching energy model [31] is described by Equation (3.2), where the switching energy is dependent on gate width W, gate length L, supply voltage V_{dd}, and switching provability α.

$$P_{switching} = \alpha \cdot C_{ox} \cdot W \cdot L \cdot V_{dd}^2 \tag{3.2}$$

3.3 Delay Model

The delay of a single logic gate can be expressed as

$$d = gh + p \tag{3.3}$$

where g and h are logical effort and electrical effort, respectively; and p is parasitic delay. In particular, we use the delay model in [31] that connects the gate delay to its sizing and operating voltages:

$$\begin{aligned} Delay = \frac{k_{tp} \cdot k_{fit} \cdot L^2}{2 \cdot n \cdot \mu \cdot \phi_t^2} \cdot \frac{V_{dd}}{(ln(e^{\frac{(1+\sigma)V_{dd}-V_{th}}{2 \cdot n \cdot \phi_t}} + 1))^2} \\ \cdot \frac{\gamma_i \cdot W_i + W_{i+1}}{W_i} \end{aligned} \tag{3.4}$$

where subscripts i and $i+1$ represent the the driver and load gates, respectively; γ is the ratio of gate parasitic to input capacitance; and k_{tp} and k_{fit} are fitting parameters.

From Equation (3.4), we can conclude that the delay increases approximately linearly with the increase of threshold voltage. Comparing to the exponential relation between leakage energy and threshold voltage, it provides us with large room to reduce the leakage energy consumption via aging.

3.4 Aging Model

We employ the aging model proposed in paper [6] for our aging process. The time dependence of V_{th} shift due to NBTI follows fractional power law of the stress time, as shown in Equation (3.5).

$$\Delta V_{th} = A \cdot e^{\beta V_G} \cdot e^{-E_\alpha / kT} \cdot t^{0.25} \tag{3.5}$$

where V_G is the applied gate voltage; A and β are constants; E_α is the measured activation energy of the NBTI process; T is the temperature; and t is the stress time.

Threshold voltage is also impacted by other aging phenomena such as positive bias temperature instabilities (PBTI) in NMOS transistors and hot carrier injection (HCI). However, the impact of these device aging mechanisms is lower than the impact of NBTI in PMOS transistors, they can be treated in similar ways, both conceptually and algorithmically, and their impacts are orthogonal. Also, several other device aging mechanisms, such as electromigration, do not impact threshold voltages and, therefore, cannot be used for reduction of leakage energy.

Kim et al. [58] compared the impacts NBTI and PBTI on threshold voltage using different materials. Probably the most effective way to apply the proposed leakage reduction technique is to age PMOS transistors during stand-by mode of operation through application of appropriate input vectors [57].

Furthermore, we note that the increase of V_{th} under static NBTI effect can be ultra fast, for example, Wang et al. [47] showed that more than 1mV threshold voltage increase can be achieved by within 10^4 seconds under stress. It is well known that the speed of aging is significantly faster if it is conducted under higher temperatures. However, in order to increase temperature one has to spend significant energy. Also, higher temperature and thermal cycling reduce the lifetime of an integrated circuit. Therefore, we do not recommend temperature-driven faster aging techniques.

As prior work, researchers from our group have conducted aging on Xilinx FPGAs. The method to age a specific LUT on a specific slice is to instantiate the LUT in HDL with a constant value to its inputs (i.e., the aging input vector that stresses the PMOS transistors), and apply location constraint in the synthesis tool to map it to a specific cell.

CHAPTER 4

Low Energy FPGA Design Flow

We introduce an aging-based post-silicon leakage optimization scheme that is integrated into the FPGA design flow. It provides us with reduced leakage energy consumption on FPGAs. Figure 4.1 shows the overall flow of our approach. There are three phases in the flow to reduce leakage energy consumption.

- *FPGA pre-processing*, in which we conduct gate-level characterization (GLC) on the FPGA to identify the CLBs that introduce high leakage energy or high delay, and mask them from cell placement. Also, the GLC results facilitate us with identifying the critical paths on the FPGA, which are later required in the aging process.

- *Post-silicon device aging*, in which we stress the PMOS transistors in unused or non-critical CLBs, while setting the critical CLBs in the unstressed mode. We select the aging input vectors by solving a SAT problem that set the corresponding PMOS transistors under stress (i.e., *Open* states).

- *FPGA post-processing*. After aging the FPGA, we conduct post-processing and validation, in which we apply low leakage energy input vectors to the unused NMOS transistors, in order to minimize their leakage energy consumption; Also, we validate the reduced leakage energy using the leakage power model and repeat the aging process if necessary.

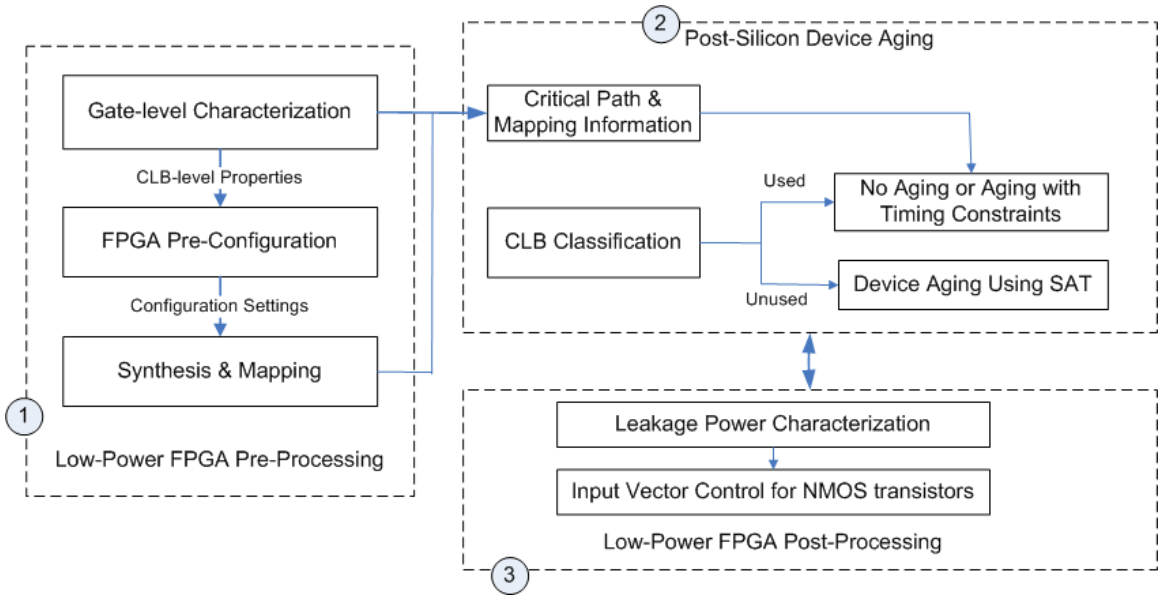


Figure 4.1: Overall flow of the aging-based low leakage energy FPGA design.

CHAPTER 5

Low Energy FPGA Pre-Processing

We pre-process the FPGA to identify the CLBs that consume ultra-high leakage energy compared to other components and mask them from the cell placement. In this way, we exclude the high leakage components from consideration prior to conducting device aging, which significantly reduces the cost of the approach. We achieve this goal by characterizing the gate-level properties, such as leakage power, which can be obtained by measuring the leakage power of the entire circuit and solving a system of linear equations. Furthermore, we employ the results of gate-level characterization to identify the critical paths on the FPGA, which are required in the device aging process.

5.1 Gate-level Characterization

In gate-level characterization (GLC) [49][51][52][48][50][54][53], we recover the gate-level IC properties from global side-channel measurements under the application of various input vectors. For example, when J input vectors have been applied on a target circuit with K gates, the gate-level leakage energy values can be solved using the following linear program (LP):

$$
\begin{aligned}
Objective: \quad & \min_{1 \le j \le J} \mathcal{F}(err_j) \\
Constraints: \quad & \Sigma_{k=1}^{K} E_{jk} = \tilde{E}_j + err_j \\
& j = 1, \ldots, J
\end{aligned}
\tag{5.1}
$$

where E_{jk} is the leakage energy of gate k ($k = 1, \ldots, K$) when input vector j ($j = 1, \ldots, J$) is applied; $\tilde{E}_j$ is the measured total leakage energy when the input vector j is applied; err_j is the measurement error; $\mathcal{F}$ is a metric for quantifying the measurement errors, such as l_1 or l_2 norm. In this LP formulation, E_{jk} can be expressed as a product of its constant nominal value $E_{nom,jk}$ and a scaling factor (due to PV) δ_k, i.e., $E_{jk} = \delta_k E_{nom,jk}$. By solving the LP with δ_k as the variables, we can obtain the value of E_{jk} for each gate k ($k = 1, \ldots, K$).

Furthermore, by following the energy models (i.e., Equations (1.1) and (3.2)), we can formulate a system of nonlinear equations and solve for the physical-level properties (i.e., threshold voltage and effective channel length) of each individual gate.

5.2 Scenario-based Critical Path Identification

Our device-aging based leakage energy reduction approach requires us to identify the unused and non-critical components on the FPGAs, so that we can avoid aging the critical components and compromising the performance. While the unused components can be obtained from synthesis and mapping of the specific application, the identification of the critical paths is non-trivial.

According to the delay model (i.e., Equation (3.4)), the critical path of an IC is impacted by the following two factors: (1) process variation, which varies

the IC physical properties, such as threshold voltage and effective channel length, from their nominal specifications. Consequently, the critical paths are different on different chips even if they are of the same design. (2) input vectors, which impacts the switching delay of a logic gate. Therefore, the critical path of a given FPGA is both chip-specific and application-specific.

In order to identify the critical path, we design a scenario-based statistical analysis method on two different levels. First, to address the impact of process variation, we generate multiple IC instances following well accepted PV models, such as Gaussian model [2] and quad-tree model [13], and determine the critical path on each design. Then, we summarize the simulation results and obtain the statistics of critical path gates, e.g., how often a specific gate would appear on critical path over a number of PV-impacted chips. The statistics provide us with intuitions how critical each gate is in terms of the performance and leakage energy consumption. Second, to address the impact of input vectors, we simulate the application-specific input vectors on the target design and calculate the resulting critical paths.

5.3 The Application-Specific Limitation

Based on the pre-processing flow, we note that the proposed approach is application-specific, because the unused or non-critical components may change when we re-program the FPGA for another application. We have the following two ways to address the issue. First, the approach can accommodate a group of domain-specific applications, instead of just one, by combining their statistics of used/unused components. Second, the application-specific approach is particularly useful in the scenario where re-programming is not required or rarely happens, such as the FPGA integrated with system-on-chips (SoCs)[17].

CHAPTER 6

Aging-based Post-silicon Leakage Energy Reduction

In this section, we discuss in details the individual steps for reducing the total leakage energy of FPGAs via post-silicon aging, following the overall flow introduced in Chapter 4.

6.1 Problem Definition

Our goal in selecting the aging input vectors is to be able to stress the unused and non-critical CLBs while keeping the critical CLBs unstressed. After obtaining the CLB information (i.e., used/unused CLBs and critical/non-critical CLBs) from the pre-processing step, we formulate a SAT problem to determine the candidate input vectors.

SAT is a problem that determines if a set of variables can be assigned to satisfy a boolean formula. In the IC domain, if the circuit netlist is known, the signal of each gate can be expressed as a boolean formula with a set of primary input signals as the variables. Therefore, the input vector selection problem that intends to set a specific gate or a set of gates to given signals can be naturally converted to a SAT instance. By solving the SAT instance, we can provably find the desirable input vectors based on the given signal requirements. In our SAT

problem formulation, we use an objective file to specify the signals for a subset of gates that need new input vectors. The gates that are not included in the objective file will be assumed as don't-care by the SAT solver.

The method we use to age a specific gate is to stress the gate, for example, setting its output signal to 1. Therefore, the very first step in achieving aging of specific gates in the circuit is to determine the input vectors that we can apply to stress the required gates and reduce the total leakage energy consumption. In other words, the problem becomes how to determine the input vectors for an IC that set the inputs of specific gates to specific signals, as formally stated as follows:

Aging Input Vector Selection Problem. Given an IC with N gates, where each gate $i(1 \leq i < N)$ has a required signal for each of its inputs, the aging input vector selection problem aims to find the input vector of the IC that satisfy the signal requirements of the gates.

6.2 SAT Problem Formulation

We note that the aging input vector selection problem can be translated to a Boolean SAT problem, where the signal of each gate is represented by a Boolean expression concerning the primary input signals. By evaluating all the Boolean expressions (i.e., clauses) to be true simultaneously, the SAT problem aims to search for the corresponding primary input vectors, namely aging input vectors. In particular, the SAT problem for aging N gates can be formulated as the following:

$$\wedge_{i=1}^{N} C_i = 1 \tag{6.1}$$

and

$$C_i = \begin{cases} O_i(PI_1, PI_2, ...PI_k), & b_i = 1; \\ !O_i(PI_1, PI_2, ...PI_k), & b_i = 0; \\ 1, & b_i = don't - care; \end{cases}$$

where C_i is the clause for setting gate i to its objective signal; $O(PI_1, PI_2, ...PI_k)$ is the Boolean expression for the signal of gate i based on inputs PI_1 to PI_k; b_i is the objective signal of gate i; N is the number of gates to age; and k is the number of primary inputs of the circuit.

After solving the SAT problem, the SAT solver, such as [41], would generate the primary input vector PI_1 to PI_k, which satisfies the SAT objective.

6.3 SAT Example

In Figure 6.1, we demonstrate the SAT problem formulation in a small example. In the objective file for the expected gate signals, we set all the gates that are on the critical path, namely critical path (CP) gates (i.e., gates 1 and 2) to signal 0 and most of the gates that are off the critical path, namely non-critical path (non-CP) gates (i.e., gates 4 and 5), to signal 1. The SAT solver obtains input vector 110 that satisfy the specified objectives. Note that although gate 3 is off the critical path, it cannot be stressed in this case due to the fact that the corresponding SAT problem is infeasible.

6.4 Justification of the Aging Approach on Industrial FPGA Designs

As discussed earlier, the method we age a particular LUT on the FPGA is by stressing its PMOS transistors (i.e., keeping them in the "closed" state con-

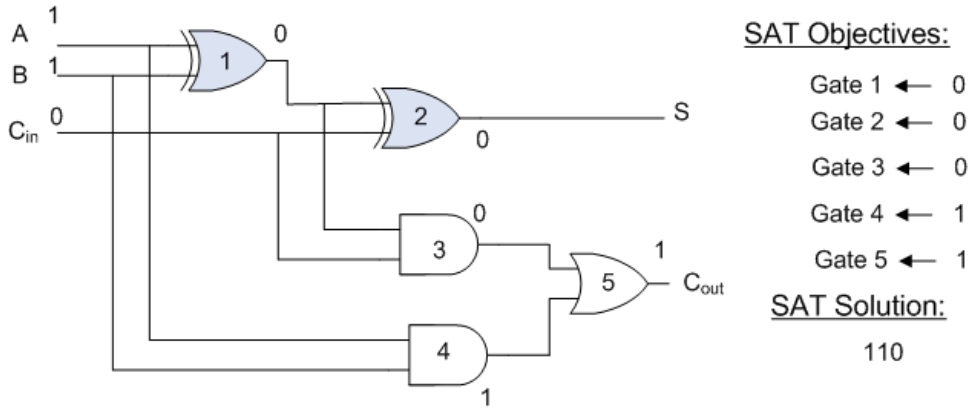


Figure 6.1: Example of SAT formulation for aging input vector selection on a 1-bit full adder. The SAT objectives are formulated to set all the CP gates to signal 0 and most of the non-CP gates to signal 1. The output from the SAT solver provides the input vectors that satisfy the objectives.

stantly). Due to the NBTI effect, their threshold voltage will increase, and the leakage energy will decrease exponentially. In particular, on a FPGA, we apply selected aging input vectors on the I/O pads that end up stressing the unused PMOS transistors. The applicability of the approach can be partly justified using related research in the FPGA testing domain [44][39]. In FPGA testing, the goal is to apply a (relatively large) set of test vectors via the I/O pads that provide a full coverage of the faults. In our case, we apply aging input vectors in the same way, but we only need one or very few aging input vectors to stress some particular PMOS transistors.

Real industrial FPGAs consist of both logic blocks and routing structures, as summarized by Betz et al. [3]. In the next two subsections, we justify the applicability of our approach by analyzing the transistor-level implementations of each component in the industrial FPGAs [3][26][25][24][23][38].

6.4.1 Justification on FPGA Logic Blocks

In most of the industrial FPGA structures, LUT is the major component for implementing the logic blocks. To justify the applicability of our approach on LUTs, we analyze a transistor-level implementation of a commonly used 4-input LUT in Figure 6.2 [20]. The 4-input LUT can be implemented with 15 PMOS and 15 NMOS transistors as shown in Figure 1. If we set the input bits (A3, A2, A1, A0) to be (0, 0, 0, 0), all 15 PMOS transistors will be conducting (i.e., under stress). Meanwhile, all 15 NMOS transistors will not be under stress, but in our approach, we do not intend to age the NMOS transistors, since NBTI has little impact on NMOS transistors.

This is just a motivational example for LUT, which can be aged using only one aging input vector. In other implementations, if the PMOS transistors cannot be aged by one input vector, we can accommodate the approach to apply multiple input vectors. For example, suppose there are n transistors $T = t_1, t_2, ..., t_n$, we can apply k aging input vectors $I_1, ..., I_k$ interleavingly, with I_i stressing T_i, where $Ti \subset T, 1 \leq i \leq k$. Eventually, we can age the set of transistors $T_{aged} = T_1 \cup ..., \cup T_k$. We assume that T_{aged} will be sufficiently large to cover T, even if it is not the case, it does not compromise the effectiveness of our approach. For leakage energy reduction, we are not aiming to age all PMOS transistors to the greatest extent. Instead, our attempt is to provide a best-effort energy reduction solution by aging as many PMOS transistors as we can, with limited costs that we can afford. The leakage energy savings are sufficiently large because of the exponential relation between the leakage energy and threshold voltage.

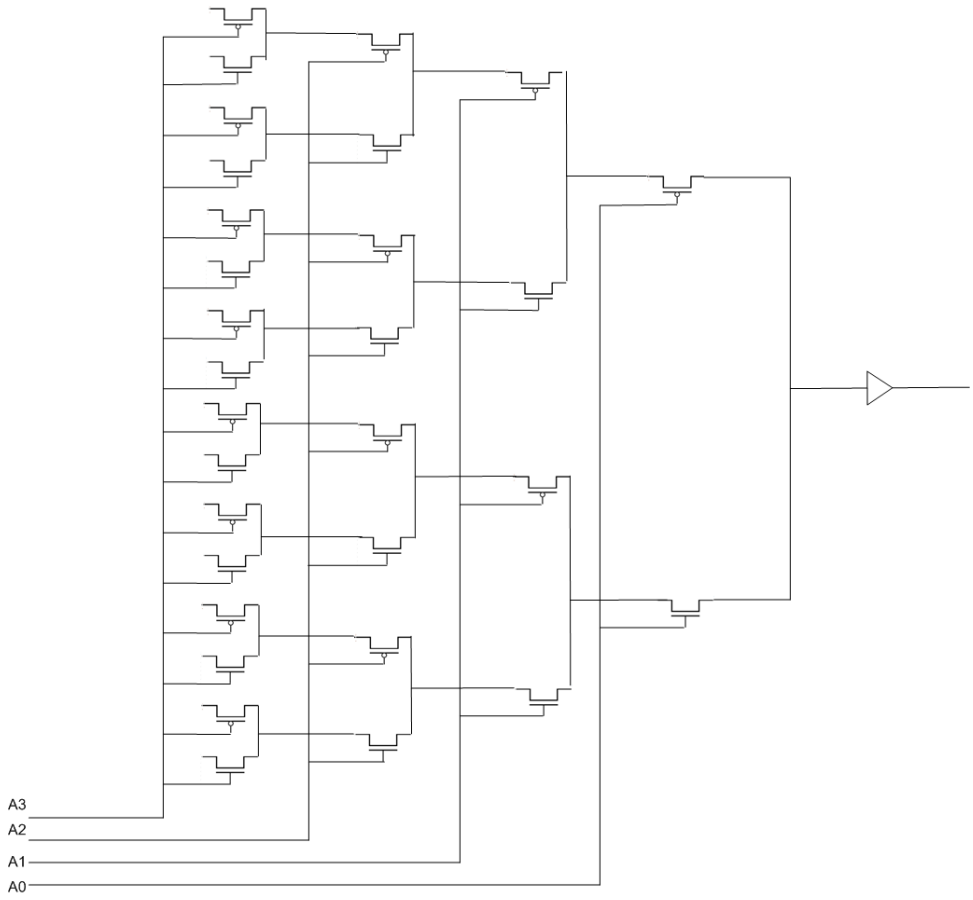


Figure 6.2: Example: A transistor-level implementation of 4-input LUT [20], where (A3, A2, A1, A0) are the 4-bit inputs (address) to the LUT.

6.4.2 Justification on FPGA Routing Structures

The routing structure on an industrial FPGA design mainly includes the connection blocks and switch blocks that are implemented by a set of programmable switches[3]. Furthermore, the programmable switches are implemented using the following components: SRAM cells, buffers, and multiplexers [3]. To justify the applicability of our approach on FPGA routing structures, we analyze the transistor-level implementation of the three components in details.

6.4.2.1 SRAM Cells

According to [3] (page 208, Figure B.1), a SRAM cell can be implemented using 2 PMOS transistors and 4 NMOS transistors. We can age this SRAM cell by applying two aging input vectors (i.e., two values loaded to prog_data). In particular, $prog\_data = 1$ will age the left PMOS transistor, and $prog\_data = 0$ will age the right PMOS transistor.

6.4.2.2 Buffers

According to [3] (page 210, Figure B.3), a multi-stage buffer can be implemented using 3 PMOS transistors and 3 NMOS transistors. We can age the buffer using two aging input vectors (i.e., two values loaded to the In bit). In particular, $In = 0$ will age the first and third PMOS transistor, and $In = 1$ will age the second PMOS transistor.

6.4.2.3 Multiplexers

In FPGAs, multiplexers have been implemented in the following three ways: (1) (in most cases) NMOS-only pass transistors [3] (Figure B.7, page 213 in [3]); (2)

CMOS transmission gates that consist of both NMOS and PMOS transistors [23] (Figure 5, page 61 in [23]); or (3) Level-restoring circuit with NMOS pass transistors [38] (Figure 2, page 24 in [38]). For our aging approach, in cases (2) and (3), we can obtain significant leakage energy savings by aging the PMOS transistors using the same method as for SRAM and buffers; however, for case (1), the NBTI aging is not effective, since the transistors are all NMOS transistors. To address this issue, we have been investigating on another major aging effect, namely hot carrier injection (HCI) [14][30], which ages both PMOS and NMOS transistors. The same approach applies to the NMOS-only based implementation of 4-input LUT [3] (Figure B.9, page 215 in [3]).We consider the detailed discussion of HCI as the future work of the thesis.

CHAPTER 7

Low Energy FPGA Post-Processing

7.1 Input Vector Control for Unused NMOS transistors

It is well known that NBTI only increases the threshold voltage of PMOS devices while having very limited impact on the NMOS transistors. Therefore, the aforementioned NBTI-based aging approach only reduces the leakage energy of PMOS transistors on the FPGA. In order to address the issue for NMOS transistors, we conduct input vector control after the aging process, where the goal is to apply input vectors that set the NMOS transistors in the low leakage mode [1]. For example, as shown in Figure 3.1, the leakage current of an NAND gate can be reduced from 454.5nA to 37.84nA when set to the low leakage mode, i.e., $A = 0, B = 0$. In this way, the PMOS transistors become dominating in terms of the leakage energy consumption of the FPGA, which has been reduced exponentially using our NBTI aging approach.

7.2 Validations on Leakage Energy Savings

We validate the leakage energy savings obtained from our aging-based approach using the leakage power model presented in Equation (1.1). In particular, we calculate the ratio between the leakage energy values before and after aging and use it for validation. Also, the calculated leakage energy saving serves as a metric

for determining whether we should repeat the pre-processing and aging processes and obtain possibly larger energy savings, under the condition that the aged components are not saturated to cause a malfunction or create new critical paths that cause delay degradations.

CHAPTER 8

Experimental Results

8.1 Evaluation Method

We implement the aging-based low leakage energy scheme into the FPGA design flow. In particular, we use Xilinx Spartan6-XC6SLX45 as the FPGA platform to evaluate our approach. We synthesize and map a set of ITC99 [18] and Opencores [36] test benchmarks on the FPGA and evaluate the leakage energy savings based on the leakage power model in Equation (1.1) and the obtained cell placement results using Xilinx ISE 13.1.

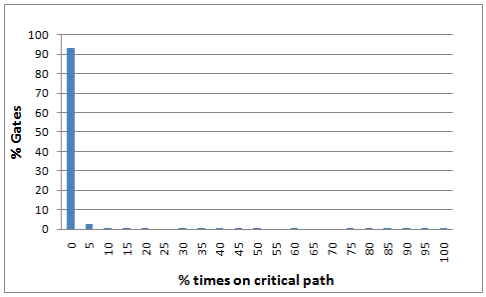
Ideally, one should measure and demonstrate the effectiveness of a new approach on actual FPGA chip. However, unfortunately, currently it is not possible to undertake such measurements on FPGAs. The issue is that the technique cannot be applied on already used FPGAs that are anyhow unintentionally aged. For example, our industrial contacts explained us that each FPGA chip, before being shipped to a customer, is thoroughly tested and that currently this testing is done in such a way that device aging consequences are ignored.

8.2 Aging Configuration

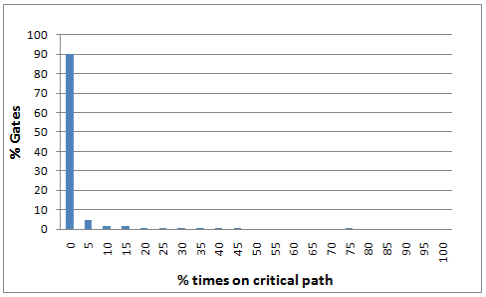
In order age the appropriate gates, we first conduct simulation on a set of chips under the presence of PV and find the impact of PV on the critical path gates.

The statistics of critical path gates provide us with an insight on which gates we should age for leakage energy reduction.

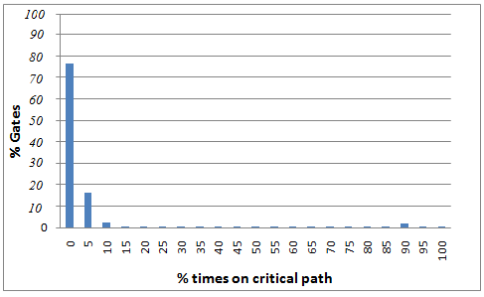
The results are shown in Figure 8.1, which indicate the following: (1) There is a large portion (i.e., over 75%) of the gates that never appear on critical path and, therefore, there is large room for leakage energy reduction via aging; (2) A small percentage of gates (i.e., less than 25%) appear on critical path in at least one chip being tested, which we should avoid aging; and (3) The critical path gates are different on different chips due to the impact of PV. For example, in all tested benchmarks, most of the CP gates appear on critical paths in less than 5% of the chips, making the CP gates on each chip vary from others in a great extend.



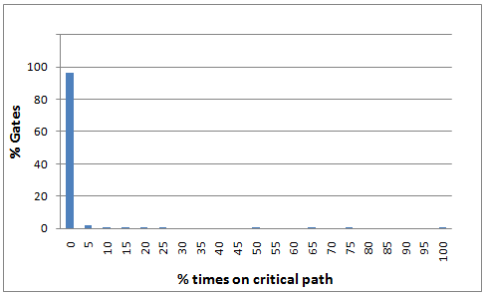
(a) C3540



(b) C5315



(c) C6288



(d) C7552

Figure 8.1: Statistics of critical path gates on multiple chips subject to process variation.

8.3 Leakage Energy Savings on FPGA

Table 8.1 shows the resource usage of the test circuits and the leakage energy savings after applying our aging-based approach. The first five columns show the benchmarks and their resource usage on Xilinx FPGA Spartan6-XC6SLX45. Here we define a "gate" as an internal FPGA gate that is used to build the target design. We observe that for most applications, a very small percentage of the LUTs and Registers are being used, leaving a large room for leakage energy savings using intentional aging of the unused components. In the last column, we include the energy saving results (i.e., the number of times compared to the original leakage energy consumption). Note that the energy saving results are obtained from aging only the PMOS transistors in unused CLBs and all the FPGA routing structures. We can obtain additional energy savings by aging the PMOS transistors in used but non-critical CLBs with timing constraints, as well as setting the unused NMOS transistors in low leakage mode via input vector control. However, the results in Table 8.1 indicate that we have already obtained a substantial amount of leakage energy savings by aging only the PMOS transistors in unused CLBs on the FPGA. Also, in this case, there is no performance degradation resulted from the aging process, since all the aged components are not being used by the application.

Table 8.1: Leakage energy savings via aging on FPGA. The first five columns show the benchmarks and their resource usage on Xilinx FPGA Spartan6-XC6SLX45. The last column shows the leakage energy savings after applying our NBTI aging-based approach.

Benchmarks	# Gates	# FFs	Used LUTs	Used Registers	Leakage Savings (Times)
b14	10,098	245	3.80%	0.45%	1.55
b15	8,922	449	7.93%	1.16%	1.51
b17	32,326	1,415	23.50%	3.60%	1.41
b17_1	39,665	1,415	23.03%	3.60%	1.41
b18	114,621	3,320	48.86%	7.80%	1.26
b18_1	108,482	3,320	48.52%	7.81%	1.27
b22	29,951	735	11.46%	1.32%	1.49
b22_1	21,772	735	12.03%	1.33%	1.49
openrisc	44,476	2,021	46.14%	12.58%	1.28
eth	35,961	10,544	10.88%	4.29%	1.49
tv80s	5,604	359	6.28%	0.73%	1.53
usb_funct	9,164	1,746	6.76%	3.08%	1.52
ac97	8,020	2,199	3.83%	2.18%	1.54

CHAPTER 9

Conclusion

We have developed an aging-based post-silicon approach for reducing the leakage energy consumption on FPGA. The approach leverages the fact that the leakage energy decreases exponentially with the increase the of threshold voltage, which can be achieved by aging the transistors. In order to minimize the delay degradation due to aging, we obtained aging input vectors that age the unused and non-critical components of the FPGA design by solving a SAT problem. Our experiments on Xilinx FPGA platform indicated substantial leakage energy reductions while maintaining the initial performance.

References

[1] Y. Alkabani, T. Massey, F. Koushanfar, and M. Potkonjak. Input vector control for post-silicon leakage current minimization in the presence of manufacturing variability. In *Design Automation Conference (DAC)*, pages 606–609, 2008.

[2] A. Asenov. Random dopant induced threshold voltage lowering and fluctuations in sub-0.1 μm MOSFET's: A 3-D "atomistic" simulation study. *IEEE Transactions on Electron Devices*, 45(12):2505–2513, 1998.

[3] V. Betz, J. Rose, and A. Marquardt. *Architecture and CAD for Deep-Submicron FPGAs*. Kluwer Academic Publishers, 1999.

[4] S. Bijansky and A. Aziz. TuneFPGA: post-silicon tuning of dual-Vdd FPGAs. In *Design Automation Conference (DAC)*, pages 796–799, 2008.

[5] F. Bruguier, P. Benoit, P. Maurine, and L. Torres. A new process characterization method for FPGAs based on electromagnetic analysis. In *International Conference on Field Programmable Logic and Applications (FPL)*, pages 20–23, 2011.

[6] S. Chakravarthi, A. Krishnan, V. Reddy, C. Machala, and S. Krishnan. A comprehensive framework for predictive modeling of negative bias temperature instability. In *International Reliability Physics Symposium (IRPS)*, pages 273–282, 2004.

[7] D. Chen, J. Cong, Y. Fan, and L. Wan. LOPASS: a low-power architectural synthesis system for FPGAs with interconnect estimation and optimization. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 18(4), 2010.

[8] D. Chen, J. Cong, F. Li, and L. He. Low-power technology mapping for FPGA architectures with dual supply voltages. In *International Symposium on Field Programmable Gate Arrays (FPGA)*, pages 109–117, 2004.

[9] T. Chen and S. Naffziger. Comparison of adaptive body bias (ABB) and adaptive supply voltage (ASV) for improving delay and leakage under the presence of process variation. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,*, 11(5):888–899, 2003.

[10] B. Cheng, S. Roya, A. Browna, C. Millara, and A. Asenov. Evaluation of statistical technology generation LSTP MOSFETs. *Solid-State Electronics*, 53(7):767–772, 2009.

[11] L. Cheng, J. Xiong, L. He, and M. Hutton. FPGA performance optimization via chipwise placement considering process variations. In *International Conference on Field Programmable Logic and Applications (FPL)*, pages 1–6, 2006.

[12] C. Chow, L. Tsui, P. Leong, W. Luk, and S. Wilton. Dynamic voltage scaling for commercial FPGAs. In *International Conference on Field-Programmable Technology (FPT)*, pages 173–180, 2005.

[13] B. Cline, K. Chopra, D. Blaauw, and Y. Cao. Analysis and modeling of CD variation for statistical static timing. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 60–66, 2006.

[14] P. Fang, J. Tao, J. Chen, and C. Hu. Design in hot-carrier reliability for high performance logic applications. In *IEEE Custom Integrated Circuits Conference (CICC)*, pages 521–531, 1998.

[15] A. Gayasen, K. Lee, N. Vijaykrishnan, M. Kandemir, M.J. Irwin, and T. Tuan. A dual-Vdd low power fpga architecture. In *International Conference on Field Programmable Logic and Applications (FPL)*, pages 145–157, 2004.

[16] J. Gregg and T. Chen. Post silicon power/performance optimization in the presence of process variations using individual well-adaptive body biasing. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 15(3):366–376, 2007.

[17] M. Hammerquist and R. Lysecky. Design space exploration for application-specific FPGAs in system-on-a-chip designs. In *IEEE International SOC Conference (SOCC)*, pages 279–282, 2008.

[18] ITC99 benchmark, http://www.cerc.utexas.edu/itc99-benchmarks/bench.html.

[19] B. Kheradmand-Boroujeni, C. Piguet, and Y. Leblebici. AVGS-Mux style: a novel technology and device independent technique for reducing power and compensating process variations in FPGA fabrics. In *Design, Automation and Test in Europe (DATE)*, pages 339–344, 2010.

[20] D. Koch. *Partial Reconfiguration on FPGAs*. Springer, 2013.

[21] A. Kumar and M. Anis. Fpga design for timing yield under process variations. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 18(3):423–435, 2010.

[22] J. Lamoureux, G. Lemieux, and S. Wilton. Glitchless: an active glitch minimization technique for FPGAs. In *International Symposium on Field Programmable Gate Arrays (FPGA)*, pages 156–165, 2007.

[23] E. Lee, G. Lemieux, and S. Mirabbasi. Interconnect driver design for long wires in field-programmable gate arrays. *Journal of Signal Processing Systems*, 51(1):57–76, 2008.

[24] G. Lemieux and D. Lewis. Analytical framework for switch block design. In *International Conference on Field Programmable Logic and Applications (FPL)*, pages 122–131, 2002.

[25] G. Lemieux and D. Lewis. Circuit design of routing switches. In *International Symposium on Field Programmable Gate Arrays (FPGA)*, pages 19–28, 2002.

[26] P. Leventis, M. Chan, Chan M, D. Lewis, B. Nouban, G. Powell, B. Vest, M. Wong, R. Xia, and J. Costello. Cyclone: A low-cost, high-performance FPGA. In *IEEE Custom Integrated Circuits Conference (CICC)*, pages 49–52, 2003.

[27] F. Li, Y. Lin, L. He, D. Chen, and J. Cong. Power modeling and characteristics of field programmable gate arrays. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 24(11):1712–1724, 2006.

[28] F. Li, Y. Lin, L. He, and J. Cong. Low-power FPGA using pre-defined dual-Vdd/dual-Vt fabrics. In *International Symposium on Field Programmable Gate Arrays (FPGA)*, pages 42–50, 2004.

[29] X. Li, F. Wang, T. La, and Z. Ling. FPGA as process monitor-an effective method to characterize poly gate CD variation and its impact on product performance and yield. *IEEE Transactions on Semiconductor Manufacturing*, 17(3):267–272, 2004.

[30] D. Lorenz, G. Georgakos, and U. Schlichtmann. Aging analysis of circuit timing considering NBTI and HCI. In *IEEE International On-Line Testing Symposium (IOLTS)*, pages 3–8, 2009.

[31] D. Markovic, C. Wang, L. Alarcon, T. Liu, and J. Rabaey. Ultralow-power design in near-threshold region. *Proceedings of the IEEE*, 98(2):237–252, 2010.

[32] Y. Matsumoto, M. Hioki, T. Kawanami, T. Tsutsumi, T. Nakagawa, T. Sekigawa, and H. Koike. Performance and yield enhancement of FPGAs with

within-die variation using multiple configurations. In *International Symposium on Field Programmable Gate Arrays (FPGA)*, pages 169–177, 2007.

[33] G. Nabaa, N. Azizi, and F. Najm. An adaptive FPGA architecture with process variation compensation and reduced leakage. In *Design Automation Conference (DAC)*, pages 624–629, 2006.

[34] S. Nassif, K. Bernstein, D. Frank, A. Gattiker, W. Haensch, B. Ji, E. Nowak, D. Pearson, and N. Rohrer. High performance CMOS variability in the 65nm regime and beyond. In *International Electron Devices Meeting (IEDM)*, pages 569–571, 2007.

[35] A. Nowroz and S. Reda. Thermal and power characterization of field-programmable gate arrays. In *International Symposium on Field Programmable Gate Arrays (FPGA)*, pages 111–114, 2011.

[36] Opencores, http://opencores.org/.

[37] P. Pant, R. Roy, and A. Chatterjee. Dual-threshold voltage assignment with transistor sizing for low power CMOS circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 9(2):390–394, 2001.

[38] A. Rahman and V. Polavarapuv. Evaluation of low-leakage design techniques for field programmable gate arrays. In *International Symposium on Field Programmable Gate Arrays (FPGA)*, pages 23–30, 2004.

[39] M. Renovell and Y. Zorian. Different experiments in test generation for XILINX FPGAs. In *International Test Conference (ITC)*, page 854, 2000.

[40] R. Rubin and A. DeHon. Choose-your-own-adventure routing: lightweight load-time defect avoidance. In *International Symposium on Field Programmable Gate Arrays (FPGA)*, pages 23–32, 2009.

[41] sat4j: http://www.sat4j.org/.

[42] S. Sivaswamy and K. Bazargan. Variation-aware routing for FPGAs. In *International Symposium on Field Programmable Gate Arrays (FPGA)*, pages 71–79, 2007.

[43] A. Srivastava, D. Sylvester, and D. Blaauw. *Statistical Analysis and Optimization for VLSI: Timing and Power*. Springer, 1999.

[44] Mehdi Tahoori. Application-dependent testing of FPGAs. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 14(9):1024–1033, 2006.

[45] J. Tschanz, J. Kao, S. Narendra, R. Nair, D. Antoniadis, A. Chandrakasan, and V. De. Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage. In *International Solid-State Circuits Conference (ISSCC)*, pages 422–478, 2002.

[46] S. Velusamy, W. Huang, J. Lach, M. Stan, and K. Skadron. Monitoring temperature in FPGA based SoCs. In *International Conference on Computer Design (ICCD)*, pages 634–637, 2005.

[47] W. Wang, S. Yang, S. Bhardwaj, R. Vattikonda, S. Vrudhula, F. Liu, and Y. Cao. The impact of NBTI on the performance of combinational and sequential circuits. In *Design Automation Conference (DAC)*, pages 364–369, 2007.

[48] S. Wei, K. Li, F. Koushanfar, and M. Potkonjak. Provably complete hardware trojan detection using test point insertion. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 569–576, 2012.

[49] S. Wei, S. Meguerdichian, and M. Potkonjak. Gate-level characterization: Foundations and hardware security applications. In *Design Automation Conference (DAC)*, pages 222–227, 2010.

[50] S. Wei, S. Meguerdichian, and M. Potkonjak. Malicious circuitry detection using thermal conditioning. *IEEE Transactions on Information Forensics and Security*, 6(3):1136–1145, 2011.

[51] S. Wei and M. Potkonjak. Scalable segmentation-based malicious circuitry detection and diagnosis. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 483–486, 2010.

[52] S. Wei and M. Potkonjak. Integrated circuit security techniques using variable supply voltage. In *Design Automation Conference (DAC)*, pages 248–253, 2011.

[53] S. Wei and M. Potkonjak. Scalable consistency-based hardware Trojan detection and diagnosis. In *International Conference on Network and System Security (NSS)*, pages 176–183, 2011.

[54] S. Wei and M. Potkonjak. Scalable hardware Trojan diagnosis. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 20(6):1049–1057, 2012.

[55] S. Wei, J. Zheng, and M. Potkonjak. Low power FPGA design using post-silicon device aging (abstract only). In *International Symposium on Field Programmable Gate Arrays (FPGA)*, page 277, 2013.

[56] H. Xu, R. Vemuri, and W. Jone. Temporal and spatial idleness exploitation for optimal-grained leakage control. In *International Conference on Computer-Aided Design (ICCAD)*, pages 468–473, 2009.

[57] L. Yuan and G. Qu. A combined gate replacement and input vector control approach for leakage current reduction. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 14(2):173–182, 2006.

[58] S. Zafar, Y. Kim, V. Narayanan, C. Cabral, V. Paruchuri, B. Doris, J. Stathis, A. Callegari, and M. Chudzik. A comparative study of NBTI and PBTI (charge trapping) in SiO2/HfO2 stacks with FUSI, TiN, Re gates. In *Symposium on VLSI Technology*, pages 23–25, 2006.

[59] K. Zick and J. Hayes. On-line sensing for healthier FPGA systems. In *International Symposium on Field Programmable Gate Arrays (FPGA)*, pages 239–248, 2010.

[60] P. Zuchowski, C. Reynolds, R. Grupp, S. Davis, B. Cremen, and B. Troxel. A hybrid ASIC and FPGA architecture. In *International Conference on Computer Aided Design (ICCAD)*, pages 187–194, 2002.