Title
Hardware Design Techniques for Securing and Synthesizing Resource-Constrained IoT Systems

Author
Wendt, James Bradley

Publication Date
2015

Peer reviewed|Thesis/dissertation

University of California
Los Angeles

Hardware Design Techniques for Securing and Synthesizing Resource-Constrained IoT Systems

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

James Bradley Wendt

2015

Abstract of the Dissertation

Hardware Design Techniques for Securing and Synthesizing Resource-Constrained IoT Systems

by

James Bradley Wendt

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 2015

Professor Miodrag Potkonjak, Chair

The Internet of Things (IoT) paradigm has enabled everyday objects to be instrumented and operated in such a way that they can be queried and controlled over the Internet. While the 1990s saw the connection of nearly 1 billion users to the Internet, and the 2000s saw an increase to 2 billion users through the proliferation of mobile phones, it is estimated that by 2020, IoT will introduce an additional 26 billion units to the Internet ecosystem.

IoT systems have been developed and envisioned for numerous environments and applications and their rapid emergence has introduced a number of unique opportunities and challenges in the space of hardware design. For example, the application of these systems in a variety of environments has lead to an increased need for new low power design solutions, specifically for remote and battery operated devices.

While low energy design is paramount for the successful deployment of resource-constrained IoT devices, their often remote and physically accessible nature has

also contributed to rendering traditional cryptographic techniques insufficient to address all of the security concerns surrounding these systems. Hence, security has become an equally important requirement. These two desiderata, security and low energy, are often conflicting requirements by nature and present a challenging scenario for design. For example, higher levels of security often require larger amounts of energy consumption.

In this dissertation we present energy-aware design methods for the synthesis and security of IoT systems. We present novel energy reduction and delay minimization techniques applied on integrated circuit subsystems of IoT applications in order to enable near-threshold computing operation with maximal energy savings and minimal speed degradation. We also present semantics-based techniques for the organization and coordination of system components in order to both reduce energy consumption as well as increase energy harvester production. Finally, we demonstrate new techniques for securing IoT systems, including intellectual property protection, trusted remote sensing, and trusted chip selection.

The dissertation of James Bradley Wendt is approved.

Chih-Kong Ken Yang

Miloš D. Ercegovac

Jens Palsberg

Miodrag Potkonjak, Committee Chair

University of California, Los Angeles

2015

To my parents.

Table of Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I would like to thank my adviser and doctoral committee chair, Professor Miodrag Potkonjak, for his support and guidance through all of my years as a graduate student at UCLA. He not only guided me through the world of academic research but also provided me with numerous opportunities to grow as a scientist, for which I am very grateful.

I would like to thank my doctoral committee members, Professors C. K. Ken Yang, Milos Ercegovac, and Jens Palsberg, for their insightful comments and advice on my prospectus, dissertation, and final defense, and for aiding me in the completion of my graduate studies.

To my professors in the UCLA Computer Science Department and my professors at Pomona College, thank you for not only providing me with a solid scientific foundation, but also for inspiring me with your passion for research and teaching.

To my hosts and colleagues at UC Riverside, JPL, AT&T, and Google, thank you for your patience and mentorship, and for introducing me to the world of engineering and research in industry.

Many thanks to my collaborators and colleagues at UCLA: Sheng Wei, Viswha Goudar, Nathaniel Conos, Jason Zheng, Jong Ahnn, Saro Meguerdichian, Teng Xu, Jia Guo, and Hongxiang Gu. Thank you for participating with me in technical discussions and debate, working late to finish joint papers and proposals, and for sharing in the graduate school experience with me. I would like to extend a special thank you to Nathaniel Conos, Saro Meguerdichian, Vishwa Goudar, and Teng Xu who collaborated with me and provided experimental results and paper writing assistance to support my work in Chapters 4, 5, 6, and 7, respectively.

Special thanks goes to Professor Potkonjak who oversaw and co-authored all of the research work that constituted the chapters in this thesis.

And finally, I would like to thank my family and friends who have supported me throughout my academic journey. A special thank you to my parents for their unconditional love and support, providing me with the means and confidence to dream big. To my sister for her love, friendship, and patience. And to Samantha for her unwavering love and encouragement, especially on the days when I needed it most. Thank you.

Vita

2005-2009	B.A. (Physics) Pomona College
2009-2011	M.S. (Computer Science) University of California, Los Angeles

Publications

P1. J. B. Wendt, "Nanocell-based (public) physical unclonable function," Master's thesis, University of California, Los Angeles, 2011.

P2. J. B. Wendt and M. Potkonjak, "Medical diagnostic-based sensor selection," in *IEEE Sensors*, pp. 1507–1510, 2011.

P3. J. B. Wendt and M. Potkonjak, "Nanotechnology-based trusted remote sensing," in *IEEE Sensors*, pp. 1213–1216, 2011.

P4. J. B. Wendt, S. Meguerdichian, H. Noshadi, and M. Potkonjak, "Energy and cost reduction in localized multisensory systems through application-driven compression," in *Data Compression Conference (DCC)*, p. 411, 2012.

P5. J. B. Wendt, S. Meguerdichian, H. Noshadi, and M. Potkonjak, "Semantics-driven sensor configuration for energy reduction in medical sensor networks," in

International Symposium on Low Power Electronics and Design (ISLPED), pp. 303–308, 2012.

P6. J. B. Wendt, V. Goudar, H. Noshadi, and M. Potkonjak, "Spatiotemporal assignment of energy harvesters on a self-sustaining medical shoe," in *IEEE Sensors*, pp. 1312–1315, 2012.

P7. S. Meguerdichian, J. B. Wendt, and M. Potkonjak, "Simultaneous trust and privacy in medical systems using public physical unclonable functions," in *Telehealthcare Computing and Engineering: Principles and Design* (F. Hu, ed.), pp. 679–698, CRC Press, 2013.

P8. J. B. Wendt, S. Meguerdichian, and M. Potkonjak, "Small is beautiful and smart," in *Telehealthcare Computing and Engineering: Principles and Design* (F. Hu, ed.), pp. 341–358, CRC Press, 2013.

P9. T. Xu, J. B. Wendt, and M. Potkonjak, "Digital bimodal function: An ultra-low energy security primitive," in *International Symposium on Low Power Electronics and Design (ISLPED)*, pp. 292–297, 2013.

P10. J. B. Wendt and M. Potkonjak, "Improving energy efficiency in sensing subsystems via near-threshold computing and device aging," in *IEEE Sensors*, pp. 555–558, 2013.

P11. J. B. Wendt and M. Potkonjak, "The bidirectional polyomino partitioned PPUF as a hardware security primitive," in *IEEE Global Conference on Signal*

and Information Processing (GlobalSIP), pp. 257–260, 2013.

P12. V. Goudar, J. B. Wendt, M. Potkonjak, Z. Ren, P. Brochu, and Q. Pei, "Leveraging human gait characteristics towards self-sustained operation of low-power mobile devices," in *World Forum on Internet of Things (WF-IoT)*, pp. 468–473, 2014.

P13. M. Rostami, J. B. Wendt, M. Potkonjak, and F. Koushanfar, "Quo vadis, PUF? Trends and challenges of emerging physical-disorder based security," in *Design, Automation & Test in Europe (DATE)*, no. 352, pp. 1–6, 2014.

P14. S. Wei, J. B. Wendt, A. Nahapetian, and M. Potkonjak, "Reverse engineering and prevention techniques for physical unclonable functions using side channels," in *Design Automation Conference (DAC)*, no. 90, pp. 1–6, 2014.

P15. J. B. Wendt, F. Koushanfar, and M. Potkonjak, "Techniques for foundry identification," in *Design Automation Conference (DAC)*, no. 208, pp. 1–6, 2014.

P16. T. Xu, J. B. Wendt, and M. Potkonjak, "Matched digital PUFs for low power security in implantable medical devices," in *International Conference on Healthcare Informatics (ICHI)*, pp. 33-38, 2014.

P17. T. Xu, J. B. Wendt, and M. Potkonjak, "Secure remote sensing and communication using digital PUFs," in *Symposium on Architectures for Networking and Communications Systems (ANCS)*, pp. 173–184, 2014.

P18. T. Xu, J. B. Wendt, and M. Potkonjak, "Ultra-lightweight symmetric-key cipher for resource constrained systems," in *IEEE Sensors*, pp. 1252–1255, 2014.

P19. J. B. Wendt and M. Potkonjak, "Hardware obfuscation using PUF-based logic," in *International Conference on Computer-Aided Design (ICCAD)*, pp. 270–277, 2014.

P20. T. Xu, J. B. Wendt, and M. Potkonjak, "Security of IoT systems: Design challenges and opportunities," in *International Conference on Computer-Aided Design (ICCAD)*, pp. 417–423, 2014.

P21. J. B. Wendt, N. A. Conos, and M. Potkonjak, "Multiple constant multiplication implementations in near-threshold computing systems," to appear in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

P22. J. Rajendran, R. Karri, J. B. Wendt, M. Potkonjak, N. McDonald, G. S. Rose, and B. Wysocki, "Nanoelectronic solutions for hardware security," to appear in *Proceedings of the IEEE*, 2015.

CHAPTER 1

Introduction

1.1 The Internet of Things

The Internet of Things is a rapidly emerging paradigm in which the essential concept is that a great variety of Things are equipped in such a way that they can be queried and operated over the Internet. This has enabled both user-to-object applications, in which users can control and sense their environments remotely, and object-to-object applications, enabling opportunities for big data learning, semantic reasoning, and coordinated operation. This paradigm is projected to have a positive impact on many aspects of everyday life, ranging from organizational and operational improvements, to better semantic reasoning and inference in personal, industrial, and global spaces.

Radio-frequency identification (RFID) tags provide a first glimpse into a very rudimentary generation of IoT devices. Companies such as Walmart and Amazon use these devices to automatically identify and track inventory. Over the course of the last half decade a number of new IoT applications and devices have emerged, including wearable medical apparatuses, home automation systems, industrial sensors and actuators, biochip transponders, and wearable accessories such as Internet-connected watches, eye wear, and shoes.

Over the course of the last half decade a number of new IoT applications and devices have emerged, including wearable medical devices, home automa-

tion systems, automotive applications, industrial sensors and actuators, biochip transponders, and wearable accessories such as Internet-connected watches, eye wear, and shoes.

By the year 2000 nearly 1 billion users worldwide had connected to the Internet. By 2010 the emergence of mobile connectivity had increased the number of connected devices to 2 billion. In 2008, the National Intelligence Council predicted IoT to be one of the top six disruptive technologies impacting the US economy out to 2025 [1]. Current estimates predict that by 2020 IoT will encompass 26 billion units [2].

In just the last few years many large technology players, including Google, Intel, ARM, and General Electric, have announced and demonstrated significant investments into the IoT space. For example, in 2014 Google acquired Nest Labs, a home automation company, whose flagship device is an Internet connected thermostat that not only allows enables remote querying and control of an individual's home temperature over the Internet, but also aims to learn user behaviors and habits with respect to temperature, power consumption, and scheduling [3]. Google has also expanded its IoT presence with the acquisition of Dropcam, an IoT home security device manufacturer [4].

In January of 2014, Intel announced the production of its Edison computer, a small-form Bluetooth and WiFi enabled development platform created specifically for IoT applications [5]. One year later Intel unveiled Intel Curie, an even smaller form adaptable hardware platform than the Edison, specifically targeted for Internet-connected wearable device development [6].

Companies such as Apple and Samsung have also joined the IoT movement in the hardware space, with devices such as their respective smart watches, and have also produced software development platforms, such as Apple's HomeKit [7]

and Samsung's IoT Platform [8] which aim to provide a framework for mobile-to-object and object-to-object communications and control.

In addition to the aforementioned consumer facing devices, there also exists a variety of industrial counterparts, such as GE's Industrial Internet project which intends to provide IoT solutions to industries such as aviation, healthcare, manufacturing, mining, oil and gas, and electrical power [9].

1.2 Challenges and Motivation

The practical realization of IoT requires the development of a number of new versions of platforms and technologies. Device identification, process tracking, sensing and actuation, communication, computational sensing, semantic knowledge processing, coordinated and distributed control, and behavioral, traffic, and user modeling are just a few of the processes that the IoT paradigm has enabled or will enable [10].

The realization of IoT systems is subject to numerous constraints including cost, power, energy, and lifetime. One should also consider a great diversity of IoT systems from fully organized to small individual nodes [11] [12] [13]. For example, Things such as cars, airplanes, and industrial equipment allow for much more expensive instrumentation with much high power and energy budgets in comparison to household IoT devices and those that are battery operated, wireless powered, or that even harvest energy from their surrounding environment.Therefore, although for full impact, generic algorithms and protocols that can be applied broadly to all IoT devices are required, different customized solutions for individual environments and applications are also needed.

In this dissertation we present methods for synthesizing and securing resource-

constrained IoT systems. Resource-constrained IoT applications, which are generally wireless, remote, and battery operated systems, have highly constrained design requirements due to their low energy budgets. A significant percentage of these types of IoT devices will also operate in a passive mode without batteries. Energy for these devices will either be harvested or received using a wireless medium. For these devices especially, ultralow energy solutions are required.

A second important design desideratum is security. There is a wide consensus that security will be one of the most challenging of requirements for the successful deployment of IoT. This is especially so since the potential for malicious attacks can and will be greatly spread and actuated from the Internet to the physical world. Already attacks have been carried out on devices such as the Nest thermostat [14].

IoT security encompasses several layers of abstraction and a number of dimensions. These abstraction levels range from physical layers of sensors, to computation and communication, to the semantic layer in which all collected information is interpreted and processed. From a research point of view, most novel attacks are on physical signals and, in particular, semantic attacks during data processing and decision making steps. It is also important to secure hardware since IoT presents the possibility for new physical-based attacks. It is important to observe that the lowest security at any level and at any dimension determines the overall security.

Hardware-based security is ideally suited to answer IoT security requirements. However, in order to realize the full potential of hardware-based security, very significant additional research and engineering issues have to be addressed in novel and creative ways. Hardware-based security provides a natural starting point for the realization of IoT protocols and procedures due to their very low

area and energy requirements. They are also naturally more resilient against side-channel and physical attacks. Also very importantly, is that they enable the creation of secure and trusted information flows [15]. Finally, they provide elegant and efficient solutions to several problems that classical cryptography has not been able to solve, such as secure location discovery.

IoT security desiderata can be grouped into two broad classes. The first class consists of required security tasks. As usual, the primary potential difficulties are related but contradictory requirements of different tasks. For instance, the strength of authentication and trust are in direct contradiction with a criterion of privacy. The second class of desiderata is related to design metrics such as cost, size, latency, and, in particular, energy requirements. As usual, the key impact of these requirements is that they greatly constrain acceptable security solutions. The most important security requirements include authentication and tracking, data and information integrity, mutual trust, privacy, and digital forgetting. We expect that a dominant percentage of computational sensing, decision making, communication, and activity organization will be conducted in data centers. Hence, there is a need for ensuring security in data centers as well coordinating security between data centers and distributed IoT devices [16].

Since many IoT systems will require very minimal hardware and constrained energy budgets, they will also require ultra compact security solutions with an ultra small footprint. This is also especially so since many IoT devices will operate in physically accessible, unprotected, and even potentially hostile environments.

1.3 Contributions and Organization

Table 1.1 highlights the major contributions and organization of this dissertation. We present both low energy design methods as well as new security applications. These two aspects of design are not only the premier design requirements for resource-constrained IoT systems, but they also present unique challenges due to their inherent conflicting requirements (e.g. higher security often requires higher power).

We organize our energy reduction techniques into two categories. The first consists of design and operational techniques applied on the integrated circuit. The second set of techniques are applied at a higher level of abstraction which encompasses system-level components, such as individual sensor nodes or harvesters.

We begin by presenting our circuit-level design techniques for energy reduction. Specifically, we advocate for the use of near-threshold computing (NTC). NTC is a technique that provides significant energy savings at the cost of performance degradation and an increase in sensitivity to process variation. It is very well suited for resource-constrained IoT applications both due to its ultralow energy consumption as well as due to the low processing power required by many IoT applications. However, the issues surrounding NTC—most notably, performance degradation and sensitivity to process variation—must be mitigated first in order to ensure its applicability to the IoT domain. In Chapter 2 we present techniques to lessen the impact of NTC's sensitivity to process variation through device aging; a technique that physically alters the threshold voltages of transistors. In Chapter 3 we propose techniques to mitigate the performance degradation effect of NTC through the use of adaptive body biasing; a technique that alters the threshold voltage of a group of gates post-silicon. In Chapter 4 we

continue to focus on improving the performance of NTC circuits, but reduce the scope of the relevant applications to those which employ the popular low power and fast multiple constant multiplication hardware architecture commonly used for applications such as image and video processing and radio transmission.

In Chapters 5 and 6 we present system-level design, organizational, and operational techniques for the realization of low power IoT systems. Specifically, we capitalize on the semantic nature of IoT devices in order to reduce the size and number of system components as well as develop subsampling configurations in order to reduce total energy consumption. We also present techniques for energy harvester placement and operation in order to create a self sufficient device. In both of these chapters we apply our techniques to a wearable health device for analysis.

The final chapters of this dissertation present a number of security solutions for IoT. These solutions are categorized under both circuit and system because while they are, for the most part, implemented at the circuit-level, the results are observed and organized in both a circuit-level and system-level context. The techniques we present include hardware obfuscation, which ensures intellectual property protection and trusted remote sensing, as well as techniques for trusted chip selection, which enables anyone to validate that a particular chip was fabricated at a trusted foundry.

	Circuit	System
Energy	• Ultralow power operation using NTC and device aging (ch. 2) • Mitigating the adverse delay effects of NTC using adaptive body biasing (ch. 3) • Ultralow power implementations of linear systems (ch. 4)	• Semantic-based component reduction (ch. 5) • Semantic-based subsampling configuration (ch. 5) • Spatiotemporal harvester assignment (ch. 6)
Security	• Intellectual property protection (ch. 7) • Trusted remote sensing (ch. 7) • Trusted IC selection (ch. 8)	

Table 1.1: Major contributions and organization of the dissertation.

CHAPTER 2

Energy Reduction through Coordinated Device Aging and NTC

In this chapter we present circuit-level design techniques for energy reduction in IoT systems. We apply the near-threshold computing paradigm in order to reduce energy consumption dramatically. NTC is a technique applied to integrated circuits in which the supply voltage is set to approximately the threshold voltage of the transistors. Since switching power of a transistor is proportional to the square of the supply voltage, any reduction in supply voltage produces quadratic gains in energy savings. Specifically, NTC has been shown to achieve energy savings comparable to sub-threshold operation while maintaining more favorable performance characteristics [17]. Many IoT systems which interact with the physical world are ideal for NTC because speed is rarely of great benefit while low energy is absolutely paramount. Many systems, for example, do not require high powered processors for data computation, but instead rely on offloading heavily intensive computational tasks to data centers.

The main obstacle to effectively apply NTC is process variation. A single gate in an IC with a low V_{th} forces operation of the circuit at a low V_{gnd} and a single gate with a high V_{th} forces operation of the circuit at a relatively high V_{dd}. Since the switching energy of the gates in a circuit is a function of $(V_{dd} - V_{gnd})^2$, the tails on the low ends and high ends of the threshold voltage distribution are

ultimately responsible for superfluous energy consumption.

In order to reduce this effect, we propose the use of device aging to effectively shift the V_{th} values of the lower gates to higher values [18]. While this slows individual gates, we benefit from an overall energy reduction over the entire circuit by enabling the increase of V_{gnd}. Furthermore, the majority of gates are often not on the critical path, so aging often has no impact on circuit delay. However, if it is crucial that a gate on the critical path is aged (thus increasing circuit delay), we subsequently increase V_{dd} to return the circuit to its original delay, before returning to aging minimum V_{th} gates.

Furthermore, we recognize that application specific integrated circuit (ASIC) solutions are often not always the most cost efficient means in which to realize subsystem designs for IoT applications. Field-programmable gate arrays (FPGAs), on the other hand, offer much more flexibility and do not require large upfront capital costs and are popular IoT prototyping platforms. Hence, we propose new aging and placement techniques for FPGA energy minimization that capitalizes on the FPGA block architecture.

2.1 Preliminaries

2.1.1 Process Variation

Gate delays and effective channel lengths in nanoscale technologies are subject to significant process variation [19] [20]. A variety of manufacturing faults emerge as a result, including but not limited to variations in doping concentrations, imperfect mask alignment, and molecular chemical and physical phenomena. These forms of process variation manifest as the deviation of IC characteristics from nominal values. For example, variations in doping concentrations and line edge

roughness alter transistor threshold voltages and effective channel lengths. These manifestations have been thoroughly studied, categorized, and modeled, yet continue to be of paramount concern [21] [22].

2.1.2 Near-Threshold Computing

In modern circuitry, energy consumption and delay is often managed by altering the supply voltage relative to ground. Since the relationship between energy and voltage is quadratic, voltage scaling has become one of the most effective and researched methods for integrated circuit power reduction [23].

Energy consumption is comprised of two components, switching energy and leakage energy, each of which are functions of the supply voltage, V_{dd}. These energy components are also dependent on other physical characteristics of each transistor, expressed through Equations 2.1 and 2.2 with respective parameters and constants defined by Dreslinksi et al. [17]. What most distinguishes these two components from one another is that switching is a function of V_{dd} and V_{gnd} while leakage energy is a function of V_{dd} and V_{th}.

$$P_{switching} = \alpha \cdot C_{ox} \cdot W \cdot L \cdot (V_{dd} - V_{gnd})^2 \tag{2.1}$$

$$P_{leakage} = 2 \cdot n \cdot \mu \cdot C_{ox} \cdot \frac{W}{L} \cdot \phi_t^2 \cdot D \cdot V_{dd} \cdot e^{\frac{\sigma \cdot V_{dd} - V_{th}}{n \cdot \phi_t}} \tag{2.2}$$

Today, the vast majority of circuits operate in the super-threshold region where $V_{dd} \gg V_{th}$. Chandrakasan's ultradynamic voltage scaling techniques operate in this region [24]. Here, switching is the dominant component of energy consumption. Traditional techniques for energy reduction apply voltage scaling at the cost of increased delay. In order to increase the speed of the circuit, V_{dd}

is raised, thus increasing the gap between V_{dd} and V_{th}, and reducing the delay of the device components as defined by equation 2.3.

$$Delay = \frac{k_{tp} \cdot k_{fit} \cdot L_{eff}^2}{2 \cdot n \cdot \mu \cdot \phi_t^2} \cdot \frac{V_{dd}}{(ln(e^{\frac{(1+\sigma)V_{dd}-V_{th}}{2 \cdot n \cdot \phi_t}} + 1))^2} \cdot \frac{\gamma_i \cdot W_i + W_{i+1}}{W_i} \tag{2.3}$$

Techniques for sub-threshold operation of circuits in which $V_{dd} < V_{th}$ have been proposed [25] [26]. Operation in this region returns substantially higher energy savings than operation in the super-threshold region. However, as V_{dd} extends below V_{th}, marginal energy savings reduces by an order of magnitude while performance penalties increase substantially (50-100×) [17]. Leakage energy also becomes a dominant component of overall energy usage. Thus, sub-threshold operation has become useful for niche applications due to its degraded performance metrics [27].

The most balanced tradeoffs between energy reduction and performance degradation are found in the near-threshold region where $V_{dd} \sim V_{th}$ [17]. Here, energy savings are 10× that of super-threshold operation while performance degradation is similar to sub-threshold operation.

Numerous design challenges have emerged as a result of NTC [28]. Most notably, NTC circuits are highly susceptible to variability due to their near-threshold operation. Inherent variations in manufacturing processes affect threshold voltage distributions, thus imposing design constraints at near-threshold. Take for example Figure 2.1. In the near-threshold case the supply voltage must be carefully set so that it does not accidentally disable any gate on the right tail of the curve. Furthermore, potential voltage fluctuations during operation must also be accounted for. In the super-threshold case, process variation has

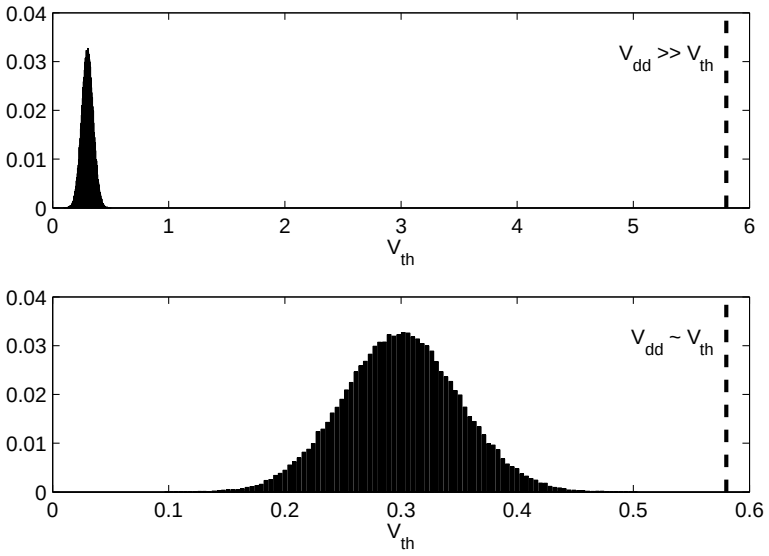


Figure 2.1: Example distribution of threshold voltages due to process variation. In the top figure the supply voltage is set in the super-threshold region where process variation has very little impact on operation. In the bottom figure process variation must be taken into account when setting the supply voltage in the near-threshold region.

little to no impact on supply voltage setting and handling of fluctuations. Post-silicon techniques have been proposed to mitigate the effects of process variation on NTC through body-biasing, soft-edge clocking, and employing dual supply voltages [28] [29], while device aging techniques have not yet been proposed.

2.1.3 Device Aging

Negative bias temperature instability and hot-carrier injection phenomena can cause significant alterations to the delay characteristics of individual gates [18] [30] [31] [32]. Systematic and constant stressing of gates can irreversibly alter the physical characteristics of the gates, resulting in an increase in their threshold voltages. While device aging is widely considered to be a detrimental effect in the lifetime and operation of general circuitry, we utilize it to our advantage

in NTC-operated circuits for overall energy reduction. This is possible because switching energy is a function of the difference between V_{dd} and V_{gnd}. By aging individual gates we can alter this distribution post-silicon in such a way that enables a reduction in switching energy.

2.1.4 Gate Level Characterization

Gate level characterization (GLC) techniques can be classified into four major groups: direct measurement approaches, schemes that employ FPGA reconfiguration, approaches that create and observe special IC structures and specialized circuity, and non-destructive techniques that construct global measurements and deduce scaling factors of each gate by solving a system of equations [33] [34] [35].

Direct measurement techniques use atomic force microscopes, electric line measurements, and optical instruments to directly measure critical dimensions (e.g. effective channel length) [33]. They are very accurate and have a wide range of speeds, however their application is often restricted to just dimensions measurements.

FPGA GLC techniques include the iterative creation of clock measurement circuitry that isolate individual blocks [34]. Other techniques include populating chips with structures such as ring oscillators that can be easily characterized through clock sweeping and counting methods [36]. The limitations of these techniques is that they can only be applied to specific types of designs.

Non-destructive GLC techniques include those that do not use any spatial correlation assumptions [35], and those that do. While these techniques are universally applicable with zero overhead and low cost, up until now they were not scalable and not able to characterize significant percentage of gates.

In general, GLC techniques are expensive. Hence, while we present methods

for coordinated device aging and NTC operation on ASIC circuits, which require GLC, in this chapter we focus primarily on methods derived specifically for FPGA platforms, which do not require any expensive GLC techniques.

2.2 Motivation and Problem Formulation

In this section we present the first approach for energy and power minimization in FGPA-based systems using device aging. We begin by explaining the motivation and our assumptions, and follow up with a presentation of our problem formulation. We explain the key optimization ideas in the new approach and present a description of our overall approach. And finally, we present a simulation-based study that indicates that energy minimization in FPGAs can be very effective.

There are a number of reasons why studying energy minimization in FPGA-based sensing systems is important. The first is that ASIC-based systems very often have unacceptable preproduction costs that can reach up to tens of millions of dollars. This means that even before the first ASIC-based sensing system is realized, one has to invest around $30 million dollars just to cover design, testing, and validation.

An alternative to ASIC-based sensing systems is to use more general purpose microprocessors or microcontrollers. However, a serious downfall of these systems is that optimizing general purpose processors for time and energy is often complex and cumbersome.

FPGAs do not impose any non-production costs on the designer and have an unprecedented level of flexibility. Recent developments have put FPGAs at the frontier of integrated systems in terms of both their feature size as well as their potential for integration. For example, FPGA Spartan chips have more than 5.8

million transistors while only Nvidia's graphics processor has a larger number of switching elements. Recently, Xilinx announced that they implemented their newest family of FPGA's at TSMC facilities using 20nm technology. This announcement means that FPGAs now use more advanced and finer technologies than both ASICs and even microprocessors.

Our objective is to develop an approach that minimizes energy spent in FPGA-based subsystems. We employ two generic mechanisms. The first is to use device-based aging for the alternation of FPGA component speeds. Specifically, we consider slices and blocks as basic structures in FPGAs and simultaneously age all transistors in a particular slice. The key observation here is that the FPGA will operate in NTC mode and therefore have a very low energy budget.

The second degree of freedom is that in almost all situations, the size of the actual design is significantly smaller than the capacity of the full FPGA. Therefore, we can move each program component by the same horizontal and vertical shift, thus moving the boundaries of the program design to a number of locations. In such a way, we can place the program in a position to avoid using slices that have transistors with threshold voltages that are not amenable for energy efficiency. Specifically, our goal is to select slices in such a way that the highest threshold voltage in any of them is as low as possible while the lowest is as high as possible. In this type of system, we reduce the voltage swing and hence reduce energy quadratically. These two mechanisms, device aging and selection of slices that only have favorable threshold voltages, are combined in such a way that achieves provably optimum minimum energy.

The key observation is that all slices can be normalized by setting them such that the highest threshold voltages among all of them are equal. This is accomplished using non-uniform device aging as dictated by process variation.

An additional and important observation is that transistor-level characterization of slices can be accomplished without using any already proposed and effective but often expensive gate level characterization techniques. We employ a new binary search technique that distinguishes between correctly operating slices and incorrectly operating slices due to either high threshold voltages that prevent low supply voltage or low threshold voltages that prevent high ground voltage. Once we characterize each slice, we search for slices that are adjacent to one another in order not to jeopardize the critical path of the design while excluding slices which have gates with too low of a threshold voltage in their structure.

Our process variation model has two components. In the first we use Asenov's model indicating that threshold voltages are proportional to the number of dopants, and therefore, follow an independent Gaussian distribution [18]. The second component corresponds to the effective channel length of transistors and follows Cline's model that includes information about correlations that are typical for modern integrated circuits [19].

2.3 Energy Reduction

In this section we describe in detail our techniques for energy minimization employing near-threshold computing and utilizing device aging for both ASIC and FPGA systems.

A typical post-fabrication distribution of threshold voltages for the s38584 benchmark circuit is depicted in Iteration 0 of Figure 2.2. Ultimately, the left and right tails of the distribution are most responsible for forcing the relatively high and low setting of V_{dd} and V_{gnd}, respectively. In order to minimize this voltage swing and thus reduce circuit switching energy we aim to increase the

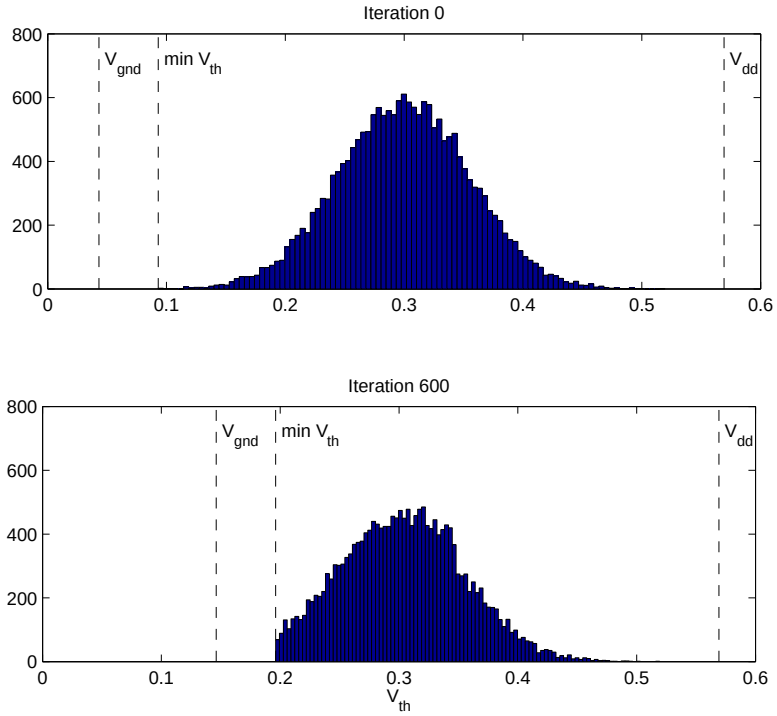


Figure 2.2: Gate V_{th} distributions of s38584 from the ISCAS'89 benchmark suite [37]. V_{dd} remains at a near-threshold voltage just above the highest V_{th} gate while V_{gnd} is set to a near-threshold voltage just below the lowest V_{th} gate. After aging the lowest 600 gates we can increase V_{gnd} and reduce overall circuit switching energy.

threshold voltages of the gates that comprise the lower tail, thus enabling a higher setting of ground. This is achieved by iteratively aging gates corresponding to the minimum threshold voltage across the circuit to that gate's maximum extent. In the case that a gate on the critical path is selected and subsequently aged, thus increasing the overall circuit delay, we adjust the supply voltage until we return to the original delay before returning to aging.

Figure 2.3 depicts the energy savings relative to normal super-threshold operation. NTC alone provides 45-60% energy savings while our techniques reduce energy consumption by an additional 15%.

In order to apply our aging-based energy minimization methods to the FPGA

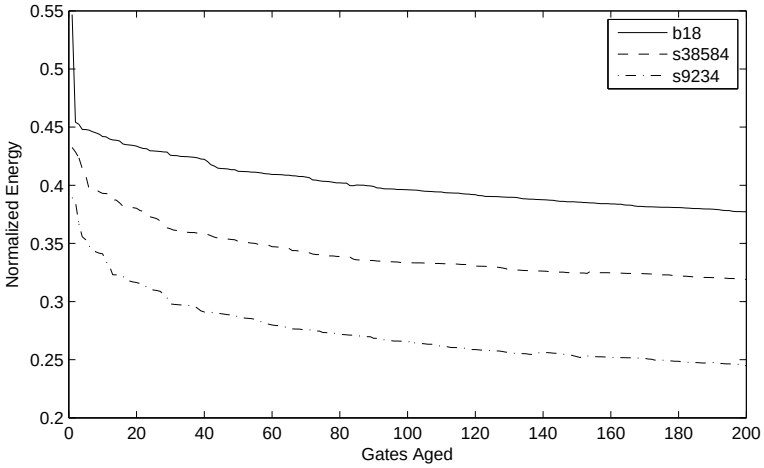


Figure 2.3: ASIC energy reduction applied using iterative minimum V_{th} aging on circuits from the ISCAS'85 and ISCAS'89 benchmark suites [37] [38].

we must first characterize the maximum and minimum threshold voltages of each slice. Existing GLC solutions are expensive and overly detailed for our purposes. Thus, we introduce a new and inexpensive binary search technique for FPGA slice characterization. For the purposes of energy minimization of the FPGA using NTC, we need only to know the minimum and maximum threshold voltages within a single slice. We measure these values by varying the supply and ground voltages until the slice is operational.This technique is described in detail in Algorithm 1.

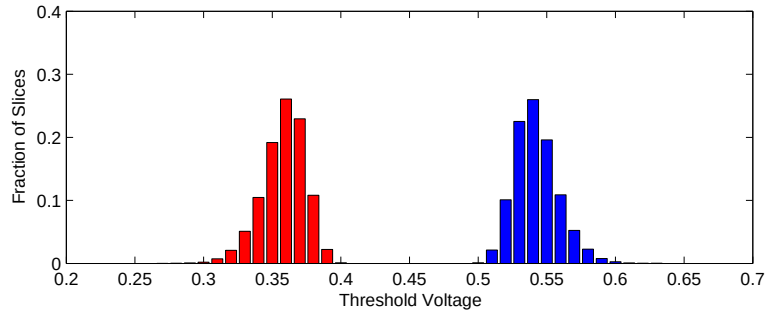
Herein, we denote the maximum and minimum threshold voltages at a particular slice i as $V_{th}^{max}(i)$ and $V_{th}^{min}(i)$. Once all FPGA slices are characterized we perform coarse grained per slice aging by stressing all gates in a particular slice uniformly. This is performed on each slice individually in order to match $V_{th}^{max}(i)$ to the FPGA-wide maximum threshold voltage. This will simultaneously increase the values of V_{th}^{min} across all slices non-uniformly according to each slice's particular V_{th}^{max}. This change is depicted in Figure 2.4.

Now, in order to minimize energy, we find a subset of physically adjacent tiles (as required by the layout of the program) whose minimum V_{th}^{min} among all covered slices is largest among all possible placements, thus reducing the voltage swing and quadratically reducing energy consumption. This is possible because most programs use a fraction of the total FPGA space. We iteratively disable slices containing the minimum V_{th}^{min} over all slices, until it is no longer possible to position the program on the FPGA board without covering a disabled slice. The last disabled slice that prevented the program from being positioned is then

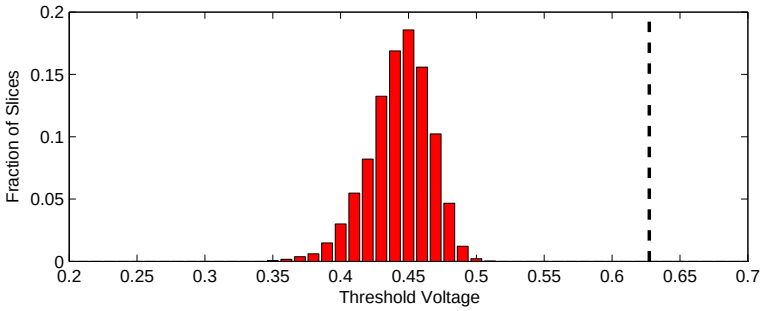
Algorithm 1 FPGA Slice Characterization

Require: $0 < \epsilon < V_{gnd} \leq V_{dd}$

$\delta \leftarrow +1$
$\Delta \leftarrow V_{dd}$
while $\epsilon < \Delta$ **or** $!circuit.IsOperational(V_{dd}, V_{gnd})$ **do**
  $V_{dd} \leftarrow V_{dd} + \delta\Delta$
  if $\delta = +1$ **and** $circuit.IsOperational(V_{dd}, V_{gnd})$ **then**
    $\delta \leftarrow -1$
    $\Delta \leftarrow \Delta/2$
  else if $\delta = -1$ **and** $!circuit.IsOperational(V_{dd}, V_{gnd})$ **then**
    $\delta \leftarrow +1$
    $\Delta \leftarrow \Delta/2$
  end if
end while
$\delta \leftarrow +1$
$\Delta \leftarrow V_{gnd}$
while $\epsilon < \Delta$ **or** $!circuit.IsOperational(V_{dd}, V_{gnd})$ **do**
  $V_{gnd} \leftarrow V_{gnd} + \delta\Delta$
  if $\delta = +1$ **and** $!circuit.IsOperational(V_{dd}, V_{gnd})$ **then**
    $\delta \leftarrow -1$
    $\Delta \leftarrow \Delta/2$
  else if $\delta = -1$ **and** $circuit.IsOperational(V_{dd}, V_{gnd})$ **then**
    $\delta \leftarrow +1$
    $\Delta \leftarrow \Delta/2$
  end if
end while
return V_{dd}, V_{gnd}



(a)



(b)

Figure 2.4: (a) Distributions of maximum and minimum threshold voltages found in slices of a typical FPGA board. (b) Distribution of minimum threshold voltages across FPGA slices after aging. The dotted line represents the maximum threshold voltage of each slice.

re-enabled. We subsequently iterate over all possible valid program assignments. The final assignment corresponds to the position in which the slices covered by the program at that location has a minimum V_{th}^{min} that is largest among all possible positions, thus yielding the smallest operational voltage swing between V_{dd} and V_{gnd} and providing the lowest energy utilization.

We compare the energy savings of our techniques for aging and placement against an NTC-operated FPGA with and without aging. We present our results in Figure 2.5. Our technique achieves energy savings of up to 35% against the aged FPGA and 75% to 84% against the traditionally operated non-aged FPGA.

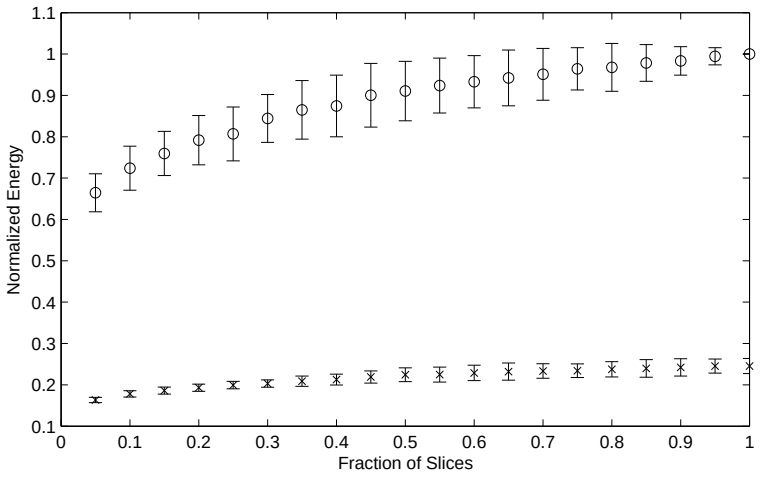


Figure 2.5: FPGA energy consumption of varying sized programs as compared to (o) the case in which the FPGA slices are aged and operated at the maximum V_{th}^{max} and the minimum V_{th}^{min} over all slices; and (x) the case in which the FPGA is operated without aging at the maximum V_{th}^{max} and minimum V_{th}^{min} over all slices. Error bars correspond to the standard deviation over all simulated FPGA instances.

2.4 Summary

Our new approach for the creation of energy efficient subsystems employs the near-threshold computing paradigm on both ASIC—where aging is conducted at the gate-level granularity—as well as on FPGAs—where aging is applied at the slice-level granularity. We have achieved energy reduction of up to 2.9× over popular ASIC benchmark circuits and up to 6× over traditional operation of FPGAs.

CHAPTER 3

Adaptive Body Biasing for Reclaiming Speed in NTC

For some systems, such as a majority of physical remote sensor networks, the increase in delay imposed by near-threshold operation is of no consequence since these systems often rely on sampling frequencies on the order of megahertz which is easily attainable, even using NTC, for current node technologies today. However, for a great majority of applications, NTC is difficult to adopt due to the huge increase in delay it imposes on the application.

In this chapter, we mitigate the effects of NTC on the overall delay of a circuit and reclaim some of that lost speed. This is accomplished while maintaining that the energy savings originally gained through NTC operation are not lost. Our techniques utilize inherent differences in process variations coupled with adaptive body biasing (ABB) in order to speed up the critical path gates, thus reducing overall circuit delay, while not affecting overall switching energy. ABB is a post-silicon technique that enables the temporary tuning of physical characteristics of a selection of gates. Essentially, the effective threshold voltage, V_{th}, of a biased gate is shifted according to the applied bias voltage. Note that while the effective threshold of these gates can be altered post-silicon, the assignment of gates to a particular ABB group must be done pre-silicon.

Our approach assumes that both the supply voltage and ground voltage are

operated at near-threshold. We select those gates that are most commonly on the critical path during NTC operation and bias them in order to reduce their delay and the overall delay of the circuit.

Furthermore, selection of critical path gates is a non-trivial task since process variation has a chaotic effect on critical path assignment when operating at near-threshold voltages. We propose both a static solution and an iterative solution for critical path gate selection pre-silicon and compare their results. Our approach is the first of our knowledge to incorporate ABB for circuit speedup in NTC.

3.1 Preliminaries

3.1.1 Near-Threshold Computing

Energy consumption in modern CMOS circuits can be minimized quadratically by lowering supply voltage relative to ground. As a consequence, voltage scaling techniques have become one of the most effective methods and widely researched topics in IC power reduction [23].

Operating in the near-threshold region ($V_{dd} \sim V_{th}$) has been demonstrated to offer balanced trade-offs between energy reduction (10×) and performance degradation (10×) as compared to super-threshold ($V_{dd} > V_{th}$) and sub-threshold ($V_{dd} < V_{th}$) operation [17]. While NTC offers desirable levels of energy efficiency, it has also introduced numerous design challenges [28] [39]. Most notably, due to near-threshold operation and the variability of circuit threshold voltages (due to process variations), NTC circuits are highly susceptible to variability.

3.1.2 Process Variation

As we enter deeper into submicron technologies and transistor sizes continue to decrease, process variation begins to prevail as a prominent design and manufacturing constraint [19] [20]. In the case of NTC it imposes an unfavorable design space since its aim is to assign a supply voltage very near to the threshold voltage of the circuit's transistors. However, process variations manifest as a distribution of threshold voltages, thus forcing V_{dd} to be set above the highest V_{th} of the circuit and, similarly, forcing V_{gnd} to be set below the lowest V_{th}.

While V_{dd} and V_{gnd} can still be set in such a way as to achieve balanced energy and delay tradeoffs, they are forced apart by the tails of the threshold voltage distribution. This means that just a small fraction of gates in the entire circuit (the gates corresponding to the tails of the distribution) are the culprits forcing a relatively high and relatively low supply voltage and ground, respectively. For reference, in a Gaussian distribution, 4% of gate thresholds impose a 2σ separation in voltage.

3.1.3 Adaptive Body Biasing

ABB enables the control of the effective threshold voltages of transistors post-silicon by applying a body biasing voltage to a select group [40] [41] [42]. By controlling the threshold voltages of various groups of transistors, one can effectively coordinate each group such that the distance of the near-threshold supply and ground voltages can be minimized, thus reducing total switching energy of the circuit. On the other hand, one can also use ABB effectively on say, those gates that are most often on the critical path, in order to increase the speed of the circuit.

In general, due to the additional logic and interconnect necessary to enable practical and manageable ABB, the number of groups of transistors that can be addressed and individually shifted using ABB should be kept minimal, contain a small number of gates, and be physically located within proximity of one another. Our methodology chooses a small number of gates by only selecting those most commonly on the critical path. Furthermore, since these gates make up the common critical path, their geographic proximity to one another is naturally very close.

3.1.4 Power and Delay Modeling

We derive our models for power and delay from Markovic et al. [43]. Equation 3.1 represents our gate-level switching energy model. As can be seen from the equation, switching energy is quadratically dependent on the difference between V_{dd} and V_{gnd}. Thus, reducing or increasing the gap between these two voltages can have a significant effect on the switching energy of the circuit. While Equation 3.1 is not directly dependent on V_{th} there does exist an indirect relationship. Since we are operating our circuits in the near-threshold region, were $V_{dd} \sim V_{th}$, the minimum and maximum values of V_{dd} and V_{gnd}, respectively, are determined by the edge of the tails of the V_{th} distribution.

Equation 3.2 represents our gate-level delay model. Altering the threshold voltages through ABB will have a non-linear affect on the individual delays of each affected gate and will almost certainly alter the critical path. We find that in NTC operated circuits especially, the critical path is very sensitive to differences in threshold voltages between gates. We discuss this notion further in subsequent sections of this chapter and take it into consideration in our techniques.

$$P_{switching} = \alpha \cdot C_{ox} \cdot W \cdot L \cdot (V_{dd} - V_{gnd})^2 \tag{3.1}$$

$$Delay = \frac{k_{tp} \cdot k_{fit} \cdot L_{eff}^2}{2 \cdot n \cdot \mu \cdot \phi_t^2} \cdot \frac{V_{dd}}{(ln(e^{\frac{(1+\sigma)V_{dd}-V_{th}}{2 \cdot n \cdot \phi_t}} + 1))^2} \cdot \frac{\gamma_i \cdot W_i + W_{i+1}}{W_i} \tag{3.2}$$

3.2 Methodology and Techniques

We use ABB to mitigate the negative effects of NTC on the delay of a circuit. This must be done delicately so as to to retain the energy savings gained through near-threshold operation.

A typical distribution of threshold voltages across the gates of a typical circuit can be modeled after a Gaussian. Due to process variations the physical characteristics of each gate deviate from nominal values, thus resulting in a distribution of threshold voltages. For maximal energy savings, the supply and ground voltages are set such that they are just above and just below the highest and lowest threshold voltages in the distribution with some buffer, respectively.

In typical NTC operation, the energy usage of the system is derived primarily from the switching energy expended by each gate; a function of per gate switching frequency and the switching voltage, i.e., the gap between V_{dd} and V_{gnd} in Equation 3.1.

Unfortunately, while NTC drastically reduces overall circuit energy consumption it simultaneously degrades circuit performance relative to super-threshold operation. In general, in order to recover this lost performance, we need not reduce the delay of every gate in the circuit, but rather only those gates which are on the critical path. If it is possible to single out those critical path gates and

group them for biasing, then the threshold voltages of those gates can be lowered simultaneously, thus reducing the delay of the critical path.

However, as the number of gates in the critical path grows, the marginal speed improvement diminishes since the critical path grouped gates are in the same V_{th} distribution as the rest of the gates in the circuit. Thus, their variance reduces the amount of bias that can be applied to the group since its magnitude depends on the lowest V_{th} of the critical path gates (i.e., it cannot be biased lower than the lowest V_{th} in the remainder of the distribution without increasing overall switching energy). Furthermore, the speed of the slowest gate on the critical path will be determined by the gate with the highest V_{th}.

In order to mitigate these adverse variational properties, we exploit various characteristics of process variation. In particular, decreasing the doping concentrations of a particular set of transistors reduces the variance of their threshold voltages [44]. Thus, if we dope those gates on the critical path at lower concentrations, we can reduce their variability and separate these gates from the greater distribution more obviously.

An example of this phenomenon is depicted in Figure 3.1a. The gates most commonly found on the critical path are doped at lower concentrations, thus reducing their nominal threshold voltages below even the lowest of the remaining gates along with their variance. In Figure 3.1b we apply ABB to this set of selected critical path gates in order to increase their threshold voltage to reside just within the larger distribution of remaining gates. This achieves similar energy consumption rates to NTC while decreasing overall circuit delay through the reduction of the critical path gate threshold voltages.

These solutions are optimal provided that the critical path gates are known pre-silicon. Unfortunately, determining the critical path gates pre-silicon is non-

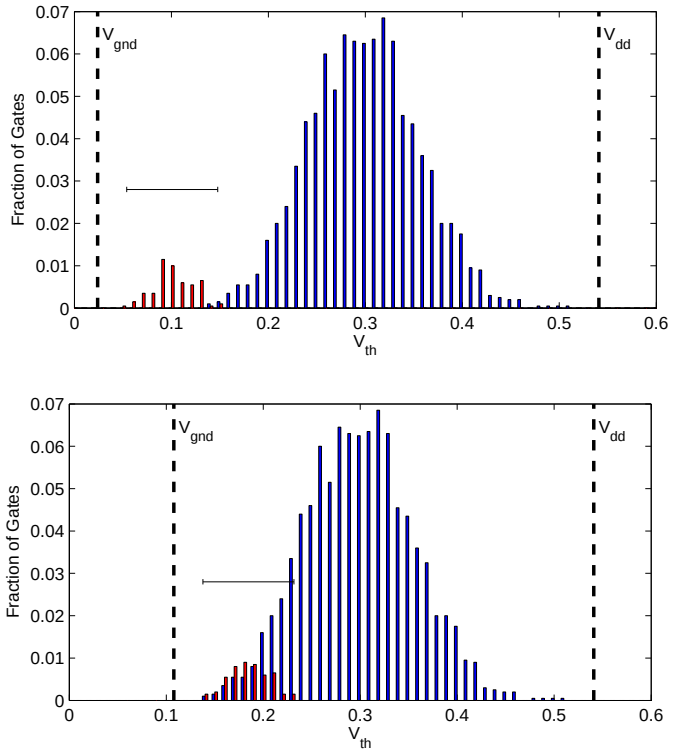


Figure 3.1: The red distribution represents gates on the critical path and the blue distribution represents the remaining gates. In this scenario we purposely dope the critical path gates at lower concentrations, thus reducing their nominal threshold voltage values and variance. In order to achieve the same energy savings as NTC, the critical path gates are biased and shifted up from (a) to (b).

trivial in NTC operated circuits and near impossible due to inherent random threshold voltage variation. Due to process variations and the near-threshold nature of the supply voltage, the set of critical path gates can vary wildly from one circuit instance to the next. Figure 3.2 depicts the probability that a given circuit is on the critical path for one thousand randomly generated instances of each benchmark circuit. The b17 circuit contains about 50 gates that have about 85% probability of being on the critical path when operated at NTC. However, the remainder of the benchmark circuits have a majority of gates (of those that are on the critical path at least once) with much lower probabilities of being on

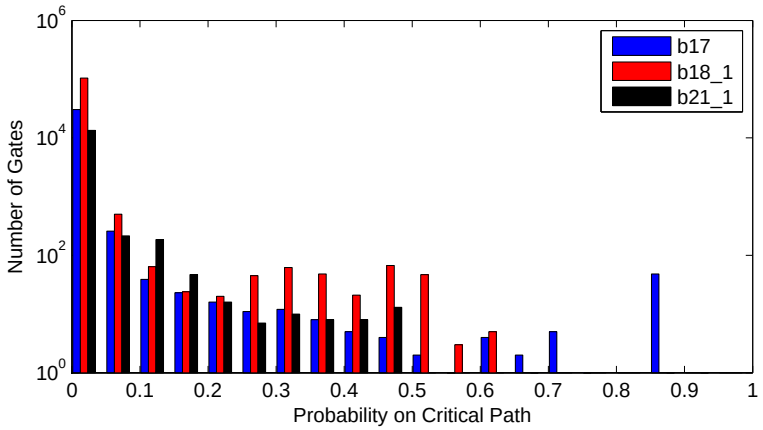


Figure 3.2: Log-distribution plot of the number of gates in each benchmark that have a probability of being on the critical path when the circuit is operated at near-threshold.

the critical path.

3.3 ABB Group Selection

Herein our solutions implement the doping methodology previously discussed. Predicted *critical path gates* are doped during fabrication to reduce their threshold voltage variance. Post-silicon we bias these gates using ABB to match the lowest V_{th} of the critical path group distribution to the lowest V_{th} of the remaining gates distribution. This ultimately maintains energy savings while increasing overall speed relative to traditional near-threshold operation. We perform our methodologies on the circuits from the ITC99 benchmark suite [45].

3.3.1 ABB Group Selection: Single Pass

Due to the variable probabilities of critical path gate groupings—a symptom of process variation and near-threshold operation—we investigate the different delay improvements for different combinations of potential critical path gates.

We first compute individual gate probabilities of appearing on the critical path for different circuit instances as depicted in Figure 3.2. For the purposes of clarity only three circuits are shown, however, we implement this procedure on the majority of the ITC99 benchmarks.

Next, we select gates to append to the *critical path* group for adaptive body biasing. We first limit membership to those gates with the highest probability of appearing on the critical path. For circuit b17, for example, this corresponds to about 50 gates that have a probability of 85% or above of belonging on the critical path. In Figure 3.3 this corresponds to a threshold of 0.85 along the x-axis. We then apply our biasing technique to the chosen gates and compute the average energy and delay over one thousand instances of the new design.

We repeat this procedure over lower thresholds for the ten documented ITC99 benchmark circuits. As can be seen in the three circuits depicted in Figure 3.3, energy usage remains constant with NTC operation while delay is reduced. For all circuits, we see the most improvement in delay occur when all potential critical path gates (gates with probability greater than 0) are appended to the ABB group. Energy savings are very near to the base NTC case while our delay improvements range from an improvement factor of 1.19 to 1.36 over NTC operation.

3.3.2 ABB Group Selection: Iterative Refinement

While our single pass ABB group selection methodology successfully reduces delay while maintaining NTC energy savings, it is limited because it only checks the probability that each gate will be on the critical path given that the circuit is operated in near-threshold without the effects of ABB. Once these gates are placed in the ABB group and biased such that energy remains at NTC levels

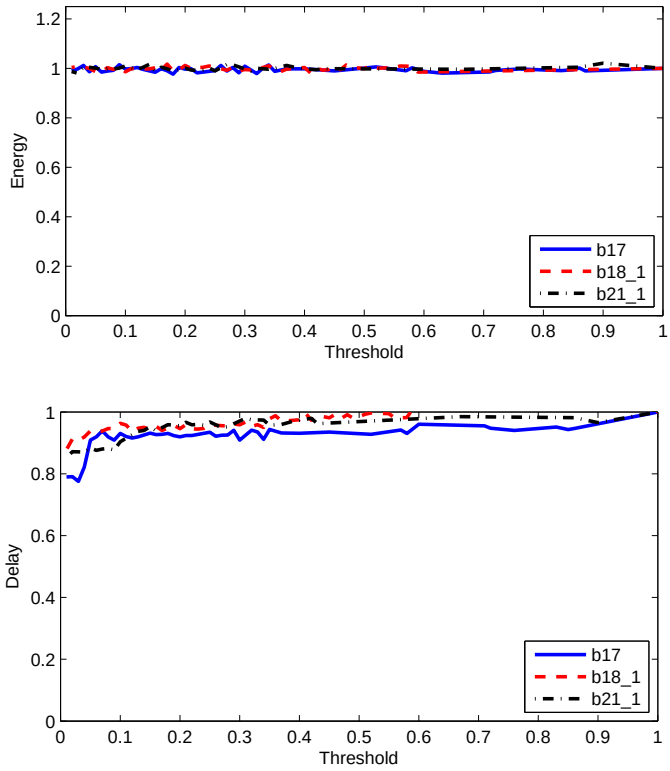


Figure 3.3: Results from single pass ABB group selection. The energy and delay are averaged values over one thousand instances. The threshold corresponds to the probability of being on the critical path in the generic NTC scenario.

and delay is reduced, the dynamics of the critical path change. This is especially true since the circuit is operated at near-threshold voltages, and fluctuations in critical path gate groupings from non-biased circuits can vary significantly (recall Figure 3.2).

Ultimately, we must consider how the dynamics of the critical path change once a particular set of gates is selected and biased. Thus, we propose an iterative refinement technique which simulates one thousand instances of a circuit, then chooses k gates with the highest probability of being on the critical path, adds them to the ABB group, then recomputes and reapplies the bias to the new group, and repeats.

The results of these iterations are depicted in Figure 3.4, where $k = 50$. Note that energy remains constant across iterations, and even decreases in some instances as the critical path gate grouping grows and ultimately begins to reduce the spread of the larger V_{th} distribution slightly. What is most impressive is the delay improvement that happens rapidly in the first iterations. Delay factor improvements of 1.6 are already achieved with ABB groupings of one thousand gates, easily surpassing the results of our single pass methodology. Table 3.1 lists the results of our iterative ABB grouping approach along with a comparison against our single pass technique.

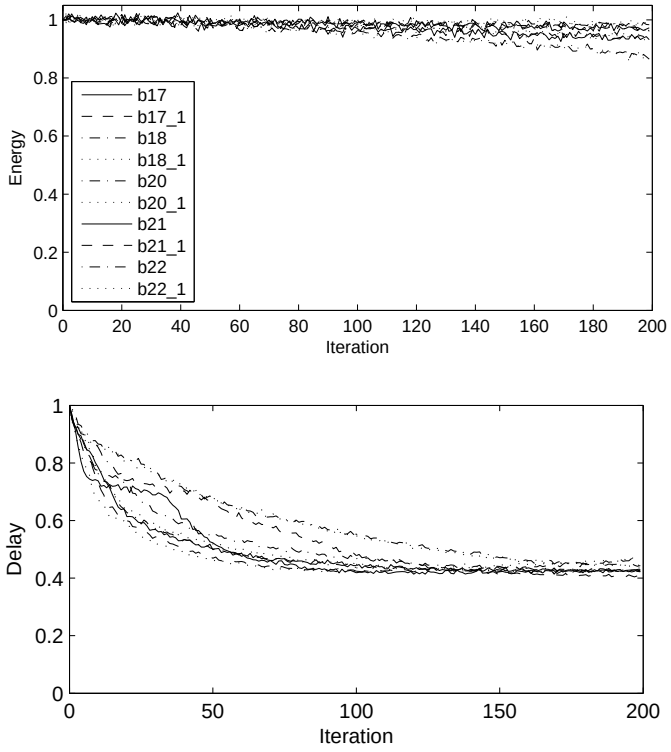


Figure 3.4: Results from iterative refinement ABB group selection. At each iteration we create one thousand instances of the individual circuit and append the k most probable critical path gate candidates from those instances to the ABB group.

Circuit			Energy Factor Improvement		Delay Factor Improvement			
Name	Number of Gates	Number of Gates in ABB Group	NTC vs. Normal	NTC+ABB vs. Normal	NTC vs. Normal	NTC+ABB vs. Normal	NTC+ABB vs. NTC	Iterative vs. Single Pass
b21_1	12,248	1,650	5.04	5.10	0.182	0.343	1.89	1.54
b20_1	12,264	1,300	5.04	5.06	0.188	0.349	1.85	1.36
b20	17,158	2,250	4.89	4.94	0.188	0.364	1.93	1.53
b21	17,482	2,300	4.85	4.86	0.189	0.372	1.97	1.50
b22_1	18,461	2,100	4.85	4.90	0.187	0.345	1.84	1.43
b22	25,460	2,600	4.73	4.71	0.189	0.346	1.83	1.51
b17	27,852	1,050	4.69	4.61	0.202	0.290	1.43	1.13
b17_1	32,971	1,400	4.59	4.60	0.194	0.267	1.38	1.12
b18_1	88,954	3,200	4.19	4.20	0.218	0.351	1.61	1.35
b18	94,249	3,100	4.15	4.15	0.221	0.348	1.58	1.33

Table 3.1: Summary of results from applying iterative refinement ABB group selection. *Normal* refers to traditional super-threshold operation. *NTC* refers to basic near-threshold operation without modification. *NTC+ABB* refers to our iterative refinement ABB group selection technique. Energy factor between NTC and NTC+ABB are near identical. We see a dramatic and expected performance degradation from Normal operation near 5×. However, we see a factor of improvement in delay between 1.38 and 1.97 when utilizing our NTC+ABB technique over NTC.

3.4 Summary

NTC has enabled tremendous improvements in energy reduction of digital circuits. However, these improvements come at a significant cost to the delay of the circuit. This has limited the overall usefulness and applicability of NTC to a majority of applications.

Thus, we have presented new algorithms for adaptive body biasing of near-threshold operated circuits in order to retain comparable NTC energy savings while reclaiming lost speed. We exploit inherent process variations in manufacturing in order to reduce variability in threshold voltage variations and couple this with our techniques for iterative critical path gate assembly. We apply our algorithms to the circuits in the ITC'99 benchmark suite which result in a factor improvement in delay ranging from 1.38 to 1.97.

CHAPTER 4

Ultralow Power Implementations of Linear Systems

Linear transformations are common and essential algorithms employed in many popular applications and often implemented on a variety of wireless devices such as smart phones, cameras, and sensor nodes. For example, the discrete cosine transform (DCT) is used in many image and video encoders and decoders such as JPEG and MPEG standards. Orthogonal frequency-division multiplexing (OFDM) is one of the most popular techniques for broadband digital communication due to its effective use of the fast Fourier transform (FFT) that greatly reduces hardware and energy requirements.

It is well known that many linear transforms and filters, including FFT, DCT, FIR and IIR filters, can be implemented using multiple constant multiplication (MCM). MCM enables low power and low latency parallel multiplications of several hard-coded constants with a single input variable using shifts, additions, and subtractions. Currently, a large body of literature on MCM focuses on minimizing their number of operations, implementation size, and depth. However, as of now, there has been relatively no effort to synthesize ultralow energy implementations, in particular, through the use of near-threshold computing. MCM implemented using near-threshold technology additionally greatly reduces energy requirements.

While NTC provides as much as 10× energy efficiency gains over traditional super-threshold operation, it also increases delay by the same factor. A key obstacle in designing circuits for NTC operation is process variation, which manifests as deviations in gate characteristics, such as delay and energy, from their nominal values. These deviations become more pronounced when operating at near-threshold. For example, small fluctuations in threshold voltages have a much higher impact on delay in near-threshold than in super-threshold. Output capacitive load also becomes of concern at near-threshold due to its ability to compound NTC performance inefficiencies.

In order to synthesize MCM in NTC systems, we propose several techniques which exploit several degrees of freedom. Among them, common sub-expression exploration is combined with simultaneous speed and energy organization through our key contribution: operation replication for load reduction to improve latency. We also propose for the creation of deep combinational logic which not only eliminates energy expensive read and write operations to storage elements, such as flip-flops, but also reduces the impact of process variation and improves yield.

4.1 Preliminaries

4.1.1 Near-Threshold Computing

As previously discussed in Chapters 2 and 3, NTC provides 10× energy savings, but speed degradation comparable to operation at sub-threshold. Furthermore, because $V_{dd} \sim V_{th}$, process variation becomes a much more prominent issue in both design and operation. We address these two issues by introducing deep combinational logic chains for process variation and minimizing the number of operations under a specified maximal logic delay. In particular, our delay model

is novel in that it considers the loads that a particular gate drives while loads are traditionally only considered in gate-level systems. To the best of our knowledge, this is the first time that they are included in high-level (operation-level) synthesis.

4.1.2 Chaining

Chaining is a powerful architectural structure in which complex arithmetic or logic units are connected in succession without intermediate storage units, thus creating a chain of deep logic. It was first applied in the CRAY-1 computer system in the mid-1970s [46]. It has since been used in behavioral synthesis, architecture, and application specific instruction set microarchitectures [47] [48] [49]. Traditionally, chaining is used for reducing the required number of clock cycles.

Our key novelty is that we use chaining for reducing the impact of process variation. At super-threshold voltages, delay is predominantly affected by the magnitude of the supply voltage. At near-threshold voltages, delay becomes exponentially affected by the difference between the supply and threshold voltages. Thus, it is paramount that the effects of process variation be considered when designing circuits for NTC operation as tiny variations can cause big discrepancies in delays between similar circuit components.

Consider Figure 4.1, a functional MCM structure computing the multiplications against a single variable in an 8-bit FFT application. In this example we assume similar design characteristics (e.g. input size, load, and function) for each operation, while internal characteristics (e.g. threshold voltage) differ due to process variation. Traditional design techniques place registers between operations in order to capitalize on rapid clock rates and high throughput. However,

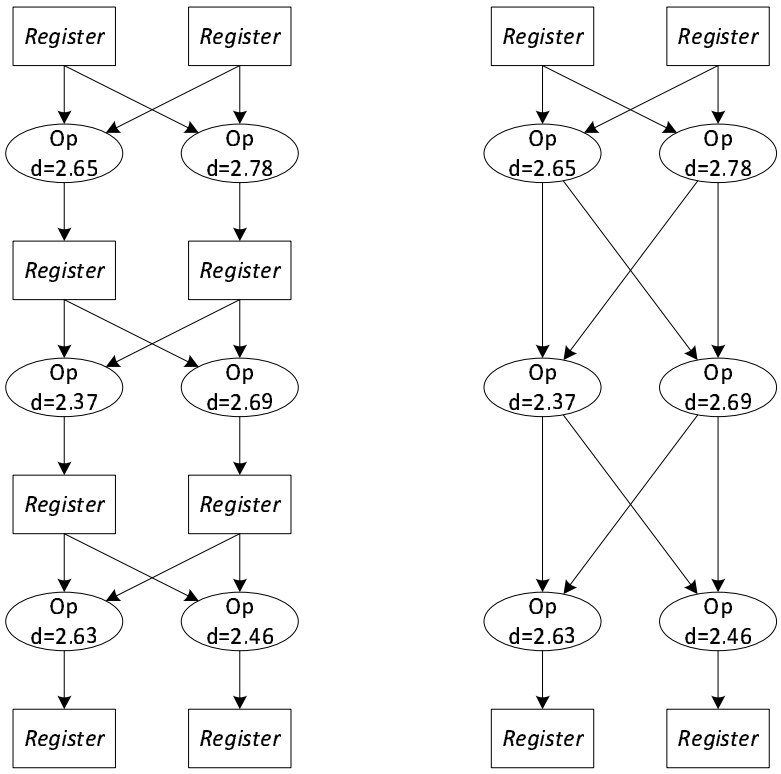


Figure 4.1: Motivational example depicting the effect of process variation on the delay of (a) multi-cycle and (b) deep chained logic circuits. In the multi-cycle case the clock rate is constrained by the maximum delay of each pair of operations. Thus, total circuit delay is 8.34. In the deep logic case, total delay is 7.93.

at near-threshold operation where process variation has an exponential impact on delay, this multi-cycle architecture can be detrimental. Due to inherent delay variations, the clock rate in Figure 4.1a is constrained to be no faster than the maximum delay of any set of operations positioned between registers. The overall circuit delay will be approximately the maximum delay times the required number of clock cycles to finish the operation to the end.

By implementing the same circuit using deep logic, as depicted in Figure 4.1b, we reduce the impact of the few extremely slow components affected by

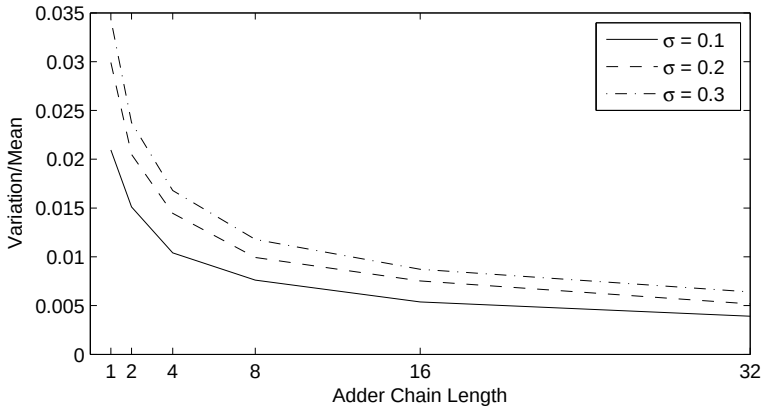


Figure 4.2: Effect of deep logic on the impact of process variation at near-threshold operation. Sigma σ values correspond to the experimental standard deviation in nominal threshold voltages.

process variation through delay averaging over long chains. Figure 4.2 depicts the quantitative results of the impact of long chains of multiple adders in succession.

4.1.3 Multiple Constant Multiplication

MCM structures were first studied in the context of minimizing the number of operations in FIR filters [50]. Cappello and Steiglitz were the first to prove that optimal MCM generation is NP-complete [51]. Consequently, several MCM generation techniques have been identified and addressed using three different types of approaches. The first treats MCM as a computer science theoretical problem in which the goal is to maximize the usage of common sub-expressions between the multiple constants [52]. The second approach treats the problem in a graph theoretical manner and utilizes minimum spanning trees [53]. While minimum spanning trees can be solved using polynomial complexity algorithms, finding edges requires potentially exponential time. The third approach has produced the best results for MCM generation and were obtained using a combination of artificial intelligence (heuristic search) and symbolic algebra (rewrite systems) by

Markus Puschel and his collaborators [54].

To the best of our knowledge, ours is the first effort to address ultralow power implementations of MCM computations in the general case by employing NTC. However, several groups have addressed minimizing power in FIR filters using common sub-expression elimination [55] [56].

4.2 Iterative Node Replication for Target Delay Yield in NTC

Our iterative node replication (INR) technique for delay minimization of MCM structures operating in near-threshold improves latency, yield, and reduces susceptibility to process variation through deep logic chaining and load reduction. Previous MCM design techniques that focus on reducing the number of operations also inherently increase average component fan-out. Higher loads induce longer delays and impose design-time assignment of larger sized components, which ultimately increases energy consumption. At near-threshold voltages these consequences can outweigh the benefits of instantiating a design with a minimal number of operations. Thus, in order to improve overall circuit delay at near-threshold, load must be a preeminent design consideration.

As mentioned in Section 4.1.3, there exists a large body of literature on algorithmic solutions for MCM optimization, especially since the problem is NP-complete. Thus, we build our solution utilizing the best heuristic tools currently available. We use the Spiral suite proposed and developed by Voronenko and Puschel and their collaborators [57]. Specifically, we begin with minimal depth MCM implementations for the set of constants corresponding to the pertinent benchmark. We specifically start with minimal depth over minimal operation be-

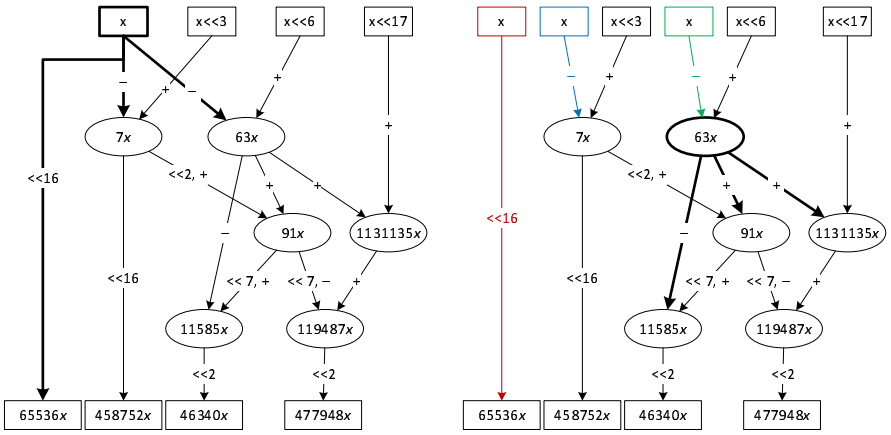
cause when operating in NTC, minimizing circuit depth is crucial for minimizing delay.

Figure 4.3 depicts an example of the starting, first, and second iterations of our INR technique for a single MCM tree corresponding to an FFT with 8 inputs. We describe the algorithm in more detail in the following section. All three implementations are functionally equivalent, but differ in terms of energy and delay when operated at near-threshold. In this example, we highlight the reduction in critical path delay that our replication technique achieves. After replicating input x and the $63x$ operation we observe a reduction in delay by about 300 ps. In this example we use approximate values for clarity in presentation. In simulation we apply the appropriate load-delay models for the appropriately sized adder modules.

4.2.1 Algorithm

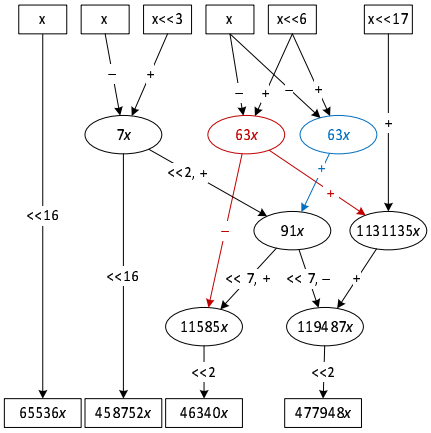
Load reduction and delay minimization is accomplished by iteratively replicating operations on the critical path that have maximal load. When considering multiple nodes that have zero slack and an equal maximal load, nodes whose transitive fan-out affect the largest set of epsilon critical paths are replicated first.

Load is distributed to the new replica by prioritizing output paths with the least amount of slack. Output paths with zero slack are swapped from the old node to the newly replicated node until no more zero slack paths can be assigned or the load of the new replica is half the load of the original. In the best case, there will exist only a single path with zero slack (i.e. a single critical path), and thus only a single output path with zero slack will be assigned to the replica node. By singling out this path and this path only we are able to maximally reduce the internal delay constraints on that path through load reduction by capitalizing on



(a) critical path delay = 13253 ps

(b) critical path delay = 12651 ps



(c) critical path delay = 12350 ps

Figure 4.3: Functionally equivalent MCM structures for a single variable of an 8 point FFT. (a) A minimal depth, minimal operation MCM tree created by the Spiral MCM synthesis tool [57]. Bolded nodes are those selected to be replicated in the following iteration. (b) A reconstructed MCM tree created by replicating inputs for x from the previous iteration. (c) The next iteration of the MCM tree created by replicating the $63x$ operation and balancing output load. Tables 4.1 and 4.2 specify the values and delays used in this example.

Constant (`float`)	Fixed-point Representation (`uint`)
1.0	65536
0.707107	46340
−0.707107	477948
−1.0	458752

Table 4.1: Multiplier constants for a single input variable used in Figure 4.3 in fixed-point representation using 16 fraction bits and 3 integer bits.

Load	Delay (ps)
1	2937
2	3238
3	3539
4	3840

Table 4.2: Approximate delay values used in Figure 4.3 for a carry-lookahead adder (cell size 2) operating at near-threshold.

the positive slack of the original node's remaining output paths.

For example, in Figure 4.3b, node $63x$ is chosen for replication because it has the maximum load among all nodes on the critical path. Once node $63x$ is replicated in Figure 4.3c, we assign only the path to $91x$ to the newly created (blue) replica since it is the only node that is on the critical path. In this way, the replicated $63x$ node (blue) that is now a part of the path with the highest constraint will have the least load between both the red and blue $63x$ nodes, thereby reducing the internal constraints as maximally as possible. For the case that more than half the paths from the replication node have zero slack, we distribute the load evenly between the original and the replica node.

Final MCM selection is accomplished by choosing a single MCM from our generated pool of iterations that has minimum energy when satisfying the target delay.

4.2.2 Results

We synthesize MCM implementations of DCT and FFT benchmarks in the presence of process variation and compare our results to multi-cycle and deep logic implementations of the Spiral solutions. A fixed width size of 16 fractional bits and 3 integer bits is used. We employ Markovic's gate level models [43] after fitting them to an industrial standard cell library [58]. Each cell is sized per capacitive load requirements and input transition slew. Our nominal threshold voltage is 0.33 V.

In Table 4.3 we compare nominal solutions corresponding to the delay of the multi-cycle Spiral solution when applying a near-threshold supply voltage. We scale the supply voltages for the deep logic and INR cases to achieve the same target delay and record the resultant energy consumption values and area requirements. Our techniques have an energy savings improvement ranging from 10% to 70% beyond even the Spiral deep logic solution. In some cases, this is achieved even when imposing additional area overhead.

Figure 4.4 depicts the circuit yield with respect to energy and delay in the presence of process variation for the 16x16 DCT application operated at near-threshold. Our techniques generate simultaneously lower energy and lower delay implementations of the required MCMs for this application. We find that not only does our load reduction technique reduce circuit delay, but also simultaneously reduces energy by eliminating the need for larger cell sizes that the original Spiral solution requires due to its ultra compact and high fan-out structure.

Figure 4.5 depicts the energy usage for operation of the pertinent synthesized benchmark application at a desired target delay. For smaller circuits we observe similar improvements to the Spiral deep logic implementation. In the case of the 4x4 DCT, the circuit is too small for replication to have a meaningful impact be-

fore the increase in the number of operations consumes too much energy without sufficient delay improvements. However, for larger applications in which MCM synthesis becomes very complex, we observe that our techniques substantially reduce energy consumption rates. This is expected since larger complex minimum depth MCM trees will contain many operations along many epsilon critical paths, thus harboring many opportunities for load reduction exploitation.

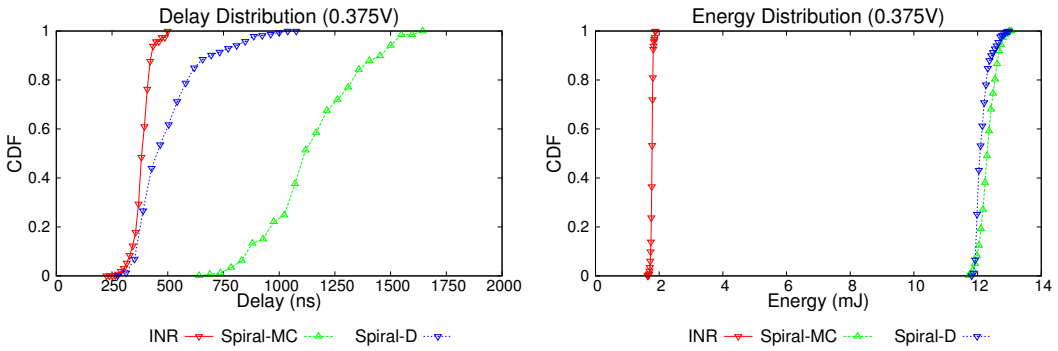


Figure 4.4: DCT–16x16 circuit yield with respect to (top) delay and (bottom) energy in the presence of process variation when operating in near-threshold with a nominal V_{th} of 0.33V. We compare our technique with the multi-cycle (MC) and deep logic (D) implementations of Spiral's heuristic solutions.

Benchmark	Solution	Target Delay (ns)	Energy (mJ)	Energy Savings	# Operations	Area (μm^2)
FFT–32	Spiral-MC	55.31	12.72	1.00×	651	2.06×10^6
	Spiral-D		3.29	3.87×	651	1.83×10^6
	INR		2.45	5.20×	692	1.90×10^6
FFT–64	Spiral-MC	55.31	60.51	1.00×	2347	8.01×10^6
	Spiral-D		14.91	4.06×	2347	7.00×10^6
	INR		10.58	5.72×	2612	7.55×10^6
FFT–128	Spiral-MC	55.31	367.56	1.00×	8555	3.19×10^7
	Spiral-D		79.78	4.61×	8555	2.80×10^7
	INR		46.49	7.91×	9204	2.82×10^7
DCT–4x4	Spiral-MC	42.91	13.58	1.00×	456	1.44×10^6
	Spiral-D		1.90	7.14×	456	1.30×10^6
	INR		1.74	7.81×	493	1.37×10^6
DCT–8x8	Spiral-MC	61.86	305.28	1.00×	5835	2.12×10^7
	Spiral-D		50.52	6.04×	5835	1.84×10^7
	INR		33.38	9.14×	6121	1.88×10^7
DCT–16x16	Spiral-MC	58.29	10822.23	1.00×	75234	3.12×10^8
	Spiral-D		1539.11	7.03×	75234	2.71×10^8
	INR		976.43	11.08×	76011	2.66×10^8

Table 4.3: Energy and area results for FFT and DCT applications synthesized using multi-cycle (MC) and deep logic (D) implementations of Spiral's heuristic synthesis tool and our iterative node replication techniques.

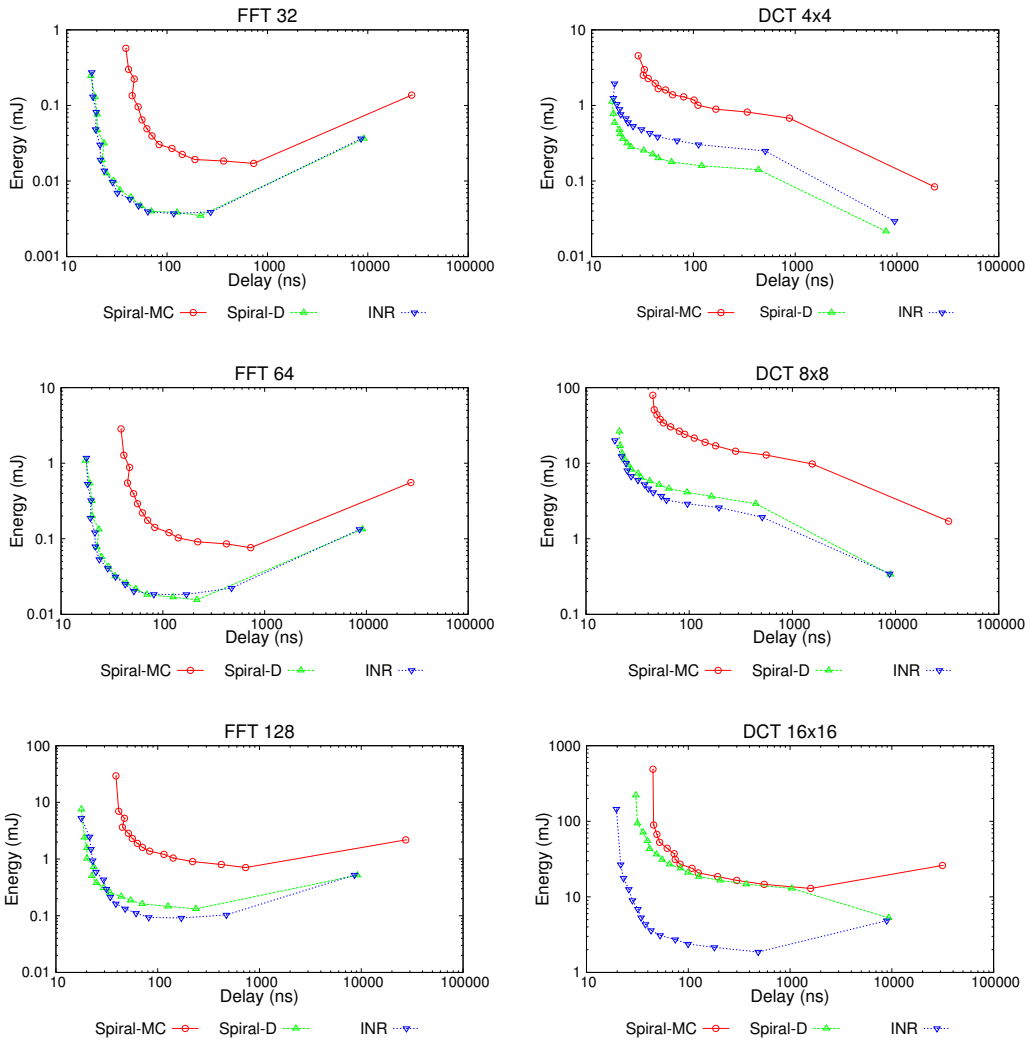


Figure 4.5: Circuit energy consumption for target delays for FFT and DCT benchmark applications using multi-cycle (MC) and deep logic (D) implementations of Spiral's heuristic solutions and implementations generated using our iterative node replication technique.

4.3 Summary

Many IoT devices employ linear systems to enable applications such as video capture, image processing, and radio communication. MCM is a popular archi-

tecture that enables such systems, however an ultralow energy implementation using near-threshold computing has not yet been presented until now. Our techniques not only reduce energy, but also recover lost speed due to NTC operation of MCM structures, through node replication, which directly reduces load in order to reduce the critical path and ultimately reduce energy by enabling smaller cell size assignments. Furthermore, we reduce the impact of process variation through the use of deep combinational logic.

We have explored our techniques for MCM optimization on the popular FFT and DCT linear transforms in the presence of process variation using accurate gate-level models and cell sizing techniques. We find that for larger applications requiring larger and more complex MCM structures, our techniques show substantial improvements in energy reduction for a range of target delays.

CHAPTER 5

Semantics-based System Configuration for Energy Reduction

The IoT paradigm holds great potential for semantic-based design, analysis, and learning, and has already created numerous opportunities for the improvement of existing semantic technologies [59]. The sheer number of devices alone will introduce new challenges in the representation, storage, organization, and search of all information generated by this large ecosystem and will create new opportunities for semantic reasoning and semantic-based operation.

In this chapter, we embrace this semantic-oriented vision of IoT to create semantics-based approaches to energy reduction by organizing and coordinating system-level components. We target a medical shoe, outfitted with numerous sensors on each sole specifically because of it semantic nature (e.g. deriving medical diagnostics from raw sensor measurements).

There is great potential for the medical community to benefit from wearable sensing systems by utilizing their capacities for remote surveillance to observe and diagnose patient ailments and disease. These Internet-connected health systems allow doctors to remove the constraint that they rely solely on in-person patient checkups and interviews in order to diagnose patient illness. By utilizing non-invasive wireless health monitoring, doctors are able to incorporate information gathered from the patient's day to day activities and routine into their

professional medical diagnoses.

However, such multisensory systems are often expensive and have rigid energy and power requirements due to their wireless framework and remote deployment. Due to the often complex design, expensive cost, and energy demands that can accompany such wireless sensor networks, wireless medical sensing systems have not yet made headway into widespread use. Medical sensing systems can contain very large sensor arrays; for example, a commercial medical shoe might contain as many as ninety-nine sensors [60]. Not only are these sensors expensive and make up a shoe that costs thousands of dollars, but they also draw power and consume energy. However, a medical shoe is inherently a mobile device, attaching a large battery pack or requiring frequent recharges are a strong deterrent to the adoption of such a medical shoe by the common patient.

In order to minimize the cost and energy demands of these sensor networks, system-level configuration techniques for sensor coverage, selection, and sampling have been proposed. These techniques often focus on maintaining full predictability of the original array while reducing the sensor count and coordinating sensor subsampling.

In a great majority of sensor network applications, readings from a single sensor are not sufficient for extraction of information, knowledge, or decision processes. Furthermore, we are rarely interested in the raw sensed data, but rather more interested in the semantic information it provides. For example, in a sensor network distributed among the trees of a wildlife preserve, we may rarely be interested in the raw temperature of many or even all points in a given area but rather more concerned with the early detection of fire.

Traditional approaches to sensor selection for energy reduction in multisensory systems remove redundant sensors from the array while maintaining full sensor

predictability [61]. However sensor predictability (i.e. raw data prediction) is not necessary when the essential information is the semantic information itself (e.g. fire detection). Thus, our key conceptual contribution is the reduction of energy using semantics-driven sensor configuration; we reduce the energy requirements of the sensor network while maintaining system relevance, in particular, for medical diagnosis.

This is achieved through a two step process, using coarse grained energy reduction techniques followed by fine grained techniques. We explain these steps in detail in Section 5.3.

In coarse grained energy reduction we introduce sensor fusion and bottom-up selection techniques that maintain semantic relevance. In Section 5.3.1 we describe our sensor placement algorithms, including: sensor selection, which iteratively chooses sensors that most improve semantic prediction accuracy; adjacent sensor fusion, in which sensors are physically or electronically combined; search space pruning for runtime reduction; and a generic method for comparison of orthogonal semantics.

Our fine grained energy reduction technique employs a very similar approach to the coarse grained technique, but applies to sampling instead of sensors. Ultimately, we find a minimal subsampling configuration that reduces energy while maintaining semantic relevance.

Our technical contributions are algorithms and methods for (i) sensor selection with an accompanying method for ranking and sorting of arbitrary semantics, (ii) sensor fusion, and (iii) sampling configuration, all of which employ the key conceptual idea that energy reduction is semantics-driven.

We use as our driving example, a medical shoe fitted with ninety-nine pressure sensors on the sole. The semantics this system is designed to compute are gait

characteristics, shown by VanSwearingen et al. to be highly correlated with disease and the risk of falling [62].

5.1 Related Work

The emergence of embedded sensor networking has introduced new scientific and engineering challenges. Much attention is now focused on energy and power reduction in wireless sensor networks due to the often large networks and their constant power demands [63]. Current energy optimization methods focus on hardware design, signal processing, and sensor selection [64] [65] [66].

Recent attention in wearable sensing systems has fostered a growing interest in medical-based sensing devices [67] [68]. Like research in other wireless sensing systems, current attention in the wireless health domain focuses on the utility and convenience of such systems as well as their cost and energy demands [67] [69] [70] [71] [72].

Investigations have been made into the application of gait analysis in wearable sensing systems such as sensor-equipped shoes [70] [71] [72]. These provide solutions for semantic computation from raw sensor data, but do not leverage gait analysis for energy optimization. Efforts employing gait analysis for energy optimization are preliminary and limited to analysis of same size sensors and prediction of only single semantic measurements [73] [74] [75].

5.2 Preliminaries

5.2.1 Semantics-driven Energy Reduction

In general, sensor selection is best applied on large multisensory arrays for maximal energy savings. Traditional approaches remove redundant sensors from the original array while maintaining full sensor predictability [61]. However, in semantics-driven devices, such as medical sensors for diagnosis, sensor predictability (i.e. raw data prediction) is not necessary. The essential information that the system is intended to measure are the diagnostic semantics (e.g. gait characteristics in a medical shoe).

Our semantics-based sensor configuration technique for energy reduction benefits sensor networks that are ultimately designed for measuring semantics from the raw data. For example, if a sensor network is required to measure temperature in a forest using thermometer sensors, traditional techniques for sensor selection are sufficient to reduce energy and cost of the system. If, however, the purpose of the forest sensor network is to detect fire through the analysis of temperature measurements, we claim that using a semantics-driven approach to sensor selection—rather than raw data predictability—yields stronger semantic prediction and energy reduction over traditional approaches.

5.2.2 Medical Shoe

Medical sensor networks are inherently semantics-driven systems. A doctor is very often not concerned with the raw measurements of the sensors (e.g. accelerometers, pressure sensors, electrochemical biosensors); but rather, more concerned with the semantic information derived from those sensors (e.g. distance/speed of an impact, gait characteristics, glucose levels).

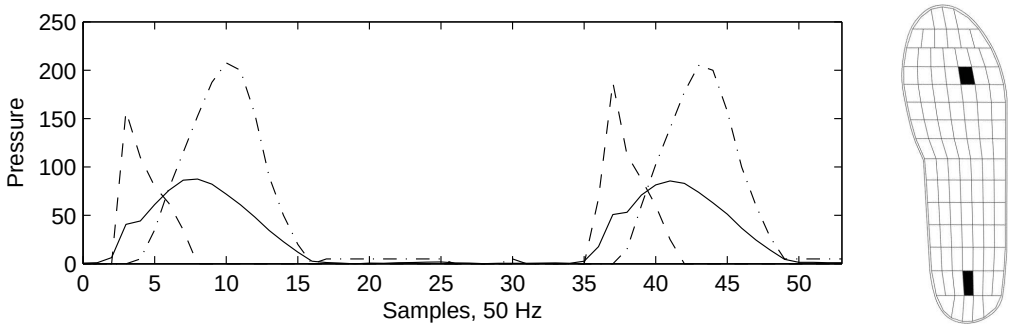


Figure 5.1: Pressure measurements of two steps of a single foot measured by sensors on the heel (dash), toe (dash-dot), and averaged over all ninety-nine sensors (solid). The heel and toe sensors are shaded on the Pedar mapping [60].

We perform our semantics-driven energy reduction methodology on a wireless medical wearable sensing system containing a large multisensory array, known as the Hermes shoe [76]. This platform is designed with the purpose of assessing balance and instability in patients through the measurements of ninety-nine passive resistive pressure sensors distributed on the sole of the foot using the Pedar plantar mapping [60]. The goal is to passively measure the required semantics and send that data back to a doctor or other medical professional for analysis periodically.

The processing unit samples data from these pressure sensors at 50 Hz using a 16-bit analog-to-digital converter. An example of the sensor mapping and typical step profile is depicted in Figure 5.1. The data consists of hundreds of steps from five subjects.

5.2.3 Gait Characteristics

Medical shoes similar to the Hermes platform have been used in a variety of applications, from sports science to elderly care. In this paper, we choose gait characteristics, such as step stride, change in step stride, maximum pressure,

lateral pressure difference, and guardedness (time between heel and toe landing) as the semantics for our study. VanSwearingen et al. have concluded that these gait characteristics correlate to a number of ailments and diseases in the elderly and directly contribute to the prediction of risk of falling in this population [62]. This strong correlation between gait and risk is a powerful means to help medical professionals diagnose these ailments with the availability of such gait data.

We normalize and extract these gait characteristics measured collectively by all ninety-nine sensors as well as measured by the individual sensors independently. Our sensor selection procedure conducts metric prediction using the metric measurements at each sensor, while our sampling solution determines the best sampling of raw data for a given set of sensors and their metric prediction model. We separate our data into training (80%) and testing (20%) subsets.

5.3 Energy Reduction

Energy reduction is achieved by applying two sequential steps to the design space: coarse grained optimization, which includes sensor fusion and selection, followed by fine grained optimization, which includes subsampling configuration.

In describing the coarse grained technique, we introduce our semantics-driven sensor selection algorithm that reduces the original multisensory array to a minimal subset of sensors that accurately predicts the semantics introduced in Sections 5.2.1 and 5.2.3. This approach is a bottom-up selection process which begins with an empty set, then iteratively adds sensors until a sufficient minimal set is found that accurately predicts the semantics to a specified threshold.

Our fine grained energy reduction technique applies a similar approach, however is performed on sensor-samples instead of sensors. A sensor-sample is a

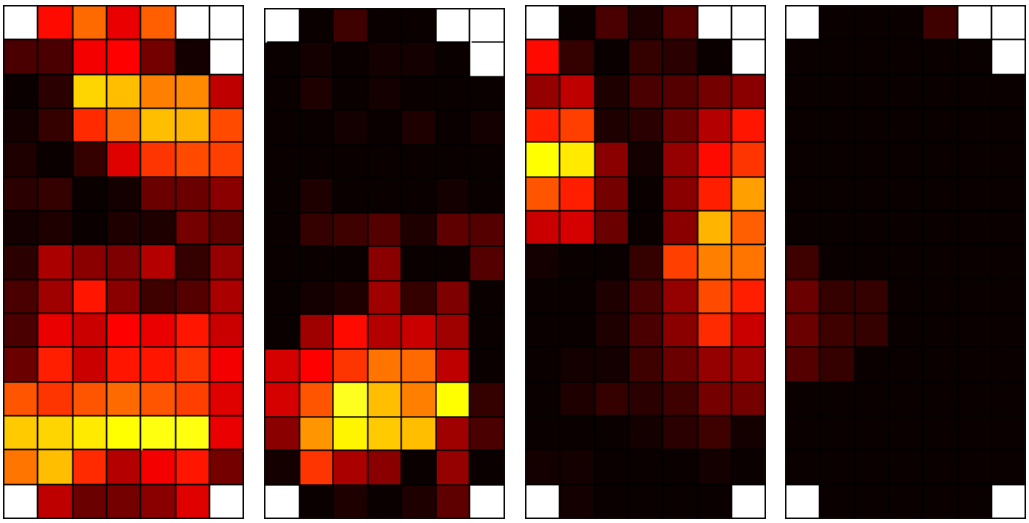
single measurement of a unique sensor at a discrete time step. In this approach, sensor-samples are iteratively removed from the minimal sensor set (found via coarse grained optimization) while maintaining semantic prediction accuracy.

Theoretically, it is possible to bypass the coarse grained sensor selection process and only apply fine grained subsampling to the original design. However, the exponential search space of this problem, along with the huge number of sensor-samples that accompanies a large multisensory array, renders this problem near impossible to solve in a reasonable amount of time. Given m samples and n sensors there are 2^{mn} possible solutions. By applying coarse grained sensor selection first, we reduce n to a small constant and reduce the problem size.

Additionally, while our primary concern is the reduction of energy usage in these medical devices, another equally important concern is the reduction of their cost, which is proportional to the number of sensors. Even if in a reasonable amount of time we could find an optimal solution using only sensor-sample selection, it is not immediately evident that this solution would ultimately reduce the number of required sensors. For example, a hypothetical solution could be that only a single sample measurement is taken at each of the ninety-nine sensors per step. While this drastically reduces energy usage, the cost of the shoe remains exorbitantly high since no sensors are removed. Coarse grained sensor selection ensures cost reduction in addition to energy reduction.

5.3.1 Sensor Configuration

Our coarse grained optimization technique reduces both energy consumption and cost of the sensor network by combining and removing sensors from the original array while maintaining prediction accuracy of the aforementioned gait characteristics. A key observation in Figure 5.2 is that while clear correlations might



(a) (b) (c) (d)

Figure 5.2: Individual sensor coefficients of determination for (a) average maximum step amplitude, (b) change in step stride, (c) lateral pressure difference, and (d) guardedness. The lighter the sensor, the more correlated it is to the metric; darker shadings denote weaker correlations.

exist between the measurements of a few individual sensors and the individual semantics, between the four semantics it is not immediately apparent which small subset of sensors can predict all semantics simultaneously well.

We explore two degrees of freedom in searching for a minimal set of optimal predictors: sensor *fusion* and sensor *selection*. Combining sensors electronically or physically reduces the total energy spent in sensing since the fused sensor acts as a single sensor and is sampled as one. We find that these fused sensors are often better semantic predictors than individual sensors.

We employ three steps to achieve coarse grained energy and cost reduction. The first step is sensor fusion. The second step is pruning, in which we eliminate combinations of sensors and fused sensors whose semantic measurements are highly correlated with one another. This ensures that pairs of sensors that

yield no significant advantage in prediction over each other are not considered for combination, thus reducing the problem size. The final step is sensor selection.

5.3.1.1 Sensor Fusion

A fused sensor is a set of adjacent sensors physically or electronically combined with one another to create a single, larger sensor whose measurements are averaged over its new area. Including sensor fusion as an additional dimension in sensor selection provides benefits in semantic prediction. A fused sensor effectively becomes a new sensor to consider, it helps reduce noise between its constituent sensors, and simultaneously reduces system energy requirements.

However, for the purposes of sensor selection, coupling the fused sensor solution space (2^n) with that of sensor selection (2^n) yields a hugely exponential search space. In order to reduce this effect we pre-select a number of fused sensor shapes. By choosing a static set of these combinations and treating them as unique sensors, we restrict the search space.

We use application specific knowledge to pre-construct a number of fused sensor shapes. For example, the correlations of determination for the semantics in Figure 5.2 reveal that potentially good sensor combinations might be squares, rectangles, or L-shapes. Thus, from the existing ninety-nine sensors, we construct five new fused sensors, as depicted in Figure 5.3, and apply them and their rotations across the sensor mapping, effectively adding 544 fused sensors to the selection pool.

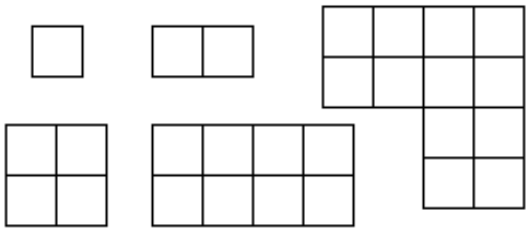


Figure 5.3: Shapes used in sensor fusion, pre-computed prior to sensor selection.

5.3.1.2 Pruning

With 643 total sensors to consider, the exponential search space remains large. Even with our approach to sensor selection that reduces the number of potential sensor groupings considered at each iteration to the constant k (described in Section 5.3.1.3), sensor selection is still a very time consuming task. By pruning the search space and reducing the number of sensors considered at each iteration, we instead use that computation time to increase the value of k and thereby drive our algorithm to find a more optimal solution. The key observation is that if two sensors give similar predictions for each metric, then there is likely no benefit in having them both in the same predictive set. In experimentation, pruning had minimal, if any, impact on sensor selection while still reducing overall computation time.

We use this observation to formulate the pruning problem as a complete weighted graph, where each sensor is a node and each edge has weight equal to the maximum difference between the predicted values for each metric by the two corresponding sensors. We then prune all edges with weight greater than ω, where ω is a specified similarity threshold. Now, adding a sensor s to a predictive set eliminates all sensors with edges to s as future candidates for that set, since those sensors will not add any additional information about the gait metrics.

5.3.1.3 Sensor Selection

Our semantics-driven sensor selection procedure, described in detail in Algorithm 2, is a bottom-up selection process. It systematically selects the best combinations of sensors until a minimal subset that accurately predicts the required semantics is found.

Figure 5.2 represents significantly strong linear correlations between a sizeable portion of sensors and three of the semantics. Thus, we use linear regression for semantic prediction; however, our method can easily be implemented using almost any prediction model. For example, it could be postulated that a linear regression model is not best for guardedness prediction given the low correlation values per sensor in Figure 5.2d. Interestingly, while no single sensor is an adequate predictor for guardedness using a linear model, after a couple iterations of sensor selection, guardedness prediction error drops drastically. This intuitive result derives from the fact that the guardedness metric is inherently derived from the measurements of at least two sensors.

The prediction error of sensors varies from metric to metric. Due to the very application-specific nature of these semantics, some are inherently harder to predict (change in step stride) while others are very well suited to the sensor and system design (amplitude). Because of these discrepancies, it can be very difficult to determine the relative prediction accuracy of a single sensor against two different metrics. We overcome this barrier by mapping the prediction error of a given sensor for a given metric to the cumulative distribution function of the prediction errors of all the sensors for that same metric. This binds the error to a normalized value that is relative to the rest of the system's prediction capabilities. Now, we are able to compare metric predictions and correctly rank our sensors by how well they predict each metric relative to one another. Our ranking function

weights each metric equally, since ultimately we are most interested in designing a medical device that can provide the doctor with the most information. This functionality is embodied in the *SortByPredictionError* and *GetPredictionError* functions in lines 10 and 11 of Algorithm 2.

While sensor fusion and pruning help to reduce the search space, they still only do so nominally. The process of combining all previously considered sensor combinations with the set of remaining sensors at each iteration still grows exponentially. Thus, at each iteration, instead of looking at all possible sensor combinations, we only investigate combinations with the best k size subset of the previous iteration's results; for every set in the best k solutions from the previous iteration, we create $O(n)$ new sensor configurations. This reduces the original exponential problem formulation complexity to $O(kn^2)$.

It is important that the k-size best subset retained between two iterations contains both top predicting sensor sets as well as a representative distribution of the current solution space. This ensures that our selection process drives to an optimal solution quickly while providing neighboring solutions that could potentially find other local minima. This is done through a greedy selection of the top $k/2$ configurations and an evenly distributed selection of $k/2$ configurations from those remaining. This functionality is embodied in the *GetGreedyRep* function in line 4 of Algorithm 2.

5.3.2 Subsampling

While sensor selection is a crucial step for reducing the cost and complexity of wearable medical sensing systems, energy is ultimately spent mainly in sampling. Therefore, the sampling strategy is of utmost importance to energy optimization of any sensor network. We conduct subsampling post-selection, based on the

Algorithm 2 Sensor Selection

Require: S_1 = set of sensor configurations of size 1
Require: k = # of configurations retained between iterations
Require: G = pruned graph of sensor-semantic correlations as described in Section 5.3.1.2, $N_G(s)$ is the of set neighboring sensors to s in G

$i = 1$
while $\epsilon > threshold\_error$ **do**
  $i = i + 1$
  T = GetGreedyRep(S_{i-1}, k)
  for $1 \leq j \leq k$ **do**
    for all sensors s, $s \notin T[j]$, $s \notin N_G(T[j])$ **do**
      Append $T[j] \cup s$ to S_i
    end for
  end for
  S_i = SortByPredictionError(S_i)
  ϵ = GetPredictionError($S_i[0]$)
end while

Ensure: Top sensor configuration, $S_i[0]$

following key observations: (i) during ambulation, the foot spends a majority of the time in the air and therefore applying no pressure to any sensors; (ii) a single sensor is sufficient to detect the start and end times of a step; and (iii) during a step, applied pressure follows multimodal behavior predictable from semantic information, as described in [61] and [77].

Based on the first two observations, we add a sensor that covers the entire sole and sample it at the full rate (50 Hz) solely to detect the start and end times of steps. Note that this sensor is large and therefore subject to a high signal-to-noise ratio, rendering it ineffective for predicting gait metrics. Without loss of accuracy, we can begin sampling all other selected sensors only when the foot lands, and stop sampling as soon as the foot leaves the ground.

We formulate the sampling reduction problem as a simple variation of our sensor selection algorithm presented in Algorithm 2. Again, this is an iterative process, but now, at iteration i, the k strongest predicting and representative sets

of $n - i$ sensor-samples are returned, where a sensor-sample is a single sampling point of a single sensor, and n is the number of coarse grained selected sensors times the number of samples in the step. Therefore, we make the following modifications: S_i now represents a set of sensor-sample configurations at iteration i, s is a sensor-sample in question to be removed, and line 7 now reads, Append $T[j] - \{s\}$ to S_i. Note that by using this approach, each sensor can have a different sampling rate, as only one sample of one sensor is removed at each iteration. Furthermore, the same strategy for pruning can be applied as in sensor selection, with sensor nodes replaced by sensor-sample nodes.

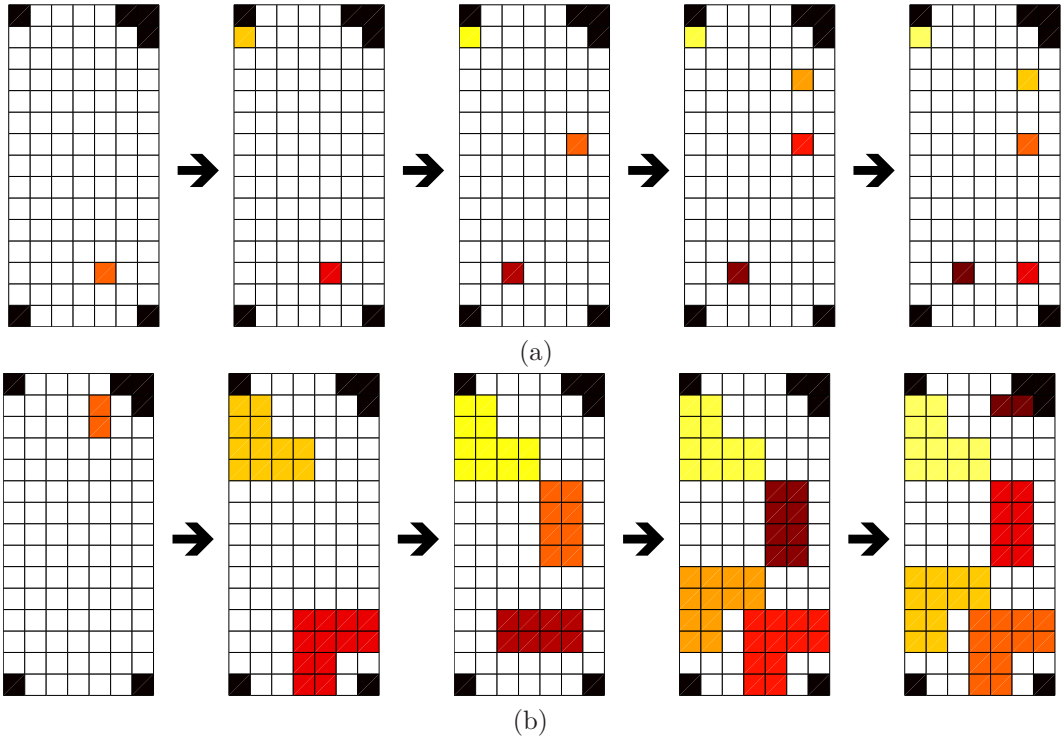


Figure 5.4: Top sensor configurations at iterations $1 \leq i \leq 5$. Solution (a) limits sensor selection to individual sensors, (b) includes sensor fusion.

5.4 Results: Sensor Configuration

We perform the semantics-driven sensor selection algorithm with pruning on individual sensors only, then incorporating sensor fusion, and compare both results to a traditional sensor selection technique that retains raw data predictability. In both configurations depicted in Figure 5.4 the selected sensors are well distributed over the inside and outside of the foot, cover both the heel and the toe, and generally cover the highest correlated areas depicted in Figure 5.2.

Semantics-driven sensor selection clearly performs better than traditional selection as seen in Figure 5.5. For the same level of energy reduction, semantics-driven selection predicts a factor of 5.8 times and 1.6 times better than the traditional approach for the change in step stride and guardedness metrics, re-

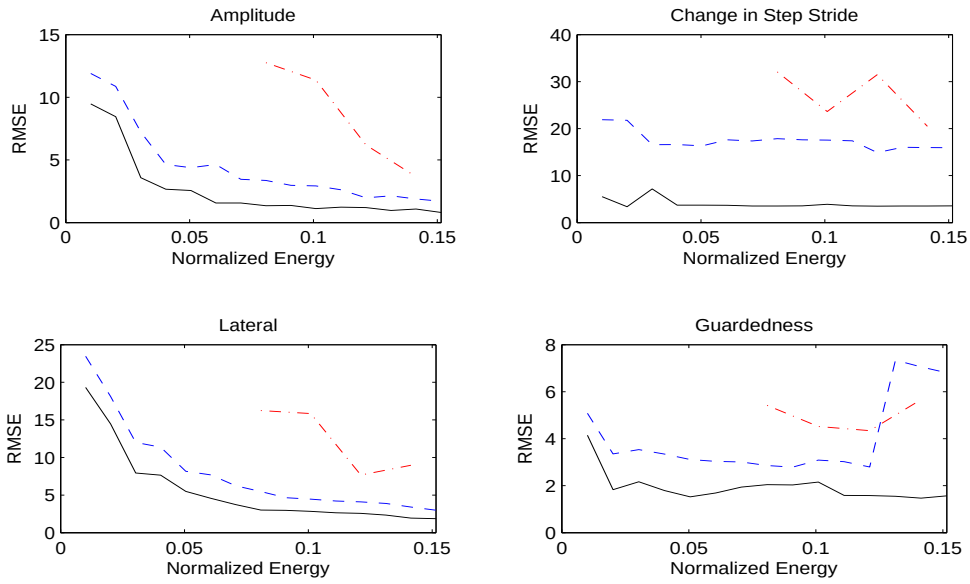


Figure 5.5: Coarse grained optimization via sensor configuration. Root mean squared testing error in prediction using best selected sensors using only single sensors (dash), using sensor fusion (solid), and results from Noshadi et al. [61] (dash-dot). Units are *pressure* for amplitude and lateral, and *samples* for step stride and guardedness.

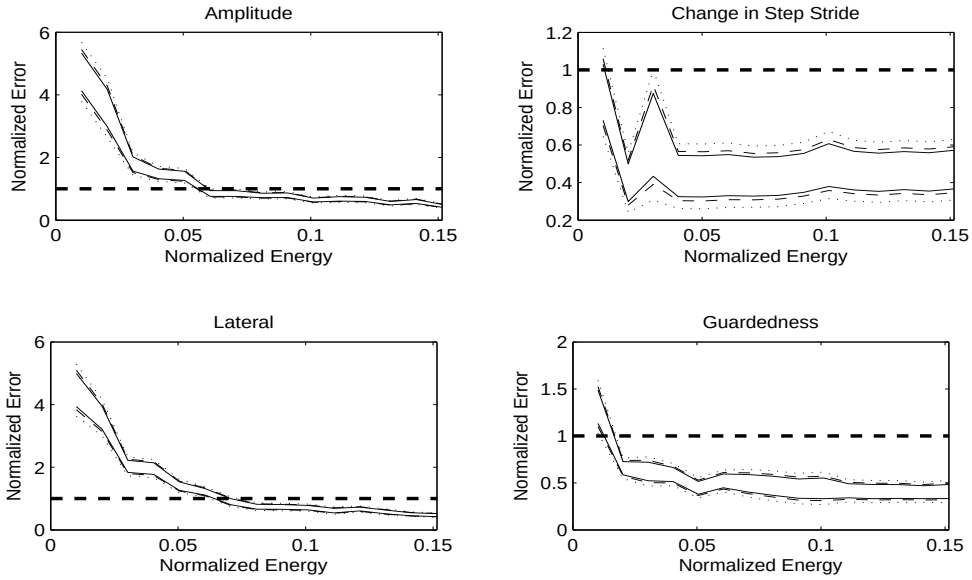


Figure 5.6: Coarse grained optimization using sensor fusion. Semantic prediction error as a percentage of desired error threshold (thick dashed); 90% confidence interval (solid), 95% (dashed), 99% (dotted). When the prediction error reduces to the threshold for all semantics, sensor selection is complete.

spectively. Analysis of the amplitude and lateral semantics shows that semantics-driven sensor selection provides a factor of about 4 times better energy reduction over traditional methods for a constant prediction error rate. Between semantics-driven selection using individual sensors and fused sensors, the latter provides much better prediction capabilities, and thus, greater energy gains.

Figure 5.6 shows that semantics-driven sensor selection reaches the desired error threshold by the seventh iteration, returning a factor of 14.1 in energy reduction over the original ninety-nine sensor system design. Recall that while the correlations between individual sensors and the guardedness metric are nearly non-existent (Figure 5.2d) the semantics-driven sensor selection process reduces the prediction error of this metric immediately in the second iteration, a testament to the semantics-driven procedure.

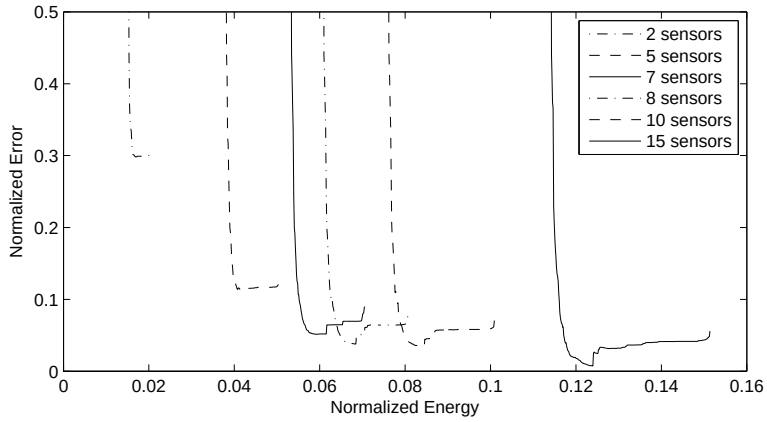


Figure 5.7: Fine grained optimization via subsampling configuration applied to the best configurations found in coarse optimization. Curves are constructed from right to left as sensor-samples are removed iteratively. The four semantic errors are normalized for equal comparison between semantics. Normalized energy is the fraction of energy expended on the new configuration over the original array of ninety-nine sensors sampled at 50 Hz.

5.5 Results: Subsampling

We apply fine grained optimization to the best sensor configurations of size 2, 5, 7, 8, 10, and 15 sensors found in the coarse grained step. In all cases, fine grained optimization further reduces energy from the coarse grained solution by the equivalent of one to three sensors sampled at 50 Hz. For example, applying fine grained optimization to the 10 sensor solution reduces the energy consumed by the configuration to the equivalent of employing 8 sensors at 50 Hz (see intersection of 8 and 10 sensors in Figure 5.7).

As we reduce energy in Figure 5.7, prediction error dips slightly, but for the most part, remains relatively flat until those sensor-samples that are most crucial for semantic prediction are removed. This result is expected due to the predictable nature of the pressure signals, gait metrics, and physiological events, in general, as observed by Noshadi et al. [61] and Meguerdichian et al. [77].

As we reduce sensor-samples the prediction error initially decreases slightly. This is because in the initial iterations, the sensor-samples that are removed are those that actually impair the prediction model. Though as expected, eventually the iterative removal of sensor-samples begins to have a negative effect on prediction and the error rises sharply. At this point we stop the fine grained optimization process and present the final configuration.

While coarse grained optimization reduces the original array to seven best predicting sensors, applying fine grained optimization further reduces energy to 5.6% of the original sensor array, a factor of 17.9 overall improvement.

5.6 Summary

The semantics-oriented vision of IoT has become one of the main impetuses for the proliferation of this technology [59] [78]. We have adopted this approach to develop semantics-driven sensor selection and subsampling configuration techniques for energy reduction in sensor networks. We leverage the key observations that the raw sensed data is unimportant, that only the metrics relevant to diagnosis are needed, and that the important metrics can be easily derived from the raw data. Consequently, our key contributions are as follows: (i) a bottom-up iterative approach to selection of a minimal set of best predicting sensors; (ii) a novel procedure for physically or electronically combining adjacent sensors to reduce sampling cost while improving prediction strength; and (iii) an extension of our sensor selection algorithm to minimize the sampling rate of individual sensors while maintaining accuracy. Our approach yields a cost reduction of 93%, and furthermore reduces energy by a factor of 17.9 over the original system configuration of ninety-nine sensors sampled at 50 Hz.

CHAPTER 6

Energy Harvesting

Wireless sensing networks have become a powerful and practical means for extracting information in a myriad of environments, ultimately extending the reach of data collection to gain new and deeper insights into new domains. Most notably, the application of sensor networks to the human body has helped foster the growth of the wireless health community. Medical professionals are now able to extend the reach of ailment and disease diagnosis beyond the walls of hospitals by gathering medical measurements during the day to day routines of their patients.

While small form, low power, and high accuracy are some of the more important design considerations for medical sensing devices, high accuracy can often lead to large, complex, and power hungry designs. However, most medical sensing devices are inherently mobile, such as walking canes and medical shoes, and should be designed with low energy and low power features as the foremost design considerations [76] [79].

While traditional approaches to optimizing power demands attempt to minimize energy requirements, we present a new procedure that employs energy harvesting techniques. Instead of a top-down approach that attempts to minimize the design to fit the energy restrictions of the power system, we propose to switch the power system to a sustainable energy source. The focus of this chapter is on creating a sustainable medical wireless sensing device, specifically, a medical shoe fitted with sensors that measure gait characteristics and abnormalities. We

present a sustainable design via spatiotemporal assignment of harvesters that extracts maximal energy determined by the gait and pressure spatiotemporal distribution of the patient.

6.1 Related Work

The emergence of wireless sensor networks has paved the way for new methods for human monitoring through a variety of applications including gait analysis, sleep observation, and emotional health monitoring [76] [80] [81]. However, some wireless health devices, such as a medical shoe, must be accompanied by a mobile power source. Even so, some wearable sensing systems are still designed with power hungry sensors and large arrays. When in operation these systems can drain their power sources quickly, often requiring frequent battery replacements or recharges, ultimately deterring patient adoption of such wearable sensing systems.

Traditional approaches to minimize the impact these power hungry devices have on patient routine range from cost minimization to energy reduction in the hopes of creating cheaper, smaller form, and lower energy devices [63] [64] [65] [66] [82] [83]. Energy reduction in particular has been accomplished through a multitude of techniques, including hardware design, sensor array reduction, and subsampling [61] [74] [75] [77]. In Chapter 5 we proposed a semantics-based approach for energy reduction at the system components level.

The topic of energy harvesting encompasses a wide range of mechanisms including solar capture techniques, electrostatic and piezoelectric transduction, and thermoelectric harvesting, among others [69] [84] [85]. In the context of human energy scavenging, walking might be the most obvious source of harvestable energy. Shoe harvesting designs date back to as far as the 1920's, however these

systems were much too bulky for practical human wear [86]. More recent solutions have utilized piezoelectrics and electromagnetic generators for energy conversion, however the latter remains too bulky for comfortable use and while the former has been one of the driving materials behind the development of energy scavenging shoes, it's energy density remains low.

Most recently, dielectric elastomers (DEs) have emerged as the premier class of material capable of energy densities ranging from 5 to 40 times the densities of piezoelectrics [87]. Similar to piezoelectrics, they posses the ability to behave as energy transducers, actuators and sensors. While piezoelectrics create an electric field when flexed due to their internal molecular structure, DEs effectively change in capacitance when strained. Thus an electric field must be applied to the device during strain and extracted after the strain has been removed in order to achieve net energy output. We discuss the intricacies of this process in Section 6.3.1.

6.2 Motivation

While DEs are a relatively new material, they have already proven to provide higher energy densities than other small form transducers capable of being mounted on wearable sensing systems. Their ability to be utilized for shoe energy scavenging has been demonstrated, however integration into existing medical systems has not yet been discussed. Meanwhile, the wireless health community continues to rely on traditional batteries for power. It is generally agreed upon that energy harvesting is an import and necessary challenge and opportunity in the wireless health domain [88].

Thus far, no combination of harvesters and sensors has been presented using a medical device. Independently, medical shoes and shoe energy harvesting have

become popular topics, however no papers present successful utilization of DEs for shoe energy harvesting while simultaneously sensing medical diagnostics.

Traditional techniques for shoe energy scavenging employ piezoelectric or electromagnetic technology. DEs promise to replace these technologies with superior energy density, low mechanical complexity, and low cost [87] [89]. This increased energy density enables the use of DEs for a self-sustaining shoe, eliminating battery replacement and wall charging altogether by allowing for near-continuous battery charging at each human step.

While DEs have and continue to demonstrate superb energy harvesting capabilities, the nature of the DE, as will be discussed in Section 6.3.1, requires precise timing information in order to operate at maximum capacity. When designing a self-sustaining shoe, not only is the placement of the harvesters paramount in maximal energy scavenging, but so is timing. A DE requires a charge to be applied at its maximal strain in order to be harvested for maximal energy gains after relaxation. Thus, we propose to use the sensing information in medical shoes to time the harvesters while simultaneously measuring medical diagnostics.

6.3 Preliminaries

6.3.1 Dielectric Elastomers

DEs are deformable polymer films built from a variety of materials, the most common of which include acrylics and silicones due to their high electric permittivity (ϵ_r) and elasticity. Their relatively high elastic energy density means they can store more energy when deformed for the same amount of material, yielding less bulky and more productive transducers [87].

When a DE is strained, the internal structure of the material changes in

capacitance following Equation 6.1, where λ is the area expansion factor. This change in capacitance allows for net energy harvesting outlined in Figure 6.1.

$$C_\lambda = \frac{\epsilon_0 \epsilon_r \lambda^2 A}{d/\lambda^2} \tag{6.1}$$

Thus, employing DEs in a shoe energy harvesting system is not as simple as placing them at the highest pressure points along the sole. Equally important is the charge timing of the DEs. While they are superb harvesters, they are most easily applied in environments with constant frequency pulsations, such as the vibrations found in machinery, bridges and buildings, where charge timing is easily predicted. While the human gait is fairly constant, it has a broad spectrum. Thus, charging and extracting energy from a shoe-mounted DE requires precise timing prediction to determine when it will reach its maximum pressure during ambulation.

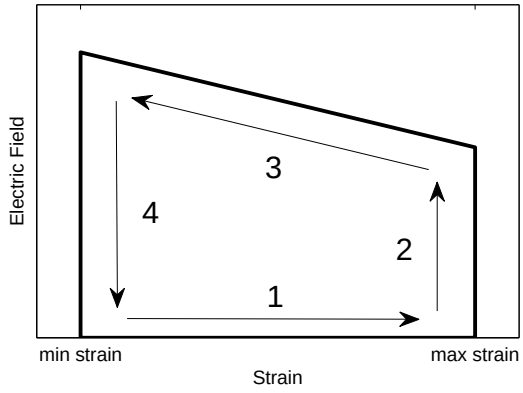


Figure 6.1: Typical DE energy harvesting cycle. (1) Stress is applied to the DE increasing its strain. (2) The loading charge is applied at the maximal achieved strain, thus creating an electric field across the DE. (3) The stress is removed from the DE and its strain is reduced. (4) The electrical energy is harvested from the DE; the energy is equivalent to the area inside the cycle. The slope of step 3 is a result of the fact that this model assumes a constant charge through relaxation.

6.3.2 Harvester Simulation

We apply the detailed model derived by Jean-Mistral et al. to compute the harvested energy per step on our medical shoe system [90]. We simulate a 9.5mm $\times$ 9.5mm VHB4910 acrylic DE manufactured by 3M with a single 1mm thick DE layer [91]. The energy generated by the dielectric elastomer is calculated using Equation 6.2. Ultimately we take into account electrical energy losses and mechanical losses and compute the harvested energy via Equation 6.3.

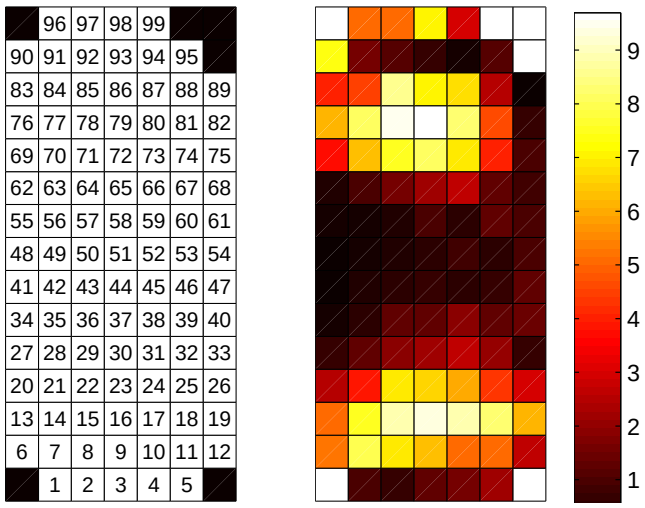
$$\Delta E_{DEG} = \frac{Q^2}{2C_1}\left(\frac{1}{\lambda_{final}^4} - \frac{1}{\lambda_{init}^4}\right) \tag{6.2}$$

$$E_{harv} = \Delta E_{DEG} - E_{loss_{elec}} - E_{loss_{mech}} \tag{6.3}$$

6.3.3 Medical Shoe

Our medical shoe consists of ninety-nine pressure sensors distributed about the sole of the foot, a processing unit, flash memory, a radio, and an ADC. The passive resistive pressure sensors are located along the sole according to the Pedar plantar mapping [60] and numbered in Figure 6.2a.

In Chapter 5 we demonstrated that gait measurements sufficient for medical diagnosis can be predicted by employing just a handful of sensors from the original array of ninety-nine. A single global sensor is all that is necessary to capture the start and end times of a human step. In this paper we also show that the single global sensor is adequate to predict optimal temporal assignment of harvesters for maximal energy harvesting.



(a) (b)

Figure 6.2: Average optimal energy output (mJ) per step at each harvester location. The applied load charge is the same at each harvester but applied at the optimal timing specific to each harvester while remaining constant across steps.

6.4 Spatiotemporal Harvester Assignment

The fine level of detail in pressure distribution provided by the medical shoe helps determine which locations are most optimal for energy harvesting. However, harvester placement not only requires spatial arrangement but also temporal assignment due to the nature of the DE energy harvesting cycle. While Figure 6.2b shows that harvesters 16, 78, and 79 have the highest potential energy, if the maximum strain timing cannot be predicted with sufficient accuracy, then the energy produced by these harvesters will be suboptimal.

Thus, we provide a solution for both spatial placement and temporal assignment of DE harvesters on the medical shoe. We accomplish this via energy profile prediction using the global sensor. As previously explained in Section 6.3.3, the global sensor is tasked with measuring the start and end times of human steps. We examine the relationship between global sensor-samples and

potential harvester-samples and generate models of each harvester energy profile given a global sensor-sample value relative to the start of the step. We compute regression models between subsequent global sensor-samples and measure their ability to predict harvester energy profiles. We choose a minimal set of global sensor-samples for prediction that maximize harvested energy. This ultimately comprises a complete mapping of global sensor-sample to harvester energy profiles and ultimately determines both the spatial placement of harvesters and their temporal assignment as determined by the best global sensor-sample predictors.

This procedure can be executed before or after sensor placement. If performed after, the harvesters are chosen from the remaining lots on the sole of the shoe. If performed before, upon choosing the physical placement of a subset of harvesters, the remaining vacant lots can be filled with medical sensors. If, however, cost is an important factor in design, the number of sensors can still be reduced using the existing sensor selection techniques mentioned previously.

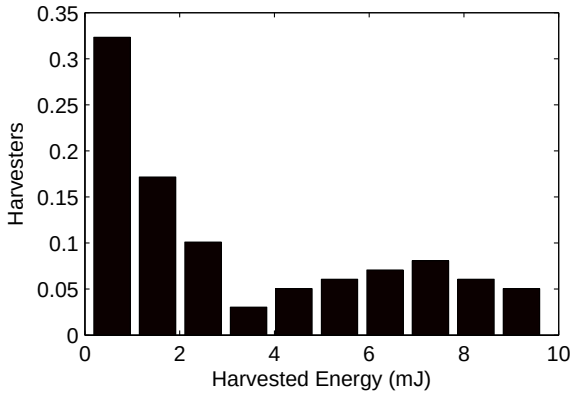


Figure 6.3: Distribution of average optimal energy output per step across all potential harvester locations on the pedar mapping assuming the load charge is applied at the optimal timing specific to each harvester.

6.5 Results

Figure 6.2b depicts the distribution of average energy gains across the foot given that the charge is applied at the optimal time per harvester. Figure 6.3 shows that about 40% of harvesters are capable of producing an average of 3mJ or more per step.

We find that the global sensor-samples have significant correlation to the top harvester energy profiles depicted in Figure 6.4 and 6.5. After training the global sensor-sample predictors for harvester energy profile prediction, we choose a covering set of global sensor-samples that predict a number of harvesters for maximal energy gains.

The best global sensor-sample predictors for each of the potential harvesters are able to harvest 93% of the harvesters within 72% to 97% of their maximum potential. Even if the timing prediction is off by one to a few samples, 90% of steps are harvested within 1mJ of optimum for the top harvester, as depicted in Figure 6.6.

Figure 6.7c depicts results from our global sensor-sample energy profile prediction algorithm along with previous results for sensor selection for medical diagnostics. Upon assigning the sensors from either Figure 6.7a or 6.7b, covering the remainder of the sole with DE harvesters and predicted using the global sensor easily powers the entire medical shoe at a human ambulation rate of 2Hz.

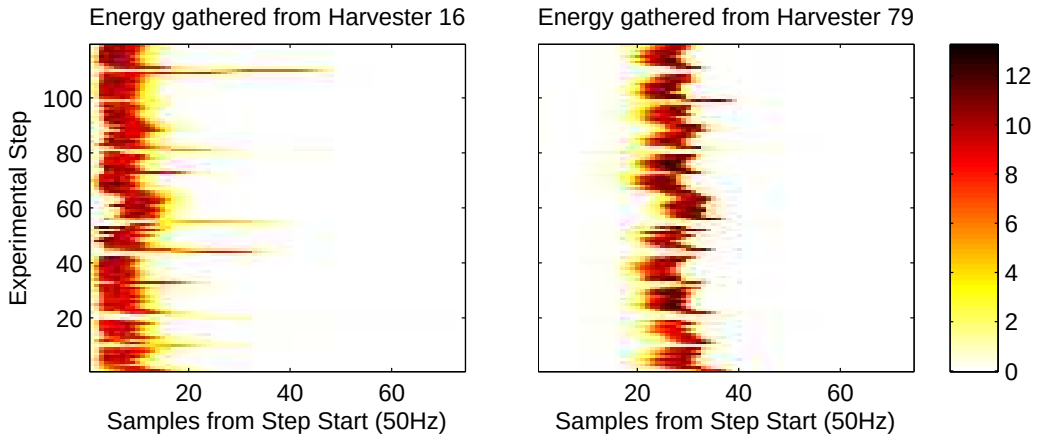


Figure 6.4: Potential energy harvesting points for harvesters 16 and 79. Assuming the top three harvesters are installed and timed optimally, and the average ambulation frequency is 2Hz, each shoe would produce about 34mW.

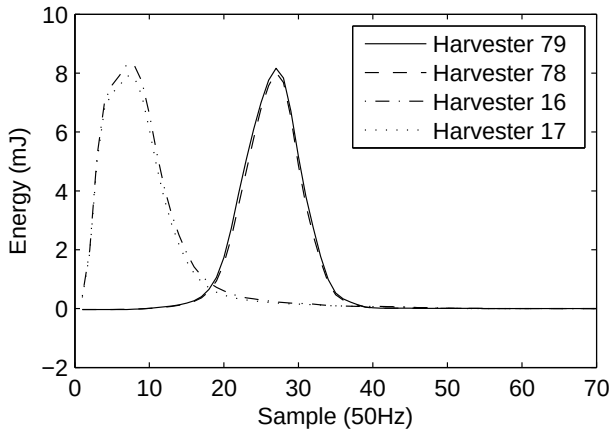


Figure 6.5: Top harvester average energy distributions given that the charge is applied to the DE at the specified sample time and the energy is harvested at the step end.

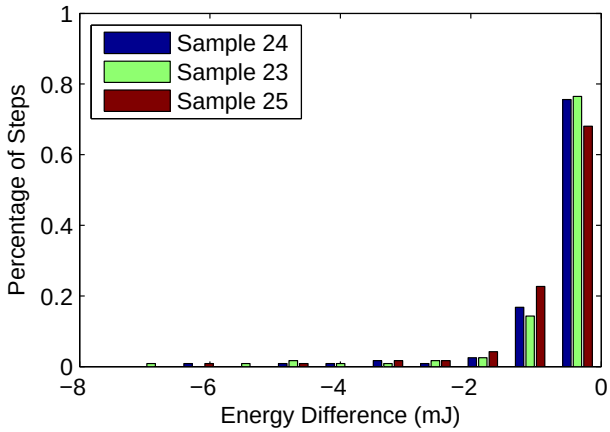
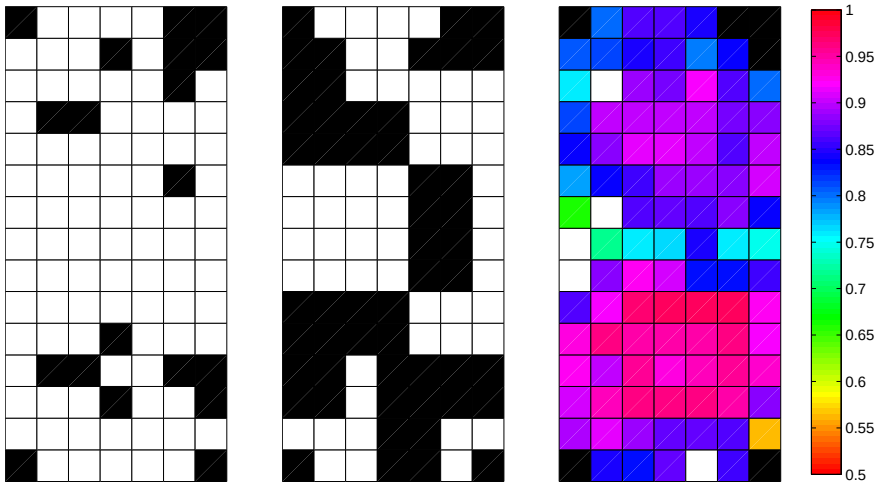


Figure 6.6: Distributions of the difference in harvested energy from the optimal potential harvested energy using the labeled global sensor predictors on harvester 79. For about 90% of all steps, each of the top three sensor-sample predictors are able to harvest within 1mJ of optimum.



(a) (b) (c)

Figure 6.7: (a, b) Minimum subset of sensors capable of measuring gait characteristics necessary for medical diagnosis found in Chapter 5. (c) Average harvested energy as a percentage of the optimal energy in Figure 6.2b. Energy is harvested when the predicted energy profile of each harvester (as predicted by the global sensor) is at maximum.

6.6 Summary

With the continuing development of IoT wearables, and more specifically, medical body sensing devices, it has become imperative that we develop new design techniques that acknowledge energy as a premier design consideration. While much attention has been given to methods for reducing power demands of wireless wearable sensing devices, incorporating the environment for power generation is still unsolved. Thus, we have presented a method for application of energy harvesters to a wireless sensing device.

Specifically, we have presented a new procedure for spatiotemporal assignment of dielectric elastomer energy harvesters for a self-sustaining medical shoe. Our approach utilizes the global sensor to predict the energy profiles of harvesters along the sole of the shoe and subsequently install harvesters in those locations and, in real-time, predict the optimal timing of the various stages of the DE energy harvesting cycle. Our technique is capable of harvesting enough energy to power the medical shoe at an ambulation rate of 2Hz.

CHAPTER 7

Hardware Obfuscation for Intellectual Property Protection and Trusted Remote Sensing

Most likely the most demanding of requirements for the widespread realization of many IoT visions is security. IoT security has an exceptionally wide scope in at least four dimensions. In terms of security scope it includes rarely addressed tasks such as trusted sensing, computation, communication, privacy, and digital forgetting. It also asks for new and better techniques for the protection of hardware, software, and data that considers the possibility of physical access to IoT devices. Sensors and actuators are common components of IoT devices and pose several unique security challenges including the integrity of physical signals and actuating events. Finally, during processing of collected data, one can envision many semantic attacks.

The state of the art in reverse engineering is so advanced that industrial chips with a billion transistors can be reverse engineered in a matter of a few weeks. There are even a number of dedicated companies that on a regular basis reverse engineer new industrial integrated circuits (ICs). Hardware obfuscation is an approach that aims to prevent the possibility of reverse engineering ICs. Recently, several conceptually new and interesting approaches in hardware obfuscation have been proposed and demonstrated. Our goal is to advance the state of the art in hardware obfuscation by presenting a technique in which reverse engineering does

not only require that each transistor be fully characterized in terms of its delay, but also enables configurable obfuscation in the sense that each integrated circuit for a given design is obfuscated in a unique way.

The starting point for our approach is the use of physical unclonable functions (PUFs) for our conceptually new task. A PUF is a hardware device that has a complex but definite mapping from inputs to outputs that is practically impossible to reverse engineer. Furthermore, the device cannot be physically cloned. Currently, most PUF designs achieve unclonability and complexity by exploiting silicon manufacturing variability that manifests as variations in circuit element properties, such as delay and leakage energy. The complete functionality of the PUF is only known if the entire input-output (challenge-response) table is enumerated or each gate's internal properties are fully characterized.

We have developed two techniques that exploit the inherent randomness of the PUF in order to obfuscate a circuit. The first technique obfuscates an arbitrary piece of random logic by replacing it entirely. Specifically, we implement the pertinent piece of logic using a physical unclonable function in conjunction with a supporting small piece of configurable fabric, such as a field-programmable gate array. The purpose of the PUF is to obfuscate the circuit, since its functionality is known only to the designer and trusted manufacturer, while the purpose of the programmable fabric is to implement the original piece of replaced logic using the PUF whose characterization is only known after fabrication.

The second technique hides circuit functionality by obfuscating signal paths in the interconnect network. Pairs of wires are merged together such that an attacker cannot determine which signal propagates along which path. In this scenario, the PUF is used to control these junctions in such a way that the original circuit functionality is simultaneously preserved and obfuscated. We

employ these two techniques in such a way that delay constraints are satisfied while security is maximized for a user specified area and energy overhead.

The most important specification in security analysis is the specification of attacks and analysis of how specific techniques are resilient against these types of attacks. In our analysis we consider two extremely powerful attacks. In both attacks we assume the adversary has complete knowledge of the circuit netlist, but does not have access to individual gate properties such as dopant concentrations and channel lengths. The first attack assumes that an adversary can simultaneously observe all flip-flops in any clock cycle. The second attack is even more powerful and allows that the adversary can not only simultaneously observe all flip-flops but can also control every flip-flop in the design. We develop heuristics that minimize the effectiveness of these two attacks along with simulation based techniques that quantify the attack time effort required in order to break our security. We apply our techniques to a number of ISCAS'89 and ITC'99 benchmarks and successfully obfuscate their functionality with an overhead of up to 10% in area.

It is very important to note that our techniques for configurable obfuscation can also be used for many other security tasks. For example, when using configurable obfuscation it is much more difficult for an attacker to place an unnoticeable Trojan horse in the circuit since he is not aware of the circuit's complete design. Also, the PUF can be used as watermarking information for each and every IC produced of a particular design.

The remainder of this paper is organized as follows. First, we survey recent research and developments in the field of hardware security. Second, we provide a brief overview of the standard delay-based PUF model which we incorporate in our techniques. Next, we describe in detail our PUF-based logic architecture and

signal path obfuscation techniques. We describe two types of attacks and present heuristics for configuration of both of our new architectures for reverse engineering prevention. And finally, we analyze our techniques in terms of security and overhead.

7.1 Related Work

While there have been a number of efforts to produce systems in many technologies which are equivalent to PUFs, process variation has enabled the creation of practical and low cost PUFs. An MIT group developed the concept and a large number of silicon prototypes in several technologies which demonstrated their advantages and limitations [92]. PUFs have been used in a number of applications including device authentication, secrete key generation, anti-counterfeiting, key distribution, and secure key storage [93] [94]. Recently, the emergence of the digital PUF has enabled its direct integration into digital logic in both ASIC [95] [96] and FPGA systems [97] [98] [99] [100]. The use of emerging nanotechnologies for PUFs has introduced an entirely new security dimension which has yet to be implemented in traditional designs [101] [102]. There are several excellent surveys on the history and state of the art of PUFs [103] [104].

In the last quarter century numerous techniques for reverse engineering and layout reconstruction on silicon chips have been proposed and demonstrated [105] [106] [107] [108]. For example, both in academia and in industry, state of the art processors, like Intel's general purpose processors, have been reverse engineered. Two excellent summaries on the state of the art in reverse engineering have been presented by Chipworks [109] [110]. Other recent efforts in reverse engineering apply logic synthesis and formula verification techniques to functionally reconstruct significant percentages of logic circuitry [111] [112]. Other interesting and impor-

tant reverse engineering efforts include imaging-based and side channel attacks [113] [114].

Hardware obfuscation is a task that aims to prevent IC reverse engineering. It can be divided into two broad groups. In the first, unique structures are added to ICs in such a way that only the designer of the circuit can enable and disable correct functional execution [115] [116]. There have also been efforts to obfuscate ICs in such a way that the physical structure of gates is difficult to deduce during reverse engineering because two or more types of gates differ in only ultra small details of implementation that cannot be easily deduced using state of the art reverse engineering techniques [117]. An NYU Poly research group has demonstrated the effectiveness of these techniques and quantified the overhead in terms of important design metrics [118]. Finally, in the late 90s there were a number of efforts to enable a designer's signature to be permanently written to hardware as proof of intellectual ownership [119].

Our work is a conceptually new application of PUFs for hardware obfuscation. While hardware obfuscation enabling and disabling techniques have been previously accomplished using PUFs, this is the first time that the PUF actually implements random logic in such a way that the functionality of the circuit is completely hidden. This new proposed technique is different from all previously proposed hardware obfuscation techniques because it reduces reverse engineering not just to the problem of identifying which gates are connected in which way, but also requires that the number of dopants in each transistor is accurately recovered (i.e. delay characterization), which is well beyond the feasibility of current reverse engineering attacks. Another important difference between state of the art obfuscation techniques and our new approach is that in each instance of an integrated circuit our approach results in a different type of obfuscation.

Therefore, even if the attacker is successful in reverse engineering a particular chip he cannot validate the functional correctness of other fabricated chips that are obfuscated.

7.2 Standard PUF Overview

Our obfuscation techniques employ the standard delay-based arbiter PUF originally designed by Suh et al. at MIT [93]. The phenomenon that enables its security and unclonability is process variation. Inherent randomness in manufacturing processes manifest as deviations in gate characteristics from nominal specifications even among designs fabricated on the same die. More specifically, each gate in the standard PUF is ultimately fabricated with different delays and cannot be controlled beyond a certain level of granularity.

The standard delay-based arbiter PUF takes advantage of these delay deviations using the architecture depicted in Figure 7.1. A challenge input of n bits, represented by the vector $\mathbf{C} = [C_0, C_1, \ldots, C_{n-1}]$ is applied, and a rising edge is sent through the PUF. The rising edge is split into two at the first junction of multiplexors. The path of each rising edge will then switch positions (top or bottom) depending on the challenge vector bit. In this example, a challenge bit of 1 will swap the paths, while a value of 0 will keep the paths propagating along their current line. The first path to arrive at the arbiter determines the output response, R, of the PUF.

For the purposes of circuit obfuscation, the functionality of the PUF must be characterized post fabrication and only by the designer or trusted manufacturer. Immediately following characterization by the trusted authority we remove the ability for further direct characterization attempts. This can be done by either

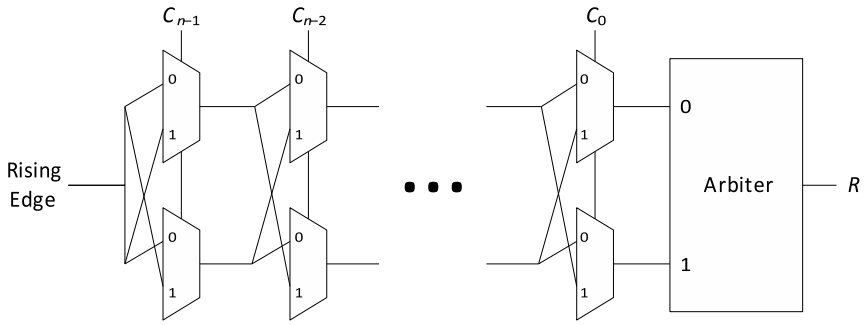


Figure 7.1: Standard delay-based arbiter PUF [93].

laser burning access wires or burning supporting fuses [120] [121].

Note that while we use the delay-based arbiter PUF because it is considered to be the standard PUF, our techniques are compatible with the majority of IC-based PUF designs.

It is known that the standard delay-based arbiter PUF can be susceptible to operating conditions (e.g. temperature variations, voltage fluctuations). However, this susceptibility is limited to specific challenge vectors whose corresponding PUF delay paths differ by extremely small amounts. Thus, fluctuations in operating conditions can change the outcome of the PUF response for these input vectors. We discuss how we handle these stability issues in Section 7.3.2.

7.3 Arbitrary Logic Replacement

Our first technique obfuscates an arbitrary circuit by replacing it with PUF-based logic. PUF-based logic directly implements the original functionality of the replaced circuit while also hiding its functionality. We explore two potential architectures, as depicted in Figure 7.2. Since the functionality of the PUF is a byproduct of manufacturing processes, its characterization is not known until after fabrication. Once fabricated, the designer characterizes the PUF through

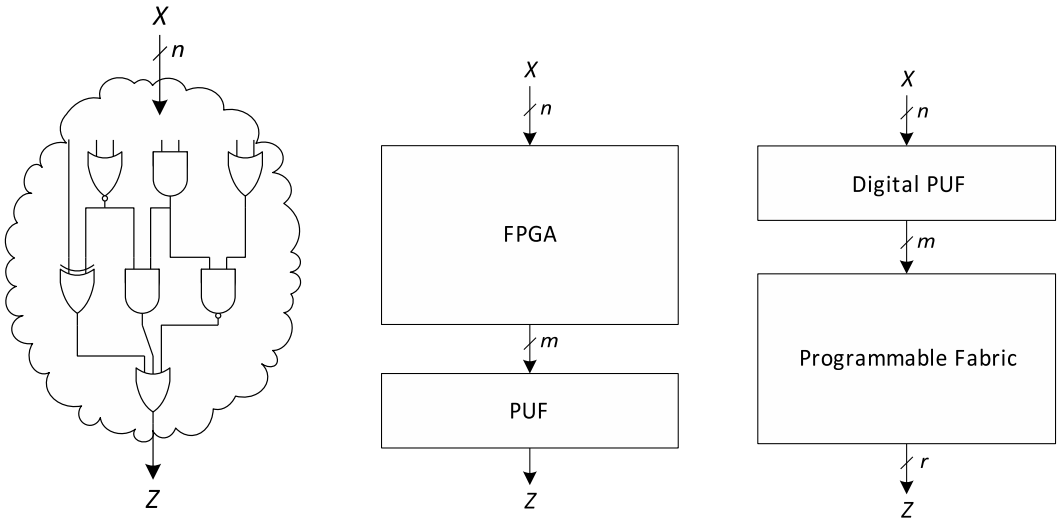


Figure 7.2: Architectures implementing the same functionality. (a) Arbitrary circuitry. (b) PUF-based logic using a preceding FPGA. (c) PUF-based logic using a preceding PUF.

special supporting hardware, then burns the pertinent access wires to disallow any subsequent unauthorized characterization. Now that the PUF's unique functionality is known only to the designer, the FPGA fabric is programmed using standard synthesis tools in order to implement the original replaced circuitry in conjunction with the corresponding PUF.

7.3.1 Programmable Fabric Configuration

Since each fabricated PUF is unique and unclonable, the supporting FPGA in the PUF-based logic architecture enables the replication of an arbitrary function. Once the PUF on a pertinent circuit has been characterized, the supporting FPGA fabric is configured accordingly given the known PUF mapping using standard synthesis tools.

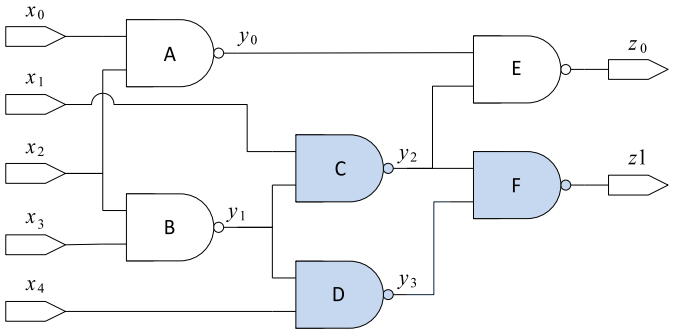
Figure 7.3 contains a motivational example in which we hide the functionality

of the c17 circuit from the ISCAS'85 benchmark suite [38] by obfuscating a portion of the circuitry. The shaded portion in Figure 7.3a is the logic to be replaced and obfuscated. The resulting obfuscated architecture is depicted in Figure 7.3b.

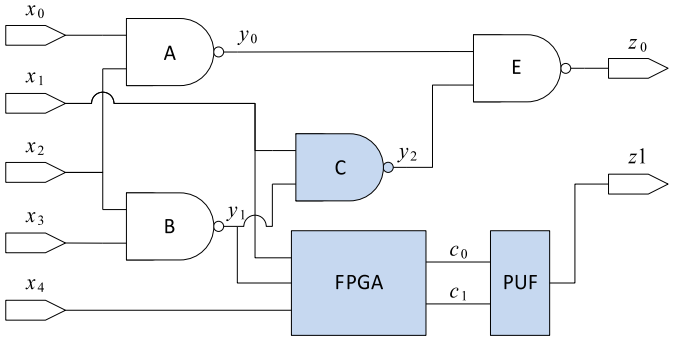
Note that gates D and F are eliminated entirely while gate C remains despite participating in the computation of z_1. This is because gates D and F only affect z_1, while the output of gate C (y_2) is also required for the computation of z_0. Thus, we cannot eliminate C entirely from the circuit. Note that while retaining gate C does increase overhead, it does not compromise security in any way since an attacker cannot know with any certainty which remaining gates in the obfuscated circuit are included in the PUF-based logic cell, if any are at all.

We present an example challenge-response table for a small PUF in 7.3c. In this example, we assume that the challenge $\mathbf{C} = [1, 0]$ is unstable. As such, we program the FPGA so that it will not produce this output.

The FPGA is synthesized such that for a particular input vector, $[x_4, y_1, x_3]$, it produces an input to the PUF that maps to the original correct output at z_1. One such specification is depicted in 7.3d. In this small example, the FPGA produces three potential output vectors which ultimately map z_1 to either 0 or 1. Without knowledge of the internal characteristics of the PUF, the mapping cannot be deduced. Of course, since this is a very small example, a brute force attack could easily enumerate all possible inputs to the PUF-based logic cell, however, we later demonstrate that this attack is infeasible for larger designs. Security can be further improved by randomizing the challenges outputted by the FPGA, especially for larger input spaces.



(a) Original circuit



(b) Obfuscated circuit

C_1	C_0	R
0	0	1
0	1	0
1	0	-
1	1	1

(c) PUF switching table

$$c_1 = y_1'$$
$$c_0 = y_1 x_1' x_4'$$

(d) FPGA implementation

Figure 7.3: Motivational example of PUF-based logic replacing a portion of (a) the c17 circuit from the ISCAS'85 benchmark suite [38]. (b) Obfuscated circuit. (c) Example characterized PUF switching table. (d) FPGA implementation enabling the replaced circuit functionality in conjunction with the PUF.

7.3.2 Stabilizing the Standard PUF

As previously mentioned, it has been observed that for some input vectors it is possible that a standard delay-based arbiter PUF produces unstable outputs. Specifically, there may exist challenge vectors which could potentially produce different responses depending on operating conditions such as temperature variations and voltage fluctuations. Hence, when employing the standard PUF, we specifically choose to use the architecture in which the FPGA precedes the PUF, as depicted in Figure 7.2b. In this way, we can configure the FPGA to eliminate potential unstable inputs.

While the architecture in Figure 7.2c is not ideal for the standard delay-based arbiter PUF due to the issue of unstable inputs, if we employ a PUF that is stable for all inputs, then this architecture is a viable option and potentially preferable. This architecture provides an additional layer of security that is inherent in the design. By placing the FPGA after the PUF, the PUF's output is not directly known and thus it is more difficult for an adversary to attack without reverse engineering the FPGA inputs from its output first.

7.4 Signal Path Obfuscation

Our second obfuscation technique utilizes PUFs to obfuscate circuit functionality by directly obfuscating the interconnect network. Specifically, we combine pairs of wires in such a way that their paths are unknown. Figure 7.4 depicts our architecture for signal path obfuscation. Similar to the PUF-based logic case, after fabrication the pertinent PUFs are characterized and the inputs to each PUF are set such that they swap their corresponding wires according to the original circuit functionality.

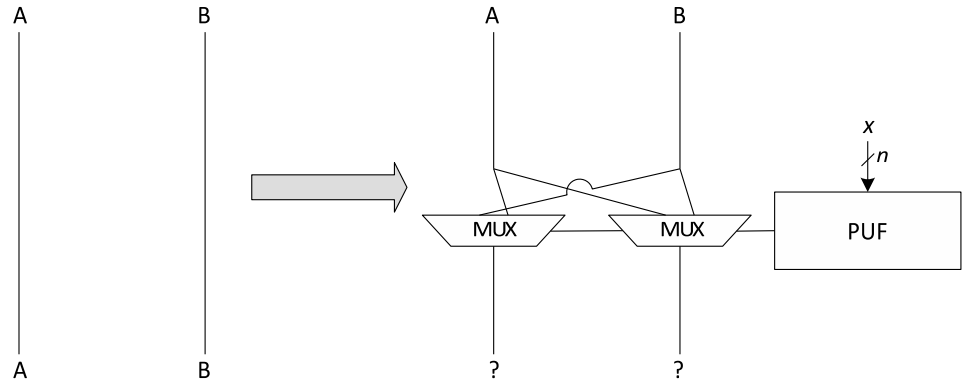


Figure 7.4: Signal path obfuscation architecture for wire swapping. The input X can only be set correctly by the designer who knows the functionality of the PUF.

This technique becomes very powerful when many wire swapping components are placed throughout the circuit in such a way that storage elements are affected by many potential swaps. For example, if the input to a single storage element is affected by k swapping components, then there exist 2^k possible configurations from which an attacker must determine the correct configuration in order to deduce the correct functionality of the circuit.

This obfuscation architecture should be placed such that the most number of flip-flops are affected the most number of times. We also take into consideration the additional delay overhead that comes with this architecture, and place these wire swapping components between gates with positive slack, so as not to affect the critical path.

It is important to note that our signal path obfuscation and PUF-based logic techniques are orthogonal and can be applied to an arbitrary circuit simultaneously.

7.5 Attacks

We assume there exist two types of attacks on the PUF-based logic and signal path obfuscation techniques. In both attacks, we assume an adversary has complete knowledge of the design of the circuit, including the functionality of any pertinent programmable logic, but does not know the input-output mapping of any PUFs. In the first attack, we assume that the adversary has the powerful ability to both read and write to all flip-flops in the circuit. The second attack assumes that an adversary has the ability to read all flip-flops, but can only write to the primary inputs.

In the case of PUF-based logic, a successful attack is one in which all PUFs are characterized. In the first powerful attack, the security of our obfuscation technique relies solely on the attacker's ability to reverse engineer the PUF through application of a complete set of inputs. By reading each corresponding output, a complete characterization table can be built. Thus, secure obfuscation relies on the size of the PUF, since the total input space grows exponentially with its size. We can make the task more difficult by using the architecture from Figure 7.2c in which the PUF precedes the FPGA. In this scenario, the attacker will not have direct access to the PUF output, but must instead reverse engineer its output values from the FPGA output.

In the case of the second type of attack, an adversary must intelligently apply inputs at each clock cycle such that he can indirectly apply as many inputs as possible to the PUF and measure the corresponding outputs. This attack is more difficult since the attacker cannot directly control all flip-flops, but instead must attempt to indirectly control them through the primary inputs. In this scenario we can make the reverse engineering task very difficult by placing the PUF in locations that are difficult to control indirectly. We discuss the specifics of these

techniques in Section 7.6.1.

In the case of reverse engineering circuit functionality in the presence of signal path obfuscation, the attacker does not need to fully characterize all PUFs, but instead needs to determine the circuit-wide configuration of all wire swapping components (i.e. whether each individual wire pair is swapped or not). An attacker can test his configuration by applying a circuit input vector with known outputs and testing to see if the resultant outputs are as expected. However, even if the circuit outputs are as expected, the attacker has still not yet completely correctly reverse engineered the circuit since he may have only correctly solved those wire swapping components that received a 0 on one wire and a 1 on the other. For the cases in which both wires are the same value, the configuration is not yet confirmed. Furthermore, for some input vectors there is a possibility that particular wire values do not participate in the final output (i.e. don't-cares). For example, even if a wire swapping component swaps a 1 and a 0, the 1 might propagate to an AND gate containing another input whose value is 0. In this case, the swap had no affect on the final output.

Ultimately, an attacker will be forced to try a large portion, if not all, of the 2^k combinations of configurations, where k is the number of wire swappings affecting a single flip-flop. By testing all of these configurations on a single set of known inputs and outputs, most likely he will find a set of partially correct configurations. The set will only be partial due to the don't-cares and correlation scenarios mentioned above. However, with these known configurations, he can repeat the same steps using the next set of known inputs and outputs to slowly build a more complete configuration.

7.6 Techniques

In this section we present our two orthogonal techniques that both utilize PUFs to obfuscate digital logic. Both techniques enable intellectual property protection by obfuscating the data path of the circuit. The logic replacement technique, which directly replaces circuitry with an FPGA and PUF, enables a second application, trusted remote sensing, by providing a unique and integrated signature of the circuit.

7.6.1 Logic Replacement

For the powerful attack case in which an adversary has write access to all flip-flops, the security of the PUF-based logic relies on the size of the PUF to create an exponential input space. Additionally, if we use a stable PUF, we employ the architecture from Figure 7.2c, in which the programmable fabric follows the PUF, to further prevent attacks by disabling direct access to the PUF output.

We prevent the second type of attack, in which an adversary can read all flip-flops but can only write to primary inputs, by replacing portions of circuitry with PUF-based logic in such a way that it is difficult for the attacker to set the PUF's inputs and thus increase his knowledge of the PUF's functionality by reading the corresponding outputs. Our placement criteria to accomplish this are the following:

- Place PUF-based logic where it is *affected by* many flip-flops, specifically by flip-flops which cannot be set directly by the attacker.
- Place PUF-based logic where it *affects* many flip-flops which cannot be set directly by the attacker.

- Place PUF-based logic where its inputs are highly correlated, thus making it very difficult for the attacker to build a large input-output table.

Since there are an exponential number of possibilities for selecting subcircuits for replacement with PUF-based logic, we simplify the task by performing multiple breadth first searches emanating from the input wires of flip-flops and traversing backwards through the circuit exploring potential PUF-based logic placements and measuring their security properties and overhead. In this way, we reduce the search space and still find many configurations that are both low in overhead and high in security. We discuss our results in detail in Section 7.7.

7.6.2 Signal Path Obfuscation

For the case of signal path obfuscation, our techniques secure the obfuscated circuit functionality for both types of defined attacks. This is accomplished using the wire swapping architecture from Figure 7.4 and combining pairs of wires (i) that affect many flip-flops, (ii) that are affected by many flip-flops, (iii) whose inputs are correlated, and (iv) that are affected by previously assigned wire swapping components. By positioning wire swapping components between wires that affect and are affected by many flip-flops we ensure that wire swapping has a large and unpredictable impact on the circuit. By choosing pairs of wires that are highly correlated we ensure that reverse engineering remains difficult since if two wires consistently have similar values it is difficult to deduce whether or not they are being swapped. And finally, by placing wire swappings along paths in which previously assigned wire swapping components are installed we ensure exponential growth in the total number of possible configurations.

We iteratively assign wire swapping components between pairs of wires throughout the circuit according to a linear evenly weighted sum of these heuristics.

Specifically, we consider (i) the union of flip-flops affected by pairs of wires as a fraction of the total flip-flops in the circuit, (ii) the union of flip-flops affecting pairs of wires as a fraction of the total flip-flops in the circuit, (iii) the coefficient of determination, R^2, between pairs of wires, and (iv) the union of wire swapping components preceding and affecting pairs of wires. Furthermore, we only consider pairs of wires which are not on the critical path and have positive slack. Thus obfuscating circuity functionality without overhead in terms of overall circuit delay.

7.7 Analysis

We analyze our PUF-based logic and signal path obfuscation techniques in terms of security and overhead by applying them to the circuits in the ISCAS'89 and ITC'99 benchmark suites [37] [45].

Figure 7.5 depicts results from our signal path obfuscation techniques on six example circuits. Specifically, we show the distributions of flip-flops that are affected by the labeled number of wire swappings on the x-axis. In all cases we successfully obfuscate a majority percentage of flip-flops using a very large number of wire swappings. This in turn creates a hugely exponential configuration search space for an attacker to reverse engineer. Note that wire swappings are applied only to wire pairs which have positive slack, thus ensuring that our final obfuscated circuit has no delay overhead.

Figures 7.6, 7.7, 7.8, and 7.9 depict the many PUF-based logic configurations that our techniques enumerate in the exponential search space for the six example circuits. Our heuristics find arbitrary portions of circuitry for PUF-based logic replacement in which there are a large number of inputs, large number of affected

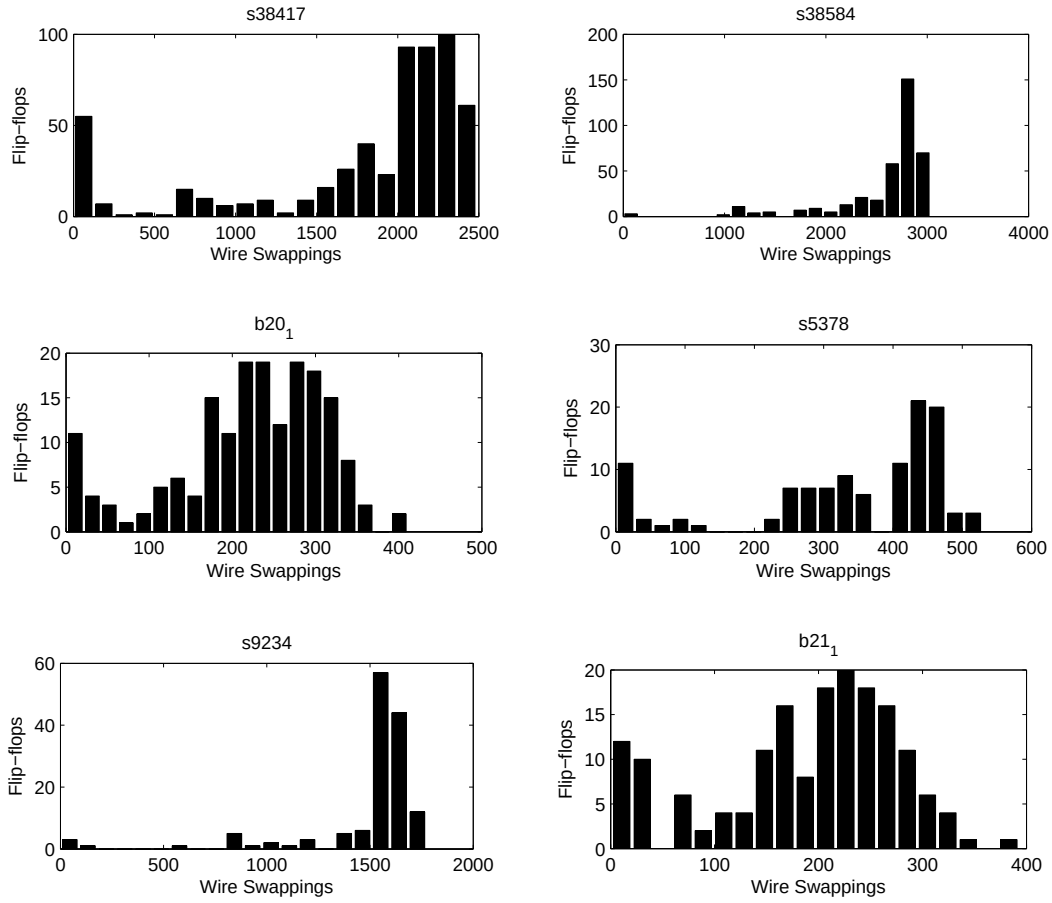


Figure 7.5: Total number of flip-flops affected by the labeled number of wire swapping components.

flip-flops, and are affected by a large number of flip-flops. These figures highlight the numerous PUF-based logic configurations that both satisfy our heuristics while simultaneously minimally impacting overhead.

Finally, we depict the security properties of our PUF-based logic obfuscation techniques in Figure 7.10. Each plot represents a single portion of arbitrary logic replaced with a single PUF-based logic architecture. We attack the resultant obfuscated circuitry by controlling the primary inputs and measuring the inputs and outputs at the pertinent PUF at each clock cycle. The figures depict the

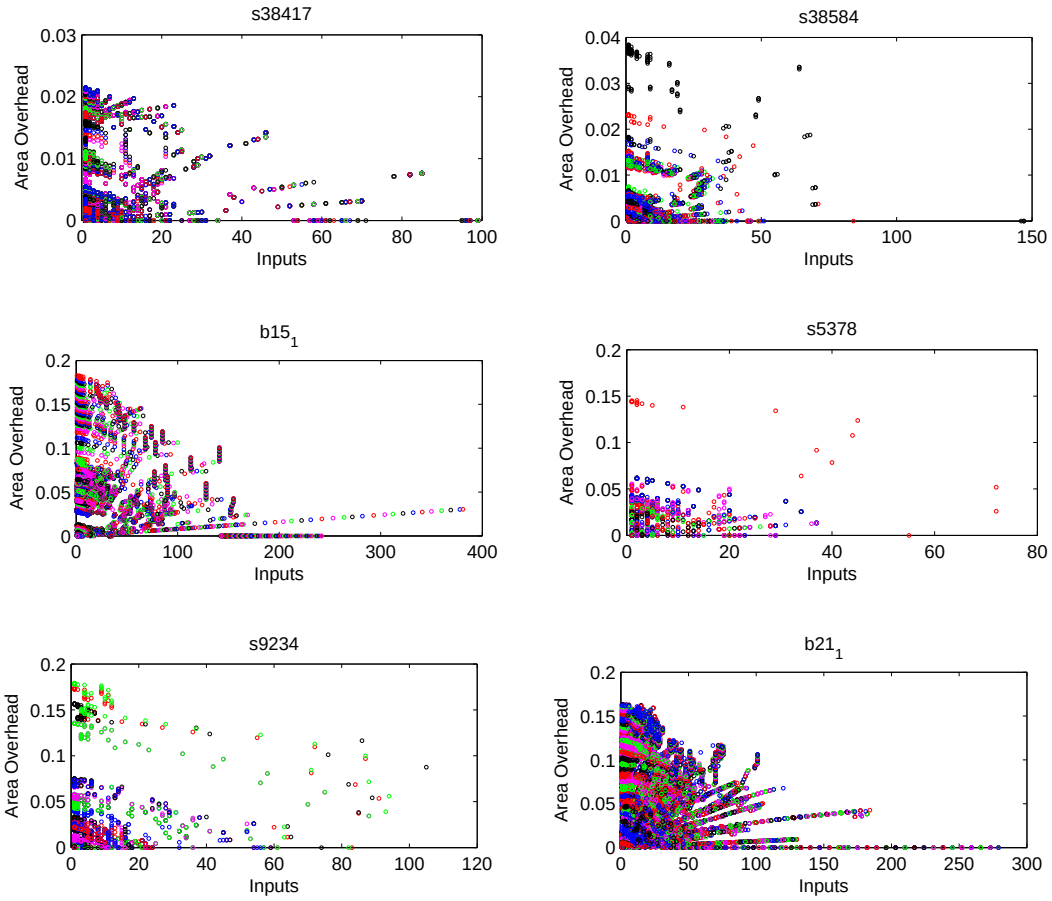


Figure 7.6: Area overhead upon replacement of circuitry using PUF-based logic with the labeled number of inputs. The different colors represent the flip-flops whose inputs come from the individual PUF-based logic component.

fractional number of correctly reverse engineered input-output mappings for the given PUF-based logic design as it relates to the pertinent PUF's number of inputs, placement in terms of depth in the circuit, number of affected flip-flops, and the number of flip-flops affecting the replaced logic. In each case, a linear increase in the corresponding heuristic causes an order of magnitude increase in difficulty of reverse engineering the circuit.

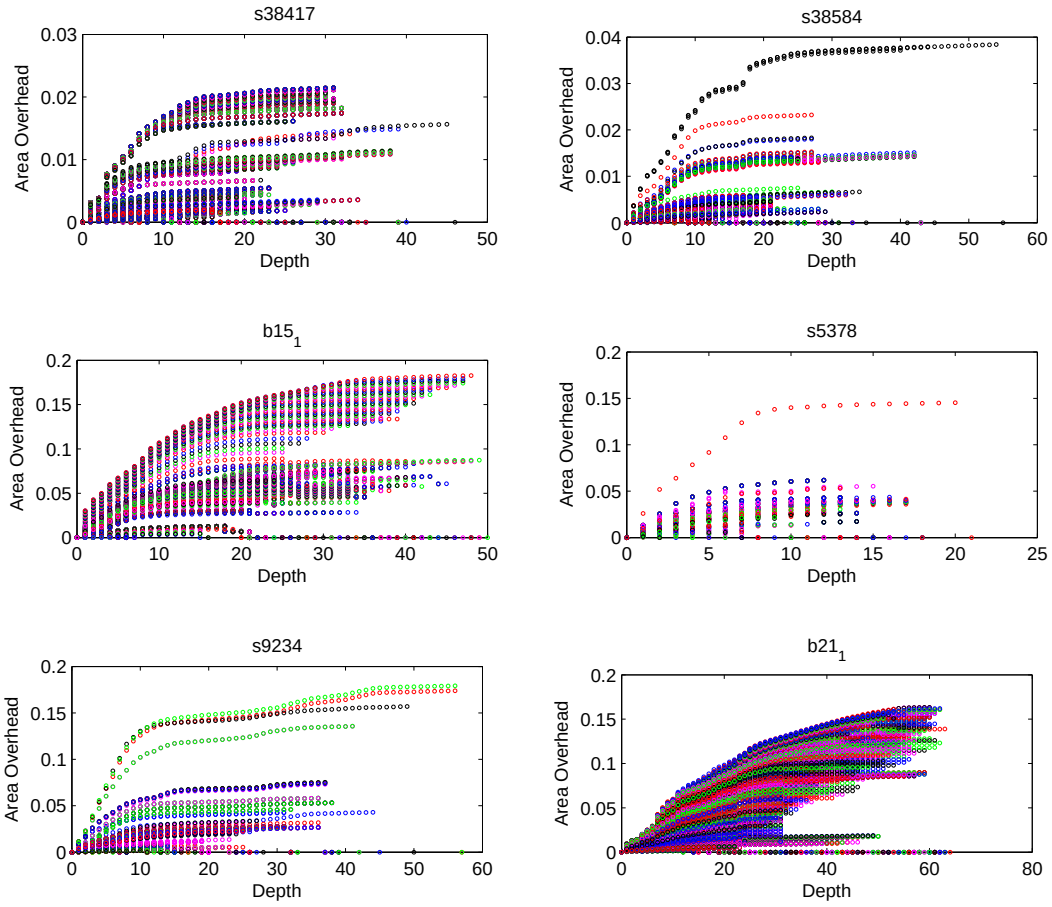


Figure 7.7: Area overhead upon replacement of circuitry using PUF-based logic with the labeled circuit depth. The different colors represent the flip-flops whose inputs come from the individual PUF-based logic component.

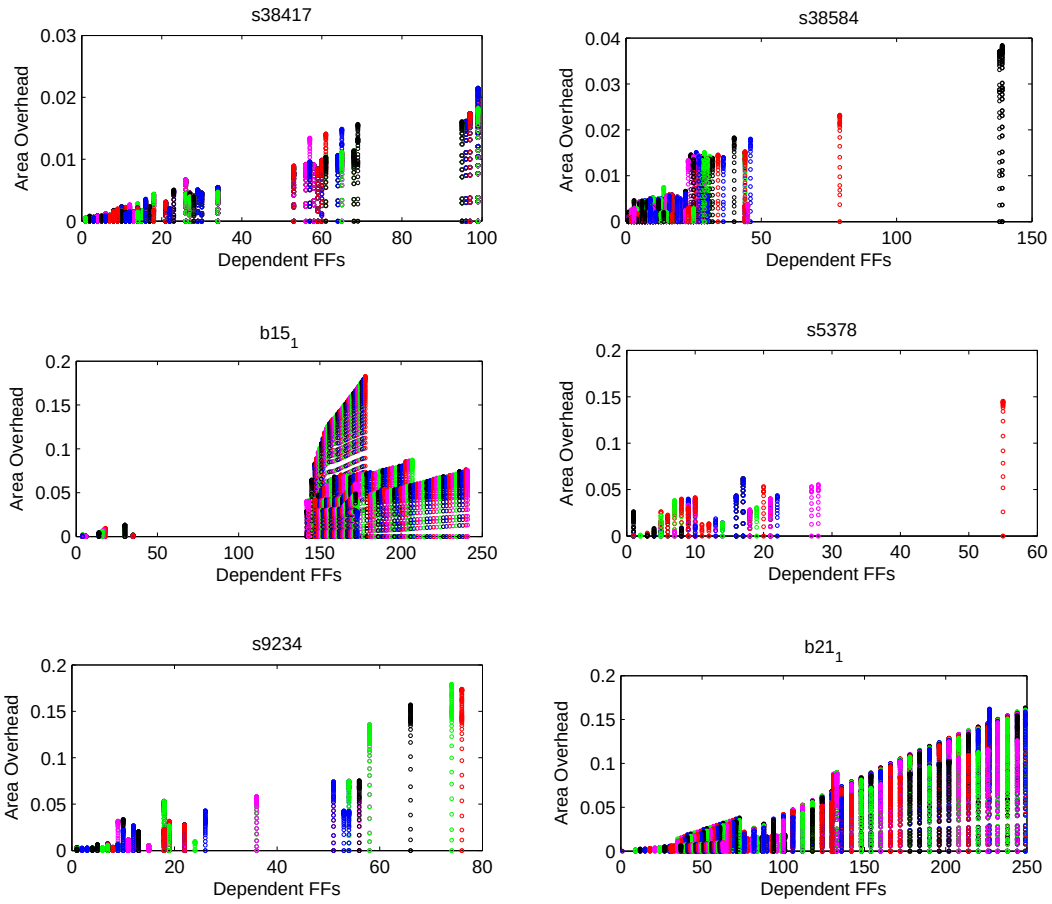


Figure 7.8: Area overhead upon replacement of circuitry using PUF-based logic that is affected by the labeled number of flip-flops. The different colors represent the flip-flops whose inputs come from the individual PUF-based logic component.

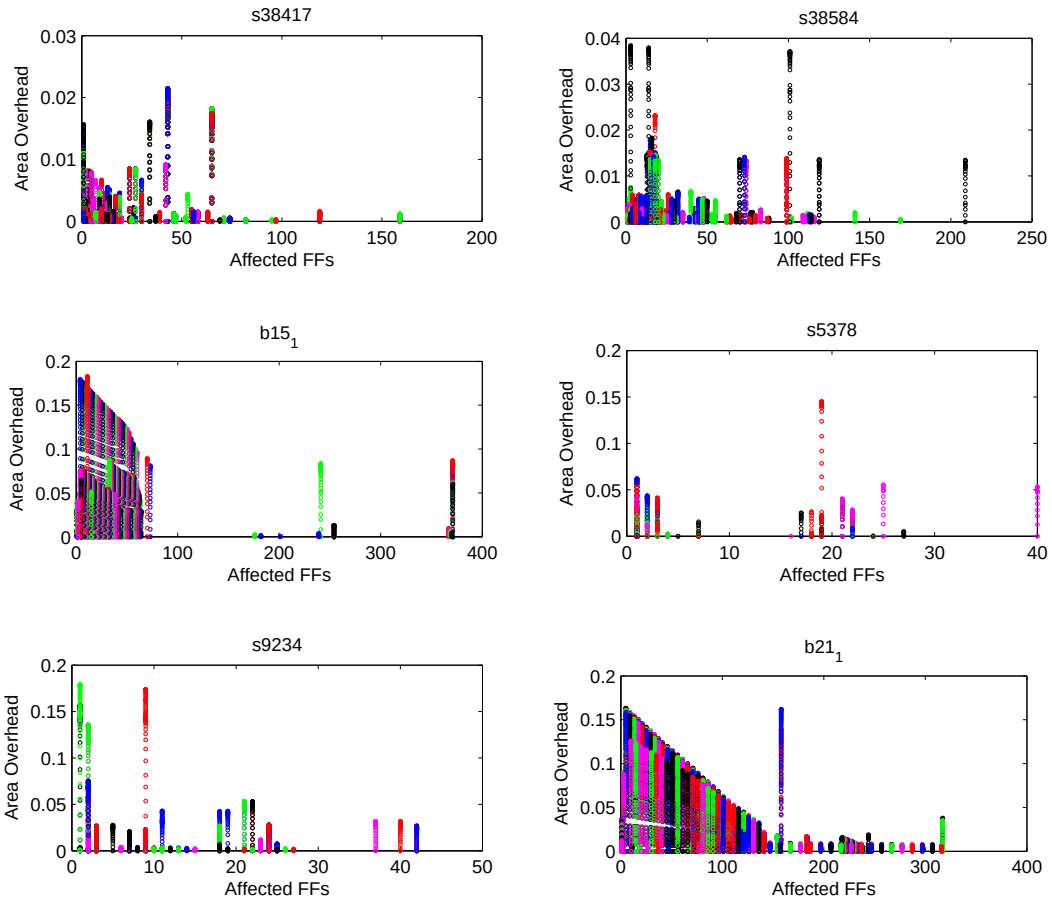
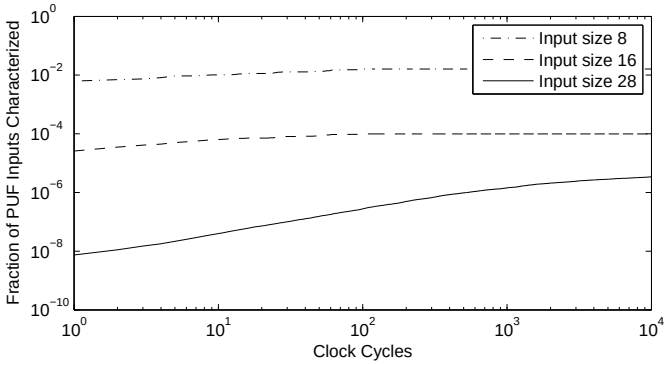
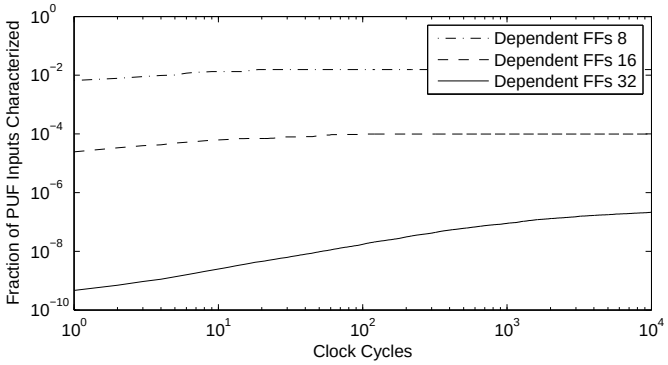


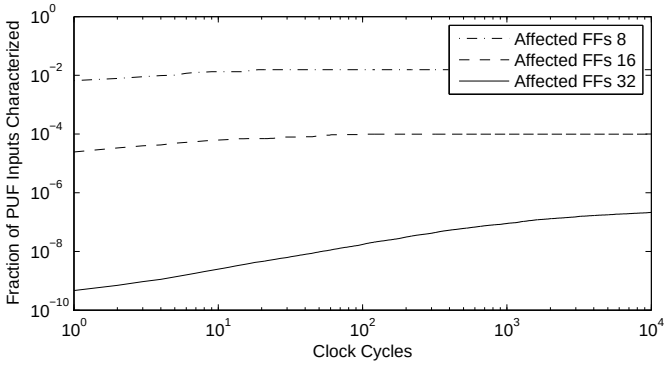
Figure 7.9: Area overhead upon replacement of circuitry using PUF-based logic that affects the labeled number of flip-flops. The different colors represent the flip-flops whose inputs come from the individual PUF-based logic component.



(a)



(b)



(c)

Figure 7.10: Fraction of PUF inputs characterized in reverse engineering an obfuscated b12 benchmark circuit.

7.8 Digital PUF Overview

The details of the architecture, initialization, and operation of the digital PUF are explained by Xu et al. [95]. It consists of a randomized LUT network which is initialized once, at power-up, by a standard delay-based PUF. The combination of these two architectures enables output randomness, satisfies the avalanche criteron, and ensures unclonability. Most importantly, the device is *digital*. Previous PUFs are analog in nature, relying on non-discrete circuit measurements, such as delay, for implementation, which are susceptible to environmental and operational conditions. Xu et al.'s digital PUF is a purely digital component once initialized and can therefore be inserted into any piece of combinational logic.

This particular characteristic enables that an even more general hardware obfuscation architecture can be applied to an arbitrary circuit. While use of the analog delay-based PUF for use in hardware obfuscation limited the placement of the obfuscated logic to be directly adjacent and connected to flip-flops, the digital PUF enables that we can place the hardware obfuscation logic anywhere. We discuss this in more detail in the remaining sections of this chapter.

Furthermore, the underlying LUT network of the digital PUF consumes energy on the order of magnitude of previous secret key cryptographic block ciphers, such as PRESENT [122], HIGHT [123], and AES [124], however enables implementation of public key protocols [97]. Additionally, the digital PUF is, as its name implies, unclonable and unique to the device, a characteristic that is not available in traditional hardware-based secret and public key systems, and is a required characteristic for our proposed applications.

7.9 Applications of the Digital PUF

In this section we present two applications of the digital PUF. The first application is logic obfuscation for intellectual property protection. In this application we replace a portion of logic within a circuit with a digital PUF and a supporting programmable fabric in order to hide the functionality of the circuit and thus eliminate the possibility of reverse engineering attacks. The second application is trusted remote sensing which enables that the base station of a sensor network can fully trust that the data collected and transmitted from each node is valid and not tampered with, and furthermore, that the node itself has not been tampered with.

7.9.1 Intellectual Property Protection

Reverse engineering attacks are so advanced nowadays that even integrated circuits containing on the order of 10^9 transistors can be reverse engineered in a matter of weeks. Techniques for hardware obfuscation attempt to prevent reverse engineering attacks by obscuring the functionality of a portion of logic within the circuit from an attacker while maintaining that the circuit performs its intended function.

Our previously described approach utilizes an analog PUF to accomplish this task. However, due to the constraints of the analog PUF, it is required that the hardware obfuscation logic be placed directly adjacent to circuit flip-flops. This can introduce additional overhead by requiring that the circuitry grow large enough to ensure secure obfuscation.

Through the use of the digital PUF we demonstrate that we can obfuscate the functionality of a circuit by obscuring any portion of any circuitry in such a

way that the original circuit functionality is maximally difficult to reverse engineer. Specifically, we combine the digital PUF with a programmable fabric that, together, implement the originally intended functionality of the original circuitry while its function remains unknown. What is most unique about our approach in comparison to previous obfuscation techniques is that our digital PUF is able to integrate directly into the circuit and actually directly obfuscate logic.

7.9.1.1 Architecture

We obfuscate a piece of arbitrary logic by completely replacing it with a digital PUF and a supporting programmable fabric using the architectures shown in Figure 7.11. Obfuscation is accomplished by connecting the original logic inputs as the challenge to the digital PUF. The configurable fabric is necessary since the actual function of the digital PUF is configured post-fabrication. This is done by first characterizing the supporting delay-based PUFs that determine the digital PUF's LUTs. Characterization of these initialization PUFs is carried out by enumerating and selecting among stable PUF inputs, then the digital PUF is configured following the procedure outlined by Xu et al. [95].

It is important to note that it is feasible to use only the standard delay-based PUF for logic obfuscation. However, given its limitations, it can only be applied using the architecture depicted in Figure 7.11a. This is due to two reasons. The first is because the standard delay-based PUF is unstable for some set of inputs. If the post-logic architecture from Figure 7.11b is used, it is possible that an input vector for which the analog PUF has an unstable output could arrive at the PUF in which case the obfuscated block would fail to produce the correct circuit functionality due to the PUF's instability. The second reason the analog PUF must be placed after the programmable fabric is because of its required arbiter

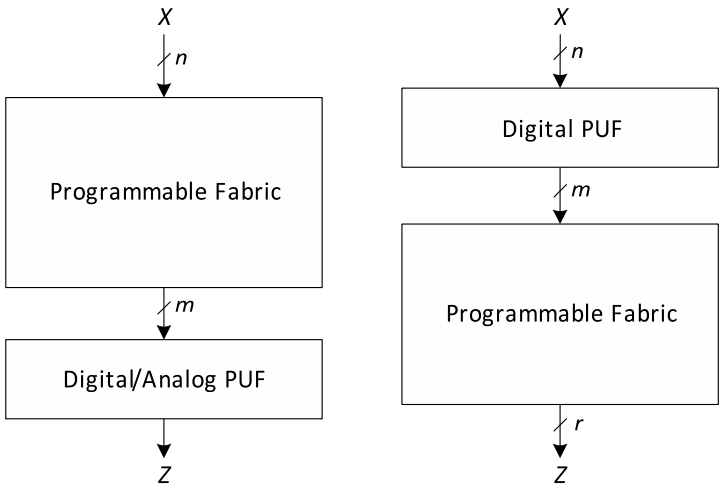


Figure 7.11: Hardware logic obfuscation architecture. (b) Pre-logic is required for the analog PUF to ensure input-output stability. (c) Post-logic is enabled through the use of the digital PUF since it is stable for all inputs.

which effectively acts as a flip-flop, thus ending the flow of logic for the given clock cycle. Assuming that an attacker can read this flip-flop and an attacker knows the structure of the configurable fabric, then his task is simplified to recording input-output pairs to build a representation of the PUF's functionality.

Since the digital PUF can be employed just like any other combinational component, it can be applied to the post-logic architecture depicted in Figure 7.11b. The biggest benefit of this architecture is that the PUF outputs cannot be measured directly as they can be in the pre-logic case when using an analog PUF. In the post-logic design, we can select the output wires of the obfuscated circuit in such a way that the remaining circuitry that the signals propagate through are difficult for an attacker to reverse engineer.

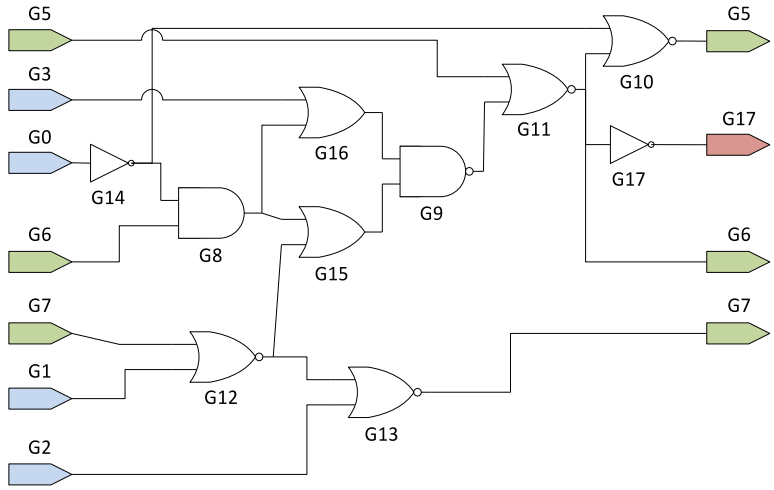
Note that utilizing the digital PUF enables us to place the PUF anywhere in the circuit without the need of flip-flops or arbiters. Also note that flip-flops in this case are not primary inputs or outputs and cannot be directly controlled.

Hence, we can obfuscate any connected subset of combinational circuitry anywhere in the design. In order to make the reverse engineering task difficult for an attacker, we select for replacement a portion of circuitry whose inputs that are difficult for the attacker to control as well as whose outputs are are difficult for the attacker to reverse engineer. We discuss the specifics of our heuristics in Section 7.9.1.3

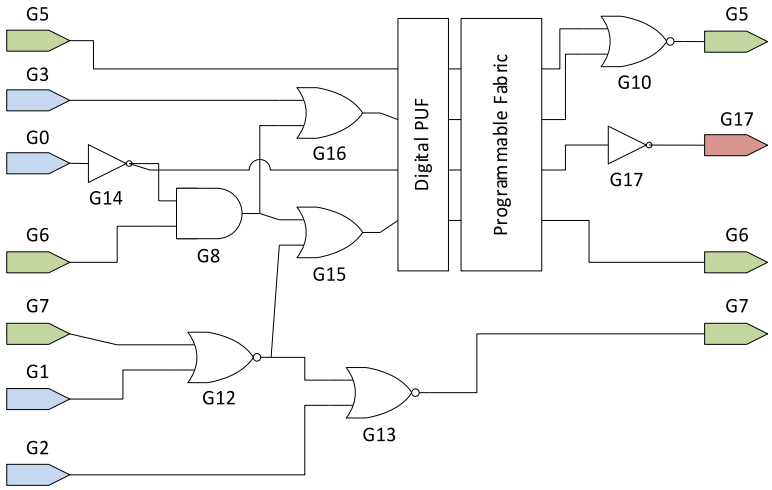
In Figure 7.12 we present a motivational example using the s27 circuit from the ISCAS'89 benchmark suite [37]. In this example we obfuscate the circuitry consisting of the G9 and G11 gates. Note that by selecting this portion of circuitry for obfuscation we affect a portion of flip-flops which cannot be directly controlled, G5 and G6. In this case G6 is simply a direct output of the obfuscated block.

This is a small circuit with as many primary inputs as their are flip-flops (specifically, flip-flops that cannot be directly controlled). In larger circuits we find it is much easier to find portions of circuitry that are influenced by a larger majority of flip-flops than primary inputs and also affect a larger number of flip-flops than primary outputs.

Once the digital PUF is configured in Figure 7.12b we synthesize the configurable fabric to map the PUF outputs to the original replaced circuitry outputs using traditional FPGA design tools.



(a)



(b)

Figure 7.12: Motivational example using the (a) s27 circuit from the ISCAS'89 benchmark suite. (b) Obfuscated form using the post-logic architecture from Figure 7.11b. The blue pins denote primary inputs. The red pin denotes a primary output. The green pins represent flip-flops.

7.9.1.2 Attack

We assume that an adversary has complete knowledge of the design of the circuit even including knowledge of the design of the supporting configurable fabric. We assume that he has read access to all flip-flops in the circuit, but only write access to those flip-flops which are primary inputs to the system.

The job of the attacker is to reverse engineer the functionality of the entire circuit. Specifically, he will focus on the part that is obfuscated. Since we allow him to know the design of the configurable fabric, this leaves him with the task of fully characterizing the digital PUF.

This task is made more difficult by strategically selecting logic for obfuscation in such a way to reduce the attacker's ability to control the inputs to the obfuscating PUF as well to diffuse the outputs of the PUF before they arrive at a readable flip-flop.

7.9.1.3 Logic Selection

The digital nature of our PUF enables us to treat it as a combinational component. This gives us almost complete freedom to select any piece of arbitrary combinational logic for obfuscation. In assigning placement of obfuscated circuitry we consider the attack outlined above. Ultimately, the obfuscated logic is a black box in which the attacker can measure the inputs and outputs but is unaware of the internal functionality (e.g. digital PUF configuration). Thus, in logic selection for obfuscation we purposefully select a portion of logic for obfuscation whose inputs are difficult to control and whose outputs are as difficult as possible to measure. In this way we prevent an attacker from reconstructing a complete input-output switching expression of the obfuscated block.

Choosing inputs is accomplished by selecting wires that are dependent upon many flip-flops. By selecting the inputs to the obfuscated block in this manner we ensure that an attacker cannot directly control its input vectors. For example, in the obfuscated circuit example in Figure 7.12b, the obfuscated block is dependent upon six flip-flops, three of which cannot be directly controlled (G5, G6, G7) and two of which are also obfuscated (G5, G6). In order to reduce delay overhead we select sets of input wires which contain positive slack and whose ASAP and ALAP delays overlap.

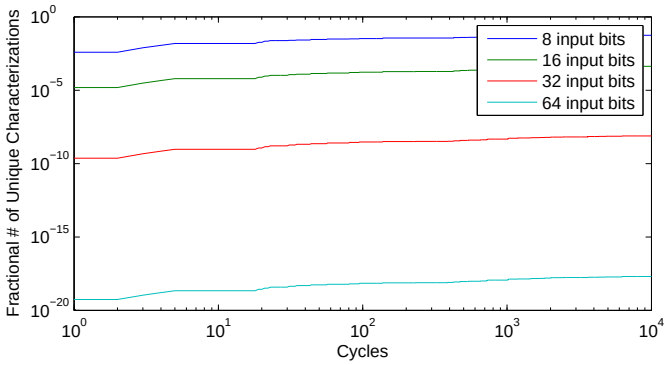
The outputs of the obfuscated block are selected in such a way so as to maximize the reverse engineering task of the attacker. We assume that an attacker has full knowledge of the netlist of the circuit as well as read access to all flip-flops. In order to hide the outputs of the obfuscated block from the attacker we select output wires that combine together through regular circuitry into one flip-flop. This forces the attacker to reverse engineer the original output through the circuitry from a minimal amount of information

Note that the reverse engineering task is equivalent to the satisfiability (SAT) problem and is thus NP-complete. Hence, we increase the level of difficulty by selecting n obfuscated logic block outputs that combine maximally to k flip-flops, thus increasing the total number of clauses and variables comprising the SAT instance that must be solved by the attacker.

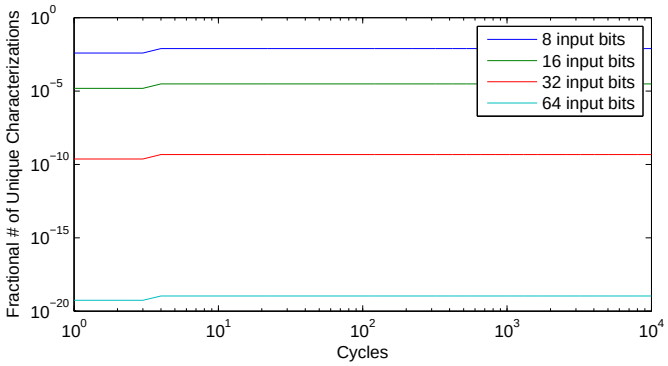
7.9.1.4 Obfuscation and Overhead

In this section we analyze and measure the overhead requirements and feasibility of attacking obfuscated circuits from the ICSACS'89 benchmark suite [37].

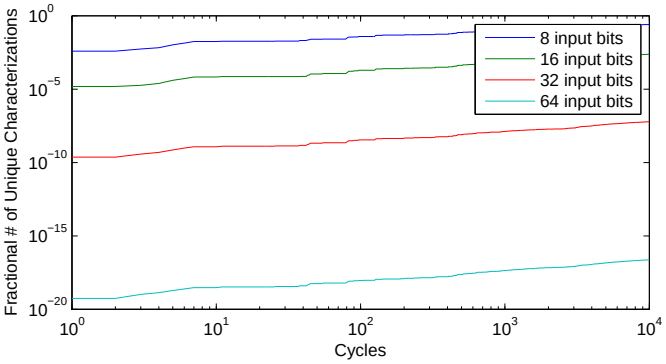
Figure 7.13 depicts the number of clock cycles required to fully characterize a fraction of input-output mappings of the pertinent obfuscated logic for three



(a)



(b)



(c)

Figure 7.13: Fraction of correctly characterized PUF obfuscated logic input-output mappings for the (a) s5378, (b) s9234, and (c) s38417 circuits from the ISCAS'89 benchmark suite [37].

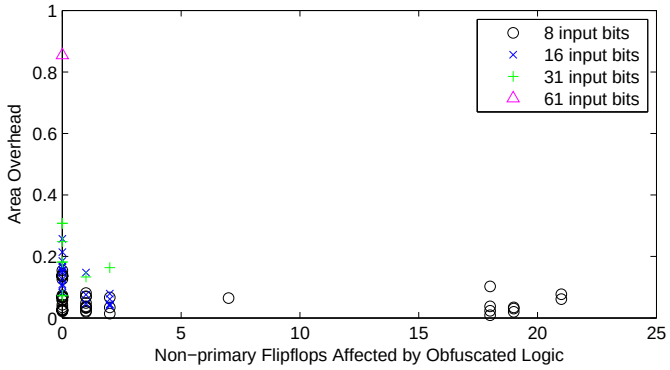
example benchmarks. In each case we analyze 100 different obfuscation configurations with the corresponding input bit size and plot the average number of characterized input-output mappings over time.

In these examples we assume an even more powerful attack than described above in which the attacker knows the output of the obfuscated block without the need to reverse engineer it. Note that even with this knowledge the number of characterized input-output mappings increases only linearly with an order of magnitude increase in cycles observed. Furthermore, by increasing the input size of the obfuscated logic block we reduce the absolute fractional number of input-output mapping characterizations by the same order of magnitude increase in input size, rendering complete specification of the obfuscated logic block infeasible.

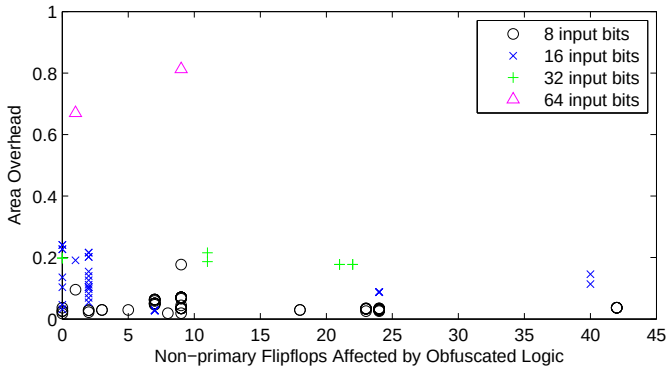
We measure the area overhead required by the varying input sizes on different gates and depict the results in Table 7.1 and Figure 7.14. Our technique ensures that area overhead remains approximately the same order of magnitude for a given input set size in absolute terms, and thus, decreases tremendously with the size of the obfuscated circuit.

Circuit	Gates	Average Area Overhead			
		8	16	32	64
s1488	653	13.55 %	58.96 %	-	-
s5378	2,779	5.48 %	11.25 %	16.09 %	85.46 %
s9234	5,597	4.14 %	11.42 %	18.87 %	74.15 %
s35932	16,065	2.27 %	2.68 %	-	-
s38417	22,179	0.82 %	2.34 %	2.69 %	4.85 %

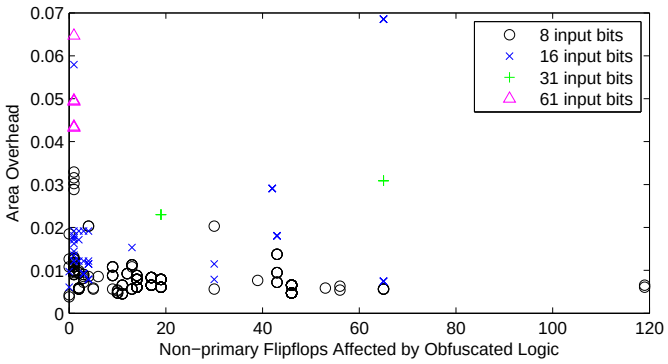
Table 7.1: Average area overhead for obfuscated logic with input sizes of 8, 16, 32, and 64 for the pertinent benchmark circuits. The dashed placeholders represent input set sizes that could not be found for the corresponding circuit.



(a)



(b)



(c)

Figure 7.14: Area overhead of circuit obfuscation as a fraction of the original size of a 90nm circuit for the (a) s5378, (b) s9234, and (c) s38417 circuits from the ISCAS'89 benchmark suite [37].

7.9.2 Remote Trust

Trust is an essential component for many remote systems. It is even more essential for sensor networks which are often left unattended and installed in potentially hostile environments. The notion of trust in such systems enables that a communicating party know with certainty that a sensor node's data being transmitted has indeed been collected by that sensor which has not been tampered or compromised in any way. While public key cryptography ensures that no information is snooped over an insecure line, it does not protect against physical attacks to the sensing node. For example, if an attacker were to move a sensing node from its intended location, the node will continue to record and send its data over a secure channel to the base station, while the base station is unaware of the attack.

The key to enabling remote trust is through the integration of the system's core functionality along with pertinent parts of trustworthy circuitry with a PUF. The idea is that by combining the PUF with these data collecting elements (i.e. sensors, GPS, clock), any tampering of the PUF and/or data elements will affect the PUF outputs, effectively changing its functionality.

Previous approaches to trusted remote sensing utilize analog PUFs as the trust mechanism [15]. In addition to a susceptibility to environmental and operational variations, these devices are also susceptible to glitching. Since these devices are analog in nature, they rely on signal path propagation races throughout the PUF network. Between clock cycles and applications, some signals remain inside the PUF, ultimately affecting the next clock cycle. A zeroing procedure has yet to be presented for these architectures that is low in latency and effective at removing glitching between uses, however it is assumed that at least half of the throughput of the device is lost in practical operation since in at least every other clock cycle it is necessary that the PUF be zeroed, possibly more.

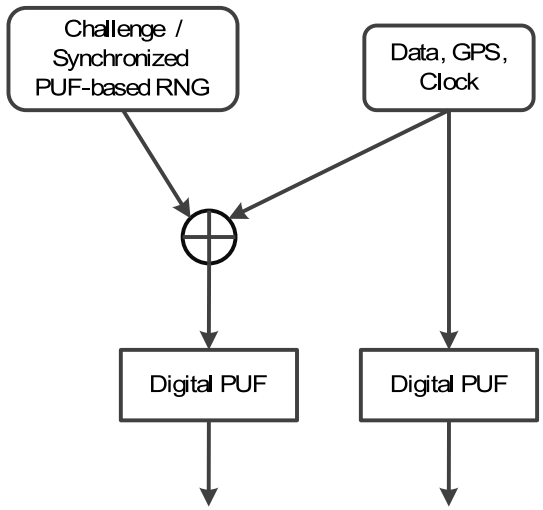


Figure 7.15: Trusted remote sensing computation flow at the sensor node. The base station provides the challenge.

The digital PUF requires only a single cycle to initialize at power-up and only a single cycle to function and requires no additional clock cycles for resetting. Not only is it a low latency, high throughput, and low energy primitive, but it is also completely integrable with digital logic. Similar to our logic obfuscation application, we can place digital PUFs in the middle of digital logic, completely integrating with data collection elements such as sensory circuitry.

Figure 7.15 depicts the trusted remote sensing computation flow performed by each sensor node. The challenge provided to the sensor node can either be sent directly by the base station or—since the digital PUF passes all NIST tests—can be supplied using a digital PUF as a synchronized random number generator (RNG). At installation, the digital PUF-based RNG is synchronized with the digital PUF-based RNG at the base station. Since the system operator is the only entity that knows the functionality of both PUFs, only he can select seeds for each PUF that synchronizes their functionality. Note that the seeds are in fact challenge vectors to the analog parts of the digital PUF for initialization and

hence, are unique. Thus, the seeds can even be made public since their use in initializing any other digital PUF would produce a completely different random number generator.

The remaining digital PUFs are also configured at installation and their functionality is recorded at the base station. At transmission time, the sensor node sends its data, timestamp, GPS coordinates, and two digital PUF outputs as illustrated in Figure 7.15. Since the base station knows the configuration of the sensor, it validates the data in a single cycle. A man-in-the-middle attack is easily caught since any alterations to any of the transmitted data will cause a vastly different PUF output which cannot be computed by an attacker without having reverse engineered the pertinent digital PUF.

7.9.2.1 Hardware Attestation

A requirement of the trusted remote sensing computation flow depicted in Figure 7.15 is that the digital PUF and circuitry (e.g. sensing, GPS, clock) must be physically coupled together to prevent physical attacks. Specifically, the boundaries between these two components should not be easily read or, more importantly, written to by an attacker.

We enable physical coupling by incorporating a variant of our hardware obfuscation architecture into the pertinent circuit as depicted in Figure 7.16. The control signal c_a selects whether the circuit operates in functional mode, in which the circuit operates as normal, or attestation mode, in which the circuit outputs a unique signature for verification.

The most important property of the generated signature is that it simultaneously and uniquely entangles the input data, logic, and digital PUF into one signal, and does so in a single clock cycle. Since the digital PUF functionality

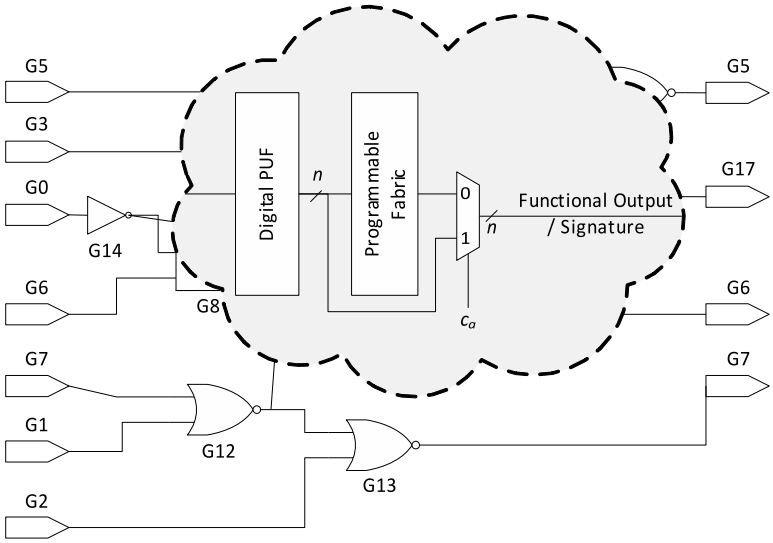


Figure 7.16: Variant of the hardware obfuscation architecture applied to the s27 benchmark suite enabling hardware attestation. The control signal c_a determine whether the circuit operates in normal functional mode or in attestation mode.

is known only by the remote operator, only he can verify the output signature given the inputs. By integrating this architecture into pertinent components of the device (e.g. sensing circuitry, GPS, clock), we enable that these components and their outputs can be remotely trusted.

CHAPTER 8

Trusted Chip Selection

Integrated circuit counterfeiting is the unauthorized manufacturing of a chip design without official consent. While this illegal practice reduces the original designer's profits, the extended effects could potentially be much more extreme. Take for example a counterfeit chip made at an untrusted foundry. If such a chip was installed in a life-critical system, such as a medical device or military equipment, the results could be catastrophic.

Since capital costs required to build semiconductor fabrication plants are upwards of one billion US dollars, it is cost prohibitive for chip designers to own and maintain their own private foundries. Instead, they must resort to outsourcing fabrication to third party foundries. Thus, it is imperative that we develop comprehensive techniques for IC trust, intellectual property protection, and counterfeit prevention.

In this paper, we introduce statistical methods for foundry identification. Our techniques enable new applications of foundry identification, including design analysis, yield calculation, and chip usage monitoring.

Our key idea is to use the manifestations of process variation (PV) in integrated circuits, and unique to semiconductor fabrication plants, for the purpose of foundry profiling. Specifically, we use distributions of the following PV-affected parameters, threshold voltage (V_{th}) and effective channel length (L_{eff}). We build

foundry profiles consisting of parameter distributions for each design that the foundry is assigned.

While nominal values of V_{th} and L_{eff} are known at design and manufacturing time, due to process variation, they deviate from their expected values [44]. In order to extract the post-silicon physical values, we propose new methods for reverse engineering these parameters. We model simultaneously both the functionality and timing of the integrated circuit. By measuring the delay values made up by a signal edge traversing many gates, we can reverse engineer their individual threshold voltages and effective channel lengths.

Unfortunately, these IC parameters are governed by non-linear models which are difficult to solve for very large systems (e.g. integrated circuits). Thus, we simplify the problem by solving as many linear parts of the system first in order to reduce the complexity and size of the non-linear portions. Specifically, we localize and activate single branches within the IC design using SAT. By measuring the delays of multiple localized branches throughout the circuit we enumerate a system of linear equations for gate delays. Once these linear equations are solved and individual gate delays are known, the non-linear systems that relate delay to V_{th} and L_{eff} are reduced to a single system per gate with as few as two equations, and thus, are much easier to solve.

Once the foundry profiles are characterized, we identify the originating foundry of a particular chip by comparing the parameter distributions of the IC with that of the foundry profile using statistical tests for distribution equality. Furthermore, we investigate and explore the effects of measurement error, sample size, and supply voltage on the identification rate.

8.1 Related Work

Several foundry identification and IC counterfeiting techniques have been developed. Physical unclonable functions utilize process variation to create unique hardware functions that enable multiple security protocols, including identification [125] [126] [127] [101] [102]. The unique power-up states of SRAM cells have been proposed as IC fingerprints [128]. And passive and active hardware metering schemes enable identification and counterfeit prevention [129] [130].

There exist a number of gate level characterization techniques including direct measurement approaches, methods that incorporate and monitor specialized hardware structures and circuitry, FPGA-based reconfiguration techniques, and modeling procedures that assemble systems of equations representing gate characteristics [131] [132] [34]. Applications of gate level characterization include hardware Trojan detection and leakage minimization through post-silicon input vector selection [133] [134] [35].

8.2 Process Variation

Gate delays and effective channel lengths in nanoscale technologies are subject to significant process variation [19] [20]. A variety of manufacturing faults emerge as a result, including but not limited to variations in doping concentrations, imperfect mask alignment, and molecular chemical and physical phenomena. These forms of process variation manifest as the deviation of IC characteristics from nominal values. For example, variations in doping concentrations and line edge roughness alter transistor threshold voltages and effective channel lengths. These manifestations have been thoroughly studied, categorized, and modeled, yet continue to be of paramount concern [21] [22].

8.3 Foundry Characterization

While the deviation of physical characteristics is inherent within a single chip, the complex and often custom procedures and machinery employed by fabrication facilities make process variations even more pronounced when comparing chips made by different foundries. We build foundry profiles based on these unique parameter deviations.

This enables the identification of an integrated circuit to a particular foundry. We accomplish this by extracting the physical characteristics of the relevant components of a design and comparing its distribution to that of the foundry profile. We investigate the use of transistor threshold voltage and transistor effective channel length distributions of particular designs as the characterizing foundry parameters.

We compare the parameter distributions of an unknown chip of known design with the parameters of a foundry using non-parametric tests for statistical significance in distribution equality, such as the Kolmogorov-Smirnov and Cramér-von Mises tests [135]. These statistical tests are found in common libraries and numerical computing environments, such as the R statistical package and MATLAB. We use the normalized asymptotic p-values (ranging from 0 to 1) to measure similarity strengths. Note, that it is common to reject the null hypothesis, and conclude that two distributions are dissimilar, if the p-value is less than or equal to 0.05.

In the following section we present our methodology for foundry identification in reverse order for clarity. We begin by discussing how to reverse engineer the V_{th} and L_{eff} values of a particular gate using known delay measurements. We then discuss how delay values can be measured for a particular set of gates within

the circuit. This discussion also includes how these sets are selected, how to determine the inputs to activate these gates and these gates only, and how to physically measure their total delay.

8.4 Extracting IC Parameters

We propose a new modeling-based approach to gate level characterization. The model we employ exhibits a non-linear relationship between the IC parameters and delay. Unfortunately, solving this system becomes prohibitively difficult as the system grows in size just beyond a few gates. Since integrated circuits are often composed of orders of magnitude more gates, solving a large set of non-linear equations is entirely impractical.

Therefore, we focus on solving for individual gate delays before solving these non-linear equations. This enables us to separate the system of non-linear equations into individual per-gate systems that can be solved independently. Our first step is to localize these gate delay values. This is accomplished by measuring path delays that consist of some number of gates from an input to an output. This is done by iteratively activating single non-branching paths in the circuit. In this manner, we are able to compose linear equations comprised of the summation of all gate delays along that path (e.g. $D_{path} = d_0 + d_1 + \ldots + d_k$, where D_{path} is the measured path delay and d_i corresponds to the unknown delay of gate i along the path). In order to activate these paths, we systematically search for pairs of inputs using a variant of SAT. We discuss the details of our techniques in the following sections.

8.4.1 Solving for Threshold Voltage and Effective Channel Length

The threshold voltage and effective channel lengths of each gate can be reverse engineered using Equation 8.1 from Marković et al. [43]. The two variables subject to the effects of process variation are L_{eff} and V_{th}. In our experiments, we vary the supply voltage (V_{dd}) and measure delay while keeping all other parameters constant.

$$Delay = \frac{k_{tp} \cdot k_{fit} \cdot L_{eff}^2}{2 \cdot n \cdot \mu \cdot \phi_t^2} \cdot \frac{V_{dd}}{(ln(e^{\frac{(1+\sigma)V_{dd}-V_{th}}{2 \cdot n \cdot \phi_t}} + 1))^2} \cdot \frac{\gamma_i \cdot W_i + W_{i+1}}{W_i} \tag{8.1}$$

Since this model contains two unknowns, it requires at least two measurements of delay and V_{dd} to create a solvable system of equations. We discuss the details of choosing values of V_{dd} in Section 8.5.4.

However, physically measuring the delay of a single gate inside of a large circuit is near impossible. Instead, one solution is to solve a system of these equations representing multiple gates whose total delay can more easily be measured. For example, if the critical path is known, then the critical path delay can be measured, which corresponds to the sum of the delays of the gates on the critical path. We discuss the details of activating many different paths for the purposes of building a solvable system of equations in the following section. Unfortunately, due to the non-linearity of this model, solving such large systems is very difficult.

8.4.2 Solving for Delay

We simplify the large non-linear system described above by reducing it to a set of linear equations comprised of gate delay unknowns (e.g. $D_{path} = d_0 + d_1 + \ldots + d_k$). Once this system of linear equations is solved, each gate's IC parameters can be

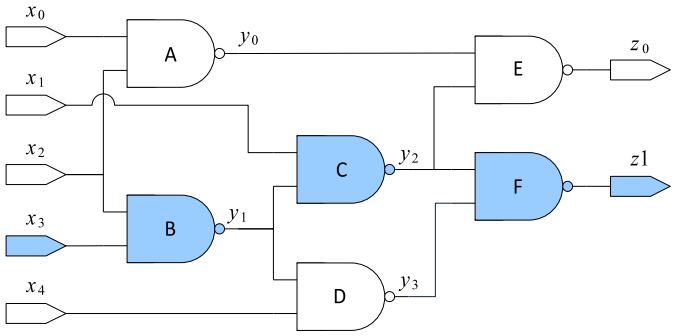
solved individually and independently using Equation 8.1. We do this because delay measurements can be localized to single non-branching circuit paths and can be physically measured accurately using clock sweeping techniques.

We build the linear system by selecting sets of gates that each constitute a single non-branching path from input to output and can be activated using appropriate input vectors. These inputs must send the signal edge through these gates and these gates only. Thus, by measuring the delay of the signal edge propagation from input to output, we know the total delay made up of all the individual gate delays along that path.

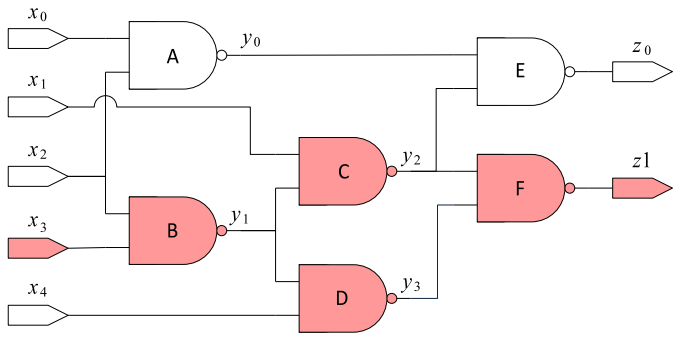
Take for example Figure 8.1a. Initializing the circuit with input P_0 followed by applying P_1 causes z_1, y_2, and y_1 to switch. By measuring the delay between sending the signal edge into the circuit (i.e. switching inputs from P_0 to P_1) and measuring the time at which z_1 switches, we can compose a linear equation of the delays of gates B, C, and F that sum to the overall delay. In other words, $D = d_B + d_C + d_F$, where d_i is the delay of gate i.

When applying the Q inputs to the example circuit in Figure 8.1b, the signal edge follows two paths (colored in red): one through C and the other through D. While this input pair successfully switches output bit z_1, due to race conditions between the C and D gate delays, it is not clear which of the two paths will cause z_1 to switch first. Therefore, the Q input pair cannot reveal any deterministic information about the circuit's individual gate delays.

Note that both the P and Q input pairs are intentionally separated by one hamming distance in order to ensure correct placement of signal edge initiation.



(a)



(b)

	x_0	x_1	x_2	x_3	x_4	y_0	y_1	y_2	y_3	z_0	z_1
P_0:	1	1	1	0	0	0	1	0	1	1	1
P_1:	1	1	1	1	0	0	0	1	1	1	0
Q_0:	1	1	1	0	1	0	1	0	0	1	1
Q_1:	1	1	1	1	1	0	0	1	1	1	0

Figure 8.1: Circuit c17 from the ISCAS85 benchmark suite [38]. The blue components in (a) correspond to the signal edge path when initialized with input P_0 followed by applying P_1. The red components in (b) correspond to the signal edge path when initialized with input Q_0 followed by applying Q_1.

8.4.2.1 Satisfiability

We employ a variant of SAT in order to find pairs of inputs that satisfy the following constraints:

- Inputs must differ by one hamming distance (i.e. one bit). This ensures

accurate placement of signal edge initiation along with precision timing.

- The signal edge must pass through and activate (i.e. switch) only the gates along a single non-branching path. There must be a single and distinct path from the switching input bit to the switching output bit.

We find input vectors that satisfy these constraints by enumerating all paths from input x to output y, then composing the relevant boolean satisfiability constraint and solving.

In Figure 8.1, only a single path from input x_0 to output z_0 exists and it passes through y_0. Equation 8.2 defines the constraint for this path. The path from input x_3 to output z_1 is represented by the constraint in Equation 8.3.

$$f(a) = \begin{cases} True, & \text{if value at wire } a \text{ switches} \\ False, & \text{otherwise.} \end{cases}$$

$$\begin{aligned} & f(z_0) \wedge f(y_0) \\ & \quad \wedge f(x_0) \wedge \neg f(x_1) \wedge \neg f(x_2) \wedge \neg f(x_3) \wedge \neg f(x_4) \end{aligned} \tag{8.2}$$

$$\begin{aligned} & f(z_1) \wedge \Big(\Big(f(y_2) \wedge \neg f(y_3) \Big) \vee \Big(\neg f(y_2) \wedge f(y_3) \Big) \Big) \\ & \quad \wedge f(y_1) \wedge \neg f(x_0) \wedge \neg f(x_1) \wedge \neg f(x_2) \wedge f(x_3) \\ & \quad \wedge \neg f(x_4) \end{aligned} \tag{8.3}$$

Because this problem is NP-complete and SAT cannot produce an optimal solution, we also conduct an exhaustive search since for a majority of our instances it is possible to enumerate enough solutions to produce a solvable system of equations. The exhaustive search is executed by iteratively applying input pairs that satisfy our first condition and checking that the signal edge activates a non-

branching path.

8.4.3 Device Aging

Device aging is a phenomena that changes the physical characteristics (e.g. threshold voltage) of circuit components over time. Even with this added complexity, we can still identify the originating foundry of an aged circuit by employing techniques and models from Wei et al . [136] and Chakravarthi et al. [30].

By measuring delay and threshold voltage before and after intentional device aging, we enumerate a set of time-dependent non-linear aging model equations as described in Equation 8.4, then solve for the initial threshold voltage, $V_{th}(t_0)$.

$$\begin{aligned} V_{th}(t_1) &= V_{th}(t_0) + K \times t_1^{0.25} \\ V_{th}(t_i) &= V_{th}(t_0) + K \times (t_{i-1} + \Delta T)^{0.25} \end{aligned} \tag{8.4}$$

8.5 Identification

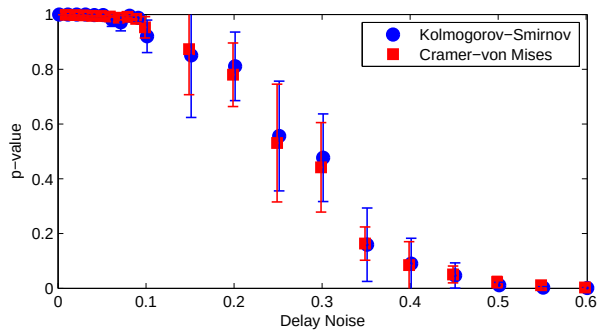
Correct identification of a circuit largely depends on the precision and accuracy of the delay measurements of each path. In this section we investigate how delay measurement errors, along with path size, supply voltage range, and supply voltage magnitude affect the ability to correctly identify the originating foundry. We also explore the capabilities of our SAT variant in localizing characterizable gates.

8.5.1 Delay Measurement Error

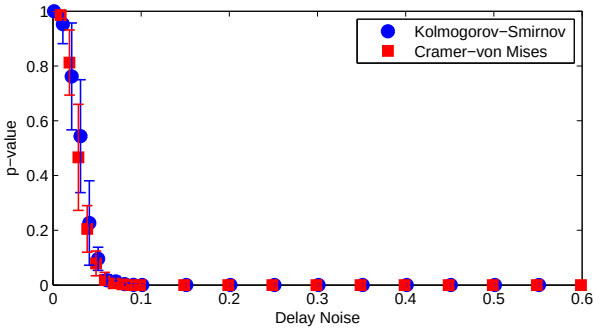
We investigate the resilience of our techniques to natural fluctuations and error in delay measurement. We introduce a gaussian error whose standard deviation is

multiplied by the expected delay and the fraction depicted on the x-axis in Figure 8.2. The figure comprises of p-values comparing the foundry profile parameters to the reverse engineered IC parameters of a branch of 200 gates with a range of supply voltages from 0.5V to 3V. The remaining figures in this paper depict p-values generated using the Kolmogorov-Smirnov test.

Correct characterization through reverse engineering of V_{th} is resilient up to an error rate of 0.4, while the characterization of L_{eff} tapers and fails the test at about 0.05. Depending on the apparatus and techniques available to us as well as the level of certainty required, either one or both of these characteristics can be utilized for identification.



(a)



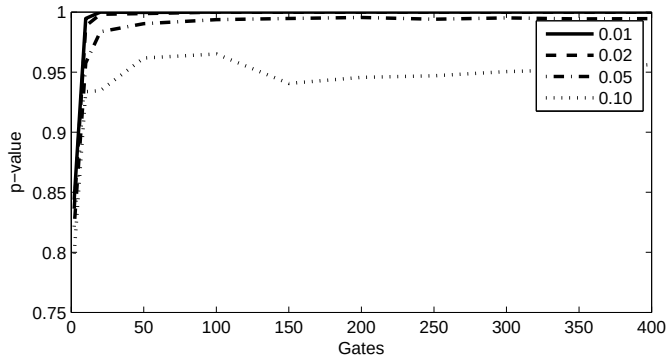
(b)

Figure 8.2: The effects of delay measurement error on the Kolmogorov-Smirnov and Cramér von-Mises two sample tests for (a) V_{th} and (b) L_{eff}. Uncertainty bars represent the standard deviation of p-values from 100 tests.

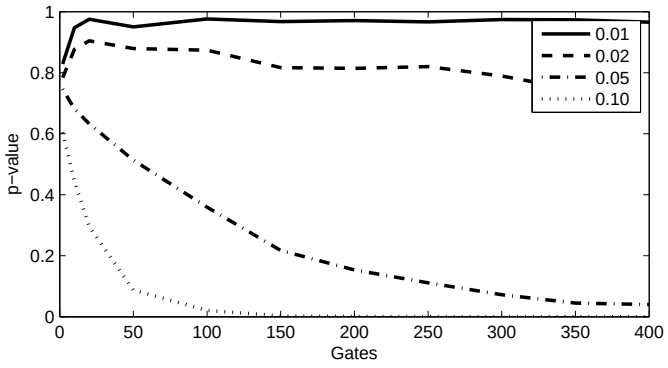
8.5.2 Sample Size

The asymptotic p-value for the Kolmogorov-Smirnov two sample test becomes very accurate for large sample sizes. It is also assumed to be reasonably accurate for sample sizes n_0 and n_1 such that $\frac{n_0 n_1}{n_0+n_1} \geq 4$. However, the measurement error in delay—which consequently translates to error in the reverse engineered values of V_{th} and L_{eff}—has a complex effect on the ability of the statistical test to measure distribution equality.

Figure 8.3 shows that even with a substantial amount of delay measurement



(a)



(b)

Figure 8.3: The effects of distribution size and delay measurement errors on correct identification using distributions of (a) V_{th} and (b) L_{eff}. Legend errors correspond to those described in Figure 8.2.

error, a circuit can be correctly identified using V_{th} parameters, while for L_{eff} the increasing error rate has a negative effect on the ability of the statistical test to correctly identify the foundry. So long as measurement error is low, we can use both characteristics to model and identify foundries. For higher error rates, V_{th} should be used.

8.5.3 Gate Delay Characterization

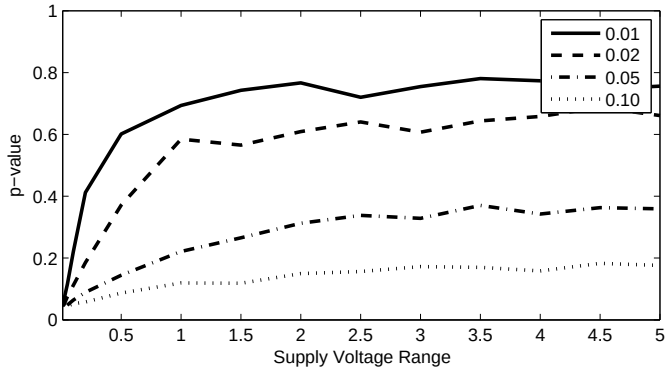
We evaluate our techniques using the ISCAS89 and ITC99 benchmark suites [37] [45]. Table 8.1 lists the total number of gates we are able to characterize using our SAT formulation. Specifically, the equations extracted comprise a solvable system for the individual gate delays of each benchmark circuit. We employ a linear solver to calculate the gate delays across a span of supply voltage magnitudes and ranges.

Circuit	Total Gates	Characterized Gates
s9234	5,597	1,165
s15850	9,772	3,994
b21_1	12,248	138
b20_1	12,264	138
s35932	16,065	4,754
b20	17,158	138
b21	17,482	138
b22_1	18,461	170
s38584	19,253	4,878
s38417	22,179	6,274
b22	25,460	154
b17	27,852	759
b17_1	32,971	860
b18_1	88,954	590
b18	94,249	582

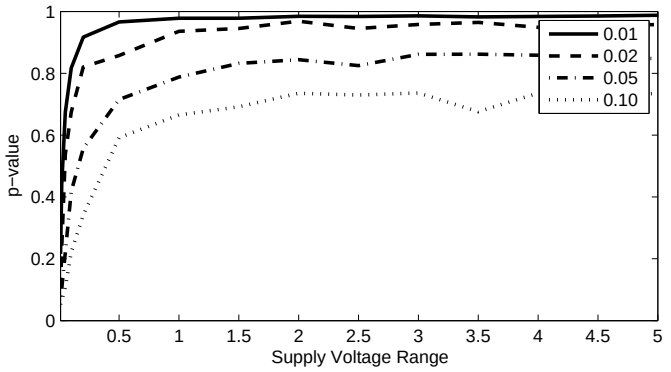
Table 8.1: Gates in benchmark circuits [37] [45] whose IC parameters can be fully characterized.

8.5.4 Supply Voltage Range and Magnitude

We find that the number of supply voltages required to create a solvable system for reverse engineering V_{th} and L_{eff} using Equation 8.1 has a very limited impact on the correct identification rate as compared to the *selection* of supply voltages. Thus, we construct our system using the minimal required number of equations and investigate the selection—focusing on magnitude and range—of supply voltages.



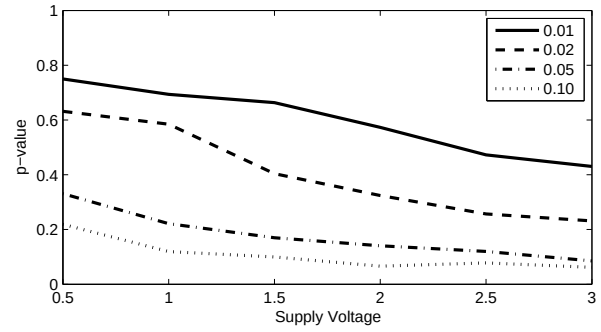
(a)



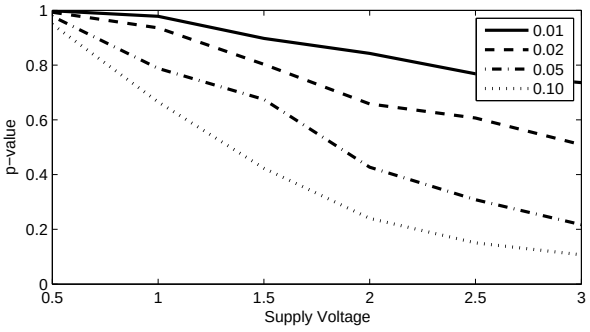
(b)

Figure 8.4: The effects of supply voltage range on correct identification using distributions of (a) V_{th} and (b) L_{eff}. The first voltage equals 1V while the second voltage differs by the value along the x-axis. Legend errors correspond to those described in Figure 8.2.

Our first observation found in Figure 8.4 confirms that increasing the distance between the two supply voltages governing the system of equations improves the overall identification rate of a circuit. Our second observation found in Figure 8.5 confirms that the magnitude of the supply voltage pair (given a predetermined range) is best applied nearer to the nominal threshold voltage rather than far away. The first observation is explained given the resulting collinear results of the non-linear model (Equation 8.1) when two supply voltages are placed close to one another. Likewise, the delay model becomes collinear at large supply voltages, while at near-threshold values, the non-linear relationship between supply voltage and delay is much more pronounced and can be solved much more accurately.



(a)



(b)

Figure 8.5: The effects of supply voltage magnitude on correct identification using distributions of (a) V_{th} and (b) L_{eff}. The first supply voltage corresponds to the value along the x-axis. The second supply voltage is 1V larger. Legend errors correspond to those described in Figure 8.2.

8.5.5 Foundry Identification

We test the overall resilience of our techniques by identifying the originating foundry of many instances of three chips in a simulated environment with a 0.05 delay measurement error rate. Simulation parameters are depicted in Figure 8.6. The foundries A, B, and C are represented by their IC parameter distributions as governed by their unique process variations. Circuits 1, 2, and 3 are example instances corresponding to foundries A, B, and an unknown site, respectively. After reverse engineering the IC parameters of circuits 1, 2, and 3 we overlay the resulting predicted IC parameters in Figure 8.6.

The Kolmogorov-Smirnov two sample test results comparing the threshold voltage and channel length distributions between each pair of circuit and foundry are listed in Table 8.2. The threshold voltage comparisons successfully identify foundries A and B as the originating fabrication facilities of circuits 1 and 2,

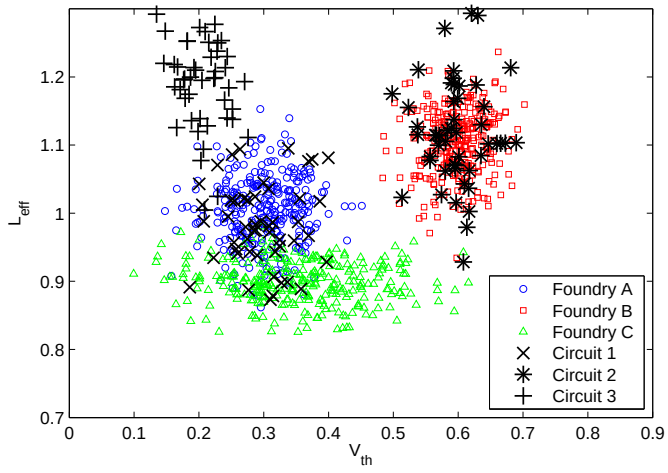


Figure 8.6: IC parameters and foundry profiles. Circuit 1 originates from foundry A, circuit 2 originates from foundry B, and circuit 3 is a counterfeit that does not originate from any trusted foundry. The circuit parameters are reverse engineered from delay values measured with a 0.05 error rate.

		Foundry A	Foundry B	Foundry C
Circuit 1	V_{th}	0.35 - 0.96	0	0
	L_{eff}	0 - 0.80	0	0
Circuit 2	V_{th}	0	0.24 - 0.76	0
	L_{eff}	0	0.02 - 0.69	0
Circuit 3	V_{th}	0	0	0
	L_{eff}	0	0	0

Table 8.2: Minimum and maximum p-values for circuit parameter and foundry profile comparisons using the Kolmogorov-Smirnov test. Foundry distributions correspond to those depicted in Figure 8.6. We test 20 instances of each circuit.

respectively, as well as rejecting circuit 3 from all three foundries. However, the effective channel length distribution tests for both circuit 1 and 2 are not as reliable as they periodically dip below an acceptable significance level of 0.05.

8.6 Summary

We have presented new statistical techniques for foundry detection by specifically identifying from which foundry a particular chip originates from. Our key idea is to consider the distributions of channel lengths and threshold voltages by assembling and solving a variant of SAT, then focus on solving the linear parts of the system as far as possible before reverse engineering the IC parameters. We then compared the IC parameter distributions using non-parametric statistical tests in order to identify the originating foundry.

We have tested our techniques on a host of benchmark circuits while investigating the effects of delay measurement error, sample size, and voltage range and magnitude on the correct identification rate. We find that reverse engineered threshold voltage distributions are resilient to high delay measurement error while effective channel lengths are resilient only at very low error rates.

CHAPTER 9

Concluding Remarks

The rapid emergence of the Internet of Things has presented numerous opportunities and challenges to address in hardware design. For resource-constrained IoT systems in particular there is a great need to develop design methods for ultralow energy operation. Security also continues to be of paramount concern and is often in direct conflict with the requirement of low energy. Furthermore, IoT devices have been envisioned and installed in a number of unique environments, which has also introduced new possibilities for new attacks.

We began our study by introducing techniques for ultralow energy design and operation applied at the circuit-level. Our first observation was that near-threshold computing is well suited for IoT due to its low energy benefits and the limited processing power required by IoT devices. We addressed issues such as process variation and performance degradation by introducing device aging, adaptive body biasing, and node organization techniques and applying them to popular circuit benchmarks as well as popular multiple constant multiplication applications, such as FFT and DCT.

We then raised our level of abstraction to the system-level to present techniques for subsystem and component organization for low energy system design and operation. Specifically, we employed a semantics-driven approach to sensor configuration and subsampling and demonstrated our methods to be more energy efficient than a purely raw data-driven approach. Furthermore, since resource-

constrained IoT devices will have very limited power budgets and are projected to become battery-less (i.e. harvest their power or receive it wirelessly), we also presented methods for harvester organization and placement on a wearable medical device.

After focusing our efforts on energy design techniques, we discussed and explored new techniques for securing these systems. Specifically, we introduced a novel hardware obfuscation technique utilizing physical unclonable functions, that not only protects the intellectual property of these systems, but also enables that the data, location, and time queried from the device can be trusted. And finally, we presented techniques for ensuring that the integrated circuits selected for use in these systems, specifically systems whose integrity of components is critical, were produced by a trusted source.

References

[1] National Intelligence Council, "Disruptive civil technologies: Six technologies with potential impacts on US interests out to 2025," in *Conference Report CR*, 2008.

[2] "Gartner says the Internet of Things will transform the data center." `http://www.gartner.com/newsroom/id/2684915`, March 18, 2014.

[3] "Nest | Home." `http://www.nest.com`.

[4] "Dropcam - super simple video monitoring and security." `https://www.dropcam.com`.

[5] "Intel® Edison platform." `http://www.intel.com/content/www/us/en/do-it-yourself/edison.html`.

[6] "Intel® Curie™ module: Unleashing wearable device innovation." `http://www.intel.com/content/www/us/en/wearables/wearable-soc.html`.

[7] "HomeKit - Apple Developer." `https://developer.apple.com/homekit`.

[8] "Samsung IoT | SAMSUNG Developers." `http://developer.samsung.com/iot`.

[9] "Industrial Internet and Internet of Things | GE Software." `https://www.gesoftware.com/industrial-internet`.

[10] A. Juels, "RFID security and privacy: A research survey," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 381–394, 2006.

[11] J.-P. Vasseur and A. Dunkels, *Interconnecting smart objects with IP: The next Internet*. Morgan Kaufmann, 2010.

[12] J. Hui, D. Culler, and S. Chakrabarti, "6LoWPAN: Incorporating IEEE 802.15.4 into the IP architecture," *Internet Protocol for Smart Objects (IPSO) Alliance White Paper*, no. 3, 2009.

[13] A. Dunkels and J. Vasseur, "IP for smart objects," *Internet Protocol for Smart Objects (IPSO) Alliance White Paper*, no. 1, 2008.

[14] G. Hernandez, O. Arias, D. Buentello, and Y. Jin, "Smart Nest thermostat: A smart spy in your home," *Black Hat USA*, 2014.

[15] M. Potkonjak, S. Meguerdichian, and J. L. Wong, "Trusted sensors and remote sensing," in *IEEE Sensors*, pp. 1104–1107, 2010.

[16] J. H. Kong, L.-M. Ang, and K. P. Seng, "Minimalist security and privacy schemes based on enhanced AES for integrated WISP sensor networks," *Journal of Communication Networks and Distributed Systems*, vol. 11, no. 2, pp. 214–232, 2013.

[17] R. G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge, "Near-threshold computing: Reclaiming Moore's law through energy efficient integrated circuits," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 253–266, 2010.

[18] B. Cheng, A. Brown, S. Roy, and A. Asenov, "PBTI/NBTI-related variability in TB-SOI and DG MOSFETs," *IEEE Electron Device Letters*, vol. 31, no. 5, pp. 408–410, 2010.

[19] B. Cline, K. Chopra, D. Blaauw, and Y. Cao, "Analysis and modeling of CD variation for statistical static timing," in *International Conference on Computer-Aided Design (ICCAD)*, pp. 60–66, 2006.

[20] S. Nassif *et al.*, "High performance CMOS variability in the 65nm regime and beyond," in *IEEE Electron Devices Meeting (IEDM)*, pp. 569–571, 2007.

[21] K. Agarwal and S. Nassif, "Characterizing process variation in nanometer CMOS," in *Design Automation Conference (DAC)*, pp. 396–399, 2007.

[22] K. J. Kuhn, "Reducing variation in advanced logic technologies: Approaches to process and design for manufacturability of nanoscale CMOS," in *IEEE International Electron Devices Meeting (IEDM)*, pp. 471–474, 2007.

[23] P. Pillai and K. G. Shin, "Real-time dynamic voltage scaling for low-power embedded operating systems," in *Symposium on Operating System Principles*, vol. 35, pp. 89–102, 2001.

[24] A. P. Chandrakasan *et al.*, "Technologies for ultradynamic voltage scaling," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 191–214, 2010.

[25] B. H. Calhoun, A. Wang, and A. Chandrakasan, "Modeling and sizing for minimum energy operation in subthreshold circuits," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 9, pp. 1778–1786, 2005.

[26] B. Zhai *et al.*, "A 2.60 pJ/Inst subthreshold sensor processor for optimal energy efficiency," in *Symposium on VLSI Circuits*, pp. 154–155, 2006.

[27] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, "Theoretical and practical limits of dynamic voltage scaling," in *Design Automation Conference (DAC)*, pp. 868–873, 2004.

[28] M. Seok, G. Chen, S. Hanson, M. Wieckowski, D. Blaauw, and D. Sylvester, "CAS-FEST 2010: Mitigating variability in near-threshold computing," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 1, no. 1, pp. 42–49, 2011.

[29] M. R. Kakoee, A. Sathanur, A. Pullini, J. Huisken, and L. Benini, "Automatic synthesis of near-threshold circuits with fine-grained performance tunability," in *International Symposium on Low Power Electronics and Design (ISLPED)*, pp. 401–406, 2010.

[30] S. Chakravarthi, A. Krishnan, V. Reddy, C. Machala, and S. Krishnan, "A comprehensive framework for predictive modeling of negative bias temperature instability," in *IEEE International Reliability Physics Symposium*, pp. 273–282, 2004.

[31] S. Wei, J. X. Zheng, and M. Potkonjak, "Low power FPGA design using post-silicon device aging," in *International Symposium on Field Programmable Gate Arrays (FPGA)*, pp. 277–277, 2013.

[32] S. Wei, J. X. Zheng, and M. Potkonjak, "Aging-based leakage energy reduction in FPGAs," in *Field Programmable Logic and Applications (FPL)*, pp. 1–4, 2013.

[33] S. Smith *et al.*, "Comparison of measurement techniques for linewidth metrology on advanced photomasks," *IEEE Transactions on Semiconductor Manufacturing*, vol. 22, no. 1, pp. 72–79, 2009.

[34] J. S. Wong, P. Sedcole, and P. Y. Cheung, "Self-measurement of combinatorial circuit delays in fpgas," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 2, no. 2, p. 10, 2009.

[35] 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)*, pp. 606–609, 2008.

[36] M. Bhushan, M. B. Ketchen, S. Polonsky, and A. Gattiker, "Ring oscillator based technique for measuring variability statistics," in *IEEE International Conference on Microelectronic Test Structures*, pp. 87–92, 2006.

[37] F. Brglez, D. Bryan, and K. Kozminski, "Combinational profiles of sequential benchmark circuits," in *International Symposium on Circuits and Systems (ISCAS)*, pp. 1929–1934, 1989.

[38] F. Brglez and H. Fujiwara, "A neutral netlist of 10 combinational benchmark circuits and a target translator in FORTRAN," in *International Symposium on Circuits and Systems (ISCAS)*, pp. 663–698, 1985.

[39] R. Dreslinski Jr, *Near threshold computing: From single core to many-core energy efficient architectures*. PhD thesis, The University of Michigan, 2011.

[40] M. Miyazaki, G. Ono, T. Hattori, K. Shiozawa, K. Uchiyama, and K. Ishibashi, "A 1000-MIPS/W microprocessor using speed adaptive threshold-voltage CMOS with forward bias," in *IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 420–421, 2000.

[41] M. Miyazaki, G. Ono, and K. Ishibashi, "A 1.2-GIPS/W microprocessor using speed-adaptive threshold-voltage CMOS with forward bias," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 2, pp. 210–217, 2002.

[42] J. W. Tschanz *et al.*, "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 11, pp. 1396–1402, 2002.

[43] D. Markovic, C. C. Wang, L. P. Alarcon, T.-T. Liu, and J. M. Rabaey, "Ultralow-power design in near-threshold region," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 237–252, 2010.

[44] A. Asenov, A. R. Brown, J. H. Davies, S. Kaya, and G. Slavcheva, "Simulation of intrinsic parameter fluctuations in decananometer and nanometer-scale MOSFETs," *IEEE Transactions on Electron Devices*, vol. 50, no. 9, pp. 1837–1852, 2003.

[45] F. Corno, M. S. Reorda, and G. Squillero, "RT-level ITC'99 benchmarks and first ATPG results," *IEEE Design and Test of Computers*, vol. 17, no. 3, pp. 44–53, 2000.

[46] R. M. Russell, "The CRAY-1 computer system," *Communications of the ACM*, vol. 21, no. 1, pp. 63–72, 1978.

[47] J. Cong, H. Huang, and W. Jiang, "Pattern-mining for behavioral synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 6, pp. 939–944, 2011.

[48] S. Chou, M.-K. Hsu, and Y.-W. Chang, "Structure-aware placement for datapath-intensive circuit designs," in *Design Automation Conference (DAC)*, pp. 762–767, 2012.

[49] K. Atasu, L. Pozzi, and P. Ienne, "Automatic application-specific instruction-set extensions under microarchitectural constraints," *International Journal of Parallel Programming*, vol. 31, no. 6, pp. 411–428, 2003.

[50] H. Samueli, "An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients," *IEEE Transactions on Circuits and Systems*, vol. 36, no. 7, pp. 1044–1047, 1989.

[51] P. Cappello and K. Steiglitz, "Some complexity issues in digital signal processing," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 32, no. 5, pp. 1037–1041, 1984.

[52] M. R. Corazao, M. A. Khalaf, L. M. Guerra, M. Potkonjak, and J. M. Rabaey, "Performance optimization using template mapping for datapath-intensive high-level synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, no. 8, pp. 877–888, 1996.

[53] O. Gustafsson, H. Ohlsson, and L. Wanhammar, "Improved multiple constant multiplication using a minimum spanning tree," in *IEEE Conference on Signals, Systems and Computers*, vol. 1, pp. 63–66, 2004.

[54] M. Püschel and J. M. Moura, "The algebraic approach to the discrete cosine and sine transforms and their fast algorithms," *SIAM Journal on Computing*, vol. 32, no. 5, pp. 1280–1316, 2003.

[55] M. Mehendale, S. D. Sherlekar, and G. Venkatesh, "Algorithmic and architectural transformations for low power realization of FIR filters," in *International Conference on VLSI Design*, pp. 12–17, 1998.

[56] A. Erdogan, M. Hasan, and T. Arslan, "Algorithmic low power FIR cores," *IEE Proceedings-Circuits, Devices and Systems*, vol. 150, no. 3, pp. 155–160, 2003.

[57] Y. Voronenko and M. Püschel, "Multiplierless multiple constant multiplication," *ACM Transactions on Algorithms*, vol. 3, no. 2, pp. 11–50, 2007.

[58] M. M. Ozdal, C. Amin, A. Ayupov, S. M. Burns, G. R. Wilke, and C. Zhuo, "An improved benchmark suite for the ISPD-2013 discrete cell sizing contest," in *International Symposium on Physical Design*, pp. 168–170, 2013.

[59] I. Toma, E. Simperl, and G. Hench, "A joint roadmap for semantic technologies and the Internet of Things," in *Proceedings of the Third STI Roadmapping Workshop*, vol. 1, 2009.

[60] Novel, "Pedar." `http://www.novel.de`, 2007.

[61] H. Noshadi, F. Dabiri, S. Meguerdichian, M. Potkonjak, and M. Sarrafzadeh, "Energy optimization in wireless medical systems using physiological behavior," in *Wireless Health*, pp. 128–136, 2010.

[62] J. M. VanSwearingen, K. A. Paschal, P. Bonino, and J.-F. Yang, "The modified gait abnormality rating scale for recognizing the risk of recurrent falls in community-dwelling elderly adults," *Physical Therapy*, vol. 76, no. 9, pp. 994–1002, 1996.

[63] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, 2009.

[64] V. Leonov, P. Fiorini, S. Sedky, T. Torfs, and C. Van Hoof, "Thermoelectric MEMS generators as a power supply for a body area network," in *International Conference on Solid-State Sensors, Actuators and Microsystems*, vol. 1, pp. 291–294, 2005.

[65] Y. Liu, B. Veeravalli, and S. Viswanathan, "Critical-path based low-energy scheduling algorithms for body area network systems," in *International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pp. 301–308, 2007.

[66] S. Xiao, A. Dhamdhere, V. Sivaraman, and A. Burdett, "Transmission power control in body area sensor networks for healthcare monitoring," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 1, pp. 37–48, 2009.

[67] K. Lorincz *et al.*, "Mercury: A wearable sensor network platform for high-fidelity motion analysis.," in *SenSys*, vol. 9, pp. 183–196, 2009.

[68] A. Krause, D. P. Siewiorek, A. Smailagic, and J. Farringdon, "Unsupervised, dynamic identification of physiological and activity context in wearable computing," in *IEEE International Symposium on Wearable Computers*, pp. 88–97, 2003.

[69] M. Rofouei, M. A. Ghodrat, M. Potkonjak, and A. Martinez-Nova, "Optimization intensive energy harvesting," in *Design, Automation and Test in Europe (DATE)*, pp. 272–275, 2012.

[70] I. P. Pappas, T. Keller, S. Mangold, M. R. Popovic, V. Dietz, and M. Morari, "A reliable gyroscope-based gait-phase detection sensor embedded in a shoe insole," *IEEE Sensors Journal*, vol. 4, no. 2, pp. 268–274, 2004.

[71] K. Oshima, Y. Ishida, S. Konomi, N. Thepvilojanapong, and Y. Tobe, "A human probe for measuring walkability," in *ACM Conference on Embedded Networked Sensor Systems*, pp. 353–354, 2009.

[72] V. Erickson, A. U. Kamthe, and A. E. Cerpa, "Measuring foot pronation using RFID sensor networks," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pp. 325–326, 2009.

[73] J. B. Wendt and M. Potkonjak, "Medical diagnostic-based sensor selection," in *IEEE Sensors*, pp. 1507–1510, 2011.

[74] J. B. Wendt, S. Meguerdichian, H. Noshadi, and M. Potkonjak, "Energy and cost reduction in localized multisensory systems through application-driven compression," in *Data Compression Conference (DCC)*, pp. 411–411, 2012.

[75] J. B. Wendt, S. Meguerdichian, and M. Potkonjak, "Small is beautiful and smart," in *Telehealthcare Computing and Engineering: Principles and Design* (F. Hu, ed.), pp. 341–358, CRC Press, 2013.

[76] H. Noshadi, S. Ahmadian, H. Hagopian, J. Woodbridge, F. Dabiri, N. Amini, M. Sarrafzadeh, and N. Terrafranca, "Hermes - mobile balance and instability assessment system.," in *BIOSIGNALS*, pp. 264–270, 2010.

[77] S. Meguerdichian, H. Noshadi, F. Dabiri, and M. Potkonjak, "Semantic multimodal compression for wearable sensing systems," in *IEEE Sensors*, pp. 1449–1453, 2010.

[78] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[79] W. Wu *et al.*, "The SmartCane system: An assistive device for geriatrics," in *International Conference on Body Area Networks*, p. 2, 2008.

[80] E. Hoque and J. A. Stankovic, "Monitoring quantity and quality of sleeping using WISPs," in *International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 370–371, 2010.

[81] R. F. Dickerson, E. I. Gorlin, and J. A. Stankovic, "Empath: A continuous remote emotional health monitoring system for depressive illness," in *Wireless Health*, p. 5, 2011.

[82] V. Goudar and M. Potkonjak, "Power constrained sensor sample selection for improved form factor and lifetime in localized bans," in *Wireless Health*, p. 5, 2012.

[83] V. Goudar and M. Potkonjak, "Energy-efficient sampling schedules for body area networks," in *IEEE Sensors*, pp. 1–4, 2012.

[84] J. A. Paradiso and T. Starner, "Energy scavenging for mobile and wireless electronics," *IEEE Pervasive Computing*, vol. 4, no. 1, pp. 18–27, 2005.

[85] P. D. Mitcheson, E. M. Yeatman, G. K. Rao, A. S. Holmes, and T. C. Green, "Energy harvesting from human and machine motion for wireless electronic devices," *Proceedings of the IEEE*, vol. 96, no. 9, pp. 1457–1486, 2008.

[86] J. A. Paradiso, "Systems for human-powered mobile computing," in *Design Automation Conference (DAC)*, pp. 645–650, 2006.

[87] R. D. Kornbluh *et al.*, "From boots to buoys: promises and challenges of dielectric elastomer energy harvesting," in *Electroactivity in Polymeric Materials*, pp. 67–93, Springer, 2012.

[88] M. A. Hanson *et al.*, "Body area sensor networks: Challenges and opportunities," *Computer*, vol. 42, no. 1, p. 58, 2009.

[89] V. Goudar and M. Potkonjak, "Dielectric elastomer generators for foot plantar pressure based energy scavenging," in *IEEE Sensors*, 2012.

[90] C. Jean-Mistral, S. Basrour, and J. Chaillout, "Modelling of dielectric polymers for energy scavenging applications," *Smart Materials and Structures*, vol. 19, no. 10, p. 105006, 2010.

[91] "3m VHB$^{\text{TM}}$ tapes," *Technical Data Sheet*.

[92] B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas, "Silicon physical random functions," in *Computer and Communications Security (CCS)*, pp. 148–160, 2002.

[93] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Design Automation Conference (DAC)*, pp. 9–14, 2007.

[94] J. Guajardo *et al.*, "Anti-counterfeiting, key distribution, and key storage in an ambient world via physical unclonable functions," *Information Systems Frontiers*, vol. 11, no. 1, pp. 19–41, 2009.

[95] T. Xu, J. B. Wendt, and M. Potkonjak, "Secure remote sensing and communication using digital PUFs," in *Symposium on Architectures for Networking and Communications Systems (ANCS)*, pp. 1–12, 2014.

[96] T. Xu, J. B. Wendt, and M. Potkonjak, "Matched digital PUFs for low power security in implantable medical devices," in *International Conference on Healthcare Informatics (ICHI)*, pp. 33–38, 2014.

[97] T. Xu, J. B. Wendt, and M. Potkonjak, "Digital bimodal function: An ultra-low energy security primitive," in *International Symposium on Low Power Electronics and Design (ISLPED)*, pp. 292–296, 2013.

[98] J. Zheng, D. Li, and M. Potkonjak, "A secure and unclonable embedded system using instruction-level PUF authentication," in *Field Programmable Logic and Applications (FPL)*, pp. 1–4, 2014.

[99] J. Zheng and M. Potkonjak, "DPUF: A reconfigurable IP protection architecture for embedded systems," in *Symposium on Architectures for Networking and Communications Systems (ANCS)*, pp. 1–2, 2014.

[100] T. Xu and M. Potkonjak, "Robust and flexible FPGA-based digital PUF," in *Field Programmable Logic and Applications (FPL)*, pp. 1–6, 2014.

[101] J. B. Wendt and M. Potkonjak, "Nanotechnology-based trusted remote sensing," in *IEEE Sensors*, pp. 1213–1216, 2011.

[102] J. B. Wendt and M. Potkonjak, "The bidirectional polyomino partitioned PPUF as a hardware security primitive," in *Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 257–260, 2013.

[103] U. Ruhrmair, S. Devadas, and F. Koushanfar, *Security based on physical unclonability and disorder*. Springer, 2011.

[104] M. Potkonjak and V. Goudar, "Public physical unclonable functions," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1142–1156, 2014.

[105] F. Luellau, T. Hoepken, and E. Barke, "A technology independent block extraction algorithm," in *Design Automation Conference (DAC)*, pp. 610–615, 1984.

[106] S. Blythe, B. Fraboni, S. Lall, H. Ahmed, and U. de Riu, "Layout reconstruction of complex silicon chips," *IEEE Journal of Solid-State Circuits*, vol. 28, no. 2, pp. 138–145, 1993.

[107] R. Nakagaki, T. Honda, and K. Nakamae, "Automatic recognition of defect areas on a semiconductor wafer using multiple scanning electron microscope images," *Measurement Science and Technology*, vol. 20, no. 7, p. 075503, 2009.

[108] Y. Ren, Y. Shi, and B.-H. Gwee, "A novel gate-level to behavior-level conversion algorithm with high microcell identifiation rate," in *IASTED International Conference*, vol. 712, p. 138, 2010.

[109] R. Torrance and D. James, "The state-of-the-art in IC reverse engineering," in *Cryptographic Hardware and Embedded Systems (CHES)*, pp. 363–381, 2009.

[110] R. Torrance and D. James, "The state-of-the-art in semiconductor reverse engineering," in *Design Automation Conference (DAC)*, pp. 333–338, 2011.

[111] P. Subramanyan, N. Tsiskaridze, K. Pasricha, D. Reisman, A. Susnea, and S. Malik, "Reverse engineering digital circuits using functional analysis," in *Design, Automation and Test in Europe (DATE)*, pp. 1277–1280, 2013.

[112] W. Li *et al.*, "WordRev: Finding word-level structures in a sea of bit-level gates," in *International Symposium on Hardware-Oriented Security and Trust (HOST)*, pp. 67–74, 2013.

[113] K. Nohl, D. Evans, S. Starbug, and H. Plötz, "Reverse-engineering a cryptographic RFID tag," in *USENIX Security Symposium*, vol. 28, 2008.

[114] D. Nedospasov, J.-P. Seifert, A. Schlosser, and S. Orlic, "Functional integrated circuit analysis," in *International Symposium on Hardware-Oriented Security and Trust (HOST)*, pp. 102–107, 2012.

[115] Y. Alkabani, F. Koushanfar, and M. Potkonjak, "Remote activation of ICs for piracy prevention and digital right management," in *International Conference on Computer-Aided Design (ICCAD)*, pp. 674–677, 2007.

[116] A. Baumgarten, A. Tyagi, and J. Zambreno, "Preventing IC piracy using reconfigurable logic barriers," *IEEE Design and Test of Computers*, vol. 27, no. 1, pp. 66–75, 2010.

[117] J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri, "Security analysis of logic obfuscation," in *Design Automation Conference (DAC)*, pp. 83–89, 2012.

[118] J. Rajendran, M. Sam, O. Sinanoglu, and R. Karri, "Security analysis of integrated circuit camouflaging," in *Computer and Communications Security (CCS)*, pp. 709–720, 2013.

[119] A. B. Kahng *et al.*, "Watermarking techniques for intellectual property protection," in *Design Automation Conference (DAC)*, pp. 776–781, 1998.

[120] C. Helfmeier, D. Nedospasov, C. Tarnovsky, J. S. Krissler, C. Boit, and J.-P. Seifert, "Breaking and entering through the silicon," in *Computer and Communications Security (CCS)*, pp. 733–744, 2013.

[121] "Implementation of security in Actel's ProASIC and ProASIC$^{\text{PLUS}}$ flash-based FPGAs." `http://www.actel.com/documents/Flash_Security_AN.pdf`, 2003.

[122] A. Bogdanov *et al.*, *PRESENT: An ultra-lightweight block cipher*. Springer, 2007.

[123] D. Hong *et al.*, "HIGHT: a new block cipher suitable for low-resource device," in *Cryptographic Hardware and Embedded Systems (CHES)*, pp. 46–59, 2006.

[124] J. Daemen and V. Rijmen, *The design of Rijndael: AES-the advanced encryption standard*. Springer Science & Business Media, 2002.

[125] N. Beckmann and M. Potkonjak, "Hardware-based public-key cryptography with public physically unclonable functions," in *Information Hiding*, pp. 206–220, 2009.

[126] M. Potkonjak, S. Meguerdichian, A. Nahapetian, and S. Wei, "Differential public physically unclonable functions: Architecture and applications," in *Design Automation Conference (DAC)*, pp. 242–247, 2011.

[127] S. Meguerdichian and M. Potkonjak, "Device aging-based physically unclonable functions," in *Design Automation Conference (DAC)*, pp. 288–289, 2011.

[128] D. E. Holcomb, W. P. Burleson, and K. Fu, "Power-up SRAM state as an identifying fingerprint and source of true random numbers," *IEEE Transactions on Computers*, vol. 58, no. 9, pp. 1198–1210, 2009.

[129] F. Koushanfar and G. Qu, "Hardware metering," in *Design Automation Conference (DAC)*, pp. 490–493, 2001.

[130] Y. Alkabani and F. Koushanfar, "Active hardware metering for intellectual property protection and security," in *USENIX Security Symposium*, pp. 291–306, 2007.

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

[132] M. Potkonjak, A. Nahapetian, M. Nelson, and T. Massey, "Hardware Trojan horse detection using gate-level characterization," in *Design Automation Conference (DAC)*, pp. 688–693, 2009.

[133] Y. Alkabani, F. Koushanfar, N. Kiyavash, and M. Potkonjak, "Trusted integrated circuits: A nondestructive hidden characteristics extraction approach," in *Information Hiding*, pp. 102–117, 2008.

[134] S. Wei and M. Potkonjak, "Scalable hardware Trojan diagnosis," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 6, pp. 1049–1057, 2012.

[135] D. E. Knuth, *The Art of Computer Programming, Vol. 3: Sorting and Searching*. Addison-Wesley Professional, 1998.

[136] S. Wei, A. Nahapetian, and M. Potkonjak, "Robust passive hardware metering," in *International Conference on Computer-Aided Design (ICCAD)*, pp. 802–809, 2011.