

Old Dominion University

ODU Digital Commons

Electrical & Computer Engineering Theses & Dissertations

Electrical & Computer Engineering

Summer 2014

IDPAL – A Partially-Adiabatic Energy-Efficient Logic Family: Theory and Applications to Secure Computing

Mihail T. Cutitaru
Old Dominion University

Follow this and additional works at: https://digitalcommons.odu.edu/ece_etds



Part of the [Hardware Systems Commons](#), [Power and Energy Commons](#), [Theory and Algorithms Commons](#), and the [VLSI and Circuits, Embedded and Hardware Systems Commons](#)

Recommended Citation

Cutitaru, Mihail T.. "IDPAL – A Partially-Adiabatic Energy-Efficient Logic Family: Theory and Applications to Secure Computing" (2014). Doctor of Philosophy (PhD), Dissertation, Electrical & Computer Engineering, Old Dominion University, DOI: 10.25777/v8qq-6742
https://digitalcommons.odu.edu/ece_etds/63

This Dissertation is brought to you for free and open access by the Electrical & Computer Engineering at ODU Digital Commons. It has been accepted for inclusion in Electrical & Computer Engineering Theses & Dissertations by an authorized administrator of ODU Digital Commons. For more information, please contact digitalcommons@odu.edu.

**IDPAL – A PARTIALLY-ADIABATIC ENERGY-EFFICIENT
LOGIC FAMILY: THEORY AND APPLICATIONS TO SECURE
COMPUTING**

by

Mihail T. Cutitaru
B.S. May 2010, Old Dominion University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

ELECTRICAL AND COMPUTER ENGINEERING

OLD DOMINION UNIVERSITY
August 2014

Approved by:

Lee A. Belfor, II (Director)

Chung-Hao Chen (Member)

Khan Iftekharuddin (Member)

Filip Cuckov (Member)

ABSTRACT

IDPAL – A PARTIALLY-ADIABATIC ENERGY-EFFICIENT LOGIC FAMILY: THEORY AND APPLICATIONS TO SECURE COMPUTING

Mihail T. Cutitaru
Old Dominion University, 2014
Director: Dr. Lee A. Belfore, II

Low-power circuits and issues associated with them have gained a significant amount of attention in recent years due to the boom in portable electronic devices. Historically, low-power operation relied heavily on technology scaling and reduced operating voltage, however this trend has been slowing down recently due to the increased power density on chips. This dissertation introduces a new very-low power partially-adiabatic logic family called Input-Decoupled Partially-Adiabatic Logic (IDPAL) with applications in low-power circuits. Experimental results show that IDPAL reduces energy usage by 79% compared to equivalent CMOS implementations and by 25% when compared to the best adiabatic implementation. Experiments ranging from a simple buffer/inverter up to a 32-bit multiplier are explored and result in consistent energy savings, showing that IDPAL could be a viable candidate for a low-power circuit implementation.

This work also shows an application of IDPAL to secure low-power circuits against power analysis attacks. It is often assumed that encryption algorithms are perfectly secure against attacks, however, most times attacks using side channels on the hardware implementation of an encryption operation are not investigated. Power analysis attacks are a subset of side channel attacks and can be implemented by measuring the power used by a circuit during an encryption operation in order to obtain secret information from the circuit under attack. Most of the previously proposed solutions for power analysis attacks use a large amount of power and are unsuitable for a low-power application. The almost-equal energy consumption for any given input in an IDPAL circuit suggests that this logic family is a good candidate for securing low-power circuits against power analysis attacks. Experimental results ranging from small circuits to large multipliers are performed and the power-analysis attack resistance of IDPAL is investigated. Results show that IDPAL circuits are not only low-power but also the most secure against power analysis attacks when compared to other adiabatic low-power circuits.

Finally, a hybrid adiabatic-CMOS microprocessor design is presented. The proposed microprocessor uses IDPAL for the implementation of circuits with high switching activity (e.g. ALU) and CMOS logic for other circuits (e.g. memory, controller). An adiabatic-CMOS interface for transforming adiabatic signals to square-wave signals is presented and issues associated with a hybrid implementation and their solutions are also discussed.

DEDICATION

This dissertation is dedicated to the memory of my father, Tudor Cutitaru.

ACKNOWLEDGEMENTS

The work presented in this dissertation is a result of many years of persistent research and would not have been possible without the help, guidance, and support from many people.

I would like to thank my advisor Dr. Lee Belfore for all the advice, direction, and patience extended to me in the past several years. His professionalism and attention to detail is reflected throughout this dissertation.

I am also grateful to my dissertation committee members, Dr. Chung-Hao Chen, Dr. Khan Iftekharuddin, and Dr. Filip Cuckov, for their help, comments, and suggestions during my research.

The Department of Electrical and Computer Engineering has provided an excellent environment for my academic growth and I am thankful for the opportunity to be a part of it. I would like to thank Deborah Kinney, Linda Marshall, and Romina Samson for all their help throughout the years in the program.

My American family, Jim and Claire Carbone, have been my pillar of support through my college years and I will be forever grateful to them for their love, care, and support.

I would also like to thank Ana Mananquil for all her love and patience during my studies and encouragement during times of need.

Last, but not least, I would like to extend my deepest gratitude to my family, my mother, Viorica, and brothers, Leonid and Rodion, for their support in the pursuit of my dreams.

TABLE OF CONTENTS

	Page
LIST OF TABLES	viii
LIST OF FIGURES	xi
Chapter	
I INTRODUCTION	1
I.1 Review of Complementary Metal-Oxide Semiconductor Devices	2
I.2 Hardware Security and Power Analysis Attacks	6
I.3 Problem Statement	9
I.4 Dissertation Contributions	10
I.5 Dissertation Overview	11
II BACKGROUND AND RELATED WORKS	12
II.1 Background in Adiabatic Computing	12
II.2 Power Analysis Attacks Background	24
III IDPAL – INPUT DECOUPLED PARTIALLY-ADIABATIC LOGIC	35
III.1 IDPAL Operating Principle	35
III.2 Results for Large Arithmetic Circuits	49
III.3 Chapter Conclusion	60
IV POWER ANALYSIS ATTACK SECURITY USING IDPAL	61
IV.1 Motivation for Combinational Circuit Security	61
IV.2 Security Metrics and Power Models Used	63
IV.3 Experimental Results	65
IV.4 Results for Large Circuits	75
IV.5 Chapter Conclusions	81
V DESIGN OF PROOF OF CONCEPT MICROPROCESSOR	83
V.1 Implementation Constraints and Customizable Features	83
V.2 Hybrid Microprocessor Design Issues	84
V.3 Implementation Details	90
V.4 Microprocessor Testing	98
V.5 Chapter Conclusion	99
VI FUTURE WORK	101
VII CONCLUSIONS	104
BIBLIOGRAPHY	106
VITA	119

LIST OF TABLES

Table	Page
1 Truth table for a 2-input NAND gate.	26
2 All 16 possible input transitions for a 2-input NAND gate and type of energy consumption for each transition in a CMOS implementation. . . .	28
3 Energy dissipation (J) per cycle for each type of gate and load at 100 MHz.	47
4 Number of each type of gate and total transistor count for the 8-, 16-, and 32-bit KSAs implemented in IDPAL.	53
5 Energy dissipation per addition operation at different frequencies and number of transistors needed for each implementation of the 32-bit KSA.	54
6 Security metrics for the NAND/AND gate for each logic family.	72
7 All 16 possible input transitions for a 2-input NAND/AND gate and energy consumption values for each transition in IDPAL.	73
8 Input test cases for the 32-bit KSA and Wallace multiplier.	77
9 Minimum, maximum, and average standard deviation values for current waveforms for a 32-bit KSA in the adiabatic families tested.	78
10 Security metrics for the 32-bit KSA for each adiabatic logic family.	79
11 Security metrics for the 32-bit Wallace multiplier for each adiabatic logic family.	81

LIST OF FIGURES

Figure	Page
1 CMOS inverter and its equivalent circuit.	14
2 2-phase sinusoidal and 4-phase trapezoidal <i>PC</i> 's with each stage marked for the first phase.	16
3 2N2P buffer/inverter circuit.	20
4 2N-2N2P buffer/inverter circuit.	22
5 CAL buffer/inverter circuit.	23
6 TSEL PMOS buffer/inverter circuit.	24
7 A 2-input CMOS NAND gate showing charging (dotted) and discharging (dot+line) paths.	27
8 A 2-input SABL NAND gate.	32
9 A 2-input WDDL NAND gate.	34
10 General gate construction in IDPAL.	36
11 IDPAL buffer/inverter.	37
12 Equivalent RC circuit for an IDPAL buffer/inverter.	38
13 Equivalent RC circuit for an IDPAL buffer/inverter with one of the inputs conducting.	39
14 Equivalent RC circuit breakdown for an IDPAL 2-input AND/NAND gate.	39
15 Simulation waveform for an IDPAL buffer/inverter.	42
16 Energy consumption plot for a single IDPAL buffer/inverter with alternating inputs.	43
17 Simulation waveform for a chain of 5 IDPAL buffers. The input is applied at $t = 0.75\mu s$ and output propagates to output of fifth buffer by $t = 3.25\mu s$	45
18 Layout of IDPAL and 2N2P AND/NAND and CMOS NAND gates in a 3-metal, 2-poly process.	46

19	IDPAL full-adder implementation.	48
20	IDPAL full-adder simulation waveforms. Inputs cycle from “111” down to “000”.	49
21	Energy usage (J)/cycle for a full-adder with 10fF load at different frequencies.	50
22	Diagram of a 32-bit Kogge-Stone Adder [31].	52
23	Energy usage (J)/operation for a 32-bit KSA with a 10fF load at different frequencies.	53
24	Energy usage for a 32-bit KSA at frequencies from 1MHz to 1GHz. . . .	56
25	Energy usage (J)/operation for a 32-bit Wallace multiplier with a 10fF load at different frequencies.	58
26	Energy usage/operation for a 32-bit KSA with a 10fF load at different frequencies in a TSMC 180nm process.	59
27	Current trace for a CMOS inverter.	66
28	Current trace for a 2-input SABL AND/NAND gate.	67
29	Current trace for a 2-input WDDL AND/NAND gate.	67
30	Current trace for a 2-input CAL AND/NAND gate.	68
31	Current trace for a 2-input 2N2P AND/NAND gate.	68
32	Current trace for a 2-input IDPAL AND/NAND gate.	69
33	Current traces for all AND/NAND gates in the experiment.	71
34	Current traces for all 16 possible transitions for the IDPAL AND/NAND gate.	74
35	Current traces for even (4) vs. odd number (5) of evaluation stages for IDPAL inverters.	76
36	Average and standard deviation current traces for 8 test cases of a 32-bit KSA in CAL, 2N2P, and IDPAL for one of the PCs.	80
37	An adiabatic-CMOS interface circuit used in the proposed microprocessor.	87
38	Output signal with and without ripples after conversion from an adiabatic input.	88

39	Block diagram of the hybrid adiabatic-CMOS microprocessor.	91
40	High-level ALU block diagram implementation.	94
41	Implementation of a 2-input XOR gate in IDPAL used as a swap gate. . .	94
42	An 8-bit bi-directional barrel shifter implementing the <i>SRL</i> , <i>SRA</i> , <i>SLL</i> , <i>ROR</i> instructions.	96

CHAPTER I

INTRODUCTION

The tremendous success of the microprocessor industry has been driven by the scalability of the transistors over the past 50 years according to Moore's predictions [5]. Heat dissipation has become a major issue in circuit design given the high transistor density, even in the area of low power design. The usual solution has heavily relied on reducing the operating voltage with each decrease in feature size, but current processes are approaching the limits of scaling due to limitations on the threshold voltage of each technology node. Landauer noted [6] that the minimum energy dissipation per switching event will ideally be no less than $kT\ln 2$ J, where k is Boltzmann's constant and T is the operating temperature in degrees Kelvin. Many techniques are currently in use to reduce the power consumption of circuits, but the current value for energy dissipation per bit erased is still almost three orders of magnitude larger than the theoretical limit calculated by Landauer. This prompts for new solutions to reduce the energy dissipation in low power microprocessors in order to extend the battery life of devices.

Continued development and miniaturization of transistor features has allowed for a surge in the number of portable devices used daily by society. Current devices provide more processing power than full-size desktops from ten years ago, which has prompted some people to move their digital identities to these devices due to many security features embedded in applications used. In recent years, extensive research has shown

that encryption modules in cellular phones, smart cards, and other specialized hardware are prone to attacks that use side channels for extraction of secret information [47, 62, 66, 72, 79, 80, 81, 82, 84, 85]. Side-channel attacks are attacks that gather secret information by exploiting channels other than the traditional communication channels in these devices. There is no indication in literature that any hardware security measures are implemented in current portable electronic devices, leaving them exposed for possible attacks because of the amount of information that an attacker stands to gain from a successful attempt. This problem suggests that power analysis attack countermeasures would be a valuable feature for low power electronic devices.

I.1 Review of Complementary Metal-Oxide Semiconductor Devices

The workhorse of the semiconductor industry for more than 50 years has been the Complementary Metal-Oxide Semiconductor (CMOS) process. CMOS circuits are built using two kinds of transistors: PMOS (p-channel metal-oxide-semiconductor field-effect transistor) and NMOS (n-channel metal-oxide-semiconductor field-effect transistor), arranged in complementary networks. The PMOS networks implement the desired function and are called pull-up networks (PUN) and the NMOS networks implement the complementary function and are called pull-down networks (PDN). CMOS circuits are digital circuits and, as such, take values at either of the two discrete potentials: High ('1' or V_{dd}) and Low ('0' or GND). Because of these two distinct values, charge is always being shuffled from the source (V_{dd}) to the output nodes to change it to a '1' or from the output nodes to the GND to change the value to a '0'. Furthermore, when no signals are

changing, CMOS circuits consume very little power, making them an attractive choice for implementation of electronic circuits.

Energy dissipation in CMOS devices occurs due to three distinct components: static/leakage, short-circuit, and dynamic [7]. Static power dissipation occurs when the value of a node is kept constant and is due to the reverse current of the PN junctions in a transistor. Leakage power is due to the subthreshold conduction current present between the source and drain of a MOS device. The subthreshold conduction current can be decreased by increasing the threshold voltage, although this also leads to the need for wider devices and higher dynamic energy dissipation. In larger technology nodes, the contribution of leakage energy dissipation is very small, however, with decreased feature sizes in modern processes, this component increases exponentially. The second component is the contribution from short circuit energy dissipation, which occurs when the output node switches values and both the PUN and PDN are briefly conducting at the same time forming a path between V_{dd} and GND . This is a very short event and has a small contribution to the overall energy dissipation. The last and largest component is the dynamic energy dissipation, also known as active losses. This occurs when a given combination of inputs causes the output node to switch values, either by charging or discharging. It is widely known that for a rail-to-rail voltage of V_{dd} and output node capacitance C , the energy required to bring a node from GND to V_{dd} is $E = CV^2$, with half of this energy dissipated in the transistor network as heat during charging. Since this component has the largest contribution to the overall power dissipation, the main focus of low power design has been to reduce it with every new technology node.

I.1.1 Low Power Devices and Techniques

Historically, CMOS circuits have been employed where low power consumption is desired. In many applications where the systems are operated at low frequencies, CMOS has performed well. The subject of low-power has not seen much interest for many years except for a small subset of devices such as medical implants. It was not until the proliferation of portable electronic devices and the continued technology scaling that the focus has shifted to low power design and challenges associated with it. Low power design was not an issue earlier since the device density was much smaller than it is now. Given Moore's Law, we expect to have a larger number of faster transistors packaged in the same area, providing an opportunity for new low power technologies to allow the trend to continue by providing more efficient heat dissipation and reducing the thermal impacts that result in circuit deterioration.

The miniaturization of transistor feature sizes has allowed for low power circuits to flourish and be used in many new areas. As an example, in the recent years we have seen explosive growth in the use of portable electronic devices, especially in the cell phone arena. It is now possible to check email, browse the Internet, and do many other useful activities from a handheld device, which was not possible a decade ago. However, this has also brought attention to many existing constraints, e.g. we want smaller, lighter, and more powerful devices than ever before. Traditional solutions, such as increasing the battery size (and consequently its weight) to provide longer battery life are not feasible [7]. Indeed, companies market external batteries to extend the battery life of a portable device by a few hours. However, this does not change the fact that with regular use, a modern

smartphone lasts less than a day without needing a charge. The general conclusion is that batteries alone will not solve the low power problem and alternate solutions are highly desirable.

One of the easiest and most effective ways to reduce dynamic dissipation is by reducing the operating voltage of a circuit, since energy dissipation is proportional to the square of the operating voltage ($E = CV^2$). Many trade-offs are associated with voltage reduction. Most significantly, a loss in performance is observed since transistors are slower because of the lower operating voltages. This is due to the non-linear scaling of the transistor threshold voltage with the operating voltage in order to avoid leakage current. Noise immunity is also reduced given a lower voltage swing. Additionally, level converters are required in order to interface low swing signals to full swing signals.

Some of the other techniques for lower power are to reduce parasitic capacitance, switching frequency, and leakage and static currents. Reducing parasitic capacitance is one of the standard steps in modern Computer-Aided Design (CAD) software, although a very thorough understanding of the full circuit design process is required. Switching frequency reduction has the same effect as reducing capacitance as it is best applied to nodes with high capacitance, so nodes with smaller capacitances can only see a small reduction in power consumption. Reducing the switching frequency also has positive impact on the reliability of a chip. Static and leakage currents are the most difficult to control since they are highly dependent on the process technology, although some level of control can be achieved by transistor sizing and layout [7].

All low power techniques attempt to reduce the energy dissipation by making strategic changes to CMOS circuit parameters; however, they do not solve the issue that energy dissipation is a function of how CMOS circuits operate. A revolutionary change in how circuits are built, clocked, and powered is necessary if any additional reductions in energy dissipation are to be expected. Several other types of logic families have been explored by the research community to replace CMOS logic for low power circuits, one of which is adiabatic computing. Adiabatic computing works under the principle that if signal voltages are changed gradually to evaluate logic functions and then gradually reclaiming charge when done, resistive losses are reduced and average power dissipation can be greatly reduced. This gradual charging avoids the high resistance imposed by the transistor network and the current peak associated with it. Adiabatic logic requires a different design flow, custom libraries, and special considerations in order for the circuits to operate properly, but it provides a significant reduction in energy consumption.

1.2 Hardware Security and Power Analysis Attacks

The second area explored in this dissertation is hardware security. Security of computation is usually incorporated at the algorithmic level in software, with most encryption algorithms being very secure from an algorithmic and mathematical standpoint. Attacks on mainstream encryption algorithms have proven very difficult and many security precautions have been taken for these algorithms to be secure at the software level. However, hardware implementations of these encryption algorithms may not always be secure and there is only a small amount of research in this area. Some of the applications of these

encryption tasks are user ID, phone card, secure authentication tokens, smart cards, and, more recently, bank cards. Oftentimes these circuits are not secured from attacks that could be staged using non-traditional communication channels, also known as side channels. This kind of security attack is also called “implementation attacks” as their main focus is the physical implementation of otherwise-secure algorithms. As society moves towards using more and more portable devices to conduct everyday business, hardware security becomes vastly more important. In recent years, we have also observed an increase in the use of smart cards being used as bank cards, mass transit cards, chips in biometric passports, and other applications [82], which makes them more prone to hardware attacks. Side-channel attacks take advantage of the power used, time elapsed, electromagnetic radiation emitted, and even sound waves from computer hardware during computation in order to extract secret information. These side-channel attacks are especially important in secure environments, e.g. military, banks, corporations, and require using systems that have built-in hardware security.

Attacks on cryptographic devices are mainly categorized based on two criteria: 1) invasive vs. non-invasive and 2) passive vs. active. Invasive attacks are the most powerful because there is no limit on what can be done to the device. Invasive attacks also require an exact copy of the device under attack, detailed implementation knowledge, and very expensive tools. Non-invasive attacks can be implemented with relatively inexpensive equipment and pose a more serious threat to the security of cryptographic devices because this attack leaves no detectable evidence of tampering. Active attacks are implemented by an attacker tampering with the device in order for it to behave abnormally, which can

lead it to reveal secret information. This kind of attacks are very powerful, but require detail knowledge and a sophisticated setup. Passive attacks, on the other hand, are staged during normal operation of a cryptographic device and are based on observing physical properties of the device. They are implemented using readily-available hardware and software, are simple to set up, and are the least costly. In this work, the focus is on power analysis attacks, which fall in the category of non-invasive passive attacks, and their countermeasures.

Power analysis attacks rely on the fact that CMOS circuits will consume a different amount of energy for different inputs to a circuit. This unequal energy consumption can be exploited by an attacker to reveal the secret key by measuring the power usage for several inputs. For example, presenting a certain set of inputs can result in an increased probability of correctly guessing part of the secret key, thus weakening the encryption and potentially compromising devices. Power analysis attacks can be implemented even on very noisy power traces or without any knowledge of the device under test, which makes this kind of attack of great concern in the future of computing hardware. Attacks can be staged on circuits of any size and power characteristics, ranging from a personal computer to smart cards used in the banking industry to implanted medical devices. Current power analysis countermeasures are mainly aimed at circuits that are not limited in their energy budget, i.e. non-battery-operated devices, and little research exists in the area of power analysis attacks countermeasures on low power devices. An example of energy-limited devices with a need for hardware security are implantable medical devices, which have limited energy due to their size and the consequences of power analysis attacks could be

fatal.

I.3 Problem Statement

The abundance of low power portable devices in use today continues to grow and this growth is expected to continue at least at its current rate. Circuit designers need to devise new techniques to make these devices use less power while delivering a higher quality user experience with the same or longer battery life. Recent developments in the areas of sensors, implantable medical devices, and wearable computing (e.g. smart watches, Google Glass) prompt for new low power solutions at small sizes. Devices with very low power dissipation could also allow for deployment of systems with solar cells as a power source, making the systems self-sustaining and decreasing the interval between battery replacements. As the traditional energy dissipation reduction rate slows down, new avenues for reducing energy dissipation in computing circuits need to be explored.

At the same time, as portable devices get even more integrated into our everyday lives, privacy and security issues become more important. As more and more users move their digital lives onto portable devices and continue assuming that hardware implementation of security algorithms are secure, security breaches can prove more devastating than ever before. Many security features are in place for protection against software attacks, e.g. viruses and other malware, but there are no indications of any hardware security countermeasures implemented in these circuits. Given the possible consequences of a power analysis attack on a cryptographic circuit, it is necessary to make circuits resistant to this type of attacks.

These two separate problems have not been investigated as one in the literature, but given the miniaturization of circuits and their tighter integration in daily lives, power analysis attacks on low power circuits become a reality with possibly devastating consequences. The few proposed solutions to this problem either solve the low power problem or the power analysis problem, but not both. A solution that covers both issues is needed in order to protect low power circuits from this kind of attacks.

I.4 Dissertation Contributions

This work aims to provide a novel solution that provides lower power operation and resistance to power analysis encryption attacks by introducing a new very low power logic family named Input Decoupled Partially-Adiabatic Logic (IDPAL). IDPAL is a new adiabatic logic family that uses a two-phase sinusoidal Power Clock and has very low variation in energy dissipation between input values. IDPAL shows promising results in reducing the energy used in computations by up to 79% compared to standard CMOS implementations and up to 25% compared to next-best adiabatic implementation. It also increases the resistance against power analysis attacks when compared to state-of-the-art secure logic families. A proof-of-concept hybrid adiabatic-CMOS microprocessor design using IDPAL is presented. Issues associated with a hybrid dual-rail microprocessor design and adiabatic-CMOS interfacing are discussed.

I.5 Dissertation Overview

The rest of the dissertation follows a traditional format. Chapter II gives a background of CMOS computing, explains the basics of adiabatic computing, and discusses some of the most well-known adiabatic logic families. It also gives some background in hardware security, power analysis attacks, and existing countermeasures. Chapter III presents IDPAL, explains its operation principle, and provides simulation results from circuits of different sizes. Chapter IV describes the application of IDPAL to hardware security, specifically as a power analysis countermeasure, using circuits of different sizes. Chapter V describes the implementation of a hybrid adiabatic-CMOS microprocessor using IDPAL for the implementation of adiabatic blocks. Chapter VI presents some areas of future work and other applications for IDPAL and Chapter VII summarizes the contributions of this dissertation.

CHAPTER II

BACKGROUND AND RELATED WORKS

This chapter gives a background on the basics of adiabatic logic, its operating principles, and clocking schemes. It also presents some of main adiabatic logic families that have shown promising results in terms of energy efficiency, both theoretically and during testing of any fabricated chips. The second half of the chapter provides an overview of power analysis attacks and briefly presents existing countermeasures at the hardware level. Several non-adiabatic dual-rail solutions are discussed, along with their respective advantages and disadvantages.

II.1 Background in Adiabatic Computing

Conventional CMOS circuits dissipate half of the input energy during gate charging and store the other half in the output node capacitance when a logic value changes. This can be illustrated using the CMOS inverter and its equivalent circuit in Fig. 1 as an example. When the input is switched from High to Low, the PMOS transistor turns ON and charges the output capacitor, where the output capacitor (C) models the fanout of the gate. When the input switches to High, the charge stored in C is discharged through the NMOS transistor to GND . During the charging process, a charge of $Q = CV_{dd}$ is transferred from the source to the output, where V_{dd} is the supply voltage. The supplied energy is then calculated as

$$E_{supplied} = QV_{dd} = CV_{dd}^2. \quad (1)$$

The current flow during charging is given by

$$i(t) = \frac{V_{dd}}{R} e^{-\frac{t}{RC}}, \quad (2)$$

where R is the equivalent transistor resistance, and the power is calculated as

$$P(t) = \frac{V_{dd}^2}{R} e^{-\frac{2t}{RC}}. \quad (3)$$

The amount of energy stored in C can be calculated by integrating the power over time and is given by

$$E_{stored} = \frac{CV_{dd}^2}{2}. \quad (4)$$

This result shows that only half of the input energy is stored in the output capacitor, while the other half is dissipated in the transistor network during charging. Furthermore, when the output changes values, the energy stored in C is not recycled back to the source but is dissipated in the NMOS transistor network. Energy dissipation in CMOS circuits is due to how these circuits are driven, i.e. using square-wave clocks, and dissipation of stored energy as heat when the output value switches to a Low, reducing the energy efficiency of the circuit.

An alternative to CMOS logic is a adiabatic switching, which ideally operates as a reversible thermodynamic process, without loss or gain of energy. Adiabatic computation works by making very small changes in energy levels of nodes in circuits sufficiently slow, ideally resulting in no energy dissipation during charging and recovering the energy from the output capacitance when discharging. However, practical implementations

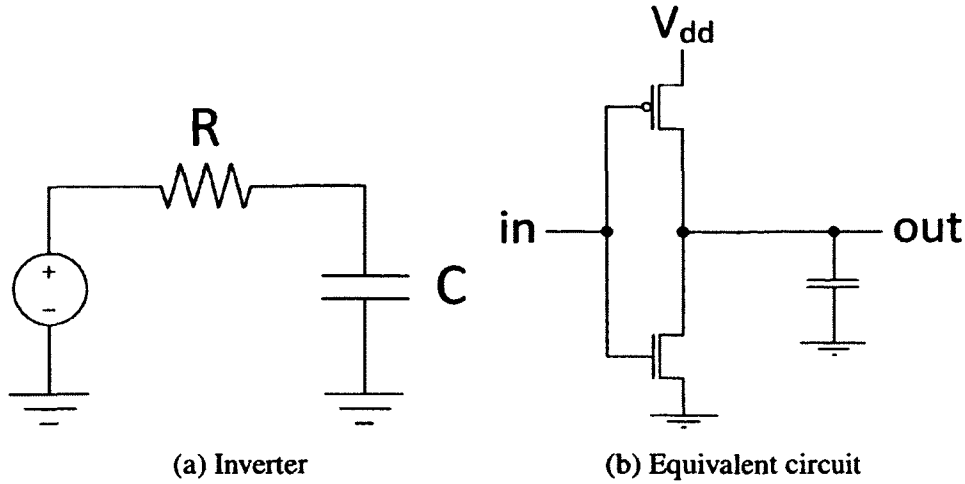


Fig. 1: CMOS inverter and its equivalent circuit.

of adiabatic circuits contain some losses, even if the circuits are driven at a small frequency. There exist two types of adiabatic computations: fully-adiabatic (circuit operates slower, loses arbitrarily little energy per operation, and almost all of the input energy is recovered) and partially-adiabatic (some energy is recovered and some is lost to irreversible, non-adiabatic operations). While fully-adiabatic circuits are very attractive from a reversible logic perspective and are most suitable for the implementation of reversible circuits, most existing adiabatic circuit designs are partially-adiabatic because of their simpler implementations and smaller circuit area.

Adiabatic switching attempts to minimize the energy wasted during charging by using a more gradual charge delivery system which is more efficient at lower frequencies. A gradual charge transfer avoids the large current peak of V_{dd}/R and reduces the power loss during output switching. Adiabatic circuits replace the square-wave clock with a time-varying Power Clock (*PC*), which supplies both power and clocking information to circuits. Unlike CMOS circuits, adiabatic circuits normally require more than one clock

phase to operate correctly. The most common types of *PC*'s used in adiabatic circuits are 4-phase trapezoidal and 1- or 2-phase sinusoidal oscillators.

Because of their continuous time-varying nature, the outputs of an adiabatic gate charge and discharge on every *PC* cycle, having an activity factor $\alpha = 1$. Given this high activity factor, adiabatic logic circuits maintain their advantage over CMOS circuits and are more suitable for circuits with moderate to high switching activity, e.g. arithmetic circuits. This continuous switching forces them to be operated in a micro-pipelined fashion, most often using multiple *PC* phases.

Adiabatic circuits work by charging the output nodes for part of the clock cycle, allowing the next clock phase to sample the output, then recovering the charge from the output node capacitances. The number of *PC*'s depends on the adiabatic family and the type of oscillator used. In general, trapezoidal *PC*'s use 4 phases, although a few designs with a single phase and some with 8 phases exist. Trapezoidal *PC*'s have 4 phases of operation: Wait, Charge, Hold, and Recover. Sinusoidal *PC*'s usually use 2 phases, but some technologies have additional circuitry to achieve single-phase operation. Sinusoidal *PC*'s have two operating phases: Charge and Recover. An example of the 4-phase trapezoidal and 2-phase sinusoidal *PC*'s is shown in Fig. 2.

II.1.1 Benefits and Disadvantages of Adiabatic Logic

Adiabatic logic circuits have a several important advantages over CMOS circuits. First, they offer very low energy consumption at a frequency of up to several hundred MHz for circuits with a moderate to high activity factor. While operation at higher speeds

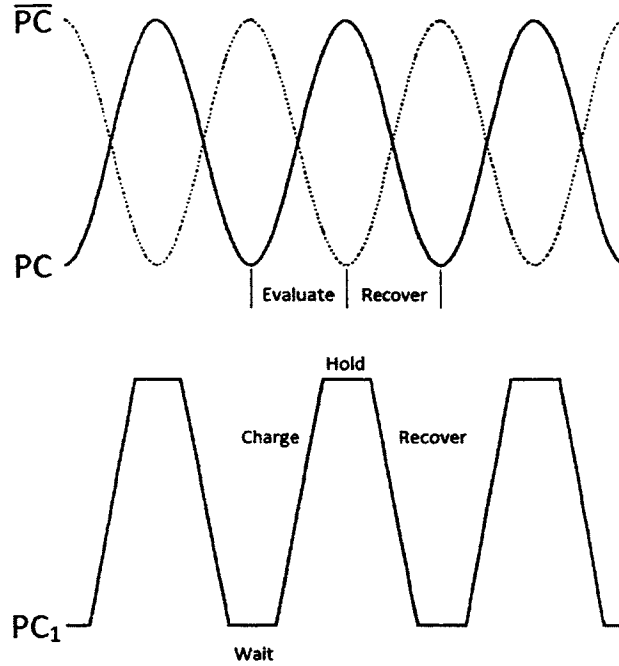


Fig. 2: 2-phase sinusoidal and 4-phase trapezoidal PC 's with each stage marked for the first phase.

is possible as demonstrated in a commercial chip [15], the advantages of adiabatic switching diminish as the power clock charging time approaches the MOS transistor charging time. Therefore, adiabatic circuits would be more suitable for use in applications that operate at frequencies below 1 GHz.

Another advantage of adiabatic logic is its inherent micro-pipelining. Unlike traditional CMOS circuits where the value of an output needs to be stored in a latch when the inputs change, adiabatic logic allows for a micro-pipelined approach, where each output is dynamically buffered and stored for one clock cycle. This allows for an adiabatic system to operate without the use of latches since the intermediary outputs are generated and immediately sampled by the next stage. Most adiabatic technologies do not have special latch circuits due to their micro-pipelining, although some flip-flop designs for adiabatic

circuits have been proposed in [21]–[23]. These designs were not true flip-flops, but were based on adiabatic buffers without any special circuitry for holding a constant value. Traditional CMOS latches and flip-flops can still be used for outputs at the adiabatic-CMOS interface.

CMOS circuits are limited in their maximum operating frequency by the critical path of a given circuit. A critical path does not exist in adiabatic logic circuits as micro-pipelining allows the evaluation of only one gate per stage. The *PC* enforces that input signals are valid as soon as a gate starts to evaluate its outputs. Delay due to the evaluation of complex gates with large input stacks is also not an issue for adiabatic circuits since the evaluation of inputs will take place before the outputs start evaluating. Adiabatic circuits are also not constrained by setup and hold times as the next gate will only evaluate after the outputs of the current gate are stable (when the *PC* is at its peak).

Adiabatic switching does not have as rigorous a constraint on voltage scaling as does CMOS switching [18]. There is a tradeoff between speed and power consumption in CMOS circuits, therefore the voltage can only be reduced to a level to which timing constraints are not violated based on the critical path. On the other hand, adiabatic switching does not have any timing constraints other than the fixed gate delay due to the threshold voltage of the charging transistors.

Another advantage of adiabatic circuits is that it strongly reduces the effects of electromigration. Electromigration is the wear-out process on lines carrying currents with a strong current density and is a predicting factor in the mean time to failure of electronic components. The effect is more likely to occur on power supply lines in CMOS circuits

where unidirectional currents are strong. In cases where the current flow is bidirectional, such as adiabatic circuits, the effects of electromigration are greatly reduced. Reducing the effects of electromigration contributes to a longer mean time to failure and also allows the clock lines to be narrower, which reduces capacitance and energy consumption.

Adiabatic circuits also have some limitations. They are dual-rail circuits by design, which requires both an input and its complement and produces the output and its complement. Dual-rail circuits often have up to double the area requirements of single rail circuits and require special care during layout for interconnects. As an example, the CMOS 2-input NAND gate can be built using 4 transistors (2 PMOS and 2 NMOS), whereas the smallest adiabatic implementation using the 2N2P logic family [8] contains 6 transistors (4 NMOS for function implementation and 2 PMOS for charging).

Adiabatic logic circuits more often than not use more than one clock phase for operation. While there are a couple of logic families that use only one clock phase (e.g. TSEL[11], CAL[10]), they also use auxiliary signals to make one-phase operation possible. Most sinusoidal adiabatic logic families use two sinusoidal clock phases, whereas the trapezoidal families use four phases. The use of multiple clock phases adds more area and routing constraints to the circuit compared to a single-phase clock and raises issues of clock synchronization across multiples phases. As the output of each stage in an adiabatic micro-pipelined circuit feeds into another stage with a different phase, the multiple clock phases have to be exactly spaced in order to compute a result properly and increase energy efficiency.

Inherent micro-pipelining can also be a disadvantage of adiabatic logic if proper care

is not taken to buffer signals in circuits that have multiple evaluation stages. Insertion of extra buffers adds to the circuit area, increases capacitance, and has an impact on the design flow as the buffers have to be accounted for in the design process. The number of transistors used in buffer circuits necessary to translate a CMOS design to an adiabatic design depends on the chosen adiabatic family and the particular implementation of a circuit. Values for the number of transistors for each adiabatic family investigate were not available from the original authors, but experimental values for large arithmetic circuits were found to range from 25% to 35% of total transistor count.

II.1.2 Summary of Adiabatic Families

Adiabatic logic families can be divided into two main categories: ones that use diodes and ones that do not. Most modern adiabatic logic families are diode-less as they are more energy-efficient because the voltage drop across the diode in older families results in lower efficiencies. There have been a number of adiabatic logic families proposed [8]–[14], [16, 17, 19, 20], [23]–[27], but only a few of them have been successful in gaining recognition and usage in the adiabatic logic research community. Most of these families use a trapezoidal *PC* and only a few use a 2-phase sinusoidal *PC*, even though the generation of a sinusoidal *PC* is more energy efficient than a trapezoidal one. Some of these adiabatic families, their operation, and advantages and disadvantages are succinctly described below.

2N2P: This partially-adiabatic logic family introduced in [8] uses a four-phase trapezoidal *PC* and the buffer/inverter implementation is shown in Fig. 3. The buffer/inverter

uses a simple NMOS pull-down network structure and contains a low number of transistors. The cross-coupled PMOSs ($P1$ and $P2$) are a feature of almost all adiabatic logic families and ensure that the voltages at the PMOS transistors' drain terminals (out and \overline{out}) are always complementary. Furthermore, the PMOS transistors provide a pathway to recover charge back to the power clock. The two NMOS transistors implement the buffer and inverter functionality by using a complementary pull-down network. Their functionality can be expanded by using a PDN structure to implement any function and its complement.

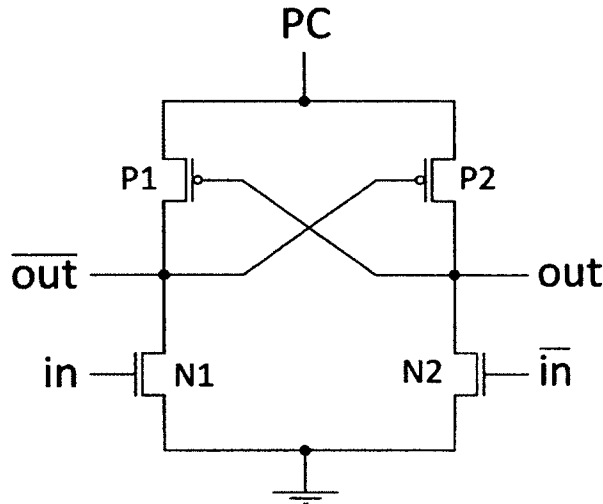


Fig. 3: 2N2P buffer/inverter circuit.

The operation of a 2N2P gate follows a traditional trapezoidal PC sequence. When PC is at GND , during the WAIT stage, the inputs are evaluated and the corresponding output is pulled to GND . In the next stage (EVALUATE), once $V_{PC} \geq |V_{TP}|$ (the PMOS threshold voltage), the other output is ramped up following the PC up to V_{dd} . Since the inputs are stable throughout the four clock phases, the output that is initially at GND does

not change values while the other charges adiabatically to V_{dd} . During the next stage (HOLD), the outputs are held at complementary values in order for the next stage of gates to evaluate. In the last phase (RECOVER), the High output is ramped down to GND by following the PC and the energy stored in that node is recovered until $V_{PC} = |V_{TP}|$. If the inputs change and cause a change at the outputs, the remaining energy will be discarded as the outputs switch values. If the inputs remain constant, the remaining energy will be used in charging the outputs during the next clock cycle, thereby reducing the amount of energy lost in the system.

2N2P achieves very good energy efficiency compared to CMOS (up to 57% less), but suffers from possible floating output nodes during the HOLD stage which reduces energy efficiency. Another disadvantage is that it uses four clock phases, adding more area and capacitance in layout. And lastly, the use of a trapezoidal PC allows for a more linear charge delivery, however, the generation of a trapezoidal PC reduces the energy efficiency of the system since it is significantly more difficult to generate than a sinusoidal PC .

2N-2N2P: The 2N-2N2P [8] logic family is an extension of 2N2P and has the same timing and clock phases. The difference is that it adds a Static Random Access Memory (SRAM)-like structure that allows for the outputs to be non-floating for all PC phases (Fig. 4). One of the disadvantages of using this SRAM-like structure is that it limits the energy efficiency at operating speeds higher than 100 MHz. In large circuit simulations, 2N-2N2P achieved an energy efficiency of 63% less than equivalent CMOS circuits, slightly better than 2N2P circuits.

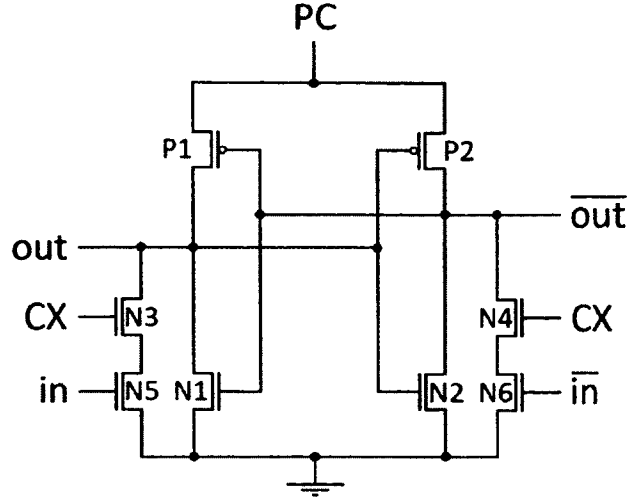


Fig. 5: CAL buffer/inverter circuit.

structure of a TSEL PMOS buffer/inverter is shown in Fig. 6. Each TSEL gate contains a reference voltage (either V_{RP} or V_{RN} , depending on the type of transistors used) as a bias voltage, which is a distinctive feature of this family. TSEL uses pre-charging of gates in order to achieve single-phase operation and contains two distinct phases: discharge/charge and evaluate. During the discharge state in the PMOS inverter, the energy stored in the *out* or \overline{out} node is recovered. At the beginning of this stage, PC is High and as it starts ramping down, both outputs are pulled towards V_{TP} and the energy is recovered. This transition is adiabatic until $V_{PC} < V_{RP} - |V_{TP}|$, which disables the cross-coupled PMOS structure. Assuming that *in* is High and \overline{in} is Low at the beginning of the Evaluate phase, V_{PC} starts rising from Low to High and turns ON P3 and P4, which turns ON P1 and P2. As long as $V_{PC} < V_{RP} - |V_{TP}|$, P3 and P4 are conducting. Since $V_{RP} > V_{PC}$ at this point, \overline{out} starts rising towards V_{RP} through P4 and the two cross-coupled PMOSs help amplify the voltage difference between *out* and \overline{out} . Once the difference between the output nodes is larger than $|V_{TP}|$, P1 turns OFF and \overline{out} charges adiabatically to V_{dd} . When $V_{PC} \geq V_{RP} - |V_{TP}|$,

means different than the ones originally intended for communication in that channel. This kind of methods that exploit unintentional information channels are called side-channel attacks. There exist several channels that have been used in the past to mount side-channel attacks, e.g. electro-magnetic radiation [43, 51], time elapsed [84], leakage currents [61, 65], acoustic emanations [81], fault injection [79], and power usage. In this dissertation, the focus is on the power analysis attacks and their countermeasures.

Power analysis attacks are a special subset of side-channel attacks that can be mounted with relative ease and without requiring very expensive hardware. These attacks are aimed at extracting secret information from a microprocessor or Application-Specific Integrated Circuit (ASIC) chip by analyzing the power consumed by the chip during encryption or decryption operations. It is likely that this kind of attacks will be more prevalent in the future due to the recent boom in portable electronic devices. As more people integrate these devices in their lives and store more and more of their personal information on them, they become a worthy target to an attacker. Moreover, since a majority of the world is moving towards implementing smart cards in the banking industry (credit and debit cards), the issue of hardware security is of even more importance.

Most of the power analysis attacks are aimed at encryption cores or microprocessors performing encryption operations. The extracted private keys could be used to mimic a particular device and take advantage of it. There have been numerous attacks staged on several kinds of encryption algorithms, including Data Encryption Standard (DES) and Advanced Encryption Standard (AES), by using power analysis attacks [40]–[42], [44, 47, 72, 75, 80, 82, 85]. As power analysis attacks are passive non-invasive attacks, it

is usually very hard for the user to be aware of the attack and protect against it.

Power analysis attacks make use of the fact that a microprocessor uses asymmetric amounts of power for different kinds of operations and instructions. A two-input CMOS NAND gate is used as an example of information leakage as asymmetric power consumption, which can be used by attacker to gain insights into the inputs to the gate. The transistor-level diagram of the NAND gate with charge (dotted line) and discharge (interrupted line) paths is given in Fig. 7. A truth table for the NAND gate is shown in Table 1. It can be observed that for 3 input combinations ($AB = "00", "01", "10"$) the gate output is High, while for 1 input combination ($AB = "11"$) the value is Low. A table with all 16 possible transitions is presented in Table 2. When the inputs change between these two different sets, the gate will consume dynamic energy ('D' in Table 2), as the output value switches. When the inputs switch between values inside each set, only static energy will be consumed ('S' in Table 2) by the CMOS gate. Since the amount of energy dissipated between dynamic and static dissipations are very different, an imbalance can be observed and unintentionally reveal information about the inputs to the gate.

Table 1: Truth table for a 2-input NAND gate.

Input	Output
00	1
01	1
10	1
11	0

Power analysis attacks are divided into two categories: Simple Power Analysis (SPA) and Differential Power Analysis (DPA). SPA attacks were introduced by Kocher [40] and

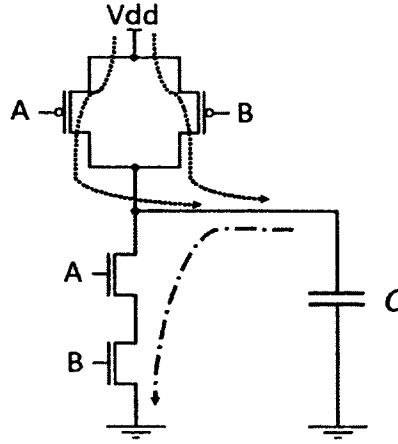


Fig. 7: A 2-input CMOS NAND gate showing charging (dotted) and discharging (dot+line) paths.

rely on deriving the secret key directly from a given power trace. This can make SPA attacks challenging in practice since an attacker might need detailed knowledge about the architecture of the device under test and the particular cryptographic algorithm used. Furthermore, if only one power trace is available, complex statistical methods are required to extract the information from the signal. An example of a viable SPA attack is when a person uses a smart card to pay for gas at a gas station. An attacker can compromise the card reader in order to extract the power trace from a user's card. It is not uncommon for a user to frequent the same gas station more than once, so more traces could be obtained and a SPA attack staged successfully. DPA attacks were also introduced in [40] and are the most popular type of power analysis attacks since they do not require in-depth knowledge of the attacked device and are effective at revealing secret keys even from very noisy power traces. In contrast to SPA, DPA attacks need a large number of power traces, usually requiring the possession of the cryptographic device for some time in order to mount an attack on it. Several successful SPA and DPA attacks were implemented in recent

Table 2: All 16 possible input transitions for a 2-input NAND gate and type of energy consumption for each transition in a CMOS implementation.

Current	Next	E_{type}
00	00	S
00	01	S
00	10	S
00	11	D
01	00	S
01	01	S
01	10	S
01	11	D
10	00	S
10	01	S
10	10	S
10	11	D
11	00	D
11	01	D
11	10	D
11	11	S

years on specialized cryptographic cores [41], Field-Programmable Gate Array (FPGA) implementations [47, 63], and smart cards [40, 82]. The relative ease of implementation and possible consequences make power analysis attacks a growing concern for portable electronic devices and call for power analysis attack resistant circuit design.

II.2.2 Existing Hardware Countermeasures

Power analysis attacks can be viewed as a problem from two different perspectives: mathematical and engineering. As a mathematical problem, the goal is to find mathematical models that describe the leakage of cryptographic devices in order to build secure systems based on these models. As a result of this viewpoint, the solutions created are mainly

software and algorithmic solutions, like masking. Masking solutions [48, 49, 50, 56] attempt to hide the bits processed in the cryptographic core by randomizing intermediate results. These solutions, however, have been proven insecure several times in the past [45, 46] in order to hide the signal. As an engineering problem, the goal is to decrease the leakage of information by decreasing the leaked signal or by increasing the noise. As a result, the solutions from this perspective lead to the development of DPA-resistant logic styles. This viewpoint has seen more research interest in the recent years as it tries to attack the problem at its root cause – at the hardware level, although more complex changes need to be made to implement these solutions. In this work we focus on the engineering approach and describe some of the proposed technologies to reduce information leakage at the hardware level.

Most of the engineering approaches to the problem try to solve the problem of balancing the amount of energy dissipated during switching between all possible inputs. Most of the time, this is impossible to achieve perfectly, but the goal is to get as close as possible to equal dissipation. In order to achieve balanced dissipation all input combinations have to use the maximum amount of energy possible for a given gate. Several logic families and other types of countermeasures have been introduced [37, 38, 52, 55, 57, 60, 64], [66]–[70], [73, 74, 76, 78, 83] and compared in [39, 53, 54, 59, 62, 71, 77]. However, almost all these countermeasures use significantly more energy than an equivalent CMOS circuit or are not designed to have built-in security against power analysis attacks. The two most well-known countermeasures against power analysis attacks are Sense Amplifier Based Logic [37] and Wave Dynamic Differential Logic [38] families and are discussed briefly

in this work.

Sense Amplifier Based Logic (SABL) [37] is one of the first logic styles proposed as a countermeasure against power analysis attacks. SABL cells are designed to have a constant internal power consumption independent of the processed logic values. Additionally, combinational SABL cells are designed such that their time-of-evaluation (TOE) is data independent, i.e. the cells only evaluate after all inputs have been set to complementary values.

SABL cells are dual-rail and use precharge and evaluate stages to achieve balanced energy dissipation. Unlike CMOS cells, all SABL cells are connected to a common clock and are precharged simultaneously, which causes very high current peaks during the precharge phase. The current peaks during the evaluation phase are smaller because not all combinational SABL cells evaluate simultaneously. Since combinational SABL cells only evaluate after all inputs have been set to complementary values, the evaluation events of different types of cells are spread out over the evaluation phase. However, due to a constant TOE, same type of gates evaluate at fixed moments in time for each clock cycle and input combinations.

The area requirements for SABL circuits are at least doubled compared to equivalent CMOS circuits, while the maximum clock frequency is halved. The power consumption of SABL circuits is significantly larger than for CMOS circuit. Experimental results with a 2-input NAND gate showed an increase of around $10\times$ compared to an equivalent CMOS gate, although this ratio is likely to fluctuate based on the type of gate tested. The DPA

resistance of SABL circuits is typically very high (if all complementary wires are sufficiently balanced), because the internal power consumption of the SABL cells is constant and the cells always evaluate at fixed moments of time during each clock cycle.

Figure 8 shows the transistor-level schematic of the 2-input NAND/AND SABL cell. In order to achieve the same number of transitions at the cell outputs *AND* and *NAND* in each clock cycle, the SABL cell consists of the differential pull-down network (DPDN) and cross-coupled inverters, similar to a SRAM cell core. The DPDN is made of NMOS transistors in such a way that the discharge paths use the same number of transistors, regardless of the input combination. The inverters are cross-coupled so their outputs are always complementary and this forms the sense amplifier part of the SABL gates. In a SABL gate, the evaluation phase starts when the clock switches to a High value. During this phase, the input signals of the DPDN are set to complementary values and the NMOS transistor *Pr3* is turned ON. As a result, the cell outputs *AND* and *NAND* are switched to complementary values. When the clock switches to a Low value, the gate is in precharge phase. During this phase, *Pr1* and *Pr2* charge all the internal nodes of the SABL gate to a High value and the outputs of the gate to a Low value, thus SABL gates fall in the category of “precharge to zero” gates. The precharge to zero is necessary to ensure the next level of gates in the cascade do not provide a path to GND during the next evaluation phase. A SABL cell can also be built from PMOS transistors in a differential pull-up network, but a circuit has to contain only one type of cell in order to provide a valid computation.

The second most common type of logic style countermeasure against power analysis attacks is Wave Dynamic Differential Logic (WDDL) [38]. WDDL cells are built as

through the combinational WDDL circuit. Since combinational WDDL cells precharge and evaluate successively, the current peaks in WDDL circuits are much smaller than those in SABL circuits but they contain higher variations. Even if the propagation delays of complementary wires in balanced WDDL circuits are pairwise identical, the TOE of the combinational WDDL cells is still dependent on the processed data, which reduces the DPA resistance of WDDL circuits.

Area requirements for WDDL circuits are at least doubled compared to CMOS circuits with equivalent functionality, similar to SABL circuits. Power consumption of WDDL circuits also increases significantly compared to CMOS, but it is lower than SABL circuits as it uses independent cells from standard libraries with smaller current peaks and does not precharge simultaneously. Maximum clock rate in WDDL circuits is similar to CMOS circuits, however, as the WDDL D-flip-flop consists of two stages of single-rail D-flip-flops, it is necessary to double the clock frequency to achieve the same throughput as in CMOS or SABL circuits.

The WDDL NAND/AND gate is presented in Fig. 9. It consists of a single-rail AND cell with true inputs and a single-rail OR cell with complementary inputs. At the beginning of the evaluation phase, all complementary inputs of a combinational WDDL cell are precharged to *GND*. When the input signals are set to complementary values, only one *GND* to V_{dd} transition occurs at the outputs of the WDDL cell per evaluation cycle, while the other output stays at *GND*. By ensuring one transition per evaluation cycle per gate, WDDL is able to have a constant number of transitions for any given input combination in a circuit, allowing it have some DPA resistance.

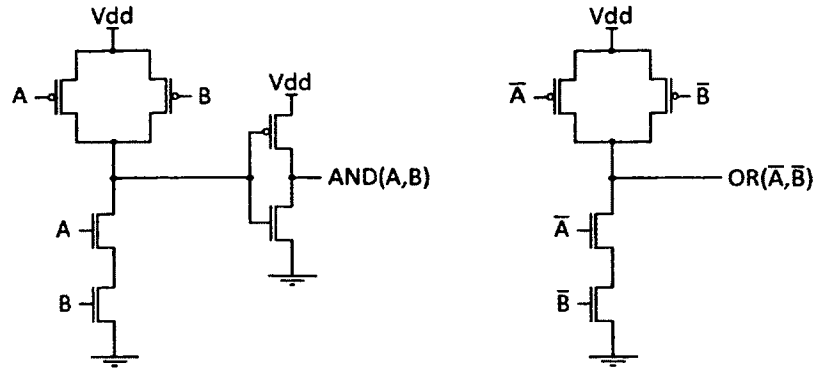


Fig. 9: A 2-input WDDL NAND gate.

Energy consumption for WDDL cells is not constant mainly for three reasons. First, different conducting paths with a varying number and arrangement of MOS transistors are established for different input combinations. Second, not all internal nodes of a WDDL cell are charged and discharged in the same way in each clock cycle for all data values, leading to charges stored at internal nodes in a cell to be dependent on the value processed (the memory effect). Third, the TOE of combinational WDDL cells depends on the input values, as mentioned earlier.

Power analysis attacks are very powerful tools for an attacker to extract secret information from a device using relatively inexpensive equipment. The proposed countermeasures found in the literature usually provide good protection against these attacks, but at a cost of much larger area and energy consumption, making these solutions unsuitable for low power applications. A new low-power solution for power analysis attacks is needed.

CHAPTER III

IDPAL – INPUT DECOUPLED PARTIALLY-ADIABATIC LOGIC

The main contribution of this work is the development of a new very low power partially-adiabatic logic family with decoupled inputs named Input Decoupled Partially-Adiabatic Logic (IDPAL). The originality of IDPAL is in its separation of a gate into two distinct parts: input network evaluation and function evaluation. In this chapter, IDPAL and several of its applications, starting from small logic gates to very large arithmetic circuits, are presented. Experimental results show that IDPAL is capable of reducing energy consumption of circuits by up to 79% compared to its equivalent CMOS implementations [1, 2, 3]. IDPAL also uses at least 25% less energy than state-of-the-art adiabatic logic families. Several large circuits are described and their energy consumption compared to other equivalent CMOS and adiabatic implementations.

III.1 IDPAL Operating Principle

IDPAL is a new very low power partially-adiabatic logic family. Similar to other adiabatic logic families, IDPAL is a dual-rail family, so it requires both an input and its complement in order to evaluate a given function. IDPAL differs from other families in the fact that gates implemented in IDPAL are composed of two decoupled parts: input network evaluation and function evaluation. A general view of an IDPAL gate is shown in Fig. 10. The inputs control two complementary switching networks to evaluate the true (F) and complementary (\overline{F}) functions. Once the inputs are evaluated, one switching

network is conductive, allowing current to flow from the *PC* to the respective NMOS transistor of the function evaluation part. The complementary switching network is OFF and its respective NMOS transistor remains OFF. In this way, the inputs are provided to the function evaluation part where the appropriate output manifests as a High by following the *PC* and the complementary output remains at Low. The two NMOS and two PMOS transistors form the function evaluation part of IDPAL gates. A unique feature of IDPAL is the connection of input networks to the *PC* rather than *GND*, unlike other adiabatic technologies, which allows the inputs evaluation portion to be actively driven by the *PC*. A more detailed explanation of IDPAL gate operation is described using the example of a buffer/inverter.

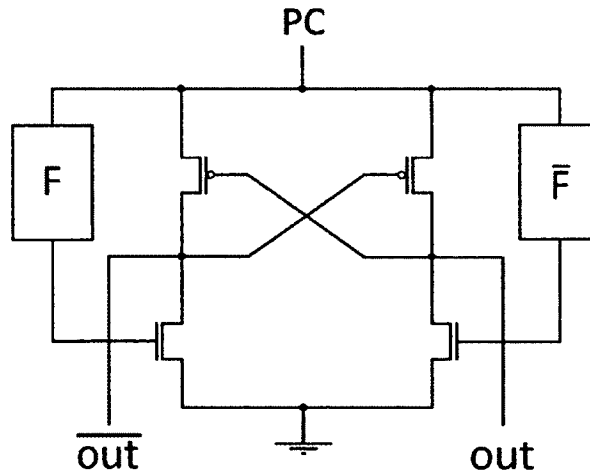


Fig. 10: General gate construction in IDPAL.

The simplest functional IDPAL gate is the buffer/inverter shown in Fig. 11 and its operation is as follows. Assuming that *in* is at V_{dd} and \overline{in} is at *GND*, when the *PC* rises from *GND* to V_{dd} , *in* will cause N1 to turn ON. When V_{PC} reaches V_{TN} (the NMOS threshold voltage), N3 will turn ON and \overline{out} will be anchored at *GND*. This event will turn P2 ON

and allow *out* to charge adiabatically to V_{dd} and the circuit will achieve the buffer/inverter function with complementary outputs. When the *PC* swings from V_{dd} to GND , the energy stored in *out* will be recovered through P2 until $V_{PC} \leq V_{TP}$. The remaining energy is discarded to GND if the inputs change, resulting in non-adiabatic losses. Otherwise, the output ramps up again as the *PC* rises, without any non-adiabatic losses. The energy loss due to the threshold voltage of the PMOS transistors is unavoidable in any partially-adiabatic family that uses the cross-coupled PMOS structure.

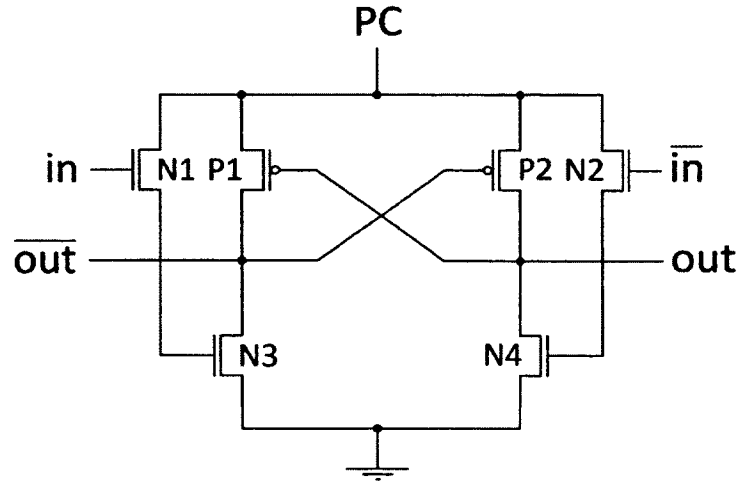


Fig. 11: IDPAL buffer/inverter.

An equivalent RC circuit for an IDPAL buffer/inverter with the main capacitances is shown in Fig. 12. Each transistor was replaced with a resistor to account for energy loss when the transistor is ON and 3 capacitors to account for the three main capacitances of a transistor (gate-to-drain/source, drain-to-substrate, and source-to-substrate). Assuming that the input *in* is High and its complement, \overline{in} , is Low, we obtain the circuit shown in Fig. 13. From this circuit, we observe three separate circuits that are part of an IDPAL

gate: input evaluation circuit, complement evaluation circuit, and function evaluation circuit. The input gate capacitance is not connected to the input evaluation circuit, instead it is connected as an additional capacitor to the function evaluation circuit in order to account for the capacitance of the subsequent gates driven by the current gate.

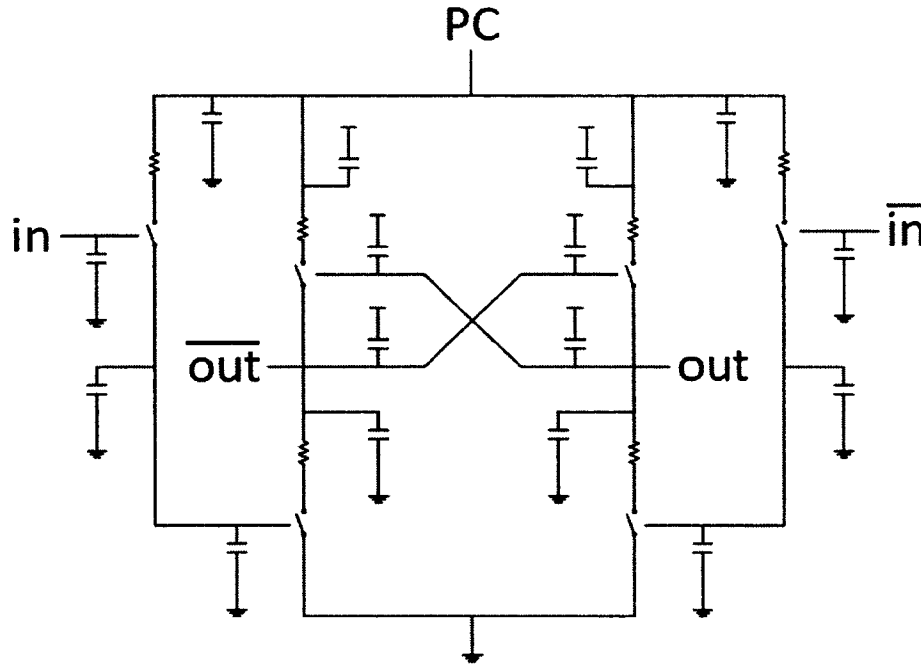


Fig. 12: Equivalent RC circuit for an IDPAL buffer/inverter.

A more complete picture can be observed by examining the RC representation of an IDPAL 2-input AND/NAND gate implementation using the three different circuits, shown in Fig. 14. In this example, the input network is composed of two series conducting transistors, whereas the other two circuits (complement output evaluation and output evaluation, respectively) are the same for all gates and circuits. If a larger gate is used in a circuit, the additional losses will take place in the input evaluation network due to the additional RC nodes. However, due to being driven by the *PC*, any additional capacitors that

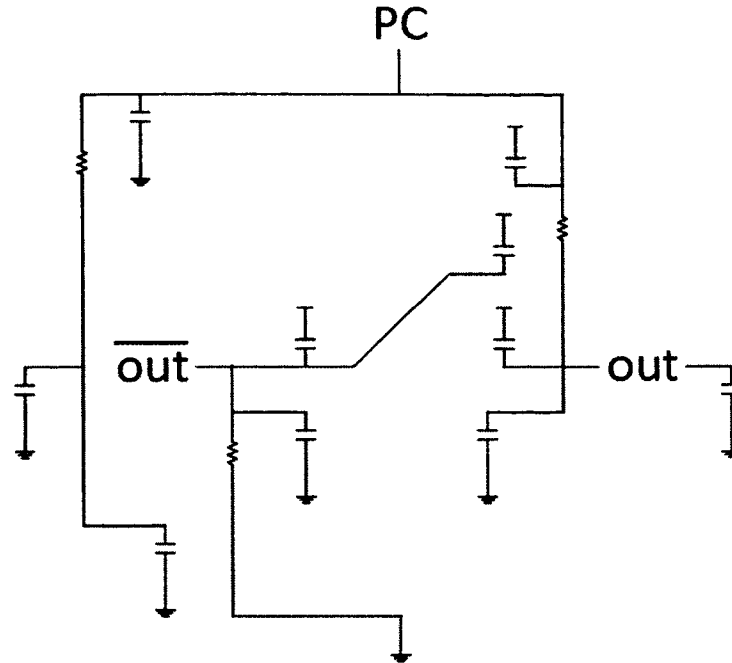


Fig. 13: Equivalent RC circuit for an IDPAL buffer/inverter with one of the inputs conducting.

are charged will be discharged to the *PC* as the charge is recycled, therefore contributing to the energy-efficiency of the circuit.

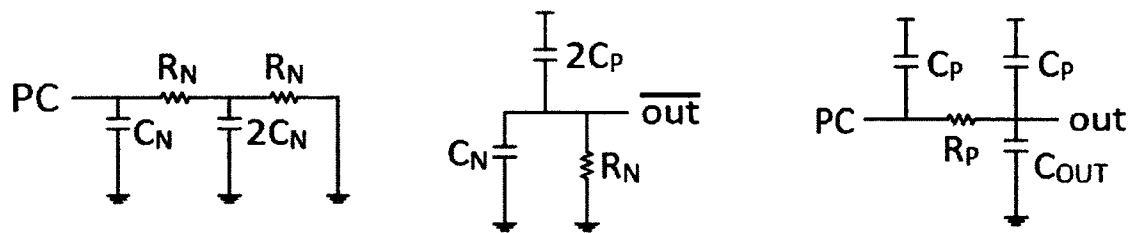


Fig. 14: Equivalent RC circuit breakdown for an IDPAL 2-input AND/NAND gate.

In order to calculate the energy usage in an IDPAL gate, each of the three circuits has to be investigated separately by calculating an expression for the current trace. This value can then be multiplied with the *PC* voltage and integrated over time in order to calculate

the energy dissipation for a given circuit. For the first sub-circuit, the complex impedance can be evaluated by:

$$\frac{1}{Z_2} = \frac{1}{R_N} + j2\omega C_N = \frac{1 + j2\omega C_N R_N}{R_N}; Z_2 = \frac{R_N}{1 + j2\omega C_N R_N} \quad (5)$$

$$Z_1 = R_N + Z_2 = R_N + \frac{R_N}{1 + j2\omega C_N R_N} \quad (6)$$

$$\frac{1}{Z} = \frac{1}{Z_1} + j\omega C_N = \frac{1 + j\omega C_N Z_1}{Z_1}; Z = \frac{Z_1}{1 + j\omega C_N Z_1} \quad (7)$$

$$Z = \frac{R_N + \frac{R_N}{1 + j2\omega C_N R_N}}{1 + j\omega C_N \left(R_N + \frac{R_N}{1 + j2\omega C_N R_N} \right)} = \frac{\frac{2R_N + j2\omega C_N R_N^2}{1 + j2\omega C_N R_N}}{\frac{1 + j2\omega C_N R_N + j\omega C_N R_N + j2\omega^2 C_N^2 R_N^2 + j\omega C_N R_N}{1 + j2\omega C_N R_N}} \quad (8)$$

$$Z = \frac{2R_N + j2\omega C_N R_N^2}{1 + j4\omega C_N R_N + j2\omega^2 C_N^2 R_N^2} \quad (9)$$

The current is then calculated as:

$$i(t) = \frac{V(t)}{Z} = \frac{V_{DD}}{Z} \sin(\omega t). \quad (10)$$

The second sub-circuit has a very low voltage swing, only up to the threshold voltage of the PMOS transistor and is controlled only by the V_{DD} . The impedance can be calculated by:

$$\frac{1}{Z_1} = \frac{1}{R_N} + j\omega C_N = \frac{1 + j\omega C_N R_N}{R_N}; Z_1 = \frac{R_N}{1 + j\omega C_N R_N} \quad (11)$$

$$Z = Z_1 + \frac{1}{j2\omega C_P} = \frac{R_N}{1 + j\omega C_N R_N} + \frac{1}{j2\omega C_P} = \frac{j2\omega C_P R_N + 1 + j\omega C_N R_N}{j2\omega C_P + j2\omega^2 C_P C_N R_N}. \quad (12)$$

The third sub-circuit contains two types of power sources: the sinusoidal PC and the V_{DD} , therefore energy dissipation from both sources has to be taken into account. From a personal interview with Ana Samolov (Ph.D. Physics), the solution to this problem appears to be non-trivial.

Simulation waveforms for a single buffer/inverter are shown in Fig. 15. The simulation was performed using LTspice in a TSMC $0.25\mu\text{m}$ process with transistor width/length (W/L) ratios of $0.36\mu\text{m}/0.24\mu\text{m}$ and $0.72\mu\text{m}/0.24\mu\text{m}$ for the NMOS and PMOS transistors, respectively. The clock frequency is 10 MHz and the input frequency is 5 MHz. From the Figure, it can be clearly seen that when an input is High for two consecutive clock cycles, the output is not taken to GND after the first clock cycle, but stays at the $|V_{TP}|$ level until the PC ramps up again. The output is taken to GND level and the energy below V_{TP} is lost only when the inputs change values. This allows the non-adiabatic energy losses to have a small contribution to the total energy consumption if the inputs are constant for very long periods.

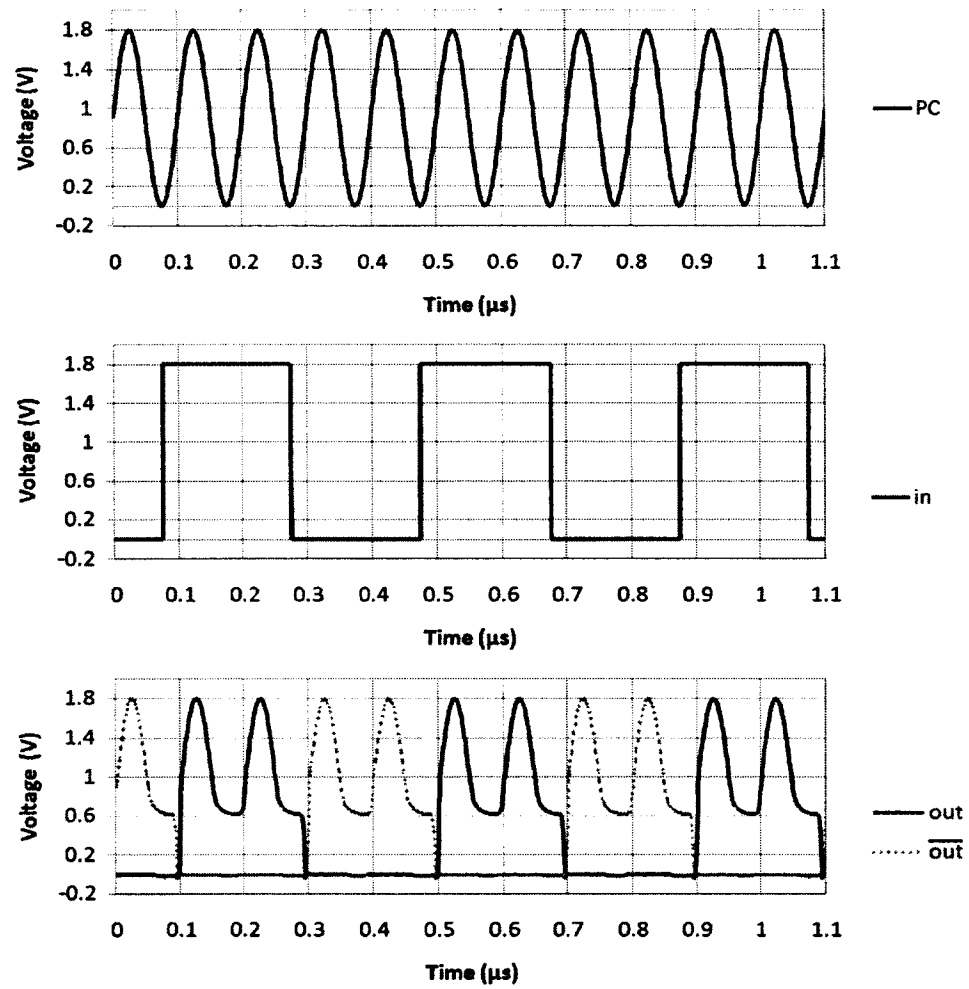


Fig. 15: Simulation waveform for an IDPAL buffer/inverter.

Energy dissipation is calculated by integrating the instantaneous power over the operating time, as was done for several previous adiabatic technologies, including [8], [10], and [11], using

$$E_{dissipated} = \int_0^T P(t)dt = \int_0^T i(t)V(t)dt. \quad (13)$$

A plot of the energy dissipation and recovery for the single buffer/inverter setup above is shown in Fig. 16. As mentioned earlier, when the input is High for two clock cycles, the energy stored in the gate output capacitance when the PC reaches $V_{PC} = |V_{TP}|$ is not lost, but recycled into the next evaluation cycle. The Figure also shows that the amount of energy lost is larger when the inputs switch values and the outputs are re-evaluated since the complementary output needs to be charged to V_{dd} from the GND level.

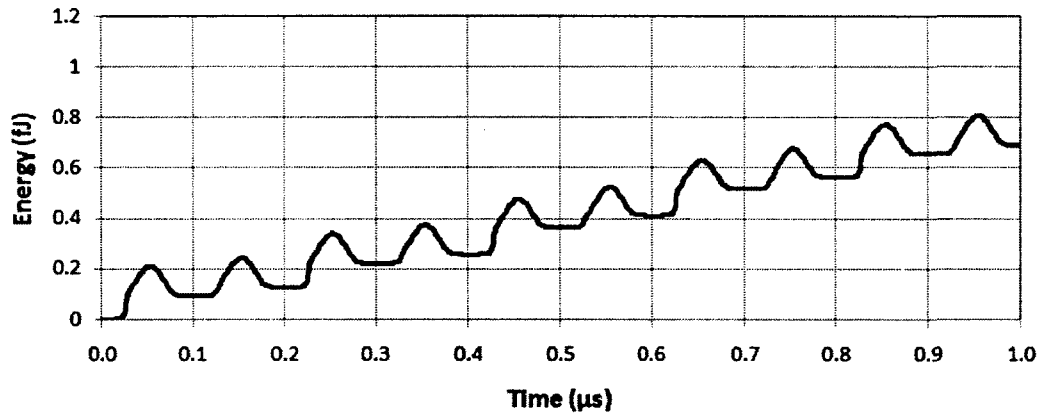


Fig. 16: Energy consumption plot for a single IDPAL buffer/inverter with alternating inputs.

Circuits with multiple layers of logic need to be controlled using alternate clock phases (PC and \overline{PC}), similar to other adiabatic families. To test the energy efficiency of IDPAL using multiple small gates, a simulation was performed using 5 buffers and results

are shown in Fig. 17. The simulation was performed using the same parameters as in the case of the single buffer/inverter. The value of the input is cascaded through each buffer/inverter and reaches the output of the fifth buffer/inverter by the third clock cycle of the first *PC*. It can be observed that when the output is applied at $t = 0.75\mu\text{s}$, it takes $50\mu\text{s}$ for it to evaluate to peak in each buffer, with the final output available at $t = 3.25\mu\text{s}$.

Layout of adiabatic circuits is very different compared to standard CMOS circuits because special care has to be taken to minimize the area taken by the *PC*'s and the larger number of transistors, so a custom cell library must be created. The layouts of the 2-input AND/NAND in IDPAL and 2N2P and NAND gate in CMOS in a 3 metal, 2 poly process are shown in Fig. 18. When compared with its CMOS equivalent, the IDPAL AND/NAND is $2.4\times$ larger, however, this is mostly due to IDPAL being a dual-rail logic family as it produces both an output and its complement. Additionally, the CMOS NAND gate does not contain any connections to a clock, although layout of larger circuits will need to add the clock and other control circuitry. When compared to the smallest layout from an adiabatic implementation (2N2P AND/NAND), the IDPAL AND/NAND uses only 20% more area, but has the added advantage of being able to use a two-phase sinusoidal *PC* instead of a four-phase trapezoidal *PC*. The type of *PC* used and its generation can greatly impact both the area used by a circuit (due to routing constraints) and the power used, as some types of *PC* are more efficient than others. In this case, IDPAL has an advantage over 2N2P as a sinusoidal *PC* can be more efficiently generated than a trapezoidal one. While adiabatic circuits use more circuit area (at most $2.16\times$ in the case of the IDPAL NAND gate), their energy efficiency ($5\times$ less energy) can justify their use in low-power

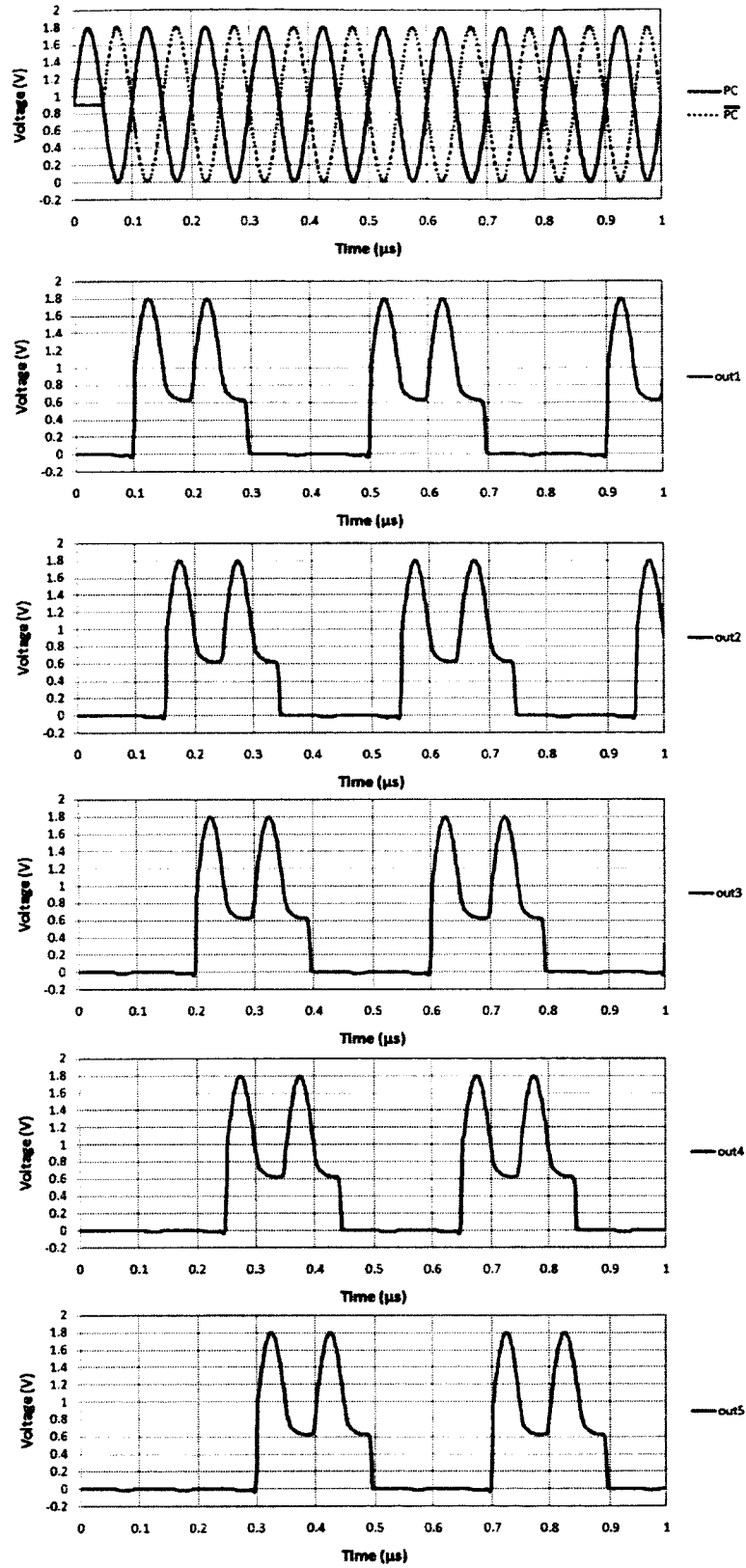


Fig. 17: Simulation waveform for a chain of 5 IDPAL buffers. The input is applied at $t = 0.75\mu s$ and output propagates to output of fifth buffer by $t = 3.25\mu s$.

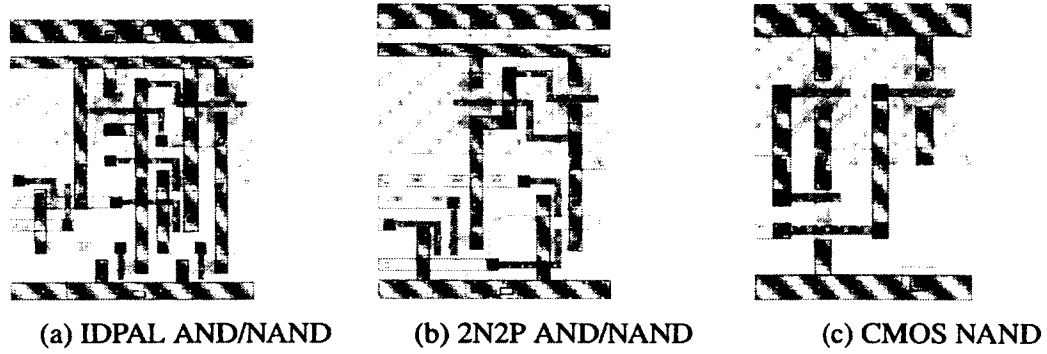


Fig. 18: Layout of IDPAL and 2N2P AND/NAND and CMOS NAND gates in a 3-metal, 2-poly process.

applications. Additionally, more complex gates would reduce the relative increase in area, making IDPAL even more attractive as a new low-power family.

Larger gates and circuits using IDPAL can be built by replacing the two input evaluation blocks (F and \bar{F} in Fig. 10) with NMOS pull-up switching networks and complement, respectively. Being an adiabatic logic family, IDPAL has all the features of other adiabatic logic families, including micro-pipelining and the need for extra buffers in the output chain in order to propagate a valid value to the output. Additionally, similar to other two-phase sinusoidal logic families, IDPAL offers an easier solution to manage the PC 's during layout, both in circuit area used and in routing issues, when compared to adiabatic solutions using more clock phases. As an example, in a two-phase sinusoidal family, a possible solution for routing the PC 's would be to have tracks for one on one side of the circuit and the other on the other side. A four-phase trapezoidal family would need to use two separate metal layers for layout to avoid overlap, which results in a less energy-efficient design.

Simulations of several other basic gates were performed using LTspice in a $0.25\mu\text{m}$

process with a two-phase sinusoidal PC oscillating between 0V and 1.8V to measure the energy consumption of each gate. The W/L ratios are $0.36\mu\text{m}/0.24\mu\text{m}$ and $0.72\mu\text{m}/0.24\mu\text{m}$ for the NMOS and PMOS transistors, respectively. Energy consumption values per cycle for single gates running at 100 MHz with a given load capacitance are shown in Table 3. Simulation values agree with expected values for these basic gates. It was expected that the AND and OR gates would have very similar values for energy consumption due to their complementary implementations at the transistor level, a fact that was confirmed in simulations. It was also expected that the inverter would have the lowest energy usage since it is the simplest of all structures, whereas the XOR gate would have the highest energy usage due to a larger input stack and thus a larger resistance and capacitance. All gates were simulated in their usual implementations without any modifications.

Table 3: Energy dissipation (J) per cycle for each type of gate and load at 100 MHz.

Gate/Load	1fF	10fF	100fF
Inverter	2.25E-17	8.46E-17	7.56E-16
AND	2.38E-17	8.61E-17	8.14E-16
OR	2.37E-17	8.59E-17	8.22E-16
XOR	2.51E-17	8.68E-17	8.35E-16

In order to confirm the energy efficiency of the proposed logic with composite gates, a full-adder (FA), shown in Fig. 19, with a 10fF output load was simulated and compared to its equivalent implementations in CMOS, CAL, 2N2P, and 2N-2N2P. Several frequencies were tested in order to observe the advantage of the proposed family over other adiabatic families and CMOS equivalents in a wide operating range. A more compact design was selected with the Sum and C_{out} being calculated in one stage instead of three stages and

several gates that would be otherwise needed for a traditional implementation. The FA in the current implementation uses 38 transistors, while the CAL, 2N2P, and CMOS full-adders use 42, 34, and 28 transistors, respectively. An even more compact implementation is possible for the *Sum* term in IDPAL by combining the inputs forming a 2-input XOR gate and reducing the total number of transistors required for the full-adder to 30, making the difference between the IDPAL and CMOS implementations even smaller.

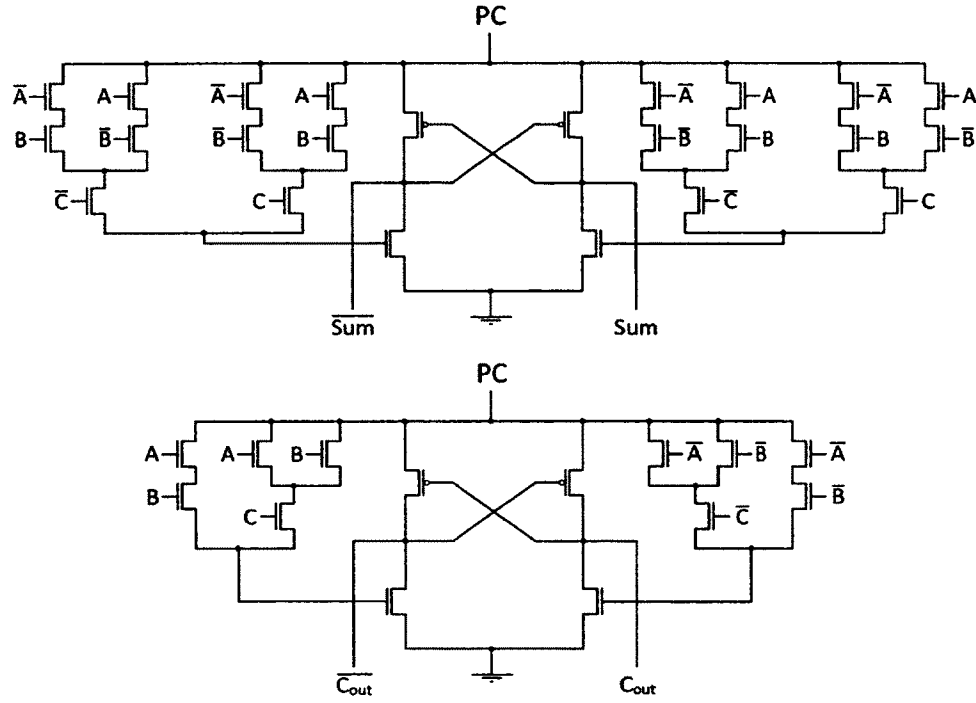


Fig. 19: IDPAL full-adder implementation.

Simulation waveforms for the FA implementation in IDPAL are shown in Fig. 20. All 8 possible input combinations, ranging from “111” down to “000” are covered to confirm correct operation. Similar to the case of the buffer/inverter, when the evaluation of inputs does not change the output from one clock to the next, the value of the output does not fall to *GND* level but stays at $|V_{TP}|$, contributing to the energy efficiency of the circuit.

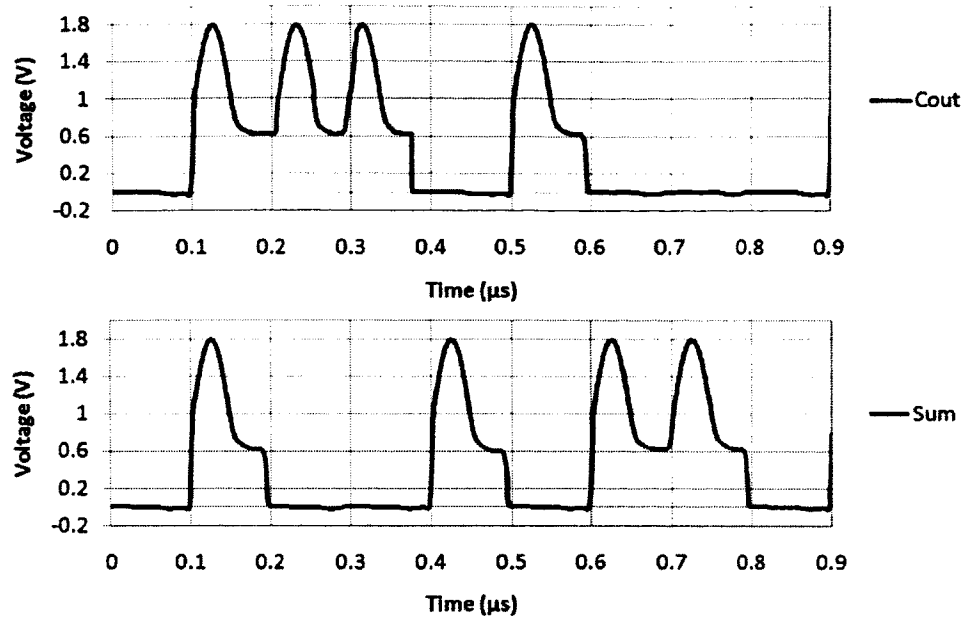


Fig. 20: IDPAL full-adder simulation waveforms. Inputs cycle from “111” down to “000”.

Full-adder simulations were also implemented for other adiabatic families and the results agree with the results reported in [8] and [10], allowing us to conclude that our models and implementations function correctly. As it can be observed from Fig. 21, the FA implemented using IDPAL is up to 67% more energy-efficient than other adiabatic full-adders (CAL FA at 100 MHz) and 79% more energy-efficient than CMOS (at 10 MHz). These results are in agreement with previous small-scale tests performed on simple gates. Although the IDPAL full-adder uses more transistors than a CMOS FA (38 vs. 28 in CMOS), it uses less energy due to a more energy-efficient operation.

III.2 Results for Large Arithmetic Circuits

Building circuits larger than a few gates in IDPAL is accomplished in a straightforward manner. While the design is relatively simple, care has to be taken in the choice of

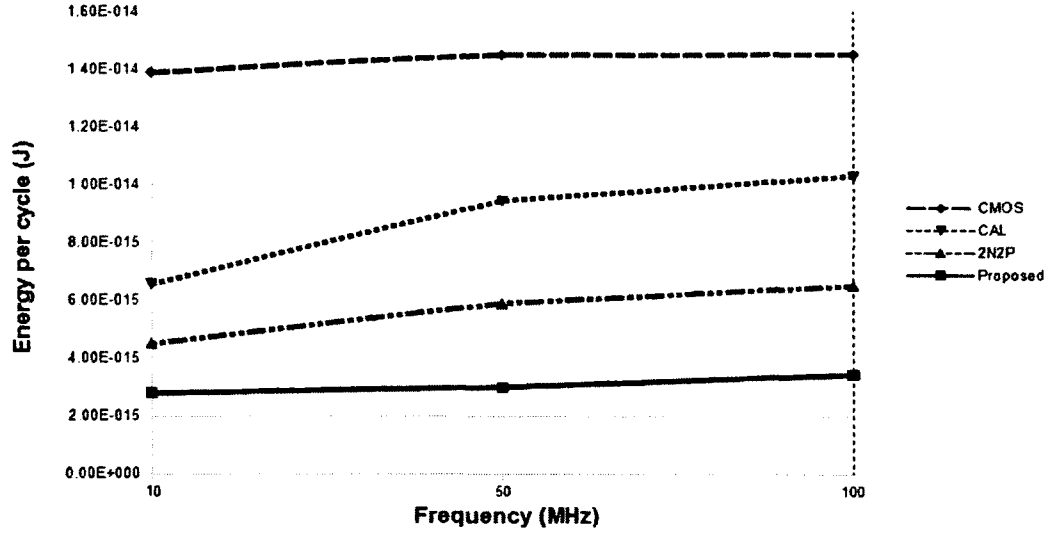


Fig. 21: Energy usage (J)/cycle for a full-adder with 10fF load at different frequencies.

implementation. Similar to other adiabatic logic families, IDPAL needs to use buffers to maintain the validity of a given signal as it moves through the micro-pipeline due to its continuously-oscillating nature. If this issue is overlooked, the resulting circuit will produce garbage outputs. Depending on the chosen implementation, buffers can contribute a significant number of additional transistors to an adiabatic circuit, so choosing the most compact implementation in terms of number of evaluation stages and number of buffers is very important.

In this experiment, a Kogge-Stone Adder (KSA) of several sizes (8-, 16-, and 32-bit) has been simulated at different frequencies with 10fF output loads to confirm the energy efficiency of IDPAL for very large circuits. KSA is a type of Carry-Lookahead Adder (CLA) proposed in [31] which produces results in $O(\log_2 n)$ time, resulting in 5, 6, and 7 micro-pipeline stages for each adder size, respectively. This adder takes more area to implement than other fast adders due to a larger number of gates used, but has a constant

fan-out of 2 at each stage, which increases performance. KSA is the industry standard for adder implementation and is considered the fastest adder design available.

The diagram of the 32-bit KSA is shown in Fig. 22 for reference. KSA uses a structure similar to a CLA with Propagate and Generate blocks, but restructures them into two new blocks, called Black Cell and Grey Cell. A Grey cell acts as a traditional CLA Carry-generation unit, whereas the Black cell includes the functionality of a Grey cell and adds the Propagate functionality from the CLA. Several KSAs were simulated and the total number of each type of gate and transistors used in each IDPAL implementation is given in Table 4. These numbers show that with each increase in the length of the adder, the main contributions in the increase of the transistor count come from non-linear variations in the numbers of Black Cells and Buffers. The KSAs were compared to similar implementations in CAL, 2N-2N2P, and CMOS. Even when using the most compact implementation of a fast adder, the number of transistors contributed by the buffers needed for an adiabatic implementation is around 25-26%, stressing the importance of choosing an appropriate design in implementation. The buffers in the diagram are only needed for the adiabatic implementations and are removed for the CMOS implementation. Micro-pipelining in adiabatic circuits also allows for the first rounds of outputs of the 32-bit KSA to be available only after 7 *PC* phase delays, however, if the pipeline does not have any stalls, the outputs are available one per *PC* cycle afterwards.

The 32-bit KSAs were simulated in LTspice using a TSMC 0.25 μm process with a W/L ratio of 0.72 μm /0.24 μm and 0.36 μm /0.24 μm for p- and n-type transistors, respectively, with *PC*'s oscillating between *GND* and 1.8 V. Several frequencies were tested

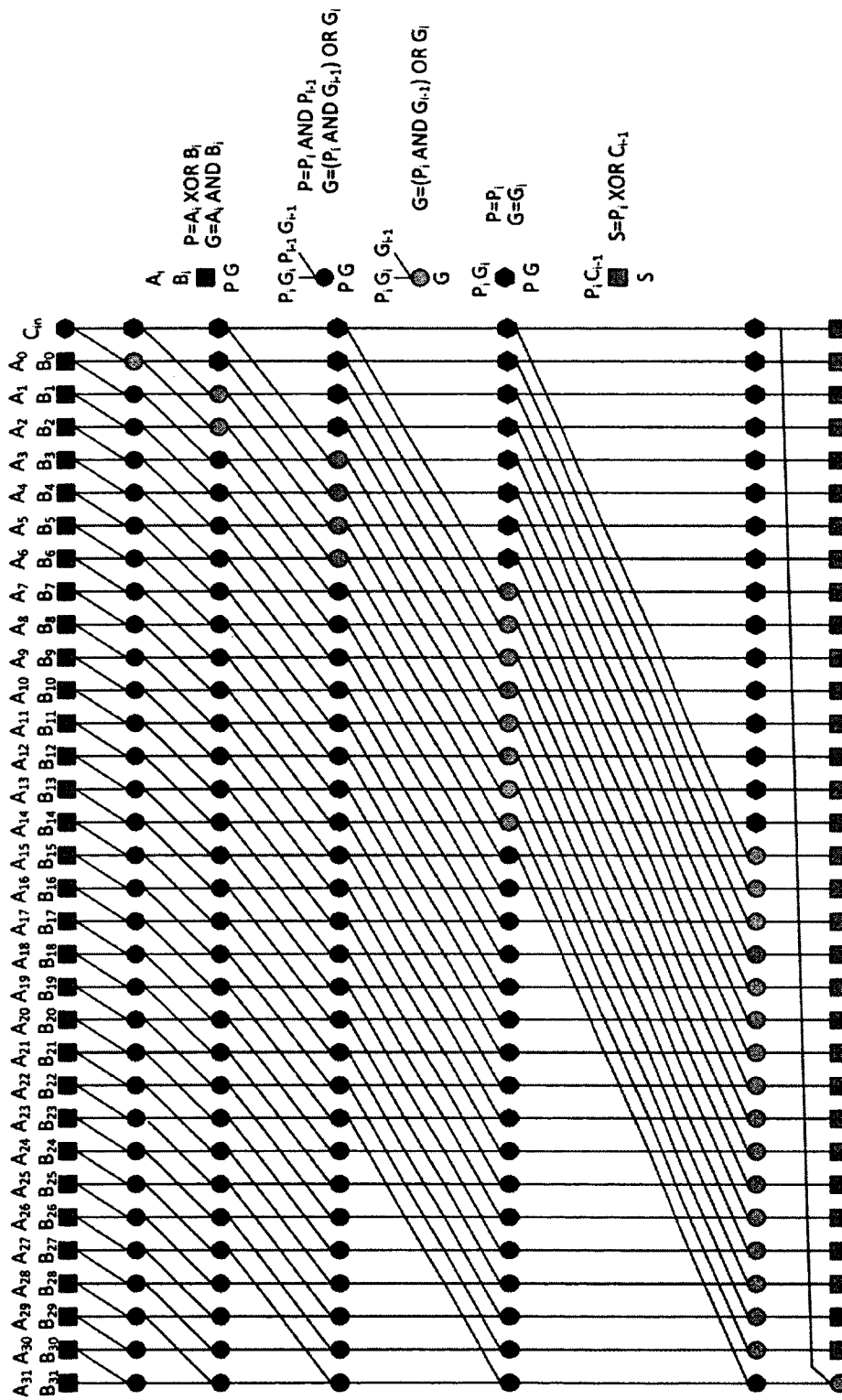


Fig. 22: Diagram of a 32-bit Kogge-Stone Adder [31].

Table 4: Number of each type of gate and total transistor count for the 8-, 16-, and 32-bit KSAs implemented in IDPAL.

	AND	XOR	Grey	Black	Buffer	Transistors
8-bit	8	16	8	13	32	762
16-bit	16	32	16	38	80	1836
32-bit	32	64	32	103	192	4350

in order to observe the energy advantages of the proposed family over other adiabatic families and CMOS equivalents and simulation results are plotted in Fig 23. As can be observed from the plot, IDPAL maintains its energy efficiency against other implementations for all frequencies tested. The proposed family uses at least 22% less energy than 2N-2N2P at 10MHz and 71% less energy than CMOS at 100 MHz. Energy consumption values for all simulations are given in Table 5.

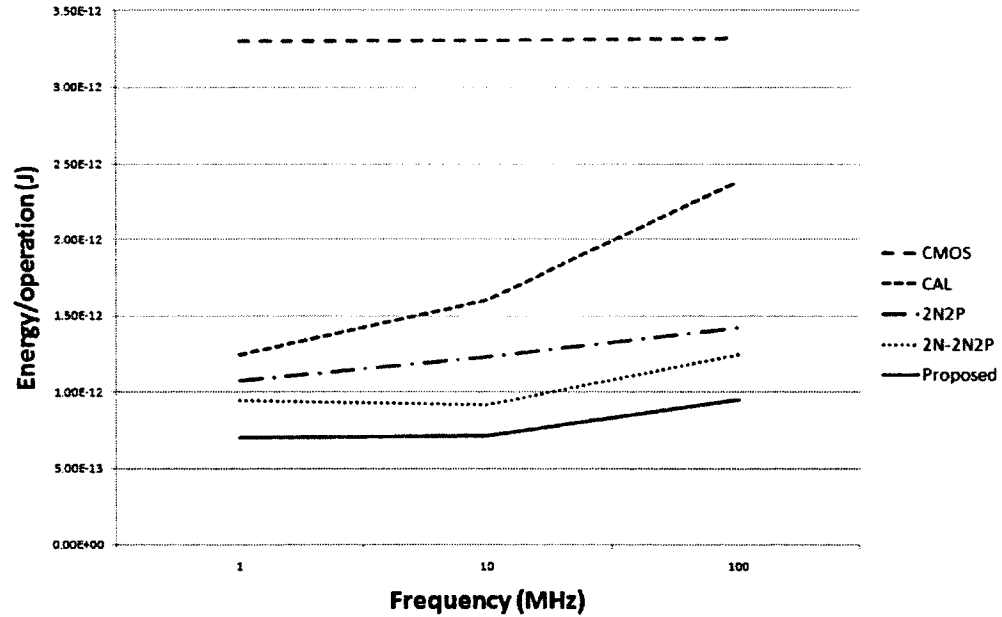


Fig. 23: Energy usage (J)/operation for a 32-bit KSA with a 10fF load at different frequencies.

The use of adiabatic logic increases the complexity of the circuit and the number of transistors used (Table 5), but the energy savings can justify this increase. The largest increase in the number of transistors between CMOS and other adiabatic implementations is due to the high number of buffers used to buffer the signal from stage to stage (around 26% of the total number of transistors). The largest number of transistors used were in the CAL implementation due to the use of two control transistors per gate in order to achieve single-phase operation. The 2N-2N2P and IDPAL implementations had the same number of transistors, however, the proposed implementation has a 22-26% better energy efficiency at the frequencies tested.

Table 5: Energy dissipation per addition operation at different frequencies and number of transistors needed for each implementation of the 32-bit KSA.

Family	Energy/op (J)			Number of transistors
	1MHz	10MHz	100MHz	
IDPAL	6.98E-13	7.12E-13	9.49E-13	4350
CMOS	3.30E-12	3.31E-12	3.32E-12	2528
CAL	1.24E-12	1.60E-12	2.38E-12	5402
2N-2N2P	9.42E-13	9.15E-13	1.24E-12	4350
2N-2P	1.07E-12	1.23E-12	1.42E-12	3298

An experiment was performed with the 4 adiabatic families investigated in this dissertation (CAL, 2N2P, 2N-2N2P, and IDPAL) with a 32-bit KSA running at 1GHz in order to observe the energy efficiency of each implementation and compare it with a CMOS implementation. Adiabatic circuits have an advantage over an equivalent CMOS implementation at lower operating frequencies due to longer charging times, allowing for smaller resistive losses in the transistor networks and, thus, increased energy efficiency. As the

frequency of a circuit is increased, the charging time is decreased proportionately. Therefore, it is expected that the energy efficiency of adiabatic circuits at speeds of 1GHz will be reduced compared to lower speeds, generally surpassing the energy consumption of the CMOS equivalent due to the larger number of switching transistors.

Experimental results (Fig. 24) show that the energy usage in all adiabatic circuits grows dramatically when running at 1GHz, as expected. The amount of energy usage between the adiabatic families are in the same order as running at other frequencies. It can be observed from the Figure that a CAL implementation has a $5\times$ larger energy usage than CMOS, whereas the IDPAL implementation uses around $2.75\times$ more than a CMOS implementation. Again, the much larger energy usage in adiabatic circuits at this frequency is due to the shorter charging time compared to other frequencies and the larger number of switching transistors compared to a CMOS implementations. These results are similar to previously-reported results for circuits running at frequencies higher than 100MHz.

Another experiment was performed only with the two most energy-efficient logic families (2N2P and IDPAL) in order to observe the approximate frequency at which the adiabatic implementations equal the energy consumption of a CMOS implementation. A 32-bit KSA was simulated at 200MHz and it was found that the 2N2P implementation uses $2.65E^{-12}$ J/operation whereas the IDPAL implementation was around 18% lower at $2.18E^{-12}$ J. Both values are very close to the CMOS energy dissipation for the same circuit at $3.3E^{-12}$ J. Based on previous energy usage values from 100MHz and 1GHz, it is estimated that both 2N2P and IDPAL would cross the energy threshold of a CMOS

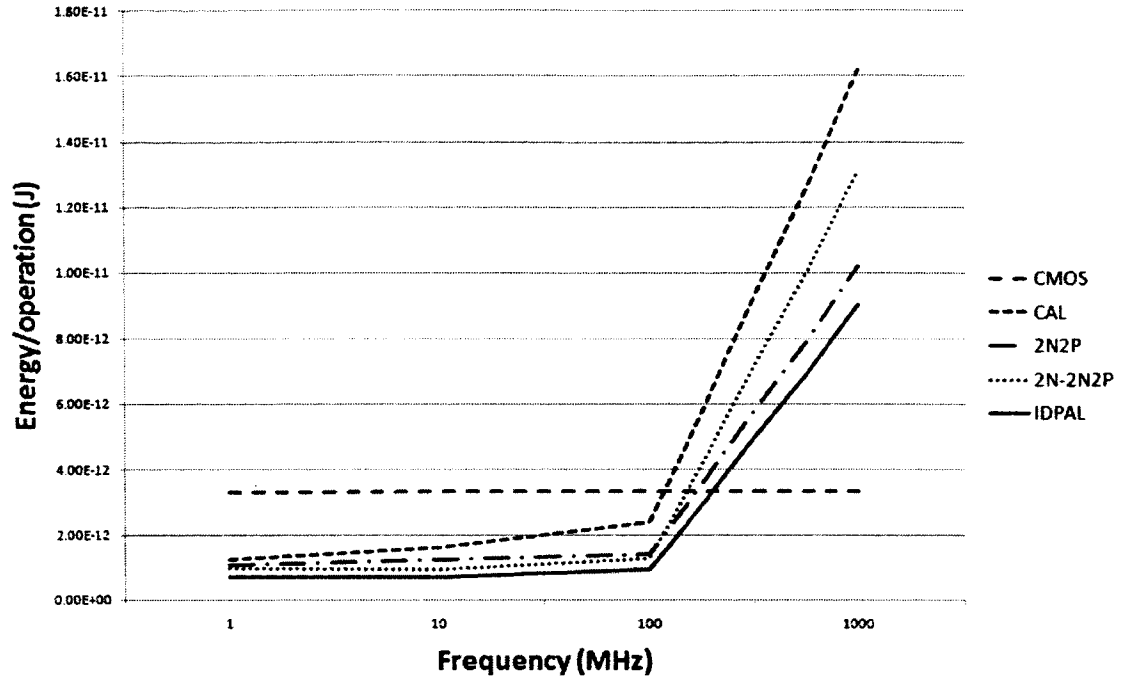


Fig. 24: Energy usage for a 32-bit KSA at frequencies from 1MHz to 1GHz.

implementation at a frequency slightly lower than 250MHz.

Lastly, an even larger design was simulated in order to observe the energy efficiency of IDPAL – 8-, 16-, and 32-bit Wallace multipliers [32]. A Wallace multiplier design was chosen because it has one of the lowest number of evaluation stages compared with other hardware multipliers, which implies that the result is available in fewer clock cycles in an adiabatic implementation and fewer buffers will be used in its implementation.

The operation of a Wallace multiplier is as follows. First, the partial products are calculated using 2-input AND gates for each pairwise bit combination of inputs. Next, a series of full adders are used to compress three rows of intermediary results of the same weight into two rows by using the following rule. If there are three rows of intermediary results available, then full adders are used on each three results, which will generate a bit

of the current weight and a bit of the next higher weight. Next, if there are two inputs of the same weight, they are input into a half adder. Lastly, any remaining intermediary results are buffered to the next layer. If three rows of intermediary results are not available, the existing intermediary results are not grouped in any half-adders but buffered to the next stage. Once the number of partial product layers reaches two, an adder is used to add them and get a final result. In this implementation, a KSA was used as the last stage for faster results. Larger multipliers follow the same idea, but include a much larger number of compression stages and transistors in their implementations.

Simulation results from the 32-bit Wallace multiplier using a $0.25\mu\text{m}$ process with W/L ratio of $0.72\mu\text{m}/0.24\mu\text{m}$ and $0.36\mu\text{m}/0.24\mu\text{m}$ for PMOS and NMOS transistors, respectively, with PC 's oscillating between GND and 1.8 V at several frequencies are shown in Figure 25. These results show that IDPAL maintains its advantage in energy efficiency even for very large circuits and would be a suitable candidate for a low-power circuit implementation. Another observation made during the building of these multipliers was that the number of buffers used at each multiplier size decreased as a percentage of the total as the size of the multiplier increased. For the 8-bit multiplier, the number of transistors used in buffers account for 13.1%, whereas for the 32-bit multiplier it was only 5.1% of the total transistor count. This is a clear example of the influence of choosing a suitable design on the number of transistors used for an adiabatic implementation and its energy efficiency.

In order to test IDPAL energy efficiency with process scaling, the 32-bit KSA was also implemented in a TSMC $0.18\mu\text{m}$ process with W/L ratios of $0.54\mu\text{m}/0.18\mu\text{m}$ and

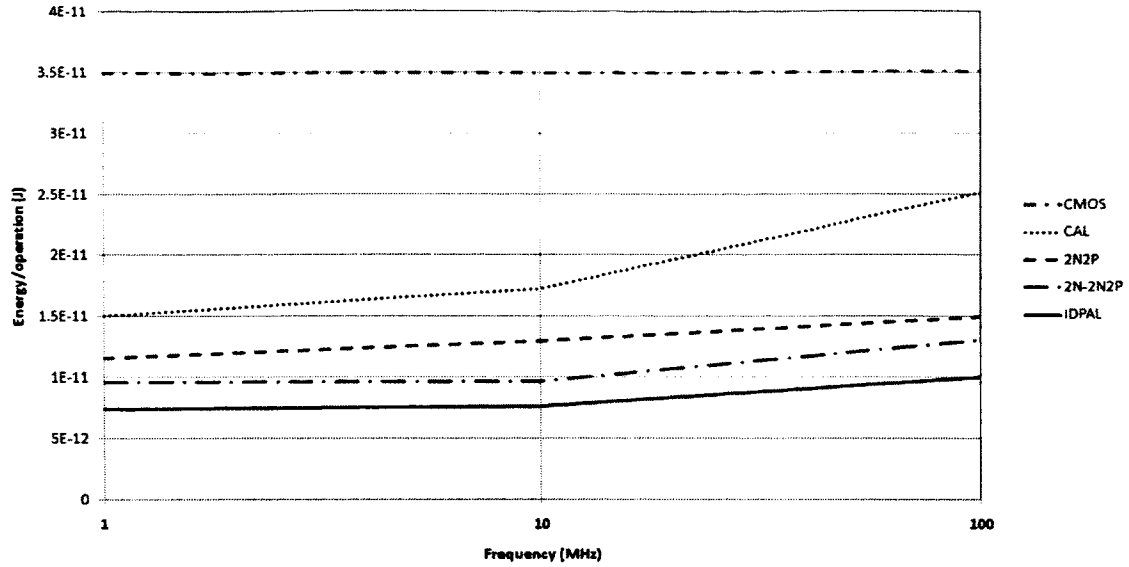


Fig. 25: Energy usage (J)/operation for a 32-bit Wallace multiplier with a 10fF load at different frequencies.

0.27 μ m/0.18 μ m. Results from these simulations with 10fF loads and *PC* oscillating between *GND* and 1.8V are shown in Fig. 26. Dennard's Law [29] indicates that a change in the feature size causes a decrease in the dissipated energy proportional to $1/\kappa$, where κ is the ratio of the original to the new feature sizes, if all other values remain constant (e.g. operating voltage). When scaling from a 0.25 μ m process to a 0.18 μ m process, we have $\kappa = 0.25/0.18 = 1.388$, making the energy dissipation for the new process approximately $0.72\times$ the old value, although in most cases in literature this is rounded to 0.7. Simulation results from the 0.18 μ m implementation of the 32-bit KSA show that the new energy dissipation values are within the range 0.69 to 0.74 of the results for the 0.25 μ m implementation. Variations are likely due to the rounding of values in calculating the energy dissipation, but are within previously reported limits on scaled circuits, including

adiabatic circuits [18]. The result agree with expected values since the two adjacent processes are both far from the current deep-sub-micron processes and leakage current and leakage energy do not have a large contribution to the overall energy dissipation. At these processes, a reduction in feature size implies a proportionate reduction in the area and capacitance of a circuit and a similar loss mechanism. Even a deep-sub-micron processes, where leakage currents are much larger, the reported energy scaling closely approximates Dennard's Law for frequencies between 1 MHz and 100 MHz [18].

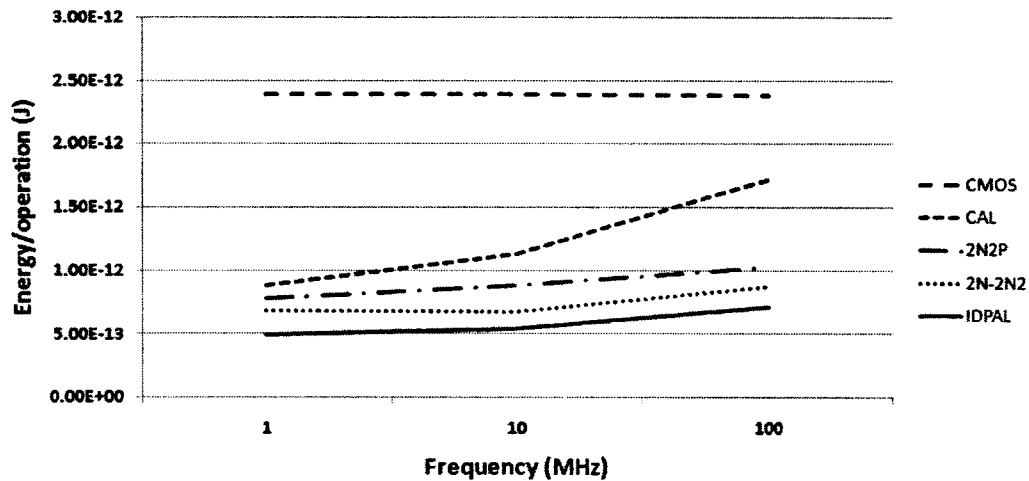


Fig. 26: Energy usage/operation for a 32-bit KSA with a 10fF load at different frequencies in a TSMC 180nm process.

Process scaling also enables circuit operation using a smaller operating voltage. One of the main drivers of lower energy dissipation has been the quadratic dependence of energy loss with operating voltage ($E = 0.5CV^2$). However, CMOS circuits are limited in the amount the operating voltage can be reduced since a lower operating voltage increases the gate propagation delay, resulting in a slower circuit. CMOS circuits are limited by design since their propagation delay depends on the critical path. Adiabatic logic is not

limited in its voltage reduction by any timing constraints since each gate operates independently and the propagation delay is negligible. The limiting factor for the operating voltage reduction in IDPAL are the 2 NMOS transistors in the function evaluation part of a gate, as they need to be conducting in order for the gate to be able to generate the proper outputs. More formally, $V_{DD,min}@IDPAL = V_{TN}$. If the operating voltage falls below this value, the circuit will malfunction and produce invalid outputs. Other adiabatic families have similar operating voltage constraints, with the operating voltage minimums ranging between $\max(V_{TN}, |V_{TP}|)$ (for 2N2P) and $2V_{TN}$ (for PFAL) [18].

III.3 Chapter Conclusion

This chapter presented IDPAL – a new partially-adiabatic logic family capable of achieving up to 79% reduction in energy compared to equivalent CMOS circuits and at least 25% less energy than state-of-the-art adiabatic families. Simulation examples ranging from small gates (buffer/inverter, AND, OR, XOR) to very large complex circuits (adder, multiplier) were presented. Results revealed a consistent reduction in energy dissipation across circuits of all sizes, making IDPAL a promising alternative to CMOS in low power circuits. An experiment using a 32-bit KSA showed that energy consumption in IDPAL circuits scales well with feature size reduction, although more experiments are required for the deep-sub-micron feature sizes. It was also shown that IDPAL, like other adiabatic logic families, does not have as strict of a limitation on the scaling of operating voltage due to its micro-pipelining.

CHAPTER IV

POWER ANALYSIS ATTACK SECURITY USING IDPAL

In this chapter, the resistance of power analysis attacks of IDPAL as applied to low power circuits is explored. IDPAL has very good energy consumption characteristics, as shown in Chapter III, but another feature of interest is the almost-constant energy dissipation, regardless of the input combinations [4]. This makes IDPAL a promising countermeasure for power analysis attacks in low power circuits. This chapter explores the security aspect of IDPAL against power analysis attacks using several security metrics and power models. It also compares its security against state-of-the-art secure logic families and other partially-adiabatic logic families.

IV.1 Motivation for Combinational Circuit Security

As power analysis attacks are based on correlations or energy differences between intermediate values, many power analysis attacks are focused in places where these values are stored – registers. This is a great opportunity for an attacker because register activity is clocked and reproducible in time, which means that register statistics will have a strong correlation with either the contents of a register or changes in its contents.

In this work, the focus is on the hardware security of combinational logic, including large arithmetic circuits, in contrast with some of the previous approaches. This work was motivated by the fact that a majority of the transistors in most circuits are used for combinational logic rather than sequential logic. For example, the AES encryption algorithm

uses around 80% combinational logic by area and power dissipation [77], with the rest being sequential logic and interconnects. However, the combinational part of the circuit is not the most frequent the target for power analysis attacks. One of the reasons for not attacking combinational circuits is the increased difficulty of the attack because the combinational gates evaluate at data-dependent times and might also include logic hazards, whose power dissipation is difficult to model.

In a system where all registers are protected against power analysis attacks, which can become a reality given all previous attempts at secure logic implementations, the only source of possible information leakage resulting from different inputs to combinational circuit. Given the nature of CMOS circuits, one may observe that in a given circuit, the registers can consume power on the rising edge and/or falling edge of the clock, whereas combinational logic consumes power shortly after the registers have evaluated. With this knowledge in mind, it would not be difficult to balance the energy consumption of registers as their number is relatively small compared to the combinational circuits. Both SPA and DPA could target combinational logic circuits if the registers are protected. In this chapter, we assume that required protections are in place for the easy targets (registers) and focus on the secure implementations of combinational circuits in order to thwart power analysis attacks in the future.

Since the other theme of this work is providing a solution for low power environments, secure logic applications studied have to also fulfill the low power requirement. Many previously-proposed technologies often use significantly more energy than a regular CMOS circuit, making them unsuitable for our applications. Some of these solutions

are compared with IDPAL and other adiabatic families in the next sections.

IV.2 Security Metrics and Power Models Used

Resistance against power analysis attacks can be measured theoretically using several methods. In a low power secure design environment, the first criteria is low power consumption. The average energy (\bar{E}) usage per operation is measured as the parameter indicative of low power performance. Along with the average energy consumption, the minimum (E_{min}) and maximum (E_{max}) energy usage for different input combinations is also tracked to give a better idea of the energy imbalance between the best and worst case scenarios for a given circuit. These values are used in a second parameter called Normal Energy Deviation (NED), calculated as $(E_{max} - E_{min})/E_{max}$. A third parameter used for resistance against power analysis attacks is the Normalized Standard Deviation (NSD), calculated as σ_E/\bar{E} , where σ_E is the dissipated energy standard deviation. The fourth parameter used in this work is the current trace of a circuit (either $i_{V_{dd}}$ or i_{PC}) which is directly proportional to the power consumption and could leak information about individual bits processed if enough traces are available for analysis.

A power model for the variability of power consumption for a circuit has to be carefully chosen in order to be able to successfully perform power analysis attacks on the given circuit. There are several methods for constructing a power model, one of which is simulation of the device under attack. This can be especially useful if the architecture is known in advance and simulation results can provide a more accurate prediction of the power consumption of the device. However, in most cases the architecture of a circuit is

not known in advance or it is infeasible to simulate it, so a more general model should be used. Two of the most common leakage models are the Hamming weight and Hamming distance models [82].

The Hamming Weight model is the most basic power consumption model and has been used in SPA and DPA attacks extensively in the past [40]. This model can be applied to approximate the power consumption of data buses and relies on the fact that a bus will consume an amount of power proportional to the number of bits switched ON on the bus. It assumes that a bus with no bits ON will consume an insignificant amount of power compared to when all bits are ON. This model can be used when no information about a particular implementation exists, but it is not very effective at showing accurate values for the power consumption of a circuit. This model is also more effective on pre-charged buses where the pre-charged value is “all zeroes”, which yields the power consumption model to depend on $W_H(0\dots0 \oplus Y_i) = W_H(Y_i)$, where Y_i is the value of an intermediate variable at time i [82].

The second most common power model is the Hamming Distance model – an extension of the Hamming Weight model. This model uses the changes in value at a given time to determine the power usage. Changes in value can occur in both combinational and sequential circuits and the power model uses the idea that the approximate amount of power consumption is proportional to the number of ‘0’ \rightarrow ‘1’ and ‘1’ \rightarrow ‘0’ transitions in the circuit. The number of bit transitions can be calculated using the Hamming weight of the XOR of the two different output values. More formally, $D_H(Y_{i-1}, Y_i) = W_H(Y_{i-1} \oplus Y_i)$, where i is the time at which the switch from one value to the other occurs. This model

gives a better approximation for the power consumption than the Hamming Weight model, but it requires a more in-depth knowledge of the device under attack.

IV.3 Experimental Results

A simple experiment to demonstrate the energy usage differences and the information that can be gathered from circuits when using CMOS gates starts the procedure. In this example, an inverter is implemented in CMOS with a 10fF load at 10 MHz and a periodic input with 25% duty cycle. The current trace is shown in Fig. 27. Note that the current amplitudes are shown as negative when the outputs are charging and positive when the outputs are discharging. It is relatively easy to notice the input transitions based on the inverter current trace and guess the type of gate used. It can be observed that the input is periodic with a 400ns period, discharging the output at $t = 475ns$ and charging 100ns later and keeping the output constant for 300 ns. If it is known that the gate is a function of one input, then it can be guessed that it most likely is an inverter. Larger gates can be analyzed in a similar manner, but require more sophisticated analysis and statistics. This small experiment shows a definite need for countermeasures in combinational circuits and the unsuitability of CMOS circuits to act as secure hardware circuits.

Another experiment was performed using 2-input NAND gates implemented in SABL, WDDL, 2N2P, CAL, and IDPAL. The gates were simulated in LTspice with a 10fF output load using a 0.25 μm process with a W/L ratio of 0.72 μm /0.24 μm and 0.36 μm /0.24 μm for p- and n-type transistors, respectively. The experiment was performed at 100MHz using a square-wave clock for SABL and WDDL, trapezoidal *PC* for CAL and 2N2P, and

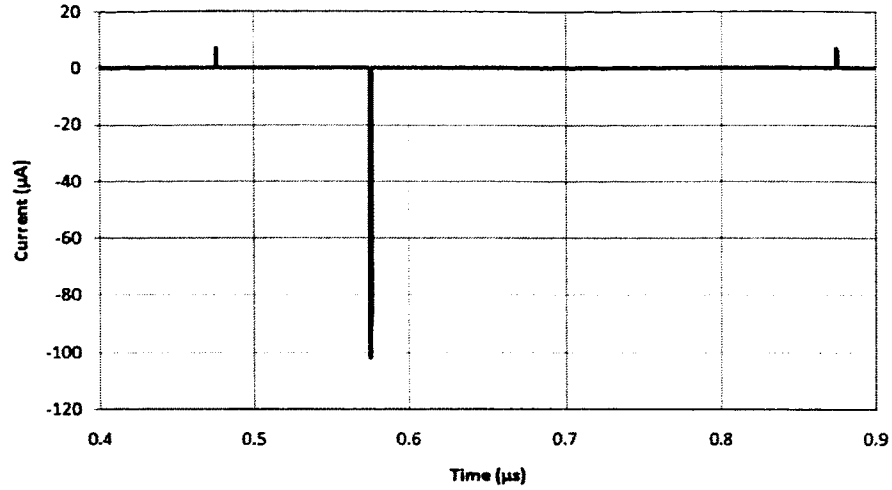


Fig. 27: Current trace for a CMOS inverter.

sinusoidal *PC* for IDPAL, oscillating between *GND* and 1.8 V.

The current drawn from the source (or *PC*, in case of the adiabatic circuits) was taken as the first security metric. Current traces from each of the simulated NAND gates cycling through the basic 4 possible inputs (“11” → “10” → “01” → “00”) are shown in Figures 28–32. The input sequence starts at time $t = 40$ ns and repeats every 40 ns. Note that the current traces in the following Figures have a negative amplitude for current supplied from the power source and a positive amplitude for the current returned to the source (only applies to adiabatic circuits). The short positive peaks for SABL and WDDL are due to input switching activity and are not considered as energy recovery but discharging of the output capacitors.

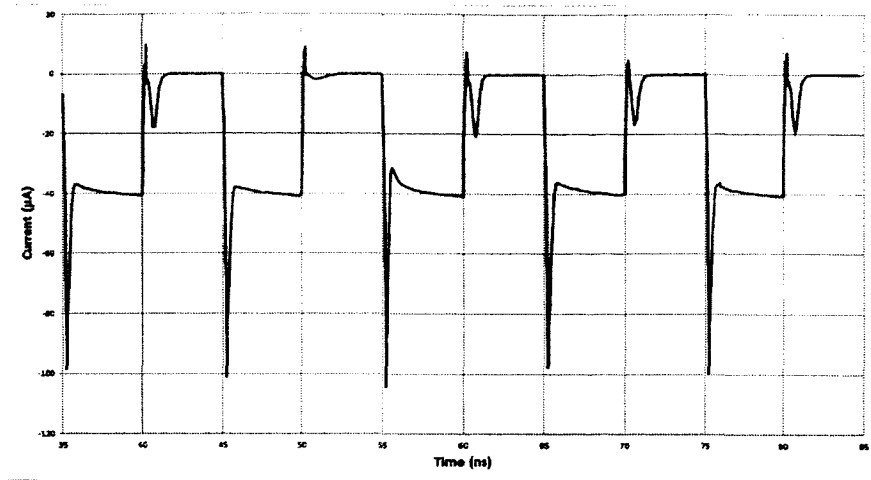


Fig. 28: Current trace for a 2-input SABL AND/NAND gate.

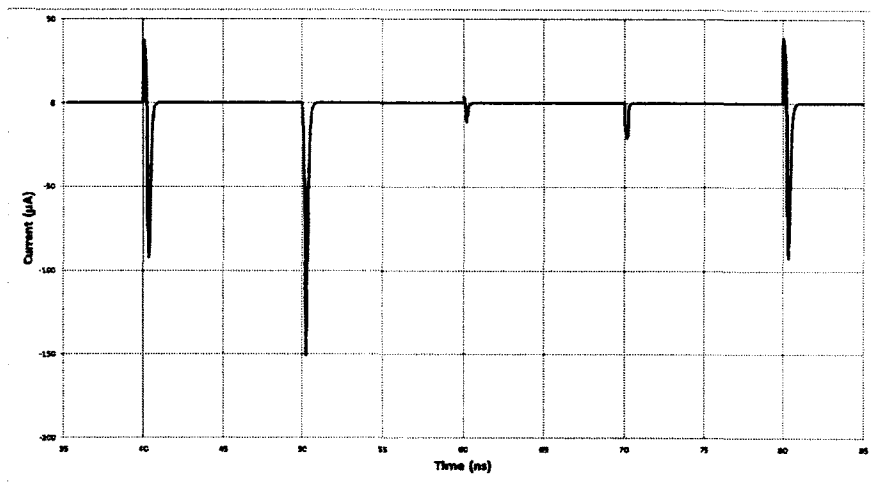


Fig. 29: Current trace for a 2-input WDDL AND/NAND gate.

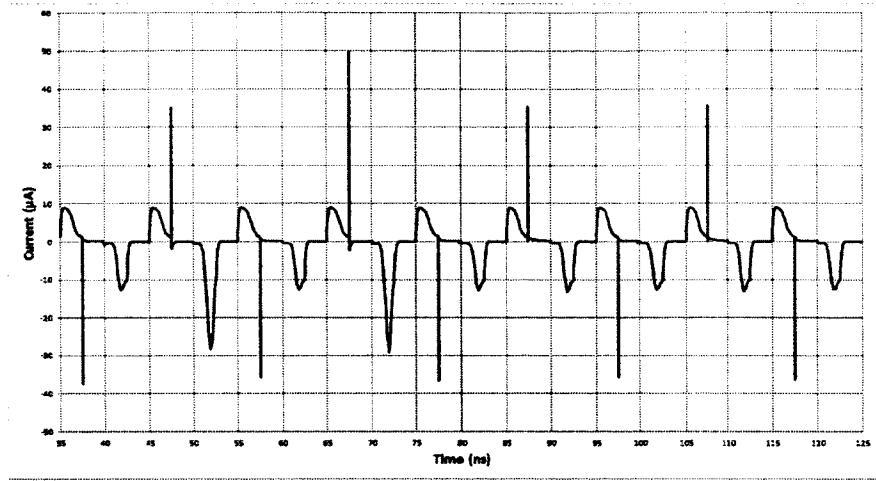


Fig. 30: Current trace for a 2-input CAL AND/NAND gate.

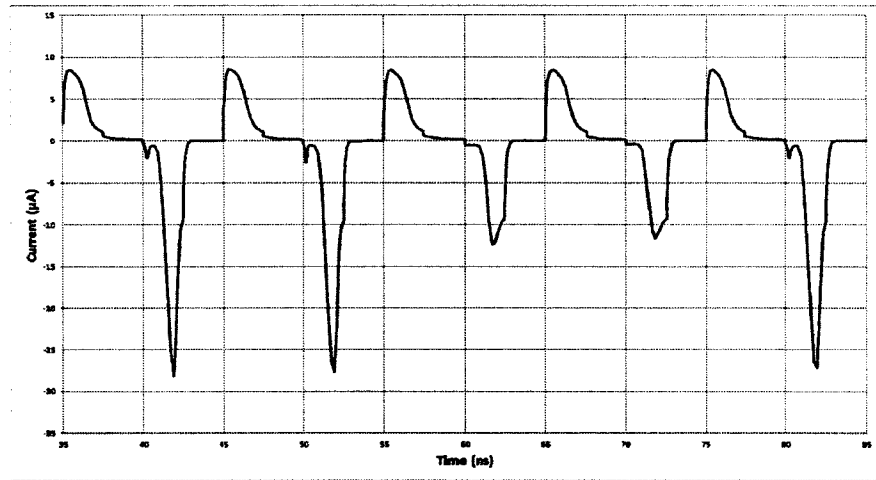


Fig. 31: Current trace for a 2-input 2N2P AND/NAND gate.

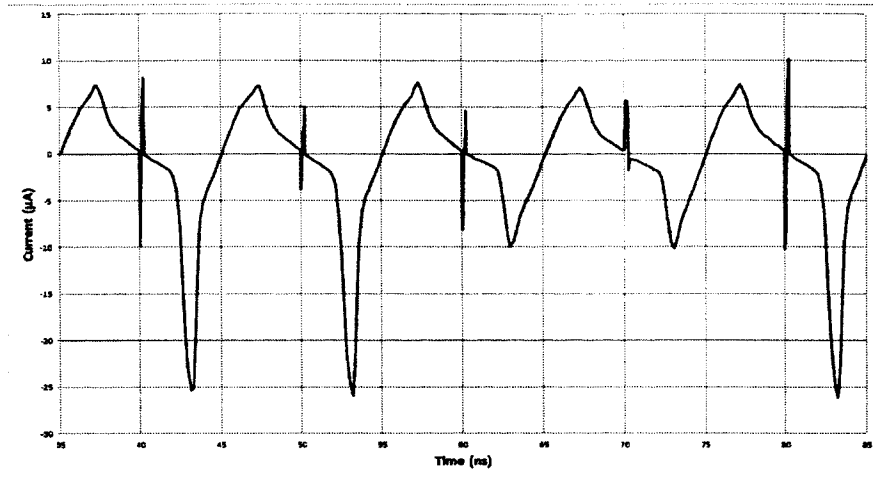


Fig. 32: Current trace for a 2-input IDPAL AND/NAND gate.

Figure 33 presents current traces from all implementations on the same scale to provide a better view of the magnitudes for each family. It clearly shows very large spikes and magnitude differences in current traces for the SABL and WDDL implementations and much smaller peaks and variations in the adiabatic implementations. This makes adiabatic implementations a promising solution in reducing both the energy consumption and improving hardware security against power analysis attacks.

Differences in the current drawn from the source can be used in SPA and DPA attacks to observe the power dissipation and extract information about a circuit. In CAL and 2N2P, the current spikes are larger than IDPAL due to using a trapezoidal *PC* instead of a sinusoidal *PC*. The sinusoidal *PC* allows for a longer charging time, leading to smoother current variations and more uniform energy dissipation. When compared to the two non-adiabatic solutions, the largest difference between current peaks is larger in IDPAL compared with the best previous implementation ($15\mu\text{A}$ vs. $6\mu\text{A}$ in SABL).

However, the largest peak current in the current implementation is almost $4\times$ less than in SABL. The significantly lower peak current is an advantage of IDPAL and could further contribute to the security of information in a given circuit, especially if the attacker does not possess tools capable of measuring at this resolution. Additionally, the basic two-input AND/NAND gate uses 12 transistors in SABL, 10 transistors in WDDL, and only 8 transistors in IDPAL. Thus IDPAL offers a method to design secure circuits while using a lower number of transistor, which implies less area and, consequently, significantly lower energy dissipation than both WDDL and SABL.

The figures of merit described earlier (NED and NSD) along with the average energy usage measure are given in Table 6. Simulation results show that information leakage in the NAND gates is very low for the SABL family ($NSD=2.71\%$), however, this comes at a very large energy consumption as the gate draws the maximum amount of energy in the pre-charge phase for every input ($\bar{E}=382\text{fJ}$), over $34\times$ the energy used in the IDPAL NAND gate. WDDL yielded much better results in terms of energy consumption (only approximately $5\times$ the energy for IDPAL), but the differences in the amount of energy used are more significant than SABL gates ($NSD = 90.08\%$). Both CAL and 2N2P offered much lower energy consumption compared with SABL and WDDL, mainly due to being adiabatic implementations, and leaked similar amounts of information between themselves. IDPAL has better energy characteristics than both CAL and 2N2P and leaks a slightly smaller amount of information ($NSD = 40.7\%$). It can be observed that the NED and NSD for SABL are much lower than any other approaches. This is mainly because these are normalized values and SABL has a very large energy consumption. Nominal

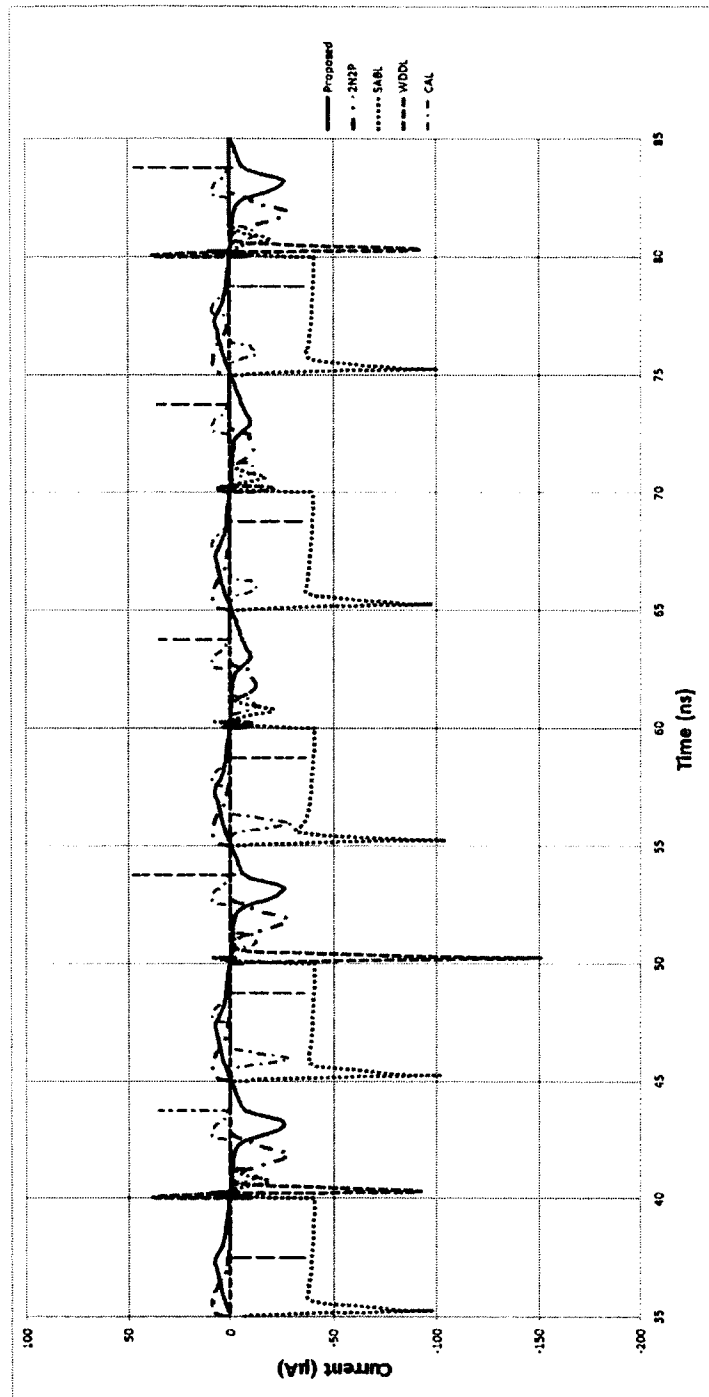


Fig. 33: Current traces for all AND/NAND gates in the experiment.

value for the energy differences for SABL is approximately equal to the deviation of 2N2P and CAL and roughly $3\times$ larger than IDPAL. In conclusion, SABL has very good information leakage characteristics compared to IDPAL, but it comes at a much larger increase in average energy consumption, making it unfit for low-power applications.

Table 6: Security metrics for the NAND/AND gate for each logic family.

	SABL	WDDL	2N2P	CAL	IDPAL
$E_{min}[\text{fJ}]$	367	7.53	7.37	13.8	7.06
$E_{max}[\text{fJ}]$	391	121.78	22.1	29.7	15.2
NED[%]	6.18	93.82	66.7	53.4	53.5
$E[\text{fJ}]$	382	58.51	14.8	21.5	11.2
$\sigma_E[\text{fJ}]$	10.4	51.71	8.32	8.84	4.55
NSD[%]	2.71	90.08	56.1	41.1	40.7

State-of-the-art secure logic families like SABL and WDDL were shown to be either effective in preventing power analysis attacks but consume a significant amount of energy (in the case of SABL) or significantly ineffective but consume a lower amount of energy (in the case of WDDL). Neither family fit the requirement of both low power and security mainly because both families are derived from traditional CMOS circuits, which do not provide a secure solution against this kind of attacks as we observed in the case of the CMOS inverter. Therefore, further analysis on secure low-power solutions focuses on adiabatic circuits only, which were shown to be low power and some provide resistance to power analysis attacks.

An additional analysis of the 2-input NAND gate can be done using the Hamming

Weight/Distance power models for the NAND gates above. For this experiment, a two-input IDPAL NAND/AND gate has been cycled through all 16 possible input combinations shown in Table 7 to account for both the Hamming Weight and Hamming Distance power models. Simulations were performed with the same parameters as before, but using a 10MHz *PC* to observe the decrease in current magnitudes at a lower frequency. Additionally, energy values for each transition were calculated to obtain a clearer picture of the energy imbalances.

Table 7: All 16 possible input transitions for a 2-input NAND/AND gate and energy consumption values for each transition in IDPAL.

Current	Next	Energy (fJ)
00	00	4.11
00	01	4.76
00	10	4.78
00	11	7.33
01	00	4.75
01	01	4.12
01	10	4.71
01	11	4.77
10	00	4.74
10	01	4.80
10	10	4.09
10	11	4.75
11	00	4.78
11	01	4.73
11	10	7.48
11	11	4.13

Figure 34 shows an overlay of the 16 current traces for this experiment. A visual inspection of the traces reveals that two of the 16 traces have peaks that are around $2\times$ larger than other peaks. The input transitions causing these peaks are “00” \rightarrow “11” and “11” \rightarrow “10”, the same two transitions that produced higher peaks in Figure 32. Similarly, more significant peaks were observed in the 2N2P and CAL NAND gates for the same

input transitions.

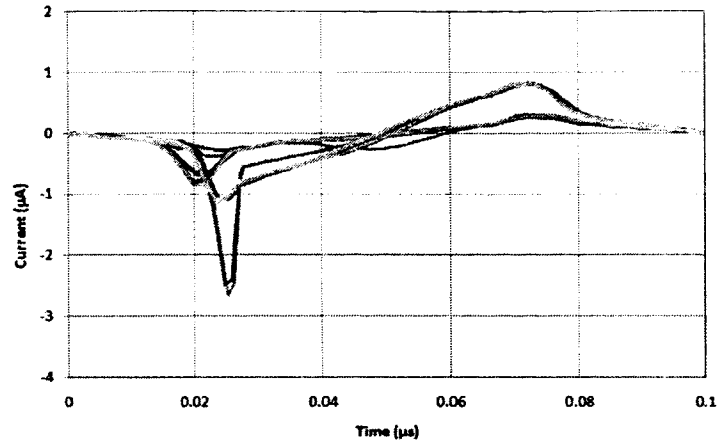


Fig. 34: Current traces for all 16 possible transitions for the IDPAL AND/NAND gate.

Applying the Hamming Weight model and comparing energy consumption of the 1 value that produce a Low value on the output of the NAND gate vs the 3 values that produce a High value, an average energy consumption of 5.80fJ is achieved for the three values, whereas the single Low-producing value uses 7.93fJ. The difference is significant (27%), however it is due to only one of the values producing a Low output. This imbalance can allow an attacker to extract the fact that the inputs to the 2-input NAND gate were “11” if only one gate is under attack, although this situation is improbable and does not produce any useful information about the circuit, a larger circuit would need to be analyzed to observe the ability of this model to predict the input values.

The Hamming Distance model used on the 16 transitions listed above is expected to give a more accurate representation of the energy consumption for each transition since the average will depend on more than one output value. Of the 16 outputs, ten have a Hamming Distance of 0 and six have a Hamming Distance of 1. The average energy consumption for the *distance* = 0 outputs is 4.49 fJ, whereas the average for *distance* = 1

is 5.64 fJ, a difference of only 1.15fJ. The larger energy consumption for the $distance = 1$ transitions is due to the two peak values being part of this average, otherwise the two average values would have been almost equal. The smaller difference in the case of the Hamming Distance model shows that IDPAL is able to successfully reduce the amount of information that can be extracted by analyzing the energy consumption of a circuit.

IV.4 Results for Large Circuits

Large circuits have been simulated and analyzed for resistance against power analysis attacks. Attacks on larger adiabatic circuits have to take into account the leakage from other PC 's as the circuits will contain more than one evaluation stages. If the number of evaluation stages is odd, one of the PC current traces can be larger than the others. Other factors, such as the use of unequal number of gates and different types of gates for different evaluation stages, will affect the magnitude of the current trace for a given PC .

To illustrate the issue of having a different number of evaluation stages for each PC , two simple simulations were performed with a chain of 4 and 5 IDPAL buffer/inverters. Since all gates are of the same type, the difference between the current traces between the first and second experiment should not be very large, but it should be visible. Current traces for each experiment are shown in Fig. 35. Results show that i_{PC} in the second experiment is approximately 200nA larger than the same trace in the first experiment since it drives 3 evaluation stages vs. 2 in the first experiment. It can also be observed that $i_{\overline{PC}}$ stays constant since it does not drive any additional gates in the second experiment. It would be possible to almost equalize two current traces in a larger circuit by adding

buffers to the circuit, but doing so would affect the energy efficiency as well as circuit area. Current trace inequality is an issue in any adiabatic logic family using more than a single phase, but no significant information can be extracted from this fact except that one phase of the *PC* drives more transistors than the other(s).

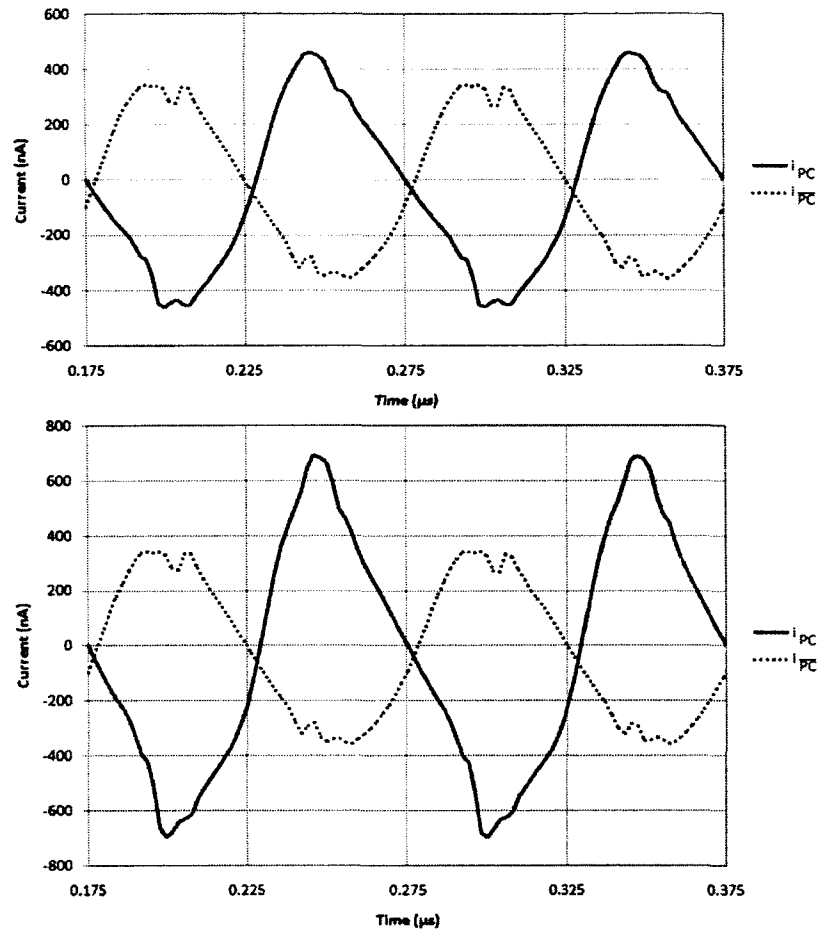


Fig. 35: Current traces for even (4) vs. odd number (5) of evaluation stages for IDPAL inverters.

A larger experiment consisted of simulating 8-, 16-, and 32-bit KSAs using the same parameters with *PC*'s running at 10MHz. As the state space even for the 8-bit KSA is very large (65,536 total possible transitions) and increases exponentially with the adder width, results from only a few representative and a few random inputs are shown in this section.

The values for the two inputs and the carry-in signals used in the 32-bit KSA are given in Table 8.

Table 8: Input test cases for the 32-bit KSA and Wallace multiplier.

1	<i>A</i>	00000000000000000000000000000000
	<i>B</i>	00000000000000000000000000000000
	<i>C_{in}</i>	0
2	<i>A</i>	00000000000000000000000000000000
	<i>B</i>	00000000000000000000000000000000
	<i>C_{in}</i>	1
3	<i>A</i>	11111111111111111111111111111111
	<i>B</i>	11111111111111111111111111111111
	<i>C_{in}</i>	0
4	<i>A</i>	11111111111111111111111111111111
	<i>B</i>	11111111111111111111111111111111
	<i>C_{in}</i>	1
5	<i>A</i>	00000000000000000000000000000000
	<i>B</i>	11111111111111111111111111111111
	<i>C_{in}</i>	0
6	<i>A</i>	11111111111111111111111111111111
	<i>B</i>	00000000000000000000000000000000
	<i>C_{in}</i>	1
7	<i>A</i>	01010101010101010101010101010101
	<i>B</i>	10101010101010101010101010101010
	<i>C_{in}</i>	0
8	<i>A</i>	11110000111100001111111100000000
	<i>B</i>	01101001011010101111111111111111
	<i>C_{in}</i>	1

Average current along with standard deviations at each data point between the 8 input cases from the 32-bit KSA for 2N2P, CAL, and IDPAL are shown in Fig. 36. Large current differences between different inputs would be an indication of lower resistance to power analysis attacks and leakage of more information. Standard deviation values among the 8 current waveforms from one *PC* of each family are shown in Table 9. Average values for the standard deviation are $3.24\mu\text{A}$ for IDPAL, $3.07\mu\text{A}$ for 2N2P, and $10.45\mu\text{A}$ for CAL. The average standard deviation is slightly lower for 2N2P compared to IDPAL most likely

because of the higher number of clock phases used in 2N2P and more homogeneous gates on each phase of the *PC*. The same observation can be made between IDPAL and CAL, although the difference in the averages is significantly higher in this case. The range of the standard deviation in the current traces for IDPAL is also lower than the other two families and the reason is thought to be a more gradual and longer charging period due to using a sinusoidal *PC* in IDPAL.

Table 9: Minimum, maximum, and average standard deviation values for current waveforms for a 32-bit KSA in the adiabatic families tested.

Family	Standard Deviation (A)		
	Min	Max	Average
IDPAL	3.8E-10	1.02E-5	3.24E-6
2N2P	7.4E-8	1.4E-5	3.07E-6
CAL	1.7E-8	1.00E-4	1.05E-5

The largest differences among the 8 input cases were observed in test case pairs (1, 4) and (5, 6) for all three families, however, these differences were relatively small. These imbalances are most likely due to the number of bits switching values, leading to the conclusion that adiabatic logic families offer some resistance, but an attacker with extremely sophisticated tools might be able to extract some information from a running circuit. Additionally, the 32-bit KSA contains 7 evaluation stages, which results in differences between the current traces of each *PC* in IDPAL and 2N2P. For IDPAL, the first *PC* drives 4 evaluation stages with a total of 2418 transistors, whereas the second *PC* drives the remaining 3 stages with 1932 transistors. This imbalance results in a difference of the current peaks between the two *PC*'s of approximately $300\mu\text{A}$, which is proportional to the difference in the number of transistors. The only information that can be extracted from this imbalance

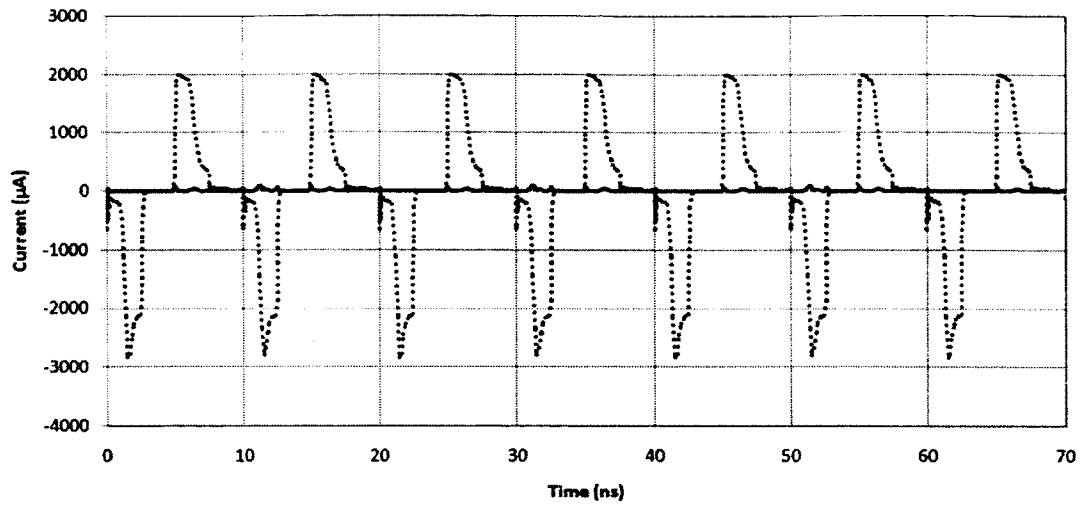
without any knowledge about the circuit implementation is that one phase of the circuit powers more transistors than the other. Additional security metrics information would be needed in order to stage an attack on a given circuit.

The figures of merit NSD and NED were also calculated and the results are shown in Table 10. Simulation results show that the average energy for CAL is highest whereas it is lowest for IDPAL, results which were expected from Chapter III. IDPAL also has the lowest NED and NSD values (5.83% and 2.03%, respectively). These results are a direct consequence of having a smaller range for the minimum and maximum energy values and energy standard deviation. This implies that IDPAL has the highest energy efficiency and resistance against power analysis attacks among the tested adiabatic logic families.

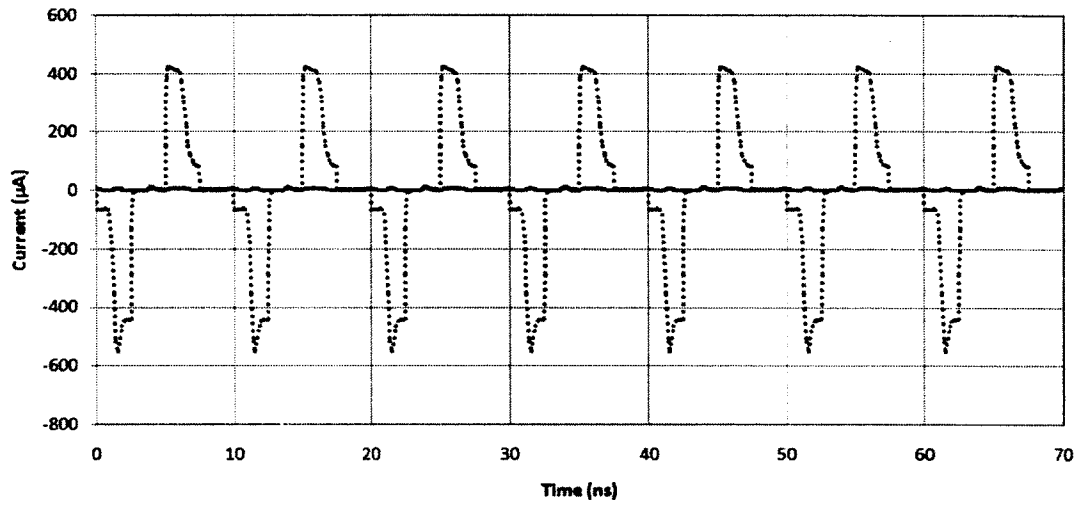
Table 10: Security metrics for the 32-bit KSA for each adiabatic logic family.

	CAL	2N2P	IDPAL
E_{min} [pJ]	1.52	1.18	0.69
E_{max} [pJ]	1.66	1.27	0.73
NED[%]	8.65	7.01	5.83
E [pJ]	1.60	1.23	0.70
σ_E [pJ]	0.05	0.03	0.01
NSD[%]	3.30	2.88	2.03

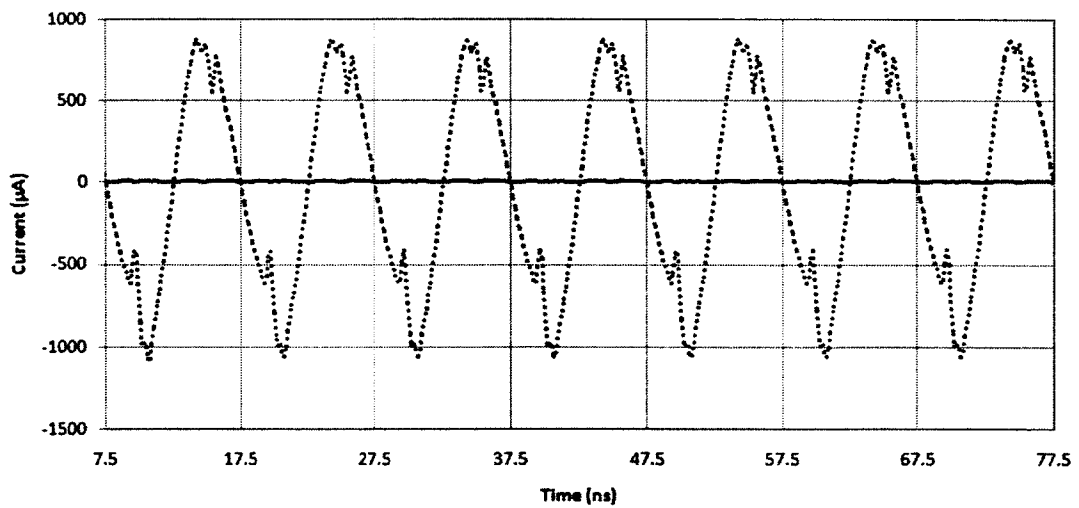
A 32-bit Wallace multiplier was also simulated and security metrics are shown in Table 11. The NSD and NED values for IDPAL were significantly lower than for 2N2P and CAL most likely due to the more even distribution of the types of gates in the 32-bit multiplier between the two *PC* phases. Since the multiplier contains 16 evaluation stages, each *PC* phase for a logic family controls an equal number of evaluation stages. This aids in helping equalize energy dissipation for each phase, although variations in the type



(a) CAL



(b) 2N2P



(c) IDPAL

Fig. 36: Average and standard deviation current traces for 8 test cases of a 32-bit KSA in CAL, 2N2P, and IDPAL for one of the *PC*s.

of gates in each evaluation stage may cause an unequal amount of energy dissipation. The 2N2P multiplier saw an increased variation in the energy consumption for each *PC* phase because all phases controlled more than one evaluation stages containing different gates. Compared to the 32-bit KSA, the NSD and NED for IDPAL for the 32-bit Wallace multiplier was only slightly higher, most likely due to the different types of gates drive by each phase of the *PC*, however IDPAL showed a lower variation in the amount of energy usage compared to other adiabatic implementations.

Table 11: Security metrics for the 32-bit Wallace multiplier for each adiabatic logic family.

	CAL	2N2P	IDPAL
E_{min} [pJ]	16.34	12.35	7.39
E_{max} [pJ]	18.06	13.52	7.85
NED[%]	9.52	8.65	5.86
\bar{E} [pJ]	17.31	12.92	7.61
σ_E [pJ]	0.69	0.43	0.18
NSD[%]	4.04	3.36	2.30

IV.5 Chapter Conclusions

This chapter explored several power analysis attack metrics applied to adiabatic and non-adiabatic circuits to measure hardware security of these circuits. It was experimentally shown that CMOS logic does not fit the requirements for a low-power and secure circuit design due to its very large differences in energy usage, demonstrating the need for more secure circuits styles. SABL and WDDL were explored in small-circuit simulations, but shown to have large energy consumption, making them unsuitable for use in a low power environment. SABL has very good security metrics (NSD=2.71%), but

very large energy consumption, whereas WDDL has better energy characteristics but high variability in energy consumption (NSD=95.2%). Adiabatic logic families showed lower energy consumption characteristics and good resistance to power analysis attacks. IDPAL has the lowest variability in energy consumption (NSD=2.03% and NED=5.83%) for a large adder, making it a suitable candidate for low power secure circuit applications.

CHAPTER V

DESIGN OF PROOF OF CONCEPT MICROPROCESSOR

This chapter presents the design of a proof of concept hybrid adiabatic-CMOS microprocessor. The bus width will be kept generic to the extent possible. Aspects that can be customized will be highlighted and implementation sizes will be estimated for different implementation configurations. All internal logic building blocks have an adiabatic implementation using IDPAL, while the controller, register file, and memory are implemented using CMOS logic. The design of the microprocessor is not pipelined, but it can use the micro-pipelining abilities of adiabatic logic.

V.1 Implementation Constraints and Customizable Features

The current implementation of the hybrid microprocessor is limited to the available die space – a 5mm^2 die in $0.5\mu\text{m}$ 3-metal 2-poly process. The goal of this proof-of-concept microprocessor is to take advantage of the best features from adiabatic and CMOS logic styles. Computation units with high switching activity (e.g. adder, shifter) are implemented in IDPAL, whereas units with low switching activity (e.g. controller, memory) are implemented using CMOS logic. This hybrid design presents several implementation constraints, but also offers many customization options.

The microprocessor is a reduced version of an OpenRISC microprocessor with a basic set of instructions in a load/store architecture. The design was based on the need for a general purpose processor capable of performing integer and logical operations as well

as implementing encryption algorithms using basic instructions. Further, this particular design was selected because the Reduced Instruction Set Computing (RISC) paradigm results in simpler data paths and control logic. The proposed microprocessor has many features typical of an implementation of a smartcard microprocessor, adapted for the constraints imposed by die size.

As noted, the microprocessor configuration will depend on the resources that are available for implementation. In particular, the silicon area will be used to assess the actual configuration implemented. Among the customizable features are the bus width, number of registers, and the complexity of the instruction set. This reduced implementation also does not include a multiplication and division unit or a floating point unit. Multiplication operations can be implemented as a series additions at the expense of a larger number of clock cycles. Division operations can be implemented using shift, subtract, and comparison operations, all of which are available in this implementation. The number of registers is at least 8, although a 16-register register file (RF) would be more optimal. The register can be implemented using a 3-ported SRAM cell, with 2 read and 1 write ports.

V.2 Hybrid Microprocessor Design Issues

A hybrid microprocessor design allows for great flexibility in the implementation of each block but also presents issues of compatibility and interfacing between each logic block. Since IDPAL and CMOS logic implementations are very different from each other and use different types of clocking circuitry, it presents several issues related to the implementation of the proposed hybrid microprocessor.

One of the main issues stemming from using adiabatic logic is the dual-rail nature of adiabatic circuits, where an output and its complement have to be made available for computation. Since CMOS circuits are single-rail, the CMOS output will need to be inverted to produce the additional complementary output required for interfacing to an adiabatic cell. Being able to invert an input instead of needing to use double the circuitry to calculate a complement makes a hybrid design very appealing and was one of the factors used in the decision to implement a hybrid microprocessor. Gate delay from the inverted output is not an issue in this case as IDPAL circuits start evaluating the outputs only when $V_{PC} > V_{TN}$ and it is highly unlikely that an inverter would present a considerable delay that would affect operation.

Clock management and synchronization can be an issue in large circuits in a silicon implementation since IDPAL uses a two-phase PC . This issue may not be too difficult to overcome, as it would be possible to interleave the two clock phases in layout, either on the same metal layer or different layers. Additionally, a two-phase sinusoidal PC design might be helpful since the load is divided among the two phases instead of relying on a single and deep square-wave clock distribution network, where clock skew issues are a big concern. All signals being latched onto an adiabatic bus have to have the same phase since all buses have to be synchronized for a signal to be transmitted correctly. If a component contains an odd number of evaluation stages, a buffer stage can be inserted in order to have its output driven by the required phase. Depending on the size of the circuit, the additional buffering layer can cause a significant increase in the number of transistors in the adiabatic implementation if a large number of units need to have their

outputs buffered. Additionally, logic can be inserted to determine when a components has finished evaluating its output in order for the circuit to continue executing other tasks.

One of the other issues in a hybrid adiabatic-CMOS processor is the interfacing between the adiabatic and CMOS circuits. Special interfacing circuitry is required because the circuit can produce invalid values if the output of an adiabatic gate is connected directly to the input of a CMOS gate or flip-flop. Since adiabatic circuits are driven continuously by the PC , the output of an adiabatic gate will fall to $|V_{TP}|$ level when the PC is at GND . This event will trigger a CMOS flip-flop to switch values, causing its output to approximate the oscillating nature of adiabatic circuits by charging and discharging every PC cycle. This solution is clearly inadequate and special circuitry needs to be constructed to allow the flip-flop to operate similar to a CMOS flip-flop. Two CMOS inverters in series have been reported to work as converters from a trapezoidal adiabatic input signal to a square-wave output signal in [28]. However, due to the longer rise time of an adiabatic input, there is a large short-circuit current in the first inverter, causing a large energy consumption.

A solution to the interfacing problem used in this work takes advantage of the dual-phase operation of IDPAL to produce a square-wave output. The flip-flop circuit does not use any additional control signals and its implementation is composed of two modified CMOS inverters connected in series. A diagram of the circuit is shown in Fig. 37. Assuming that the input in is controlled by PC , the interface circuit then has to be controlled by the complementary phase, \overline{PC} . When in is at GND and the \overline{PC} is at GND and starts rising towards V_{dd} , the intermediary output T takes on a High values since both PMOS

transistors are ON. Once \overline{PC} rises past the $|V_{TP}|$ threshold, the output may no longer be changed since the PMOS control transistor is OFF. When in is at V_{dd} , the NMOS network starts conducting once $V_{\overline{PC}} > V_{TN}$, pulling the intermediary output T to GND . The PMOS network is disconnected since in is at V_{dd} and does not have any influence on T . The intermediary output T is the converted and inverted version of the input in and approximates a CMOS signal. However, due to the oscillating adiabatic input and \overline{PC} , the intermediary output contains small ripples, as illustrated in Fig. 38a. In order to obtain an equivalent square-wave version of the original oscillating input signal without any ripples, the T signal is then passed through a pure CMOS inverter, as shown in Fig. 38b. The input in in the Figure has a 75% duty cycle to show the behavior of the adiabatic-CMOS interface under both input cases. The use of the two control transistors controlled by \overline{PC} removes the large short circuit current by blocking the formation of a path from the PMOS to NMOS transistors during switching and results in a better design for the adiabatic CMOS interface.

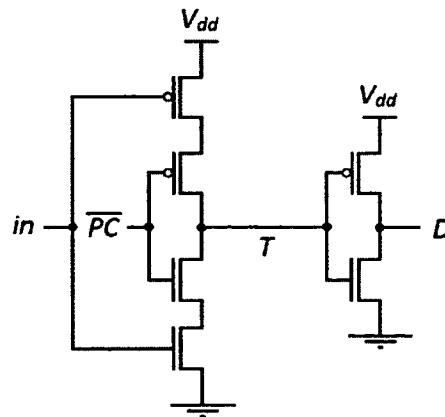
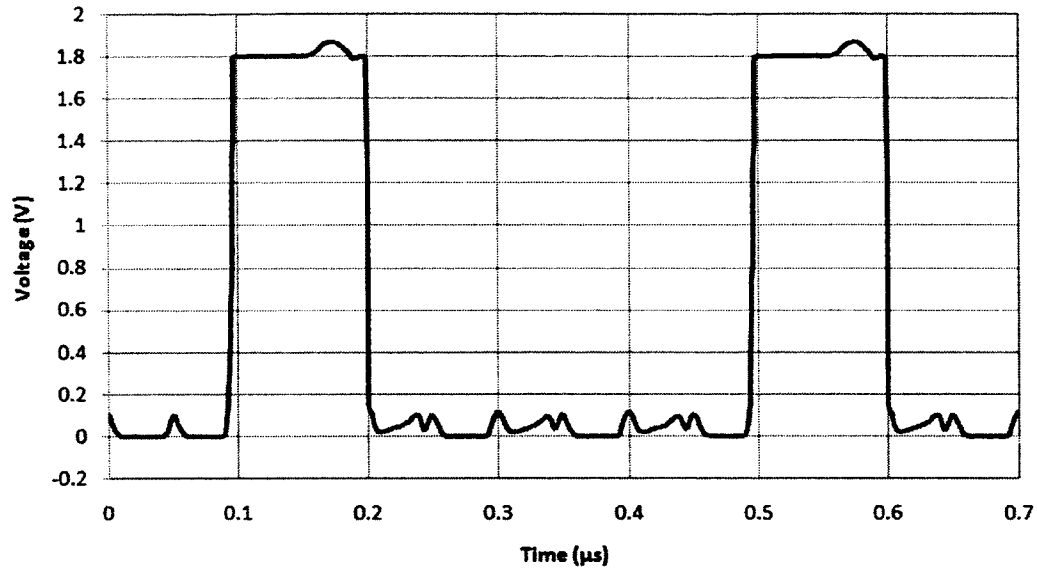
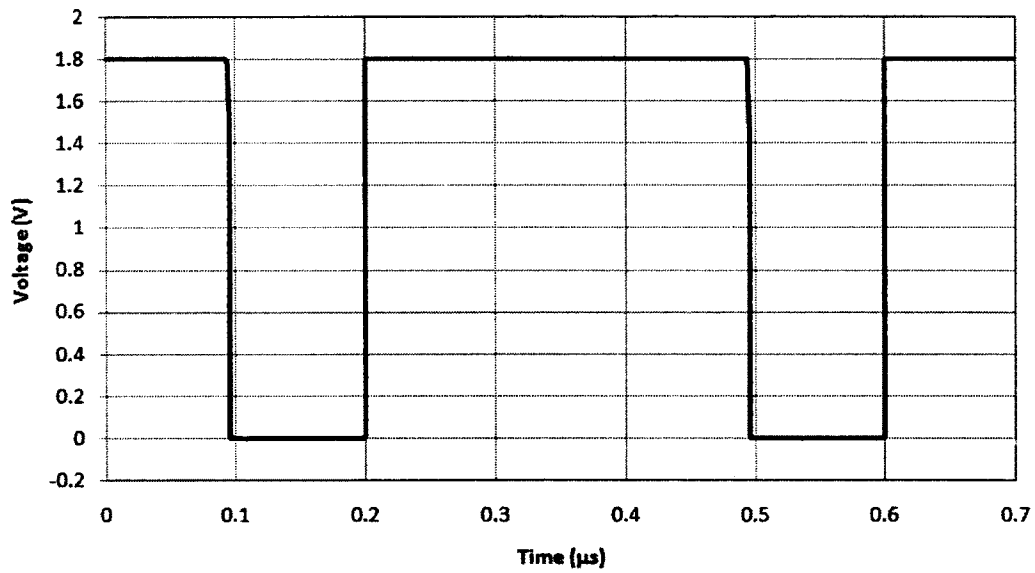


Fig. 37: An adiabatic-CMOS interface circuit used in the proposed microprocessor.



(a) Output T containing ripples.



(b) Output D without ripples.

Fig. 38: Output signal with and without ripples after conversion from an adiabatic input.

This interface has the advantage of not needing any additional control circuitry other than the signals already available. It also uses a small number of transistors in its implementation compared with other implementations [28], which increases its energy efficiency. The interface circuit was tested and shown to be operational on both trapezoidal and sinusoidal PC 's, however its performance is better on a sinusoidal PC , where the square-wave output is valid slightly earlier than on the trapezoidal PC . This difference is mainly due to the earlier evaluation of sinusoidal PC 's compared to the trapezoidal ones.

A disadvantage of all adiabatic-CMOS interfaces encountered in literature, including the one described above, is that due to the time-varying nature of the adiabatic input and the PC 's, the square-wave output is only available approximately 1/4 of the way into a PC cycle. The reason for the delay is that the adiabatic output only starts evaluating once $V_{PC} \geq |V_{TN}|$.

Because some of the operations in the microprocessor may not have a set number of clock cycles required for execution (i.e. they are asynchronous), a fully adiabatic implementation might not be the best solution due to its micro-pipelining. To accommodate this issue, a square-wave implementation using a flip-flop is used in this microprocessor for the memory read/write operations. The proposed approach uses the adiabatic-CMOS interface circuit in order to generate a square-wave output signal to be written to memory. When a memory operation is complete, a signal will be triggered to indicate the end of the operation, allowing the next unit to continue evaluation using the new input.

V.3 Implementation Details

The microprocessor block diagram is shown in Fig. 39. Dotted outlines show adiabatic implementations whereas solid outlines show CMOS implementations. There are a few instances of the adiabatic-CMOS interface being used, denoted with an interrupted outline. As can be observed, the ALU and all temporary registers are fully implemented using IDPAL, whereas CMOS implementations include the register file, control circuitry, and memory. A square-wave clock can be generated from either of the two adiabatic phases by passing it through an instance of the adiabatic-CMOS interface presented earlier. The control circuitry is implemented as a hard-wired ROM due to the simplicity of implementation. Although not shown in the Figure, the two source buses ($S1$ and $S2$) are driven by PC , whereas the destination bus (D) is driven by \overline{PC} .

Due to the limited size of the implementation, the OpenRISC instruction set was reduced to only the most basic instructions that would allow implementation to be characterized as OpenRISC also being able to implement an encryption algorithm. A description of the building blocks of the proposed microprocessor follows.

The Program Counter (PC, not to be confused with PC – the Power Clock) uses an adiabatic buffer for storage of the next instruction address. A buffer is inserted between the D bus and the PC in order to have the correct phase drive the PC on the $S1$ and $S2$ buses. The ALU has pass-through capability in order to allow the value from PC (or any other value from the $S1$ and $S2$ buses) to be passed onto the D bus.

The memory is CMOS-based and is hosted off-chip due to limited die space. An additional off-chip square-wave clock is used for control along with any other memory

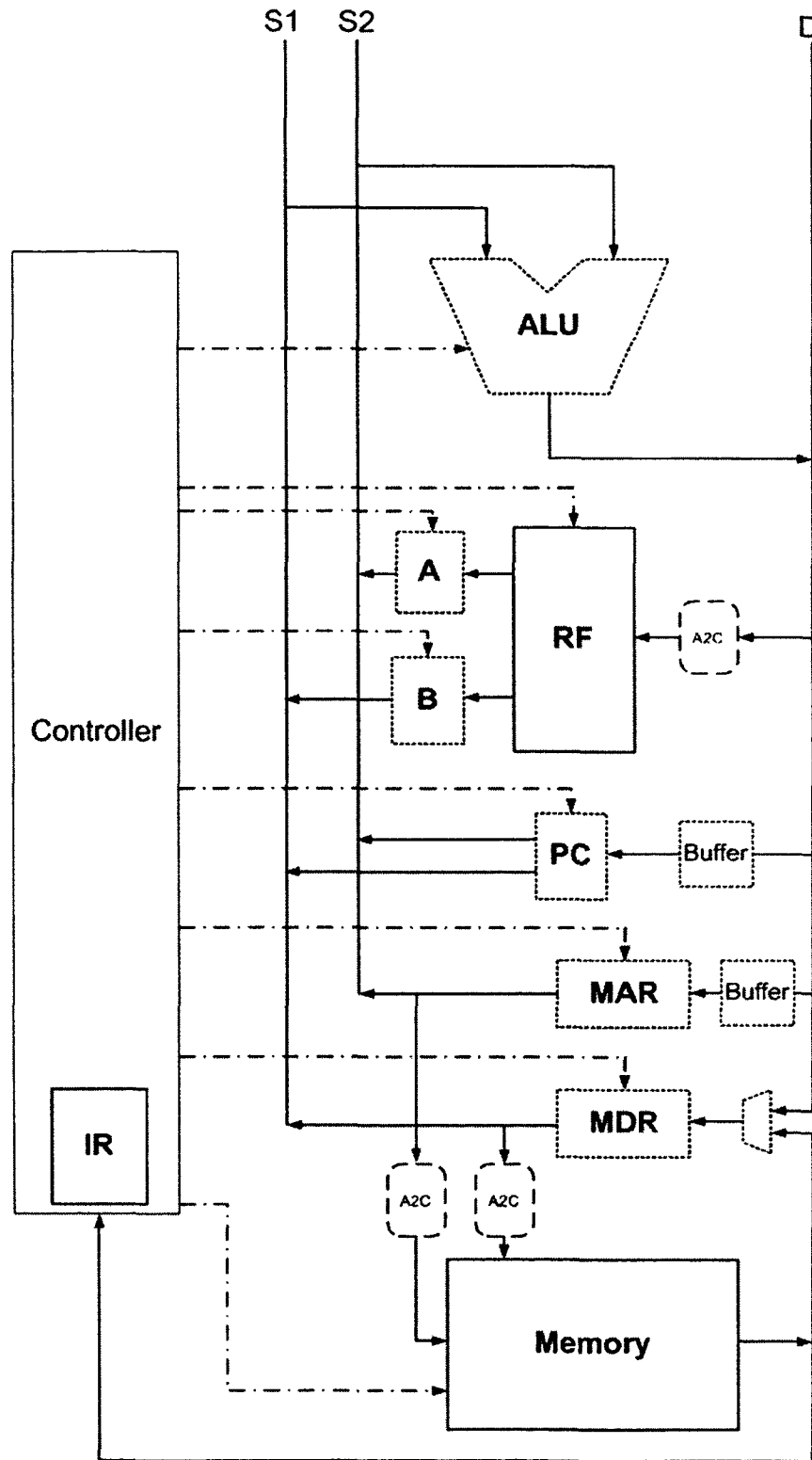


Fig. 39: Block diagram of the hybrid adiabatic-CMOS microprocessor.

control signals coming from the chip. If the memory were to be placed on chip, an adiabatic implementation similar to the one in [30] could be used. The memory has to be byte-addressable in order to be able to execute instructions that operate on single bytes. As the proposed microprocessor does not contain advanced features, the memory contains both instruction and program data. Additionally, no cache or memory management units are implemented in order to maintain the simplicity of this proof-of-concept implementation.

Although the microprocessor is not pipelined, we can take advantage of the micro-pipelining abilities of adiabatic logic if multiple instructions of the same type are executed consecutively. For example, in an encryption operation, many shift, XOR, and multiply operations take place. In a micro-pipelined environment, the results of these operations could be available one per *PC* cycle once the first result is available, assuming that the operations are independent of each other. Encryption operations and some digital signal processing operations could make use of this micro-pipelining feature of adiabatic circuits to obtain results faster than performing individual instructions sequentially. If the operations are not independent, issues that affect regular pipeline operations have to be considered before implementation (e.g. data hazards).

V.3.1 Implementation of Main Instructions

For demonstration purposes, the following discussion is based on an 8-bit implementation of the hybrid microprocessor. The discussion can be extended to a 16- or 32-bit implementation without many changes. All instructions follow the same execution stages

as a in a traditional CMOS non-pipelined implementation, although the ALU has micro-pipelining abilities due to using IDPAL for implementation.

Addition and subtraction operations are implemented using an 8-bit KSA, similar to the one discussed in Chapter III. A block diagram for the ALU is shown in Fig. 40. Unlike traditional adder/subtractors, where a XOR gate is used in conjunction with the C_{in} signal to generate the inverted version of one of the inputs, adiabatic logic signals have both the input and its complement available and only need to have different routing paths. A swap gate that has the functionality of a 2-bit multiplexer, implemented as a 2-input XOR gate using 10 transistors (Fig. 41), is inserted before the KSA in order to account for the subtraction operation. This gate is controlled by a signal which allows the inputs to be swapped when it is asserted. The second input is presented in its regular form and needs to be buffered for an additional stage due to the swap gate. The overhead associated with the subtraction operation then becomes 16 transistors for each input bit pair for a total of 128 transistors, or 14% of the total number of transistors in an adder/subtractor. The 8-bit KSA had 5 evaluation stages before the addition of the layer of swap gates, which requires a stage of buffers at the output of the adder in order to synchronize its outputs with the bus. Thus, the addition of the swap gates stage adds the extra evaluation stage that synchronizes the outputs as well as provides the subtractor functionality, allowing for an even lower overhead compared with a pure adder.

An improvement over this design would be to have logic that reads the next instruction in the micro-pipeline and is able to detect whether it is a subtraction operation or not. If the next instruction is a subtraction, then the results of the current instruction, regardless

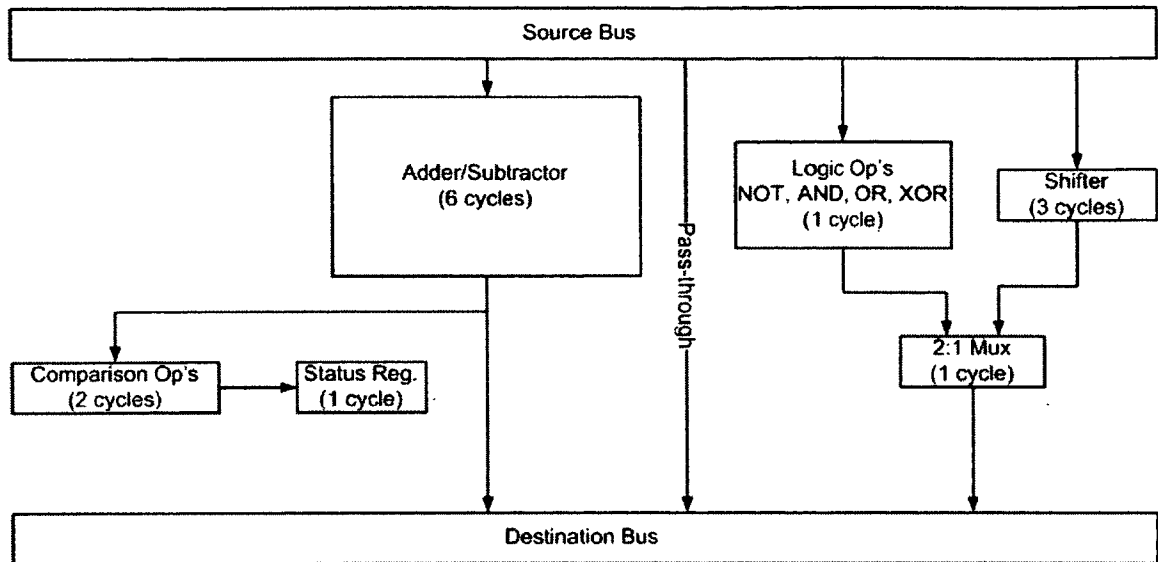


Fig. 40: High-level ALU block diagram implementation.

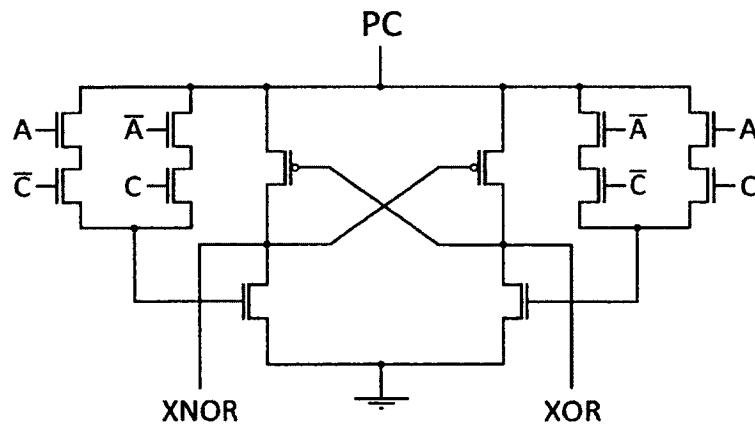


Fig. 41: Implementation of a 2-input XOR gate in IDPAL used as a swap gate.

of what it is, can be routed to opposite registers in the register banks using two 2-to-1 multiplexers, thus swapping the input values with their complements and eliminating the need for a new layer of logic in the KSA. Due to the additional transistors used in multiplexers, a more detailed analysis of the particular use of this microprocessor would need to be performed in order to choose the most optimal implementation. Additionally, this design would not be feasible on the current version of the microprocessor as the KSA needs to produce a result in an even number of phases.

Shift instructions (*SRL*, *SRA*, *SLL*, *ROR*) are implemented using a bi-directional barrel shifter implemented with modified 8-to-1 and 4-to-1 multiplexers. A diagram of the shifter design is shown in Fig. 42. The 4-to-1 multiplexers form the majority of the multiplexers used and only use 3 inputs, whereas the 8-to-1 multiplexers only use 5 inputs. The rest of the available inputs to each kind of multiplexer are “don’t cares” as their values will not influence the output of each multiplexer and are not shown in the diagrams for simplicity. The few 8-to-1 multiplexers used are needed in order to shift constant values for some of the instructions that do not have a more regular structure. The control signals S_{x0} and S_{x1} , where $x = 0, 1, 2$, are the same for both types of multiplexer and the signal S_{x2} is only used for the 8-to-1 multiplexers for the non-regular cases. An additional layer of buffers is required at the shifter output in order to use an even number of clock phases for correct operation. The first input to each type of multiplexer allows the previous input to pass through the multiplexer unchanged. The second input produces the *ROR* operation, as well as *SRL* and *SRA* on the 4-to-1 multiplexers. These two instructions are different on the 8-to-1 multiplexer as a 0 needs to be shifted in for the *SRL* operation (input 5) and

the value of the first bit needs to be shifted in for the *SRA* operation (input 4). The third input on both multiplexers is for the *SLL* operation, which has hard-wired connections to *GND* for the lowest bit positions and the multiplexers do not require any additional modifications.

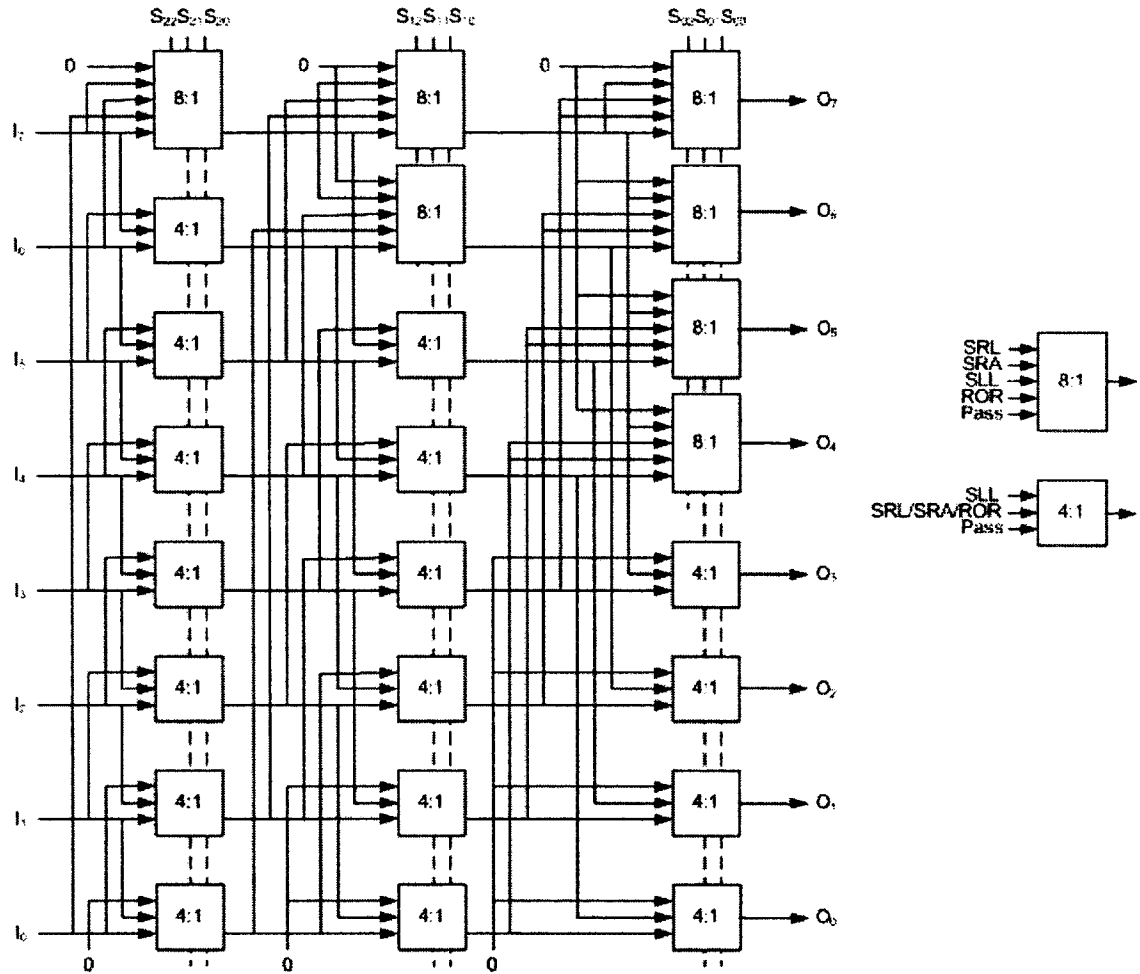


Fig. 42: An 8-bit bi-directional barrel shifter implementing the *SRL*, *SRA*, *SLL*, *ROR* instructions.

Comparison operations use a special 5-bit status register to store the result of a comparison operation. Several hardware comparator options were available, although due to

space constraints it was decided to use the 8-bit KSA implemented earlier with a few modifications to account for comparison operations. Comparison can be implemented using a KSA performing a subtraction operation. Assuming the subtraction operation is $A - B$, to check if $A > B$ or $A < B$, we check the carry-out (C_{out}) signal of the KSA. If $C_{out} = 1$, then $A > B$, otherwise $A < B$. To check for equality ($A = B$), we check the C_{out} and also pass the output through an OR gate and check the result. If $C_{out} = 1$ and the OR gate result is a 0, then $A = B$. Since the $A = B$ operation involves two steps, it adds two layers of logic to the computation. The other two comparison operations ($A \geq B$ and $A \leq B$) can be implemented from the two basic operations $A < B$ and $A > B$, respectively, by inverting their values.

Load and store instructions use an instance of the adiabatic-CMOS interface described earlier and a flip-flop in order to have a strong CMOS signal to write to and read from the memory. Address and data to be written are stored in *MAR* and *MDR*, respectively. An additional buffer was inserted between the *D* bus and the *MAR* for synchronization with the operation of *MDR* due to the 2-to-1 multiplexer. Memory is off-chip and is clocked using a separate clock running at the same frequency as the *PC*. Data is loaded from the *MDR* into the flip-flop by using a memory control signal supplied by the controller, which is then written to memory in the next clock cycle. The micro-pipeline can be padded with enough buffers in order to match the memory access time and avoid using more complex control circuitry, e.g. an interrupt. A similar scenario takes place during a load instruction and an interrupt is triggered when the load operation is complete. The *PC* and memory clocks do not have to be synchronized as the flip-flop load signal and the memory write

signal will be spaced by 1 clock cycle, allowing the address and data to be written to flip-flops and then available for the memory in the next clock cycle. Since the micro-pipeline depth will match the required access times for a read operation, available data can be passed to the multiplexer to be loaded into the *MDR* during the next \overline{PC} cycle

Basic branching instructions also are implemented in this microprocessor. Unconditional branches are implemented by simply latching the address of the next instruction in the PC register. Conditional branching instructions using the results of conditional set-flag instructions can be readily implemented using the flags in the special 5-bit status register and multiplexers. Other branching instructions (e.g. jump-and-link) will contain special control circuitry in the controller to achieve their particular implementations.

V.4 Microprocessor Testing

The proposed microprocessor has been tested in VHDL for behavioral and dataflow functionality. It is currently in layout stage using the Cadence layout suite in preparation for fabrication in silicon. Rigorous testing on the energy efficiency and hardware security of the microprocessor will begin once fabricated parts are available.

Several steps need to be taken before testing of the fabricated parts is possible. First, a special test fixture that will be able to supply the clocking information for the *PC* and \overline{PC} to the microprocessor needs to be developed. The test fixture will contain circuitry necessary for the generation the two *PC* phases (either a RLC circuit or a phase-locked loop circuit) tuned to the specifications of the microprocessor extracted from layout. It is expected that the microprocessor will be tested in the 10-100 MHz frequency range,

so an appropriate oscilloscope will need to be used to properly sample signals from the microprocessor. Second, an external memory capable of running at the specified speeds will need to be acquired and the appropriate memory control signals, including a new square-wave clock, will need to be generated.

The fabricated microprocessor will be evaluated using programs ranging from a simple logical operation to a full state-of-the-art encryption algorithm (e.g. DES, AES). Energy efficiency will be tested from measurements only on the two clock phases (PC and \overline{PC}) as the square-wave clock is not part of the adiabatic implementation. An accurate 1Ω resistor will be placed in series with the PC line in order to measure the current supplied and recovered in the circuit.

V.5 Chapter Conclusion

In this chapter, a new hybrid adiabatic-CMOS microprocessor partially implemented using IDPAL was proposed. A general description of the microprocessor using a block diagram and a more detailed explanation of each block was given. The chapter also described an adiabatic-CMOS interface circuit capable of producing square-wave outputs from oscillating inputs using a small number of transistors without any external signals. Due to the limitation on the size of the die for fabrication, a memory was not implemented in the proposed microprocessor, but an interface for using external memory was described.

A short summary of the implementation of the main instructions was presented, giving a more detailed view of the arithmetic unit. It was shown that for correct operation of the ALU, each internal component has to produce an output in an even number of clock

intervals, which is due to the source ($S1$ and $S2$) and destination (D) buses being operated using opposite phases. In many cases this restriction is not a limiting factor in the performance of the microprocessor as many of the instructions already use an even number of clock intervals. A large number of more complex instructions can be implemented using the basic instructions described in this chapter, although their implementation will take more clock cycles than having specialized hardware for specific instructions. Overall, the proposed microprocessor makes use of the advantages of adiabatic logic for circuits with high switching activity and uses CMOS logic for the other circuits.

CHAPTER VI

FUTURE WORK

This dissertation presented promising results which can be implemented successfully in the design of larger low-power adiabatic circuits. IDPAL has been tested in large circuit simulations as described in previous chapters and its energy efficiency scaled well with increased circuit size. This work could be further expanded to the implementation of a 32-bit RISC processor using IDPAL. Previous adiabatic processors were fabricated at 8- and 16-bit length [33]–[36], which do not offer a full range of options for testing the processor against standard benchmarks. A 32-bit processor would allow for an equal comparison with equivalent commercial processors. The proof of concept microprocessor can be implemented without any pipelining in order to confirm energy efficiency in silicon, with subsequent pipelined versions fabricated in the future. The processor could implement encryption operations using a provided instruction set and/or a dedicated AES cryptographic co-processor could be implemented in hardware. Implementing encryption using two different ways allows for a more in-depth comparison of energy efficiency and power analysis attack resistance both in a RISC and an ASIC. A fabrication run would also yield more than one processor, so an extension to this work can be an analysis of multiple dies to be able to gauge inter-die variations and their effects on energy efficiency and power analysis attack resistance.

The arithmetic units in this work (adder and multiplier) were implemented and translated from their CMOS implementation manually. An extension to this work would be to

create an automatic translation tool for converting single-rail CMOS designs (from VHDL code or netlists) to any type of adiabatic logic family given the details of a particular family and other design criteria. The tools should be able to optimize a design based on area or energy usage and any other constraints set by the designer.

IDPAL circuit scalability has only been investigated at 180nm and it was found that it scales similar to CMOS and other adiabatic implementations between these two adjacent technology nodes. An extension of this work would be to compare energy efficiency in more modern processes in the deep-sub-micron region (e.g. 22nm, 16nm) and analyze power analysis resistance at these technology nodes. Leakage currents are very strong and leakage power is often a very large contributor to the overall power dissipation at these feature sizes. An analysis at this level would also allow for the implementation of leakage power analysis attacks.

Sequential circuits were not investigated in most adiabatic logic families encountered in literature due to the micro-pipelining abilities of adiabatic logic. The few attempts that were made used only modified buffer designs and did not contain special circuitry for stand-alone latches and flip flops. One of the disadvantages of relying on micro-pipelining is the inability of buffers to hold a value when the buffer input is disconnected by a control transistor and such a case would require special circuitry for a self-sustaining adiabatic flip-flop. New control circuitry would need to be developed in order to interface the flip-flops with the rest of the adiabatic circuit and with square-wave output circuitry for energy-efficient operation.

The ultimate goal of this work is to help design a framework for cost-effective, reliable, and secure identification, monitoring, and tracking system which could be used in hospital environments. Current technology, starting from hospital admission to drug delivery and monitoring to discharge paperwork, is very convoluted and time-consuming and ends up in discomfort to the patient, missed procedures, and reduced efficiency. A vision for the future is to use IDPAL in smart cards, sensors, and implantable medical devices for better and secure patient care and decreased costs. Smart cards could be used for patient identification by storing patient information, health history, and other patient data locally and securely. This approach would allow for much shorter admission times and more immediate care. Sensors could monitor patient movement throughout the hospital, drug admission and patient response, and securely make the data available to attending doctors. Finally, implantable medical devices, such as pacemakers, can transmit real-time data to the doctor, their batteries can last a longer time, and the devices themselves be secure against power analysis attacks. Overall, adoption of next-generation technology, including circuits implemented using IDPAL, can provide for more immediate care for patients, decrease hospital and patient costs, and increase security of both devices and patient information.

CHAPTER VII

CONCLUSIONS

This dissertation presented IDPAL – a new very low power partially-adiabatic logic family. The unique feature of IDPAL is in the decoupling of input networks from the evaluation networks, which allows the input stack to evaluate the inputs to the evaluation network long before the evaluation stage is enabled. This gate design was shown to reduce energy consumption compared to equivalent CMOS circuits by up to 79% and at least by 25% than other state-of-the-art adiabatic logic families in large circuits (32-bit adder and multiplier). IDPAL circuits continue to maintain their energy advantage with circuit scaling. In an experiment where minimum feature size was reduced from 250nm to 180nm, an IDPAL 32-bit adder scaled similar to CMOS circuits and other adiabatic circuits.

IDPAL also has good power analysis attacks resistance characteristics due to its almost-constant energy dissipation for any input combination. In small circuit simulations, all adiabatic logic families tested showed a larger normalized energy deviation than state-of-the-art secure logic families. However, since the adiabatic families studied have a much smaller absolute magnitude deviation, they are a better choice for implementation of power analysis attack resistant circuits. The energy deviation between maximum and minimum values is $17\mu\text{A}$ lower for IDPAL than SABL, which can increase the resistance of IDPAL against power analysis attacks. When compared to other adiabatic logic families, IDPAL increases power analysis attack resistance by at least a 32%. These qualities allow IDPAL to be used as a low power power analysis attack countermeasure.

A hybrid adiabatic-CMOS microprocessor implemented using IDPAL for the adiabatic circuits was also proposed. The microprocessor takes advantage of the reduced energy consumption of IDPAL circuits for circuit blocks with high switching activity (e.g. ALU) and uses CMOS circuits for circuit blocks with lower switching activity (e.g. controller, memory). An adiabatic-CMOS interface circuit that operates without any external signals to convert an adiabatic signal to a square-wave signal was also discussed.

The use of IDPAL in low power circuits could extend the life of battery-powered devices, avoid expensive and dangerous surgeries in the case of implantable medical devices, and provide a cheaper solution for fault-tolerance applications. Given that adiabatic circuits are more suitable for medium to high switching activity applications, some other applications include digital signal processors, sensor networks, and displays.

REFERENCES

- [1] M. Cutitaru, L. A. Belfore, II, "New Single-Phase Adiabatic Logic Family," *Proc. Intl Conf. Computer Design (CDES'12)*, pp. 9-14, 2012.
- [2] M. Cutitaru, L. A. Belfore, II, "Arithmetic Circuits Using New Single-Phase Partially-Adiabatic Logic Family," *Proc. IEEE Intl Midwest Symp. Circuits and Systems*, pp. 13-16, 2013.
- [3] M. Cutitaru, L. A. Belfore, II, "An Energy-Efficient 32-bit Kogge-Stone Adder Using a Dual-Phase Partially-Adiabatic Logic Family," *Proc. Intl Conf. Computer Applications in Industry and Engineering*, pp. 113-116, 2013.
- [4] M. Cutitaru, L. A. Belfore, II, "A Partially-Adiabatic Energy-Efficient Logic Family as a Power Analysis Attack Countermeasure," *Proc. IEEE Asilomar Conf. Signals, Systems, and Computers*, pp. 1125-1129, 2013.
- [5] G. Moore, "Cramming More Components onto Integrated Circuits," *Electronics*, Vol. 38, Issue 8, pp. 114-117, 1965.
- [6] R. Landauer, "Irreversibility and Heat Generation in the Computing Process," *IBM J. Research and Development*, Vol. 5, Issue 3, pp. 183-191, 1961.
- [7] G. Yeap, "Practical Low Power Digital VLSI Design," *Kluwer Academic Publishers*, Dordrecht, The Netherlands, 1998.

- [8] A. Kramer, J. Denker, B. Flower, J. Moroney, "2nd Order Adiabatic Computation with $2n-2p$ and $2n-2n2p$ Logic Circuits," *Proc. Intl Symp. Low Power Design (ISLPED'95)*, pp. 191-196, 1995.
- [9] V. G. Oklobdzija, D. Maksimovic, "Pass-transistor Adiabatic Logic using Single Power-clock Supply," *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 44, Issue 10, pp. 842-846, 1997.
- [10] D. Maksimovic, V. G. Oklobdzija, B. Nikolic, K. W. Current, "Clocked CMOS Adiabatic Logic with Integrated Single-phase Power-clock Supply," *IEEE Trans. VLSI Systems*, Vol. 8, Issue 4, pp. 460-463, 2000.
- [11] S. Kim, M. Papaefthymiou, "True Single-Phase Adiabatic Circuitry," *IEEE Trans. VLSI Systems*, Vol. 9, Issue 1, pp. 52-63, 2001.
- [12] S. G. Younis, T. F. Knight, "Practical Implementation of Charge Recovering Asymptotically Zero Power CMOS," *Proc. IEEE Intl Workshop Low Power Design*, pp. 177-182, 1994.
- [13] Y. Ye, K. Roy, "QSERL: Quasi-static Energy Recovery Logic," *IEEE J. Solid-State Circuits*, Vol. 36, Issue 2, pp. 239-248, 2001.
- [14] M. Allam, M. Elmasry, "Dynamic Current Mode Logic (DyCML): A New Low-Power High-Performance Logic Style," *IEEE J. Solid-State Circuits*, Vol. 36, Issue 3, pp. 550-558, 2001.

- [15] V. Sathe, S. Arekapudi, A. Ishii, C. Ouyang, M. Papaefthymiou, S. Naffziger, "Resonant-Clock Design for a Power-Efficient, High-Volume x86-64 Microprocessor," *IEEE J. Solid-State Circuits*, Vol. 48, Issue 1, pp. 140-149, 2013.
- [16] C.-S. Gong, M.-T. Shiue, C.-T. Hong, K.-W. Yao, "Analysis and Design of an Efficient Irreversible Energy Recovery Logic in 0.18 μ m CMOS," *IEEE Trans. Circuits and Systems I: Regular Papers*, Vol. 55, Issue 9, pp. 2595-2607, 2008.
- [17] N. Reddy, M. Satyam, K. Kishore, "Cascadable Adiabatic Logic Circuits for Low-Power Applications," *IET Circuits, Devices and Systems*, Vol. 2, Issue 6, pp. 518-526, 2008.
- [18] P. Teichmann, "Adiabatic Logic," *Springer Series in Advanced Microelectronics*, Springer Netherlands, 2012.
- [19] K.-M. Keung, V. Manne, A. Tyagi, "A Novel Charge Recycling Design Scheme Based on Adiabatic Charge Pump," *IEEE Trans. VLSI*, Vol. 15, Issue 7, pp. 733-745, 2007.
- [20] V. Sathe, J.-Y. Chueh, M. Papaefthymiou, "Energy-Efficient GHz-Class Charge-Recovery Logic," *IEEE J. Solid-State Circuits*, Vol. 42, Issue 1, pp. 38-47, 2007.
- [21] J. Hu, D. Zhou, L. Wang, "Power-Gating Adiabatic Flip-Flops and Sequential Logic Circuits," *Proc. Intl Conf. Communications, Circuits and Systems*, pp. 1016-1020, 2007.

- [22] J. Hu, J. Dai, W. Zhang, Y. Wu, "Pre-Settable Adiabatic Flip-Flops and Sequential Circuits," *Proc. Intl Conf. Communications, Circuits, and Systems*, pp. 2299-2303, 2006.
- [23] H. Jianping, C. Lizhang, L. Xiao, "A New Type of Low-Power Adiabatic Circuit with Complementary Pass-Transistor Logic," *Proc. Intl Conf. ASIC*, Vol. 2, pp. 1235-1238, 2003.
- [24] A. Vetuli, S. Pascoli, L. Reyneri, "Positive Feedback in Adiabatic Logic," *Electronics Letters*, Vol. 32, Issue 20, pp. 1867-1869, 1996.
- [25] N. Anuar, Y. Takahashi, T. Sekine, "Two Phase Clocked Adiabatic Static CMOS Logic," *Proc. Intl Symp. System-on-Chip*, pp. 83-86, 2009.
- [26] M. Chanda, A. Dandapat, H. Rahaman, "Low-Power Sequential Circuit using Single Phase Adiabatic Dynamic Logic," *Proc. Intl Conf. Computers and Devices for Communication*, pp. 1-4, 2009.
- [27] C. Monteiro, Y. Takahashi, T. Sekine, "Robust Secure Charge-Sharing Symmetric Adiabatic Logic Against Side-Channel Attacks," *Proc. Intl Conf. Telecommunications and Signals Processing*, pp. 732-736, 2013.
- [28] J. Hu, L. Wang, H. Dong, "Interface Circuits between Adiabatic and Standard CMOS Circuits," *Proc. Midwest Symp. Circuits and Systems*, pp. 562-565, 2007.
- [29] R. Dennard, "Design of Ion-implanted MOSFETs with Very Small Physical Dimensions," *IEEE J. Solid State Circuits*, Vol. SC-9, Issue 5, pp. 256-268, 1974.

- [30] S. Nakata, H. Hanazono, H. Makino, H. Morimura, M. Miyama, Y. Matsuda, "Increase in Read Noise Margin of Single-Bit-Line SRAM Using Adiabatic Change of Word Line Voltage," *IEEE Trans. VLSI Systems*, Vol. 22, Issue 3, pp. 686-690, 2014.
- [31] P. Kogge, H. Stone, "A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations," *IEEE Trans. Computers*, Vol. C-22, Issue 8, pp. 789-793, 1987.
- [32] C. Wallace, "A Suggestion for a Fast Multiplier," *IEEE Trans. Electronic Computers*, Vol. EC-13, Issue 1, pp. 14-17, 1964.
- [33] Y. Takahashi, D. Tsuzuki, T. Sekine, M. Yokoyama, "Design of a 16-bit RISC CPU Core in a Two Phase Drive Adiabatic Dynamic CMOS Logic," *Proc. IEEE Region 10 Conf.*, pp. 1-4, 2007.
- [34] G. Yemiscioglu, P. Lee, "16-bit Clocked Adiabatic Logic (CAL) Logarithmic Signal Processor," *Proc. IEEE Intl Midwest Symp. Circuits and Systems*, pp. 113-116, 2012.
- [35] Y. Shin, C. Lee, Y. Moon, "A Design of 16-bit Adiabatic Microprocessor Core," *ETRI J. Semiconductor Technology and Science*, Vol. 26, Issue 6, pp. 194-198, 2004.
- [36] W. C. Athas, N. Tzartzanis, L. Svensson, L. Peterson, "A Low-power Microprocessor Based on Resonant Energy," *IEEE J. Solid-State Circuits*, Vol. 32, Issue 11, pp. 1693-1701, 1997.

- [37] K. Tiri, M. Akmal, I. Verbauwhede, "A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Differential Power Analysis on Smart Cards," *Proc. 28th European Solid-State Circuits Conf.*, pp. 403-406, 2002.
- [38] K. Tiri, I. Verbauwhede, "A Logic Level Design Methodology for a Secure DPA Resistant ASIC of FPGA Implementation", *Proc. Design, Automation and Test in Europe Conf. and Expo.*, pp. 246-251, 2004.
- [39] C. Monteiro, Y. Takahashi, T. Sekine, "Resistance Against Power Analysis Attacks on Adiabatic Dynamic and Adiabatic Differential Logics for Smart Card," *Proc. Intl Symp. Signal Processing and Communication Systems*, pp. 1-5, 2011.
- [40] P. Kocher, J. Jaffe, B. Jun, "Differential Power Analysis," *Proc. Annual Intl Cryptology Conf. (CRYPTO'99)*, Lecture Notes in Computer Science, Vol. 1666, pp. 388-397, 1999.
- [41] K. Kulikowski, M. Karpovsky, A. Taubin, "Power Attacks on Secure Hardware Based on Early Propagation of Data," *Proc. IEEE Intl On-Line Testing Symp.*, pp. 131-138, 2006.
- [42] S. Mangard, "A Simple Power-Analysis (SPA) Attack on Implementation of the AES Key Expansion," *Proc. 5th Intl Conf. Information Security and Cryptology*, pp. 343-358, 2003.
- [43] S. Mangard, "Exploiting Radiated Emissions – EM Attacks on Cryptographic ICs," *Proc. Austrochip 2003*, pp. 13-16, 2003.

- [44] S. Mangard, "Hardware Countermeasures against DPA – A Statistical Analysis of Their Effectiveness," *Proc. RSA Conf. 2004*, Lecture Notes in Computer Science, Vol. 2964, pp. 222-235, 2004.
- [45] S. Mangard, T. Popp, B. Gammel, "Side-Channel Leakage of Masked CMOS Gates," *Proc. RSA Conf. 2005*, Lecture Notes in Computer Science, Vol. 3376, pp. 351-365, 2005.
- [46] S. Mangard, N. Pramstaller, E. Oswald, "Successfully Attacking Masked AES Hardware Implementations," *Proc. Intl Workshop Cryptographic Hardware and Embedded Systems*, Lecture Notes in Computer Science, Vol. 3659, pp. 157-171, 2005.
- [47] S. Örs, E. Oswald, B. Preneel, "Power-Analysis Attacks on FPGAs – First Experimental Results," *Proc. Intl Workshop Cryptographic Hardware and Embedded Systems*, Lecture Notes in Computer Science, Vol. 2779, pp. 35-50, 2003.
- [48] T. Popp, S. Mangard, "Masked Dual-Rail Pre-Charge Logic: DPA-Resistance without Routing Constraints," *Proc. Intl Workshop Cryptographic Hardware and Embedded Systems*, Lecture Notes in Computer Science, Vol. 3659, pp. 172-186, 2005.
- [49] T. Popp, S. Mangard, "Implementation Aspects of the DPA-Resistant Logic Style MDPL," *Proc. IEEE Intl Symp. Circuits and Systems*, pp. 2913-2916, 2006.
- [50] N. Pramstaller, E. Oswald, S. Mangard, F. Gürkaynak, "A Masked AES ASIC Implementation," *Proc. Austrochip 2004*, pp. 77-82, 2004.

- [51] J.-J. Quisquater, D. Samyde, "Electromagnetic Analysis (EMA): Measures and Counter-Measures for Smart Cards," *Proc. Intl Conf. Research in Smart Cards*, Lecture Notes in Computer Science, Vol. 2140, pp. 200-210, 2001.
- [52] G. Ratanpal, R. Williams, T. Blalock, "An On-Chip Signal Suppression Countermeasure to Power Analysis Attacks," *IEEE Trans. Dependable and Secure Computing*, Vol. 1, Issue 3, pp. 179-189, 2004.
- [53] H. Marzouqi, M. Al-Qutayri, K. Salah, "Review of Gate-level Differential Power Analysis and Fault Analysis Countermeasures," *IET Information Security*, Vol. 8, Issue 1, pp. 51-66, 2014.
- [54] T. Sundström, A. Alvandpour, "A Comparative Analysis of Logic Styles for Secure IC's Against DPA Attacks," *Proc. 23rd NORCHIP Conf.*, pp. 297-300, 2005.
- [55] A. Shamir, "Protecting Smart Cards from Passive Power Analysis with Detached Power Supplies," *Proc. Intl Workshop Cryptographic Hardware and Embedded Systems*, Lecture Notes in Computer Science, Vol. 1965, pp. 71-77, 2000.
- [56] Z. Toprak, Y. Leblebici, "Low-Power Current Mode Logic for Improved DPA-Resistance in Embedded Systems," *Proc. IEEE Intl Symp. on Circuits and Systems*, Vol. 2, pp. 1059-1062, 2005.
- [57] K. Tiri, I. Verbauwhede, "A Digital Design Flow for Secure Integrated Circuits," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 25, Issue 7, pp. 1197-1208, 2006.

- [58] J.-W. Lee, S.-C. Chung, H.-C. Chang, C.-Y. Lee, "Efficient Power-Analysis-Resistant Dual-Field Elliptic Curve Cryptographic Processor Using Heterogeneous Dual-Processing-Element Architecture," *IEEE Trans. VLSI Systems*, Vol. 22, Issue 1, pp. 49-61, 2014.
- [59] A. Fournaris, "Toward Flexible Security and Trust Hardware Structures for Mobile-Portable Systems," *IEEE Trans. Latin America*, Vol. 10, Issue 3, pp. 1719-1722, 2012.
- [60] Y. Chen, Z. Xuecheng, L. Zhenglin, H. Yu, Z. Zhaoxia, "Energy-Efficient and Security-Optimized AES Hardware Design for Ubiquitous Computing," *IEEE J. Systems Engineering and Electronics*, Vol. 19, Issue 4, pp. 652-658, 2008.
- [61] M. Alioto, L. Giancane, G. Scotti, A. Trifiletti, "Leakage Power Analysis Attacks: A Novel Class of Attacks to Nanometer Cryptographic Circuits," *IEEE Trans. Circuits and Systems I: Regular Papers*, Vol. 57, Issue 2, pp. 355-367, 2010.
- [62] J. Wu, Y. Shi, M. Choi, "Measurement and Evaluation of Power Analysis Attacks on Asynchronous S-Box," *IEEE Trans. Instrumentation and Measurement*, Vol. 61, Issue 10, pp. 2765-2775, 2012.
- [63] M. Alioto, M. Poli, S. Rocchi, "A General Power Model of Differential Power Analysis Attacks to Static Logic Circuits," *IEEE Trans. VLSI Systems*, Vol. 18, Issue 5, pp. 711-724, 2010.

- [64] R. Muresan, S. Gregori, "Protection Circuit against Differential Power Analysis Attacks for Smart Cards," *IEEE Trans. Computers*, Vol. 57, Issue 11, pp. 1540-1549, 2008.
- [65] M. Alioto, S. Bongiovanni, M. Djukanovic, G. Scotti, A. Trifiletti, "Effectiveness of Leakage Power Analysis Attacks on DPA-Resistant Logic Styles Under Process Variations," *IEEE Trans. Circuits and Systems I: Regular Papers*, Vol. 61, Issue 2, pp. 429-442, 2014.
- [66] S. Mangard, E. Oswald, F.-X. Standaert, "One for All – All for One: Unifying Standard Differential Power Analysis Attacks," *IET J. Information Security*, Vol. 5, Issue 2, pp. 100-110, 2011.
- [67] M. Alioto, M. Poli, S. Rocchi, "Differential Power Analysis Attacks to Precharged Buses: A General Analysis for Symmetric-Key Cryptographic Algorithms," *IEEE Trans. Dependable and Secure Computing*, Vol. 7, Issue 3, pp. 226-239, 2010.
- [68] J.-W. Lee, J.-H. Hsiao, H.-C. Chang, C.-Y. Lee, "An Efficient DPA Countermeasure With Randomized Montgomery Operations for DF-ECC Processor," *IEEE Trans. Circuits and Systems II: Express Briefs*, Vol. 59, Issue 5, pp. 287-291, 2012.
- [69] N.-H. Zhu, Y.-J. Zhou, H.-M. Liu, "Employing Symmetric Dual-Rail Logic to Thwart LPA Attack," *IEEE Embedded Systems Letters*, Vol. 5, Issue 4, pp. 61-64, 2013.

- [70] P.-C. Liu, H.-C. Chang, C.-Y. Lee, "A Low Overhead DPA Countermeasure Circuit Based on Ring Oscillators," *IEEE Trans. Circuits and Systems II: Express Briefs*, Vol. 57, Issue 7, pp. 546-550, 2010.
- [71] W. Shan, X. Chen, B. Li, P. Cao, J. Li, G. Gao, L. Shi, "Evaluation of Correlation Power Analysis Resistance and Its Application on Asymmetric Mask Protected Data Encryption Standard Hardware," *IEEE Trans. Instrumentation and Measurement*, Vol. 62, Issue 10, pp. 2716-2724, 2013.
- [72] Y. Oren, A. Shamir, "Remote Password Extraction from RFID Tags," *IEEE Trans. Computers*, Vol. 56, Issue 9, pp. 1292-1296, 2007.
- [73] A. Moradi, M. Kirschbaum, T. Eisenbarth, C. Paar, "Masked Dual-Rail Precharge Logic Encounters State-of-the-Art Power Analysis Methods," *IEEE Trans. VLSI Systems*, Vol. 20, Issue 9, pp. 1578-1589, 2012.
- [74] D. Sokolov, J. Murphy, A. Bystrov, A. Yakovlev, "Design and Analysis of Dual-Rail Circuits for Security Applications," *IEEE Trans. Computers*, Vol. 54, Issue 4, pp. 449-460, 2005.
- [75] A. Bogdanov, I. Kizhvatov, "Beyond the Limits of DPA: Combined Side-Channel Collision Attacks," *IEEE Trans. Computers*, Vol. 61, Issue 8, pp. 1153-1164, 2012.
- [76] C. Tokunaga, D. Blaauw, "Securing Encryption Systems With a Switched Capacitor Current Equalizer," *IEEE J. Solid-State Circuits*, Vol. 45, Issue 1, pp. 23-31, 2010.

- [77] S. Guilley, L. Sauvage, F. Flament, V.-N. Vong, P. Hoogvorst, R. Pacalet, "Evaluation of Power Constant Dual-Rail Logics Countermeasures Against DPA With Design Time Security Metrics," *IEEE Trans. Computers*, Vol. 59, Issue 9, pp. 1250-1263, 2010.
- [78] K. Tanimura, N. Dutt, "HDRL: Homogeneous Dual-Rail Logic for DPA Attack Resistant Secure Circuit Design," *IEEE Embedded Systems Letters*, Vol. 4, Issue 3, pp. 57-60, 2012.
- [79] D. Karaklajic, J.-M. Schmidt, I. Verbauwhede, "Hardware Designer's Guide to Fault Attacks," *IEEE Trans. VLSI Systems*, Vol. 21, Issue 12, pp. 2295-2306, 2013.
- [80] S. Clark et al., "Current Events: Identifying Webpages by Tapping the Electrical Outlet," *Proc. European Symp. Research in Computer Security*, Lecture Notes in Computer Science, Vol. 8134, pp. 700-717, 2013.
- [81] D. Genkin, A. Shamir, E. Tromer, "RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis," <http://eprint.iacr.org/2013/857.pdf>, 2013.
- [82] S. Mangard, E. Oswald, T. Popp, "Power Analysis Attacks: Revealing the Secrets of Smart Cards," *Springer New York*, New York, USA, 2007.
- [83] J.-L. Danger, A. Guilley, S. Bhasin, M. Nassar, "Overview of Dual Rail with Precharge Logic Styles to Thwart Implementation-Level Attacks on Hardware Cryptoprocessors – New Attacks and Improved Counter-Measures," *Proc. Workshop Secure Control Systems*, pp. 1-8, 2009.

- [84] P. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," *Proc. Annual Intl Cryptology Conf. (CRYPTO'96)*, Lecture Notes in Computer Science, Vol. 1109, pp. 104-113, 1996.
- [85] S. Chen, R. Wang, X. Wang, K. Zhang, "Side-Channel Leaks in Web Applications: A Reality Today, A Challenge Tomorrow," *Proc. IEEE Symp. Security and Privacy*, pp. 191-206, 2010.

VITA

Mihail T. Cutitaru

Department of Electrical and Computer Engineering

Old Dominion University

Norfolk, VA 23529

EDUCATION

- B.S. Computer Engineering, Old Dominion University, Norfolk, VA, May 2010.

AWARDS

- Dean's Doctoral Fellow, Frank Batten College of Engineering and Technology, Old Dominion University, Norfolk, VA, 2011-2013.

PUBLICATIONS

- M. Cutitaru, L. A. Belfore, II, "A Partially-Adiabatic Energy-Efficient Logic Family as a Power Analysis Attack Countermeasure," *Proc. IEEE Asilomar Conf. Signals, Systems, and Computers*, pp. 1125-1129, 2013.
- M. Cutitaru, L. A. Belfore, II, "An Energy-Efficient 32-bit Kogge-Stone Adder Using a Dual-Phase Partially-Adiabatic Logic Family," *Proc. Intl Conf. Computer Applications in Industry and Engineering (CAINE13)*, pp. 113-116, 2013.
- M. Cutitaru, L. A. Belfore, II, "Arithmetic Circuits Using New Single-Phase Partially-Adiabatic Logic Family," *Proc. IEEE Intl Midwest Symp. Circuits and Systems (MWSCAS13)*, pp. 13-16, 2013.
- M. Cutitaru, L. A. Belfore, II, "New Single-Phase Adiabatic Logic Family," *Proc. of Intl Conference on Computer Design (CDES12)*, pp. 9-14, 2012.