Old Dominion University
ODU Digital Commons

Electrical & Computer Engineering Theses & Disssertations

Electrical & Computer Engineering

Spring 2016

IDPAL - Input Decoupled Partially Adiabatic Logic Family: Theory and Implementation of Side-Channel Attack Resistant Circuits

Matthew Edward McAllister
Old Dominion University

Recommended Citation

McAllister, Matthew E.. "IDPAL - Input Decoupled Partially Adiabatic Logic Family: Theory and Implementation of Side-Channel Attack Resistant Circuits" (2016). Master of Science (MS), thesis, Electrical/Computer Engineering, Old Dominion University, DOI: 10.25777/ck0r-q716
https://digitalcommons.odu.edu/ece_etds/2

IDPAL – INPUT DECOUPLED PARTIALLY ADIABATIC LOGIC FAMILY: THEORY AND IMPLEMENTATION OF SIDE-CHANNEL ATTACK RESISTANT CIRCUITS

by

Matthew Edward McAllister

A.S. December 2012, Tidewater Community College

B.S. December 2014, Old Dominion University

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

MASTER OF SCIENCE

ELECTRICAL AND COMPUTER ENGINEERING

OLD DOMINION UNIVERSITY
May 2016

Approved by:

Lee A. Belfore, II (Director)

Oscar R. Gonzalez (Member)

ChunSheng Xin (Member)

ABSTRACT

IDPAL – INPUT DECOUPLED PARTIALLY ADIABATIC LOGIC FAMILY: THEORY AND IMPLEMENTATION OF SIDE-CHANNEL ATTACK RESISTANT CIRCUITS

Matthew Edward McAllister
Old Dominion University, 2016
Director: Dr. Lee A. Belfore, II

The Input Decoupled Partially Adiabatic Logic (IDPAL) family was developed by Cutitaru to consume less power than other logic families as well as producing a resistance to side-channel attacks. With modifications made to IDPAL, the side-channel attack resistance is being revisited and quantified. The three logic families are compared in the work are CMOS, 2N2P, and IDPAL. An AND/NAND gate was created using each logic family and compared with two tests: 1) a simulated side-channel attack and 2) an energy analysis. In this work, a side-channel attack is the ability to predict the inputs of a logic circuit based on the electrical current waveform. For the Test 1, a higher prediction error suggests a higher resistance to attack. IDPAL produced the highest error in this test at 50.000%, which is 40.625% higher than in CMOS and 28.125% higher than in 2N2P. In Test 2, two primary statistics that were observed the variance in current trace (NSD and NED). Lower values of these measures implies a higher chance of a model resisting a side-channel attack. For the individual logic gates, the IDPAL model showed a lower variance in one of the two measures. For the Kogge-Stone adder, a more complex circuit, the IDPAL model was superior in both tests. With the results of the small and larger scale experiments in agreement, the final conclusion is that IDPAL does, indeed, resist side-channel attacks in a stronger fashion than other logic families.

This is dedicated to my parents, Edward and Margaret McAllister.

ACKNOWLEDGMENTS

This work is the culmination of research and testing done by myself, Dr. Lee Belfore, and a number of students and faculty, both here at Old Dominion University and at other universities. It is my hope that my work with adiabatic logic circuits will be useful to future students. First, I must thank my advisor, Dr. Lee Belfore, for all he has done for me while I have attended Old Dominion University both as an undergraduate student and a graduate student. I am very grateful for the opportunities I have had to work with him over my time as an ODU student. I would also like to thank Dr. Oscar Gonzalez and Dr. ChunSheng Xin, my thesis committee members. Their comments and suggestions were instrumental in the completion of this work, as it helped me to see my work from a different angle.

The Department of Electrical and Computer Engineering, as a whole, has been a place of growth for me. Everyone played a part in my development, not only as a student, but as a person. For everyone in this department, I am appreciative of your daily support.

Finally, but most importantly, my family. My parents, Edward and Margaret, told me that I could do anything I set my mind to. My siblings (Ashli, Kristin, Holly, and Joseph) and brothers-in-law (Eric and Robert) have always supported me and encouraged me to give everything I had. Without their love and support, I would not have gone as far as I have today.

TABLE OF CONTENTS

Page

Chapter

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

The research field of adiabatic logic families is constantly growing, as it shows the potential for creating digital circuits that consume lower power than commercially available products. One such family that is current in development is Input Decoupled Partially Adiabatic Logic (IDPAL), which also has shown promise in low power applications. Because IDPAL is dual rail, where logic circuits receive true and complementary input and generate true and complementary logic outputs, analysis conducted by Cutitaru suggested IDPAL could be used to implement encryption technologies and may be resistant to side-channel power analysis attacks [13]. Thus, IDPAL was developed with two intended properties: consuming very little power compared to other technologies and providing a resistance to power analysis attacks. To address the first characteristic, the components from this logic family recycle unused energy back into the power supply. The second characteristic was examined in his dissertation. As he concluded from his work, IDPAL performed well in both categories when compared to other logic families.

This work will build upon Cutitaru's work. Three logic families (CMOS, 2N2P, and IDPAL) will be studied and directly with two tests: one from Cutitaru's work and the other being a simulated side-channel attack. The results from each analysis technique should agree with each other in order to draw conclusions about IDPAL resistance to side-channel attacks.

I.1 SECURITY OF DIGITAL INFORMATION

Consider an individual who owns a wireless router in their home. This digital information can range in importance from their full name and street address to their credit card information. On a larger scale, such as a company's private server, the importance might range from their employee's information to trade secrets and intellectual information. Regardless of the importance of the data being protected by a digital system, it is the purpose of the digital system to prevent unauthorized users from accessing its data.

In order to address the circuit's security, it is important to understand what techniques might be used against it. The primary intrusion method, for this work, will be side-channel attacks. These attacks can be implemented with or without physical access to the digital system, as it retrieves any usable information to infer the current state of the target system. The information is not inherently given by the digital system; the intruder observes minute characteristics or properties of the system. This includes access to the power supply of the digital system itself. Once the leaked information is retrieved, an attack can be prepared within a short amount of time and under a reasonable budget. Side-channel attacks, themselves, can be broken down into different types of attacks, such as timing attacks, simple power analysis attacks, and differential power analysis attacks [1-5]. All three require a source of leaked information to begin the attack. Timing attacks rely on the amount of time a system takes to perform a task, while the two power analysis attacks rely on the amount of power consumed by the system.

Encryption algorithms often process the key into an extended series of subkeys that are used in the encryption process. Since the subkeys are derived from the encryption key, one may be able to infer bits in the encryption key using a suitably crafted side-channel attack. Assume the encryption scheme uses a 256-bit key or 32 bytes. Thus, each subkey is derived from these 32 bytes. Using the appropriate techniques in the selected side-channel attack, each subkey is determined using the leaked information. The attack ends when each subkey has been determined. Each subkey is then combined into the secret key for the encryption [3].

I.2 HARDWARE SIMULATION AND EMULATION

In order to observe the behavior of the circuits, a software program was required to simulate them. Developed by Linear Technology, Inc., LTspice® is an electronic circuit building and simulation software provided free by the company. Furthermore, LTspice® can be used to build electronic circuits from basic principles or import SPICE models from another program. Both features were utilized for this work. Finally, LTspice® has the capability to export circuit data (voltages, currents, etc.) as text files, which can be imported into other programs.

I.3 PROBLEM STATEMENT

As previously stated, Cutitaru sought to develop IDPAL into an adiabatic logic family that would qualify as a very-low power logic family as well as resist power analysis attacks. He conducted a preliminary analysis that suggested IDPAL is resistant to power analysis attacks. However, his analysis did not address a realized power analysis attack targeting the digital circuits in his research. The testing methods he used observed statistical properties linked to the current draw of each digital circuit. Further, the model used for IDPAL has changed since Cutitaru's analysis. In order to build upon the groundwork that he has established, it is proposed that IDPAL is reanalyzed in order to verify that it does resist side-channel attacks.

In order to accomplish this, three logic families (CMOS, 2N2P, and IDPAL) will be directly compared using Cutitaru's analysis techniques as well as a realized side-channel attack. If the results of the analysis methods agree, a conclusion can be made to IDPAL's resistance to side-channel attacks.

I.4 CONTRIBUTIONS OF THIS WORK

This work extends into both electrical and computer engineering concepts. As a general starting point, research was initiated to understand the concepts illustrated by Cutitaru. Through various conference papers, course textbooks, and Cutitaru's work, the following information was obtained: the theory and implementation of each logic family (CMOS, 2N2P, and IDPAL), the metrics used in Cutitaru's analysis, and an understanding of side-channel attack methods.

Previous work in this field of research had been done here at Old Dominion University by Cutitaru. This work presents three principal contributions. The first was a method of using current draw information to predict the inputs of logic gates. The second was a method of determining a Kogge-Stone adder's resistance to a side-channel attack using the variance in its current draw. The third came in the form of a MATLAB tool to perform these two analysis techniques. In order to obtain the Kogge-Stone adder models, a VHDL modeling framework capable of exporting spice models was provided by Belfore [21]. From this framework, three different technological implementations of Kogge-Stone adders were studied.

I.5 THESIS OVERVIEW

This document will follow a standard format. Chapter II will address background information necessary to understand the analysis sections presented in this work. This includes the three logic families (CMOS, 2N2P, and IDPAL), the side-channel attack, and the energy analysis. Chapter III will cover a small-scale example, using AND/NAND gates created in LTspice® to serve as a baseline for each logic family's expected performance. Chapter IV will look at a larger-scale example: 4-bit Kogge-Stone adders. Using the models obtained from the previously mentioned VHDL framework, this chapter will look to confirm the conclusion made in Chapter III. Chapter V will observe potential work that might be done with the findings of this work. Finally, Chapter VI will present the final thoughts and appropriate conclusions derived from the work.

CHAPTER II

BACKGROUND

In Chapter 1, the starting point for this work, along with some preliminary information and the problem statement, was established in order to prepare the reader for the two technical chapters (Chapters 3 and 4). In this chapter, the groundwork for those technical chapters will now be presented.

II.1 CHAPTER OVERVIEW

In this section, the chapter will be overviewed. Section 2 reviews CMOS technology in a general sense. It includes a brief review of transistors and their behavior. This lays the foundation for Section 3, which shows how CMOS technology can be used to create digital logic elements as well as its performance. Section 4 introduces adiabatic computation, illustrating general properties of adiabatic logic circuits. This is immediately followed by two subsections that serve as the background for the two selected adiabatic logic families: 2N2P and IDPAL. This includes how they are implemented in basic digital logic elements as well as their performance. After this, the concept of side-channel attacks is covered in Section 5. This includes the technique used in the next chapter of this work. Finally, Section 6 addresses how this research can be used in two encryption schemes, being the RC4 and Advanced Encryption Standard (AES) encryption schemes.

II.2 COMPLEMENTARY METAL-OXIDE SEMICONDUCTOR (CMOS) TECHNOLOGY REVIEW

In order to understand the deeper workings of digital systems, understanding their behavior at the transistor level can be valuable. Here, a brief summary of the operation of transistors and their performance characteristics is provided. Transistors are multi-terminal semiconductor devices. Two common types of transistors are metal-oxide semiconductor field effect transistors (MOSFET) and bipolar junction transistors (BJT) [6]. In all of the circuits built for this work, they are entirely made up of MOSFETS (both PMOS and NMOS). MOSFETs have two notable advantages over BJTS. MOSFETs have a lower average power consumption than BJTs, and MOSFETs are significantly smaller than BJTs. Figure 1 provides the schematic symbols that are traditionally used for NMOS and PMOS transistors. Incidentally, these specific symbols are used in LTspice® [7].

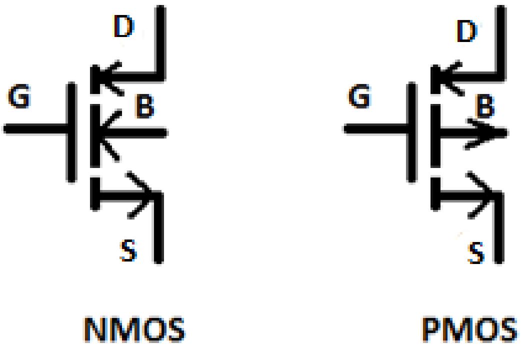


Figure 1: N-type and P-type MOSFET Symbols

A MOSFET has the following four terminals: a source (S), a gate (G), a substrate (B), and a drain (D). The behavior of MOS transistors can be described as follows: if the voltage at the gate is less than a specified threshold value (for NMOS transistors), no current will flow between the drain and source terminals. Otherwise, the current at the source will flow through the MOSFET from the drain to the source. In particular, there are two operating modes for a MOSFET: triode and saturation. The triode region allows the current to flow continuously. This is based on the set overdrive voltage (V_{OV}), which is the voltage drop between the gate and source connections (V_{GS}) minus the threshold voltage (V_t). This mode of operation only takes place when the voltage drop between the drain and source terminals (V_{DS}) is less than V_{OV}. When this is the case, the current flowing through the drain is as follows: $i_D = k'_n \left(\frac{W}{L}\right)\left(V_{OV} - \frac{1}{2}V_{DS}\right)V_{DS}$. If V_{DS} is equal to or greater than V_{OV}, it will be in saturation mode. This causes the current flow to become confined to a particular value, as described by the following: $i_D = \frac{1}{2}k'_n \left(\frac{W}{L}\right)V_{OV}^2$ [8, p. 243].

In order to simplify the understanding of MOSFETs for the purpose of this report, the information will be directly applied, as the current will behave according to both of the MOSFET operative modes. The MOSFETs used in these electrical circuits are being utilized to represent the behavior of digital logic, behaving as voltage controlled current switches. When the gate voltage exceeds the threshold voltage, a channel (an n-channel for NMOS) is created to allow the current to flow. PMOS transistors operate in a complementary fashion [6,7]. In basic applications and SPICE simulations, the substrate (also called the “bulk” or “body” connection) is not addressed in detail. Because of the physical properties of MOSFETs, this connection does not heavily impact the MOSFETs performance, as only specific voltage values are accepted for proper behavior. By grounding the substrate connection of the NMOS transistors and linking V_{DD} to the PMOS transistors, each one will operate as expected, minimizing spiking behavior [8, p.234].

Virtually every digital integrated circuit is built using CMOS technology. The name implies a complementary balance between NMOS and PMOS transistors. CMOS technology has two notable strengths: low power consumption and a good resistance to noise. Those two features have made CMOS a more desirable option than other technologies, such as transistor-to-transistor logic (TTL).

II.3 CMOS LOGIC FAMILY

As discussed in a previous section, CMOS is the logic family that will serve as a baseline for all of the logic families shown in this work. In order to create basic logic gates with CMOS, there are two elements that are fundamental in the design: pull-up networks (PUNs) and pull-down networks (PDNs). Both are based on the concept of switching networks, which use digital logic control the flow of current through the circuit. The PUN, which is comprised of PMOS transistors, is usually connected to the input voltage source, and the PDN, which is made up of NMOS transistors, is connected to a lower voltage source (usually ground). Both networks are designed to be complementary to each other, creating the desired logic function [7]. For CMOS technology, both networks only receive true inputs to produce the proper switching. To illustrate these points, Figure 2, which is a CMOS inverter, can be found below.

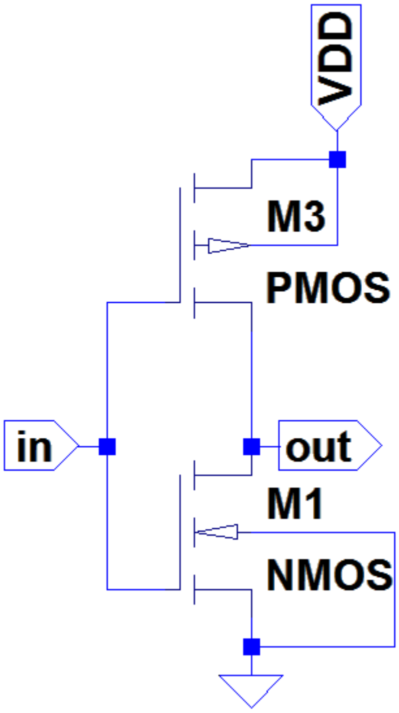


Figure 2: CMOS Inverter

For the inverter shown in Figure 2, whenever the voltage input is logically HIGH (input voltage exceeds threshold voltage), the bottom MOSFET is turned on and the top is shut off, linking the output to ground. Likewise, a logical LOW (gate voltage lower than threshold voltage) input will result in the PUN being opened and the PDN shut off, linking the output to the supply voltage directly. For CMOS logic gates, the substrate connections are made in the following manner: link the substrate connection of PMOS transistors to the power supply, V_{DD} (which will be 3.0VDC for all of the circuits in this work), and link the NMOS transistor substrate connections to the ground [7].

Building off of the inverter, a CMOS AND/NAND gate will be shown. As a review, Table 1 shows the truth table for both functions.

Table 1: Truth Table for AND/NAND Gate

A	B	F(AND)	F(NAND)
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

As a reference point, the table shows logical LOW as a ‘0’ and a logical HIGH as a ‘1’. Thus, the only time the output “F” should be logically HIGH is when both inputs A and B are also logically HIGH. In order to construct a CMOS AND/NAND gate, a CMOS NAND gate will be built, and its output will be directly fed into the CMOS inverter that was just covered. Equations 1 and 2 describe the Boolean expressions for both functions.

$$F(AND) = AB \tag{1}$$

$$F(NAND) = \overline{AB} \tag{2}$$

Using DeMorgan's theorem on Equation 2, it is possible to simplify the right hand of the equation, producing Equation 3.

$$F(NAND) = \bar{A} + \bar{B} \tag{3}$$

To design the logic for the NAND gate, both the PUN and PDN must be specifically designed. As previously stated, the PUN is connected to the source voltage, and the PDN is connected to ground. Using the truth table as a guide, the PDN should connect the output directly to ground when both inputs (A and B) are logically HIGH. This means the PUN should be designed to directly link the output to the source voltage when both inputs are not logically HIGH. To accomplish this, it would be advisable to observe the basic switching network configurations that will be used. While other configurations exist, four primary switching networks will be covered here. Each one can be found in Figure 3, followed by a discussion of how each works.

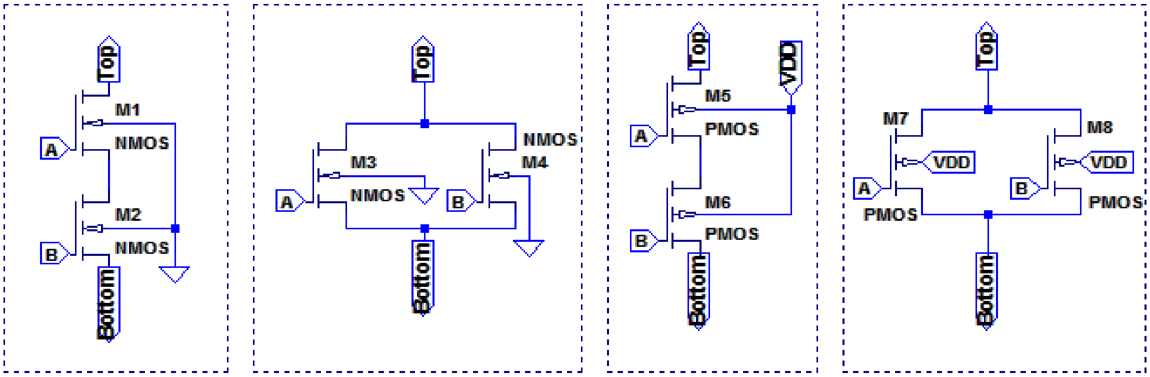


Figure 3: Basic Switching Network Configurations

Each network will be covered from left to right. For each of the networks, it will be assumed that there is a current that will attempt to flow from the top connection to the bottom. Further, A and B are the true input voltages, and VDD is a DC voltage which, as stated previously, is 3.0V. The leftmost network is only allows current to flow when both A and B are above the MOSFET's threshold voltage. This matches the function shown in the truth table for the AND function. The second network allows current to flow when either A or B are above the threshold voltage. That matches the logical OR operation. The third network allows current to flow only when A and B are below the threshold voltage. This represents the logical NOR operation. Finally, the rightmost network allows current to flow when either A and B are below the threshold voltage. This represents the NAND function. Having covered a few common switching networks, the concept of a dual switching network should be mentioned, as it is used in the adiabatic implementations. A dual switching network is one in which the MOSFET type and the input type is switched. For example, the NMOS AND switching network shown above requires two NMOS transistors in series, and the true inputs, A and B, are fed into the respective MOSFET gates. The dual of this network would be comprised of two PMOS transistors, with the complementary inputs, An and Bn, fed into the respective MOSFET gates. This, as stated previously, will become relevant in the adiabatic models, but it deserved mention here as switching networks were the topic of discussion here.

With these common network configurations covered, the design of a CMOS AND/NAND gate can be addressed, starting with the CMOS NAND gate. The gate is, again, comprised of a PUN and PDN. The PUN is connected to the source voltage and the output, and it is made up of the PMOS NAND switching network as described previously. The PDN is connected to the output and ground, and it consists of the NMOS AND switching network. The node between the PUN and PDN is the output of the NAND gate. By connecting an inverter to that node, the AND function is produced. Figure 4 below illustrates the final CMOS AND/NAND gate.

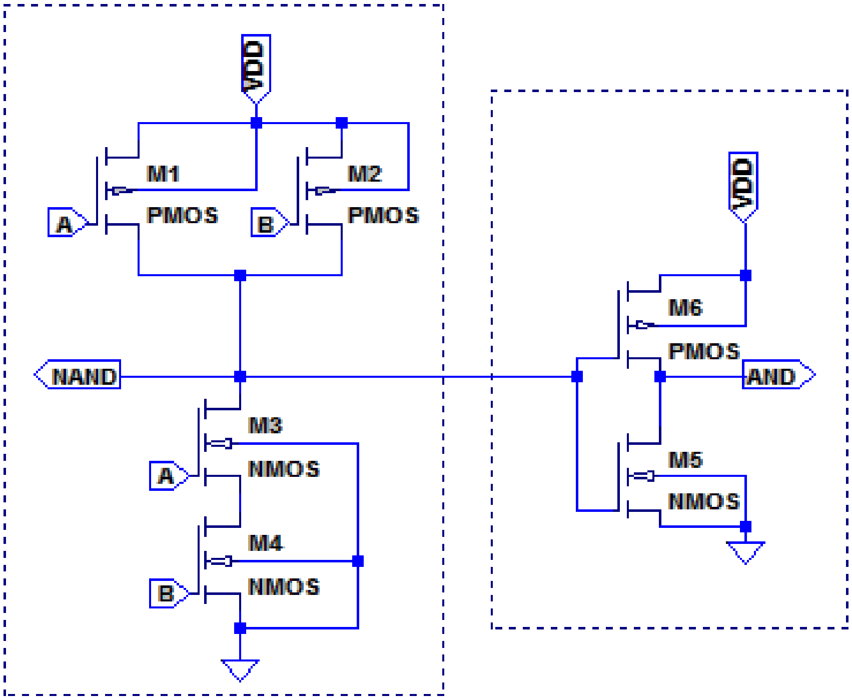


Figure 4: CMOS AND/NAND Logic Gate – Schematic

The left rectangular box shows the CMOS NAND gate as described above. As previously stated, the output of the NAND gate is labelled to show the output signal to be inverted. The block on the right, then, is the CMOS inverter as previously seen. The AND function is derived from this signal inversion. All of the substrate connections are shown in this figure, linking the NMOS connections to ground and the PMOS connections to the power supply (V_{DD}). In order to determine if the logic gate is functioning as designed, the input and output voltages can be compared. Figures 5 and 6 contain the waveforms showing the behavior of this CMOS AND/NAND gate.

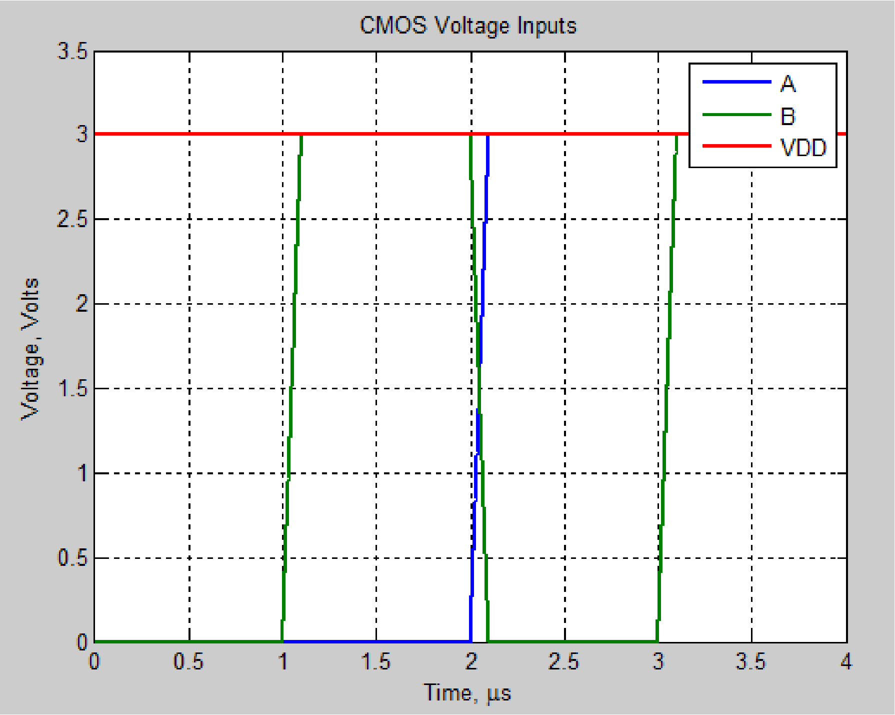


Figure 5: CMOS AND/NAND Gate – Input Voltage Waveforms

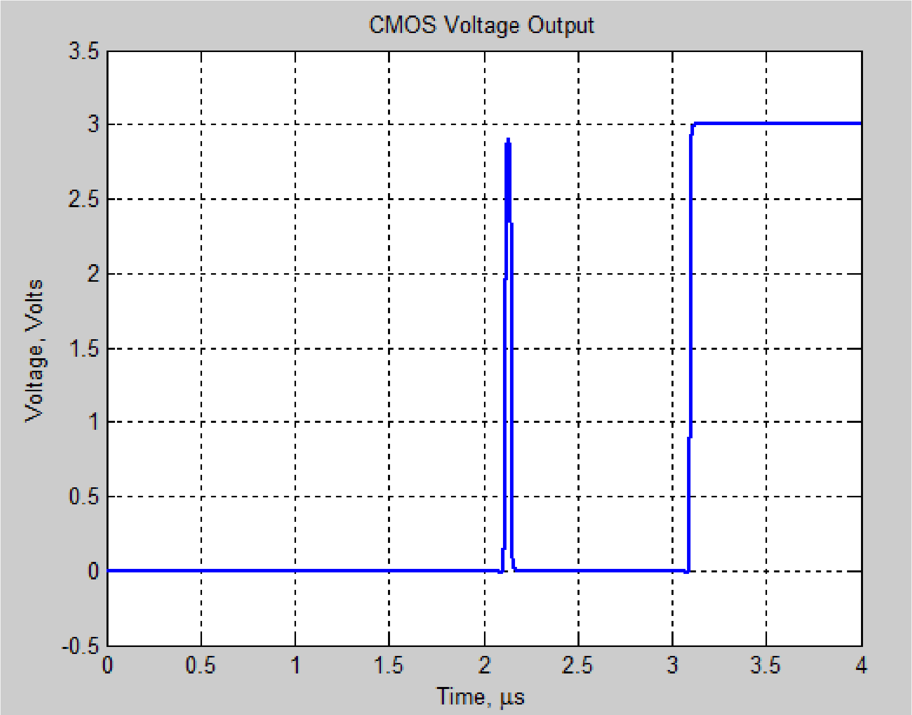


Figure 6: CMOS AND/NAND Gate – Output Voltage Waveform

Following the truth table listed at the beginning of this chapter, the behavior of this circuit can be visually inspected. For this and the following AND/NAND gate designs, the primary (true) output is the AND function and the secondary (complementary) output is the NAND function. Recall that the AND function should only be logically HIGH when both of the inputs, A and B, are also logically HIGH. The waveforms illustrate that point, as the output is only HIGH during that instance.

Now that the circuit has been proven to work, the current draw can be directly observed. To do this, the cases in the truth table (AB = 00, 01, 10, 11) were used as input combinations for the CMOS AND/NAND gate. Figure 7 is the current draw waveform that results from this sequence of inputs.

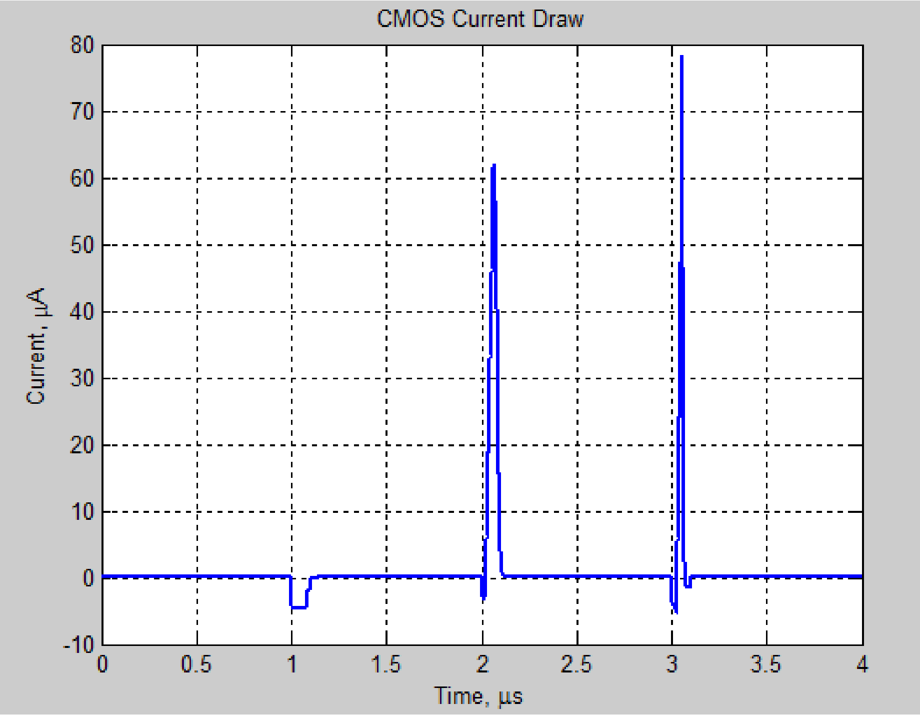


Figure 7: CMOS AND/NAND Gate – Current Draw Waveform

II.4 INTRODUCTION TO ADIABATIC COMPUTATION

An adiabatic proves is a process in which heat is neither gained nor lost. The same principle can be used in digital logic circuits. In a perfect world, all of the energy would be recycled. Energy recycling, however, does not necessarily imply adiabatic performance in these circuits. In order for these circuits to retrieve unused energy, a different method of powering the circuit is used. The power source slowly charges and discharges the digital system, which moves the charge into and out of the system without significant energy dissipation. As more research is conducted into optimizing or creating new configurations, the intention is to bring that fraction as close to the ideal case as possible [10,11]. There are many different technologies for specifying adiabatic logic circuits. Each one has unique traits and subtle differences, but each implementation is focused on reducing power consumption.

In general, adiabatic logic circuits operate with dual rail signals. This means that any given adiabatic logic circuit will receive a number of inputs and their complements, and the logic circuit will produce true and complementary outputs. An example would be appropriate in illustrating this point. Suppose two inputs, A and B, are to be fed into an ***AND*** function to produce an output, F. In order to make this happen, the complements of A and B must also be fed into the circuit: An and Bn. This will produce the complementary output: Fn. Figure 8 illustrates this, as well as illustrate one other useful feature.

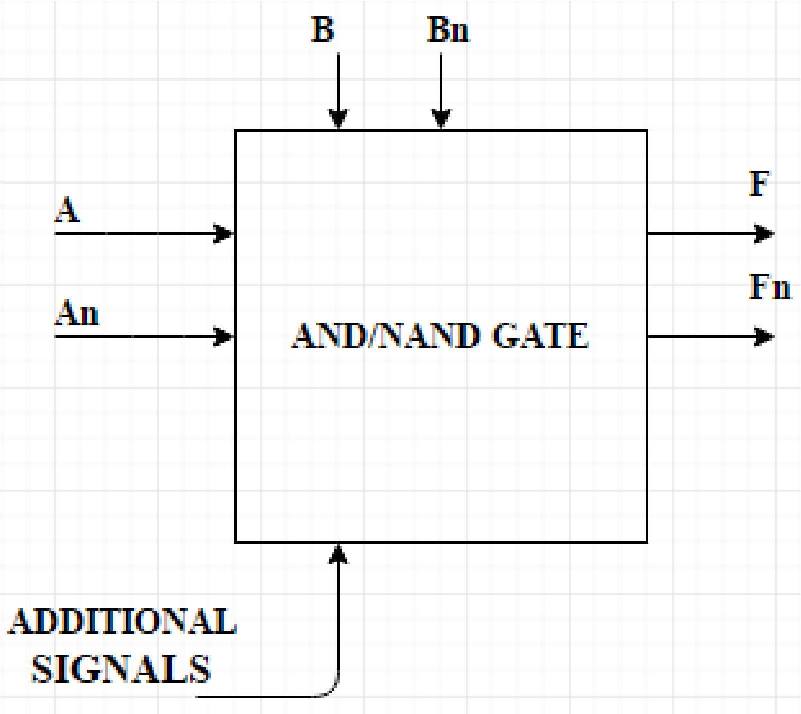


Figure 8: Adiabatic Logic Circuit Example – 2 Input AND/NAND Gate

As seen in the block diagram, the two inputs (A and B) and their complements (An and Bn) produce an output (F) and its complement (Fn). As stated in the beginning of this example, this circuit is to behave like an AND gate. Because of the complementary output, however, this circuit also behaves like a NAND gate. Thus, the adiabatic logic AND gate is also a NAND gate. This characteristic is true with the other basic logic gates (buffer/inverter, OR/NOR, XOR/XNOR). If this feature is utilized properly, it might be possible to optimize a digital logic system, reducing the total number of individual logic gates while maintaining normal operation.

The other signal that is shown in Figure 8 is labeled “Additional Signals”. Some configurations require one or more input signals to clock the adiabatic logic circuit. The primary signal fed into this area is called a power clock. The power clock is used to synchronize the circuit as well as supply the digital system with energy. In order to illustrate this signal and its impact on a circuit's performance, Figure 9 shows the current draw in a 2N2P AND/NAND gate alongside the power clock.

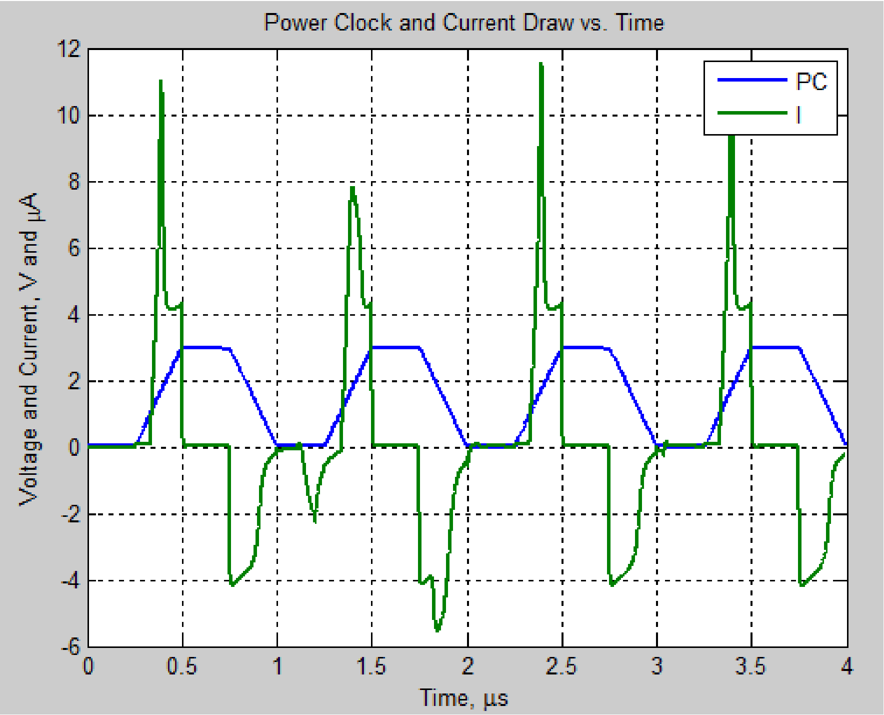


Figure 9: Power Clock Illustration – 2N2P AND/NAND Gate Current Draw

The blue curve shows the power clock signal, and the green curve shows the resulting current trace in a 2N2P AND/NAND gate. Each phase will be addressed. During the "off" and the “on” phases, no current is drawn into the circuit. There is a strong amount of current is pulled in during the “charge” phase, and a negative current is shown in the “discharge” phase. This negative current represents the unused charge being returned to the power supply.

II.4.1   2N2P LOGIC FAMILY

In this subsection, 2N2P will be the first of two adiabatic logic families discussed. In order to understand how 2N2P logic gates are designed and how they operate, please refer to Figure 10, which is a 2N2P buffer/inverter gate [11].

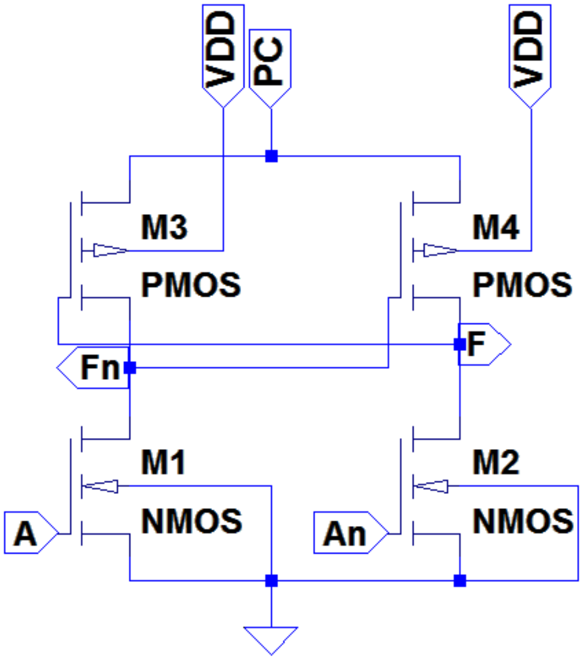


Figure 10: 2N2P Buffer/Inverter

In order to cover the circuit's performance, the structure will now be explored. The gate of each PMOS transistor is cross-fed into the drain of the opposite PMOS transistor. Assume the power clock voltage is constantly greater than 0.0V. If the true input, "in" is logically LOW, the output will be linked directly to ground. This will cross over to the opposite PMOS transistor's gate, opening a path between the complementary output and the voltage source, PC. Thus, the outputs will be complementary when the power clock is logically HIGH. When it is not, both outputs will read logically LOW. Once again, the substrate connections are not clearly shown in the figure. The substrate connections are the same as the CMOS model [20].

With the 2N2P buffer/inverter in mind, 2N2P AND/NAND gate will now be considered. The design of this logic gate follows a similar train of thought to the design of the CMOS AND/NAND gate and the 2N2P buffer/inverter. Unlike the CMOS AND gate, the 2N2P AND/NAND gate utilizes complementary logic. In contrast to the 2N2P buffer/inverter, the bottom half of the circuit (the NMOS transistors) are the output logic blocks. The schematic for this design can be found in Figure 11 below.

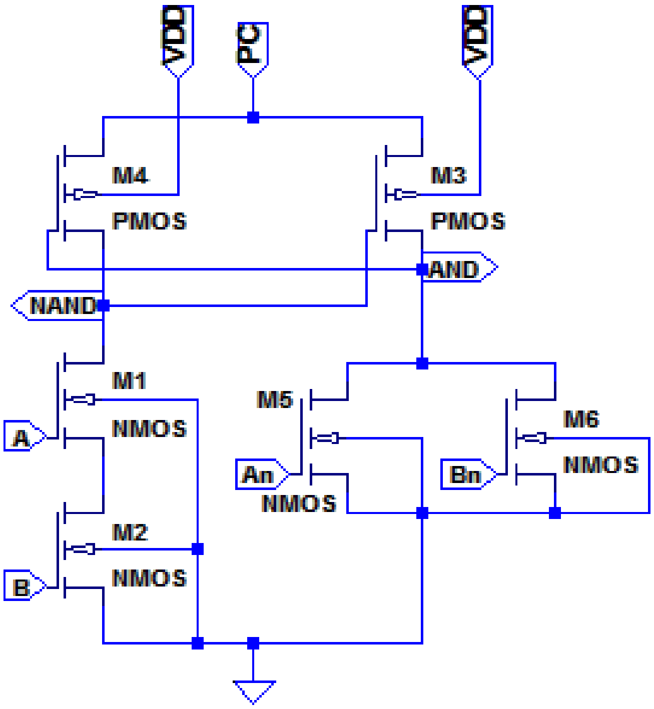


Figure 11: 2N2P AND/NAND Logic Gate – Schematic

As seen with the CMOS AND gate design, the logic for an AND gate is two NMOS transistors connected in series (receiving true inputs), and the logic for a NAND gate is two NMOS transistors in parallel (receiving complementary inputs). Further, the core of the 2N2P AND/NAND gate is similar to the 2N2P buffer/inverter circuit. The only difference between the two is the usage of the said output logic blocks for AND/NAND. The primary power source, VPC, is where the current trace will be taken from. V_{DD} is simply a DC power supply that generates the bias on the PMOS substrate connections. In order to verify this circuit's performance, Figures 12 and 13 contain the circuit's input and output waveforms.

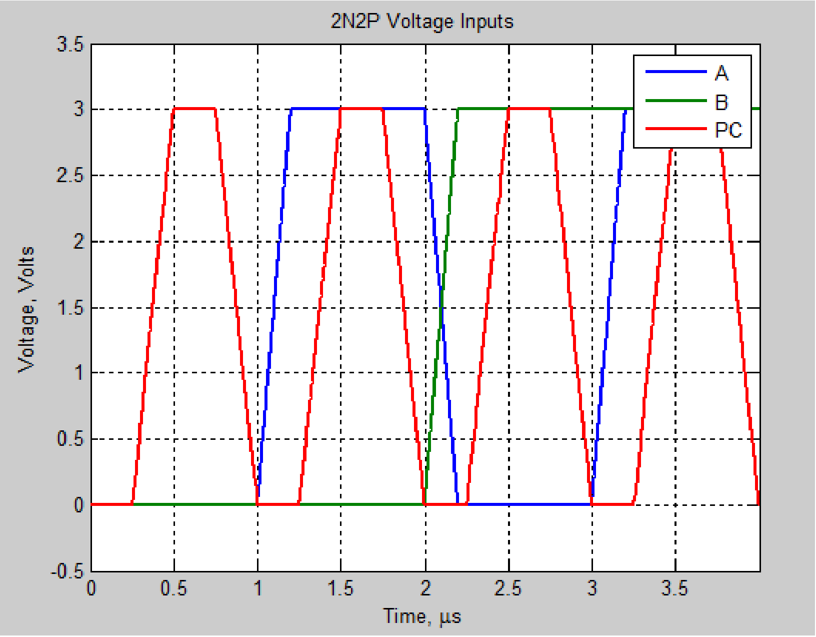


Figure 12: 2N2P AND/NAND Gate – Input Voltage Waveforms

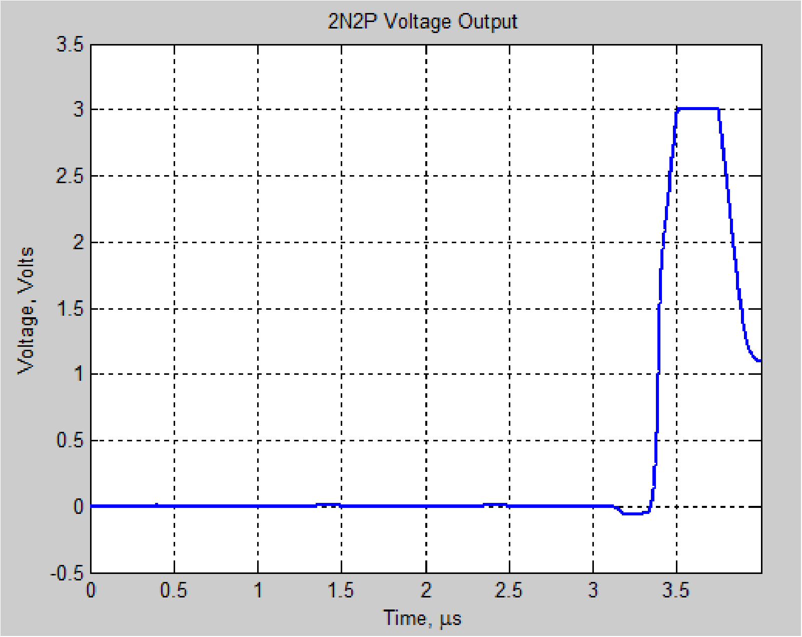


Figure 13: 2N2P AND/NAND Gate – Output Voltage Waveform

By visually inspecting these waveforms and comparing the inputs and outputs to those in the truth table, it can be determined that this circuit is behaving as expected. With this in mind, Figure 14 provides the current draw waveform that describes this circuit's performance.

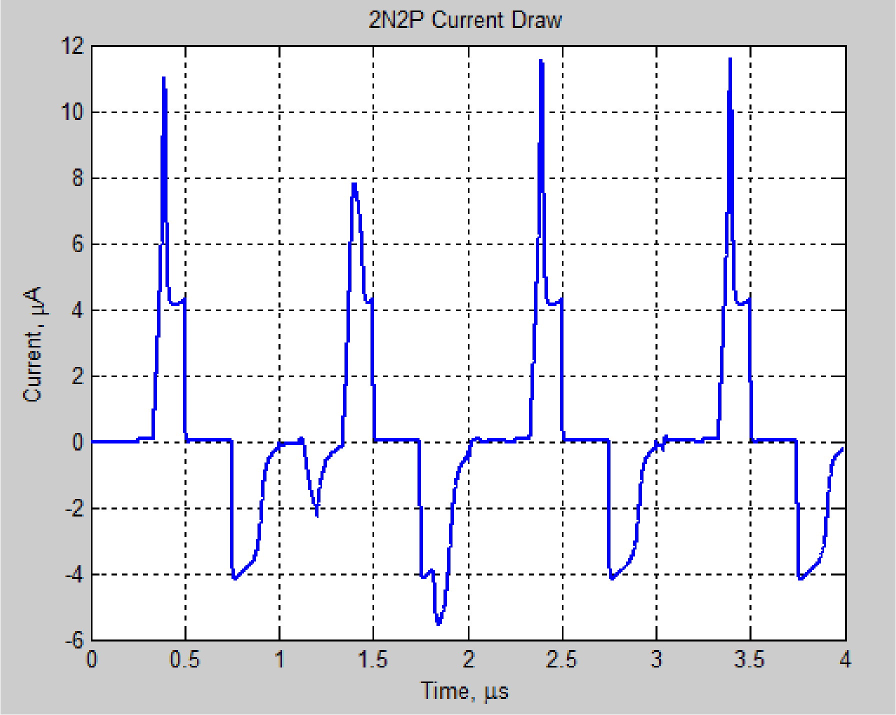


Figure 14: 2N2P AND/NAND Gate – Current Draw Waveform

From a quick visual inspection, there seems to be a lot more activity compared to the CMOS model. However, there are also moments in which the current draw is negative. This is the unused charge being drawn back into the power supply. Like the CMOS model, Figure 13 will be explored in depth using the testing methods provided in Chapter 3.

II.4.2 IDPAL LOGIC FAMILY

Unlike the CMOS or 2N2P logic families, IDPAL is still being actively researched and developed. [12] As such, there is very little information publicly available on this technology. With this in mind, a starting point for IDPAL logic gates would be ideal. Figure 15 illustrates an IDPAL buffer/inverter gate [13].

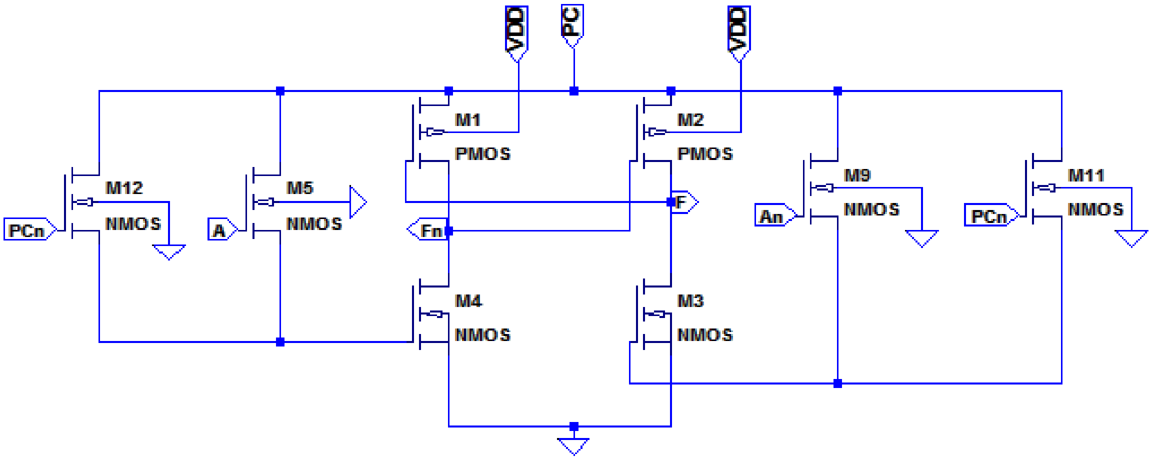


Figure 15: IDPAL Buffer/Inverter

The IDPAL buffer/inverter gate behaves as follows. The true input to this circuit is “A”, and its complementary input is “An”. In terms of the true input, the true output, “F”, is the buffer output. The complementary output, “Fn”, is the inverter output. Thus, “F” will produce a buffered version of “A”, and “Fn” produces an inverted version of “A”. The core of the circuit remains similar to that of the 2N2P model. The outside NMOS transistors are the switching networks that determine when the bottom core NMOS transistors link a particular output to ground. The additional NMOS transistor on the outside of the switching networks is called a draining MOSFET. In past research and computer simulations, it was determined that, if the inputs changed too abruptly, an electrical charge would become trapped at the gate of the bottom core NMOS transistor. This would cause the circuit to malfunction, resulting in erroneous voltage and current outputs. These draining MOSFETs operate with a phase shifted version of the power clock, called PCn. It has all of the same timing and magnitude properties as PC, but it is shifted 180°. Using this as a basis for IDPAL circuits, the AND/NAND gate can be presented. This is illustrated in Figure 16 [14].

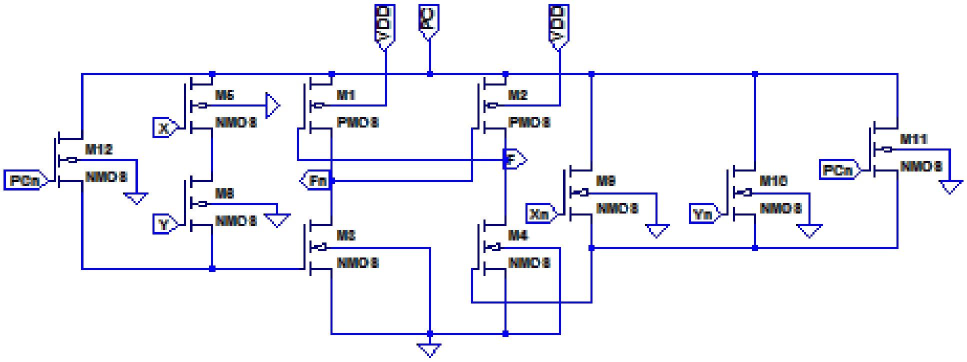


Figure 16: IDPAL AND/NAND Logic Gate – Schematic

This is similar to the buffer/inverter gate found in Figure 14. The difference is in the NMOS transistors that make up the switching networks. The switching network, utilizing the true inputs, is the NMOS AND configuration as presented previously. The other switching network, using complementary inputs and Equation 3, is the NMOS OR configuration. This, as previously shown, as derived using some Boolean algebra and DeMorgan's theorem. The substrate connections for the NMOS and PMOS transistors are all identical to that of the 2N2P model. In order to verify this circuit's performance, Figures 17 and 18 display the voltage inputs and the corresponding output.

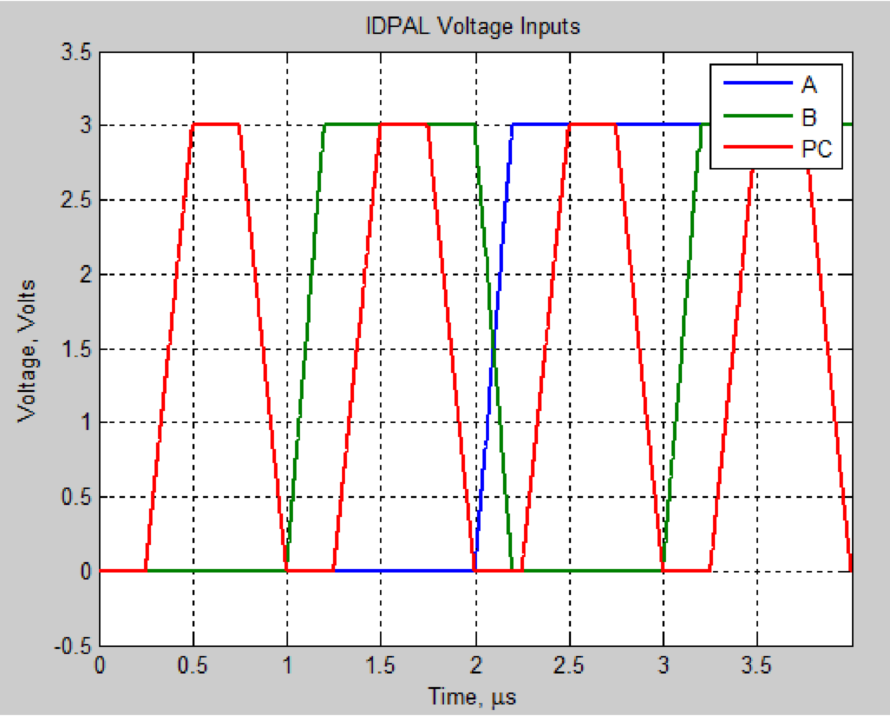


Figure 17: IDPAL AND/NAND Gate – Input Voltage Waveforms

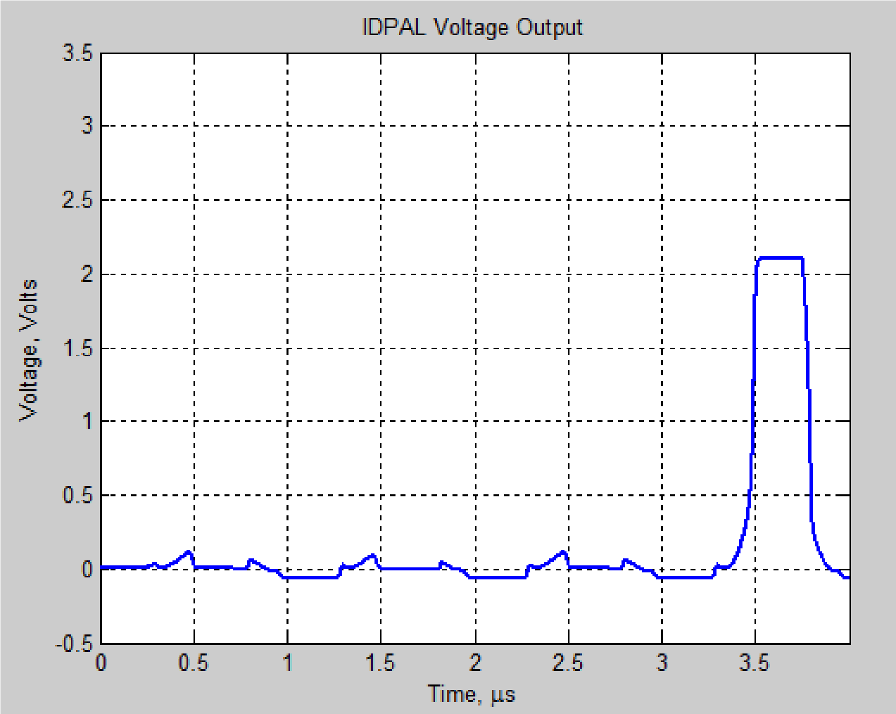


Figure 18: IDPAL AND/NAND Gate – Output Voltage Waveform

What is important to note is the power clock, which is identical to the one used in the 2N2P simulation. By slowly charging and discharging the circuit, the charge can be drawn off adiabatically. Paired with the draining NMOS transistors, the performance of the circuit will not be affected by the trapped charge issue. The other important feature of the input changes are when they occur. Inputs A and B only change values when the PC signal is 0.0V. This is to ensure the inputs have completely stabilized before the power clock begins to energize the circuit. With this in mind, the current analysis can be observed. Figure 19 shows the current trace for this IDPAL AND/NAND gate.

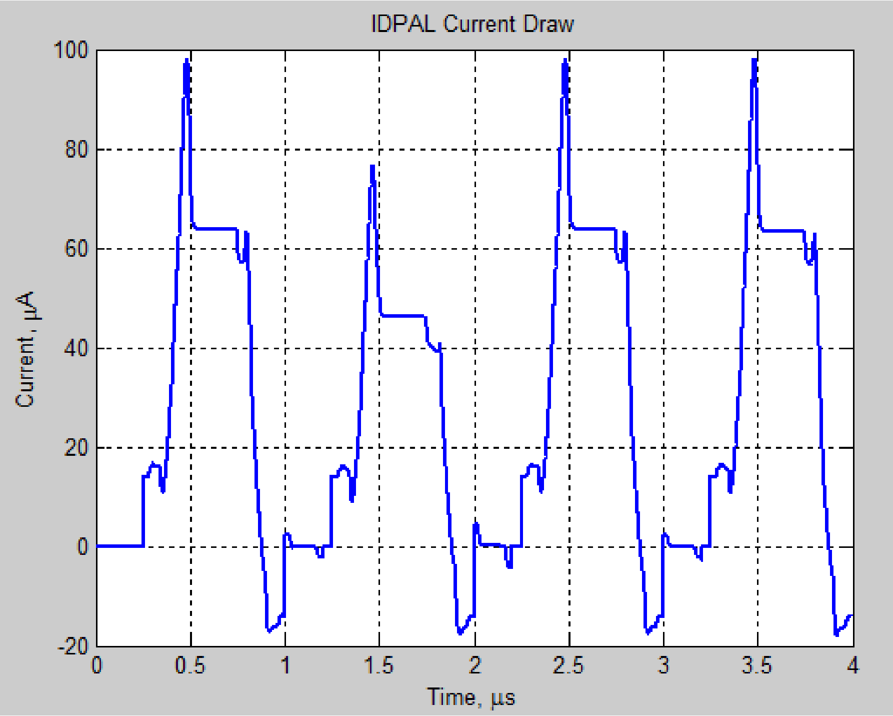


Figure 19: IDPAL AND/NAND Gate – Current Draw Waveform

II.5 SIDE-CHANNEL ATTACK TECHNIQUES

With the desire to increase a digital system's tolerance to intrusion attempts, it is important to consider what methods might be used against it. The general form of these attacks that will be considered are called side-channel or "side-band" attacks [1]. These attacks are designed in a straightforward manner, and they are quite effective against digital systems. In order to carry out an attack against a system, there must be a source of leaked information. Depending on the attack used, this information can take a variety of forms. The method that will be considered is called power analysis. This type of attacks uses leaked voltage and current information to infer the state of a digital system. In order to gain insight on how these attacks are structured and carried out, one such attack will be covered.

A report was published by a company called Cryptography Research, Inc. called "Differential Power Analysis". The purpose of their report was to implement power analysis attacks against an encryption scheme called Data Encryption Standard (DES). While DES and its components will not be covered in detail here, it can be summarized as a 16-round implementation of the Feistel Cipher. In order to determine information about this scheme, two power analysis techniques are used: simple power analysis (SPA) and differential power analysis (DPA). In both the SPA and DPA attacks, the digital system's current draw is observed while the encryption process is taking place. With knowledge of the encryption process and the system's current draw, it is possible to infer information about the present state of the encryption process. This is used to determine which specific operations are performed in each round and which are not. Having the insight on which the current draw waveform for each particular operation, the encryption key can be inferred from the current draw [5].

If this attack were to be carried out in reality, a combination of both hardware and software would be used. In order to do this, a resistor of low value (usually between 1-100Ω) is spliced into a series connection between the input voltage source and the target digital system. This resistor will show variances in the digital system's current draw as the system performs it functions. Each function will demand a specific amount of current, which is pulled from the input source. This current will pass through the spliced resistor. In order to observe this current draw, an oscilloscope is attached to the broken section of the circuit to close the loop. Figure 20 serves as an illustration of how the current draw measurement would be made in hardware.

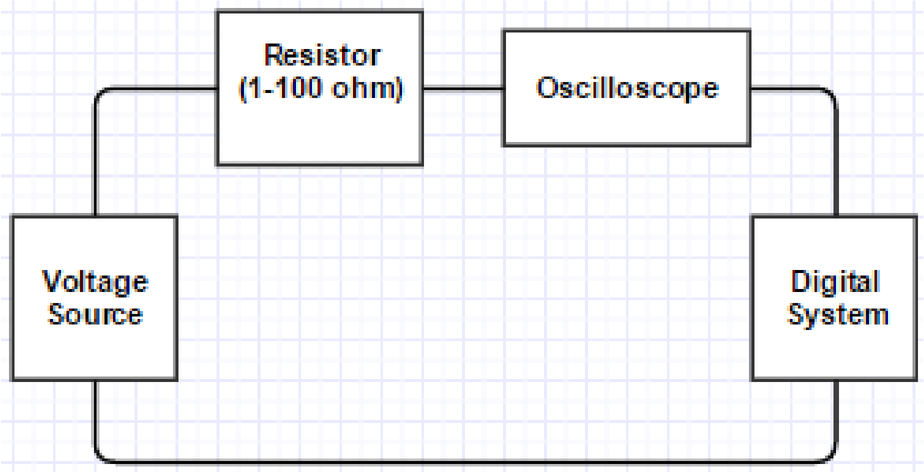


Figure 20: Current Draw Measurement in Hardware

While Cryptography Research, Inc.'s report was based on an actual encryption scheme, it is possible to implement these attacks on a variety of digital systems. The purpose of these side-channel attacks is to determine secret characteristics or inputs to the digital system. In the case of the DES encryption scheme that was analyzed, the encryption key was the target of the side-channel attack. This does not imply that the attack is limited to this target, as the leaked information might be used to determine additional information based on the targeted system. This might include additional encryption keys, plaintexts, or basic digital inputs.

The success rate of these attacks is dependent on numerous factors, such as the quality of the hardware and/or software used in the side-channel attack. If the oscilloscope in the Cryptography Research, Inc. report did not have a high enough level of precision to discern between two functions, it would more difficult to determine the encryption key. Thus, while side-channel attacks are designed to be quite powerful against digital systems, the intruder is limited by numerous factors, including his knowledge of the targeted system and the financial investment into cracking the system.

II.6 ENCRYPTION SCHEMES

In order to gain insight on how these technologies might be helpful, it is important to consider a useful method of protecting digital information: encryption schemes. There are many encryption techniques in existence, but they are implemented with the same fundamental digital logic components. A few examples will be considered. The first is found in the RC4 encryption scheme. Created by Ronald Rivest in 1987, this encryption scheme is used in wireless encryption protocols such as WEP and WPA [15]. In the first half of the encryption process (called the key-scheduling algorithm), modulo-256 adders are implemented. Because adders come in many configurations, it would be possible to implement Kogge-Stone adders (as highlighted in Chapter 4). The second half of the encryption process (called the pseudo-random generator algorithm) requires XOR gates and additional adders [16].

The second example comes from the advanced encryption scheme (AES). Based on the Rjindael algorithm, AES is a block cipher that serves as today’s standard in encryption. There are four fundamental functions that are repeatedly called during the encryption process [17]. In the first function (called AddRoundKey), a straightforward element-wise XOR between the input and key blocks takes place. The fourth function (called Mix Columns) works as a matrix multiplication between the current block and a fixed polynomial block. This can be defined with a network of digital logic gates [18].

With both the RC4 and AES encryption schemes, it is possible to implement the side-channel attacks previously discussed. This is because a hardware implementation of these encryption schemes would require fundamental logic gates and basic components, both of which will be examined in Chapters 3 and 4.

CHAPTER III

PROOF OF CONCEPT AND/NAND GATE

In the previous chapter, the background information for this work was presented. This included a review of CMOS technology, the presentation of the 2N2P and IDPAL logic families, a detailed overview of side-channel attacks, and two encryption schemes that this might benefit from these technologies. In this chapter, the analysis of the AND/NAND gate will be covered.

III.1 CHAPTER OVERVIEW

This section will serve as an overview for this chapter. Each of these sections will cover different implementations of the AND/NAND gate. The first section, followed by the appropriate subsections, is Section 2, which illustrates the two tests used to evaluate the different AND/NAND gate models. Sections 3 through 5 cover the CMOS, 2N2P, and IDPAL AND/NAND gate models, respectively. Section 6 directly compares the results found in the previous three analysis sections. Finally, Section 7 states the conclusions derived from the comparison of the technologies as found in Section 6.

III.2 SIDE-CHANNEL ATTACK RESISTANCE TESTS

In order to thoroughly examine these circuits for resistance to side-channel attacks, two tests will be utilized. The first is a simulated side-channel attack using power analysis techniques. The second is a modified version of Cutitaru’s energy analysis. Both tests are explained in detail in the following subsections.

III.2.1 CURRENT TRACE ANALYSIS

In order to verify an implementation’s resistance to side-channel attacks, a MATLAB tool was developed. The tool implements a side-channel attack in the form of a simple current analysis. To understand the concept behind both analysis techniques, the equations for electrical power and electrical energy are shown below as Equations 4 and 5, respectively.

$$P = VI \tag{4}$$

$$E = \int_{T_0}^{T_F} P dt \tag{5}$$

By directly inserting the right-hand side of Equation 4 into Equation 5, Equation 6 is obtained.

$$E = \int_{T_0}^{T_F} VIdt \tag{6}$$

This relation between the energy and current can be analyzed in a straightforward manner. Further, it is safe to assume knowledge of the source voltage (V_{DD} or PC), as the input current is assumed to be known in these simulations. The energy dissipated must be calculated in this manner due to the assumption that all output currents and voltages are obscured completely. This means that the energy dissipated will be computed for the entire digital system. With adiabatic logic, the objective is to reduce this value by recycling unused energy back into the power supply. Finally, it is assumed, for these attacks, that the intruder has knowledge on the digital system they are attempting to break into. With this concept shown, the design of the script file can be addressed.

By importing the current draw data from the spice models, the script file compares the current draw waveforms in various situations. In order to illustrate how this works, a flowchart can be found on Figure 21, followed by a discussion on the program's performance.

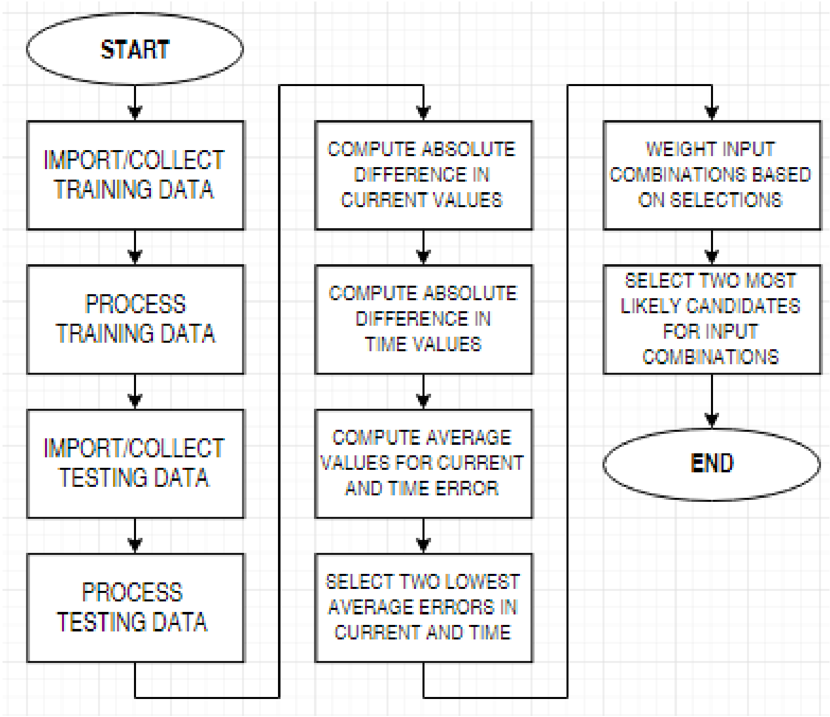

START
IMPORT/COLLECT TRAINING DATA
PROCESS TRAINING DATA
IMPORT/COLLECT TESTING DATA
PROCESS TESTING DATA
COMPUTE ABSOLUTE DIFFERENCE IN CURRENT VALUES
COMPUTE ABSOLUTE DIFFERENCE IN TIME VALUES
COMPUTE AVERAGE VALUES FOR CURRENT AND TIME ERROR
SELECT TWO LOWEST AVERAGE ERRORS IN CURRENT AND TIME
WEIGHT INPUT COMBINATIONS BASED ON SELECTIONS
SELECT TWO MOST LIKELY CANDIDATES FOR INPUT COMBINATIONS
END

Figure 21: Current Trace Analysis Flowchart

When called, the script file first imports the data that it will be evaluating: the training data set and the testing data set. The training data set is comprised of multiple current trace values. They are based on two separate stages for the digital system: charging and steady. When either the system has not been turned on or has no stored charge prior to the data collection, the system will be charged. When the appropriate charge exists in the circuit prior to data collection, the system will be steadier. In order to determine the total number of required training data sets, Equation 7 can be used:

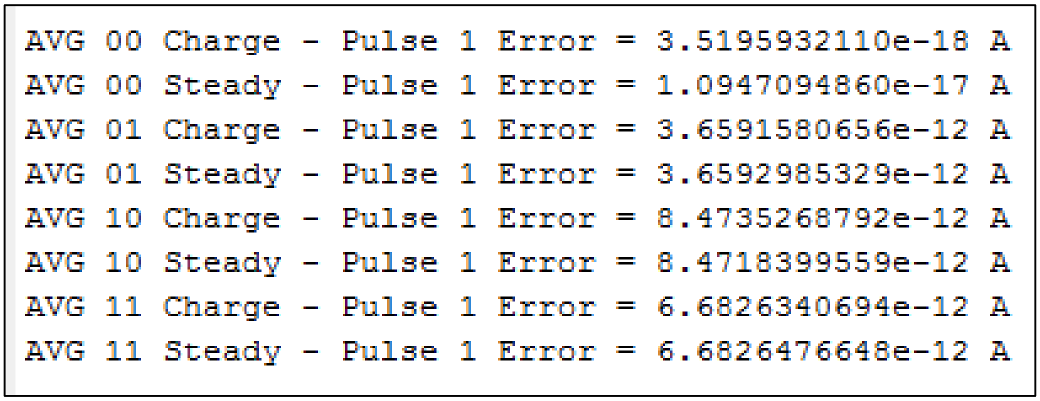
$$\#\, Training\ Data\ Sets = 2^{(\#\, true\ input\ bits+1)} \tag{7}$$

Because the circuit to be analyzed is has two true input bits, a total of 2^3 or 8 training data sets are required to enable the script file to predict input combinations. All training data sets were simulated with the same parameters to directly compare each model. Each simulation that produced the training data was run for 2.0µs: the first microsecond being the charging phase, and the second microsecond containing the steady phase. To process the data, each time and current series is split at 1.0µs and labeled for the input combination that produced them.

The testing data is then imported into the program. It is run in integer multiples of 1.0µs and split at each multiple. This is done to make each data series is exactly one microsecond in duration. It is separated by time period (called "pulses" in the script file), starting with the first pulse and up to the final pulse. In the simulations here, a maximum of four pulses were used. From here, the testing data is processed, which also reevaluates the training data to ensure all of the arrays holding the data sets are of the same length. If not, the data sets are linearly interpolated up to the size of the largest data set. Once this step has concluded, the training and testing data is fully processed and ready to be evaluated.

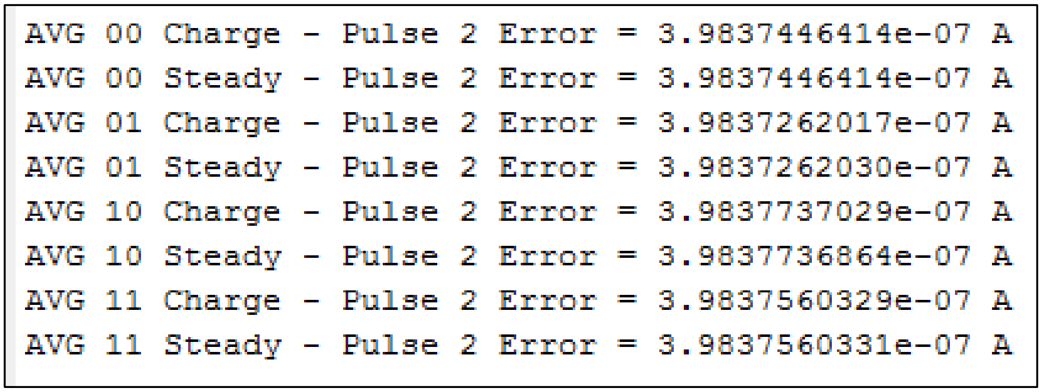
The next step in the procedure is to evaluate each testing period against all training data sets. With all of the training and testing arrays being the same length, a direct comparison is found by taking the absolute difference between each data point. The absolute difference is taken to avoid two errors, one positive and one negative, cancelling each other out. By doing this, these error values are stored in an array that is the same size as the training and testing data arrays. This comparison is performed for each possible input combination and state (charging or steady), and the process is repeated for each testing period. To further analyze the data, both the current and time arrays are broken down and measured in this manner. All of the current and time error arrays are then individually averaged in order to determine which has the lowest average absolute error. The two lowest values are selected in both current and time arrays. Based on which input combination produced the training set that was selected by this process, a collection of variables is used to store each input combination's "votes". These votes are assigned in the following manner: 51 for being selected in the current trace, and 50 for being selected in the timing trace. This dual approach allows for the current trace to be used as designed, but a timing trace can be used as a tiebreaker between options that appear to be equivalent in the absolute average current error. By using both current and timing errors, the proper input combination can be properly identified. The two input combinations with the most votes at the end of this process are selected as the guesses for the testing period's input combination.

A straightforward example of this program's performance will now be shown. The following sequence of inputs will be fed into a 2-input CMOS AND/NAND gate: 00, 01, 00, 01. As stated in the program description, each input will follow the period of 1.0μs. This means that each pulse lasts for that time duration. Each pulse is analyzed by each data point until an array of error values exists. As a result of this analysis, Figures 22 through 25 are derived from the resulting values displayed in MATLAB's command window.



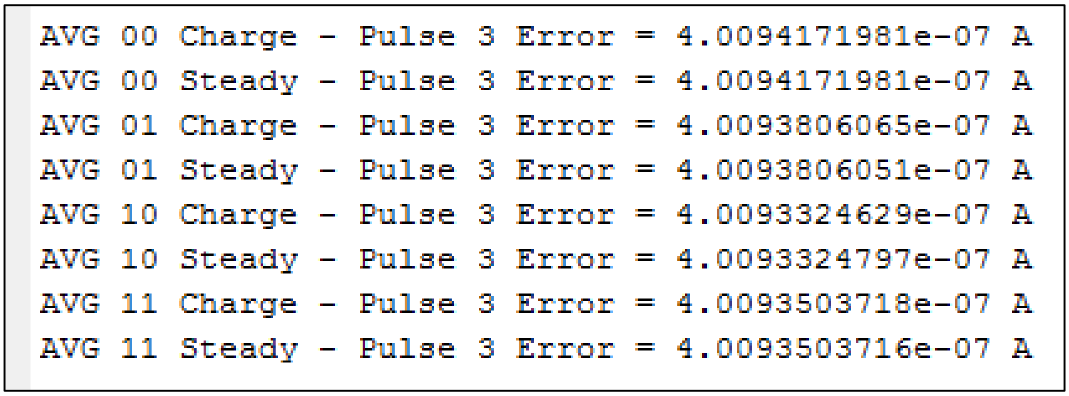
```
AVG 00 Charge - Pulse 1 Error = 3.5195932110e-18 A
AVG 00 Steady - Pulse 1 Error = 1.0947094860e-17 A
AVG 01 Charge - Pulse 1 Error = 3.6591580656e-12 A
AVG 01 Steady - Pulse 1 Error = 3.6592985329e-12 A
AVG 10 Charge - Pulse 1 Error = 8.4735268792e-12 A
AVG 10 Steady - Pulse 1 Error = 8.4718399559e-12 A
AVG 11 Charge - Pulse 1 Error = 6.6826340694e-12 A
AVG 11 Steady - Pulse 1 Error = 6.6826476648e-12 A
```

Figure 22: Current Trace Analysis Example – Pulse 1 Results



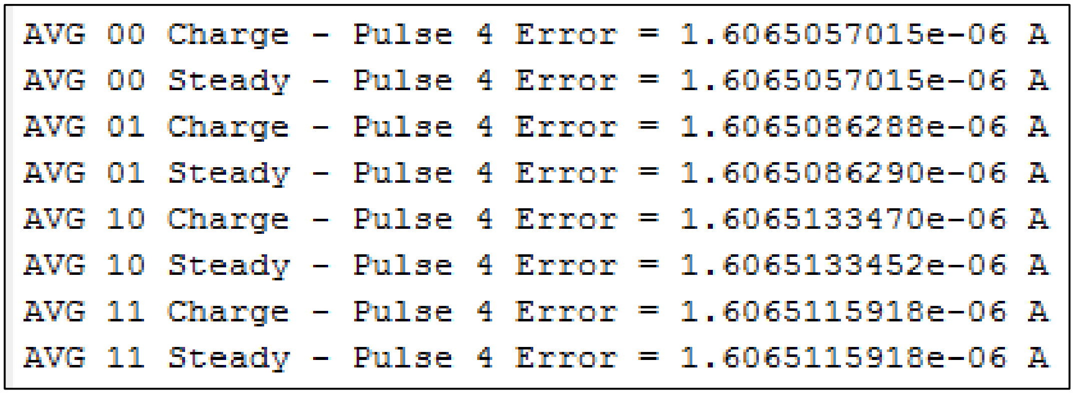
```
AVG 00 Charge - Pulse 2 Error = 3.9837446414e-07 A
AVG 00 Steady - Pulse 2 Error = 3.9837446414e-07 A
AVG 01 Charge - Pulse 2 Error = 3.9837262017e-07 A
AVG 01 Steady - Pulse 2 Error = 3.9837262030e-07 A
AVG 10 Charge - Pulse 2 Error = 3.9837737029e-07 A
AVG 10 Steady - Pulse 2 Error = 3.9837736864e-07 A
AVG 11 Charge - Pulse 2 Error = 3.9837560329e-07 A
AVG 11 Steady - Pulse 2 Error = 3.9837560331e-07 A
```

Figure 23: Current Trace Analysis Example – Pulse 2 Results



```
AVG 00 Charge - Pulse 3 Error = 4.0094171981e-07 A
AVG 00 Steady - Pulse 3 Error = 4.0094171981e-07 A
AVG 01 Charge - Pulse 3 Error = 4.0093806065e-07 A
AVG 01 Steady - Pulse 3 Error = 4.0093806051e-07 A
AVG 10 Charge - Pulse 3 Error = 4.0093324629e-07 A
AVG 10 Steady - Pulse 3 Error = 4.0093324797e-07 A
AVG 11 Charge - Pulse 3 Error = 4.0093503718e-07 A
AVG 11 Steady - Pulse 3 Error = 4.0093503716e-07 A
```

Figure 24: Current Trace Analysis Example – Pulse 3 Results



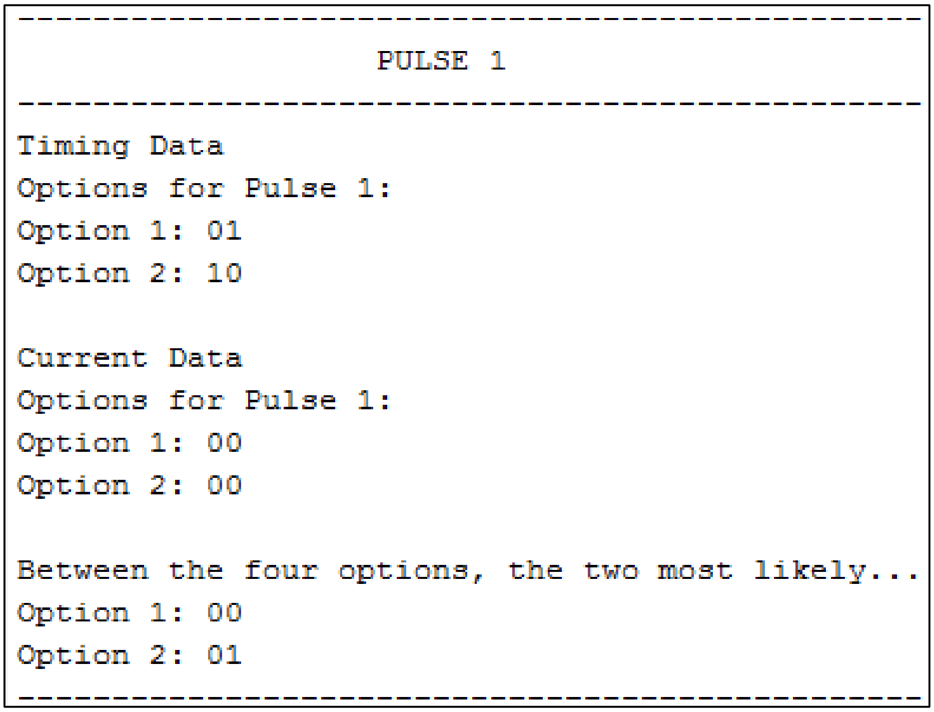
```
AVG 00 Charge - Pulse 4 Error = 1.6065057015e-06 A
AVG 00 Steady - Pulse 4 Error = 1.6065057015e-06 A
AVG 01 Charge - Pulse 4 Error = 1.6065086288e-06 A
AVG 01 Steady - Pulse 4 Error = 1.6065086290e-06 A
AVG 10 Charge - Pulse 4 Error = 1.6065133470e-06 A
AVG 10 Steady - Pulse 4 Error = 1.6065133452e-06 A
AVG 11 Charge - Pulse 4 Error = 1.6065115918e-06 A
AVG 11 Steady - Pulse 4 Error = 1.6065115918e-06 A
```

Figure 25: Current Trace Analysis Example – Pulse 4 Results

As stated previously, each input combination has two different stages: a charge state and a steady state. Each of these states are compared to each input pulse, which, as stated before, is the testing data broken into 1μs intervals. This is why there are two tests for each input combination. It is important to note that, since these are all separate training data sets, the program could identify the current of the same input combination twice. This is a unique property that was discovered in the CMOS traces, as it happened on a fairly regular basis. Each of the numerical values are the absolute average in the error trace. These are the values that are directly compared in order to select the input combination. The two lowest values are listed by the MATLAB program as candidates for the input combination.

The results of this analysis will be discussed, starting with the first pulse. The absolute average error values found in Pulse 1 strongly indicate the input combination to be ‘00’ as the first two choices. The options after that are much closer in comparison, being separated by roughly 2-3 pA each. While this is true, there is one characteristic of the digital system being measured: the circuit itself has no pre-stored charge in it. Thus, the first small duration of time (usually in the pico- to nanosecond range) is used to energize the system. This takes place for all training data samples as well as the testing data sets. As a result, the first pulse should be predicted correctly on a fairly accurate basis. This is not to say it will always predict the input combination correctly, as all programs have their limitations.

The remainder of the pulses, in terms of the absolute average error traces, are much closer in value to each other. All of their differences are within a few picoamps of each other, which makes it more difficult to determine whether a particular input combination is the correct choice. This, in addition, places a limitation on the hardware and software used in a side-channel attack implementation. In a real world application, the level of precision that some oscilloscope might have would not be able to discern between each of the error traces, making the guesses completely ambiguous. To illustrate how the program lists the selected choices, Figure 26 shows both the time and current error choices.



```
------------------------------------------------
                   PULSE 1
------------------------------------------------
Timing Data
Options for Pulse 1:
Option 1: 01
Option 2: 10

Current Data
Options for Pulse 1:
Option 1: 00
Option 2: 00

Between the four options, the two most likely...
Option 1: 00
Option 2: 01
------------------------------------------------
```

Figure 26: Current Trace Analysis Input Selection Example

III.2.2 Energy Analysis

The energy analysis performed in Dr. Cutitaru’s dissertation will be recalled and implemented as a secondary test for showing a technology’s resilience to side-channel attacks. There are four basic metrics that are computed initially: the minimum, the maximum, and the mean, and the standard deviation. From here, two additional statistics are computed to illustrate a system’s side-channel attack resiliency: the normalized standard deviation (NSD) and the normalized energy deviation (NED). Also known as the coefficient of variance (CV), the NSD is computed using the Equation 8.

$$NSD(\%) = \frac{\sigma}{\mu} \cdot 100\% \tag{8}$$

As it can be seen, the standard deviation of the provided data set, σ, is divided by the data set's mean value, μ. Multiplication by 100 produces a percentage. In terms of a system's resistance to side-channel attacks, this statistic should be minimized. If the standard deviation becomes significantly smaller than the average, the data set's variability decreases, making it more difficult to detect changes in the digital system's state. Otherwise, the variability will increase, making any state changes easier to detect [22]. The other statistic to be computed is the NED, which is found with Equation 9.

$$NED(\%) = \frac{E_{max} - E_{min}}{E_{max}} \cdot 100\% \tag{9}$$

The NED detects the range of energy consumption values and scales it according to the maximum energy consumption value. This is then multiplied by 100 to produce a percentage. As with the NSD, if the NED is minimized, the range of energy consumption values will become closer to each other. As with the NSD, this makes it more difficult to detect input combination changes and the state of the digital system [13, p.63]. In minimizing the NSD and NED, the intruder must utilize higher levels of precision, which may increase the time and financial investment in breaking the digital system. By directly inspecting these statistics in the energy analysis, each technology will be evaluated and compared to each other to determine which may be better at resisting side-channel attacks.

III.3 CMOS AND/NAND GATE ANALYSIS

In order to properly determine each model’s resistance to side-channel attacks, the following experimental setup will be used. This chapter serves as a small scale experiment (single logic gates). Each model will be tested using the two methods listed in the previous subsections. In this, and the following two, sections, these will be shown as 1) the current trace analysis, and 2) the energy analysis. In order for a model to show a stronger resistance to side-channel attacks with the first technique, the error value must increase. In the second testing method, the two desired statistics (NSD and NED) must be minimized, as they are directly related to the current draw. Minimizing these values will, theoretically, reduce the quality of leaked information, resulting in a higher error rate for side-channel attacks. In order to gather the appropriate data from LTspice®, the voltage and current of the power supply (V_{DD} for CMOS, PC for 2N2P and IDPAL) must be exported.

With the application of the two tests against CMOS, the input combination sequence is as follows: 00, 01, 10, 11. This input combination sequence produced the waveform found in Figure 7 in Chapter 2. This waveform is vital in determining what behavior is taking place in the circuit. This, again, assumes the power supply is the only source of leaked information in the digital system being broken into. In order to determine how resistant the circuit is against side-channel attacks, the simple current analysis and energy analysis tests were applied in an exhaustive manner. This was done to clearly show the behavior of the technology, which is being used as a reference point for the adiabatic technologies. As previously stated, the necessary training and testing data sets were prepared and imported into the MATLAB script. The results of the simple current analysis can be found in Table 9, as it is directly compared to the adiabatic technologies. For now, the percentages of accurate predictions (one guess and two guess) were 75.00% and 90.625%, respectively. These values serve as the benchmark for the adiabatic technologies, with lower prediction rates being desirable. In regards to the energy analysis, the same input combination sequence was used, and the energy statistics were computed in a straightforward manner. Table 2 contains the information from the energy analysis.

Table 2: CMOS AND/NAND Gate – Energy Analysis

Statistic	Min (J)	Max (J)	Mean (J)	Std Dev (J)	NED (%)	NSD (%)
Value	-6.47E-11	9.41E-10	1.55E-10	2.79E-10	4.28E-04	7.18E-04

These values, once again, will serve as the benchmark values for the adiabatic technologies. A direct comparison between these values and the ones found in the adiabatic technologies can be found in Section 6 of this chapter.

III.4 2N2P AND/NAND GATE ANALYSIS

Unlike the CMOS AND gate, there is a more consistent current draw present in this simulation. In order to determine how much information is leaking from this implementation, the exhaustive current trace and energy analysis was performed on this AND/NAND gate implementation using the current draw waveform in Figure 14. Starting with the current trace analysis, the following information was derived from it: the program would correctly guess the input combination (one guess, two guess) at a rate of 56.250% and 78.125%, respectively. Table 3 illustrates the 2N2P AND/NAND gate's performance as measured by the energy analysis.

Table 3: 2N2P AND/NAND Gate – Energy Analysis

Statistic	Min (J)	Max (J)	Mean (J)	Std Dev (J)	NED (%)	NSD (%)
Value	-4.73E-11	4.93E-11	9.17E-13	1.65E-11	7.84E-04	7.20E-04

While the discussion of how both test results is reserved for an upcoming section, one thing that would be important would be a direct comparison of the currents themselves. In Figure 27 below, the CMOS and 2N2P current waveforms are overlaid to show their differences.

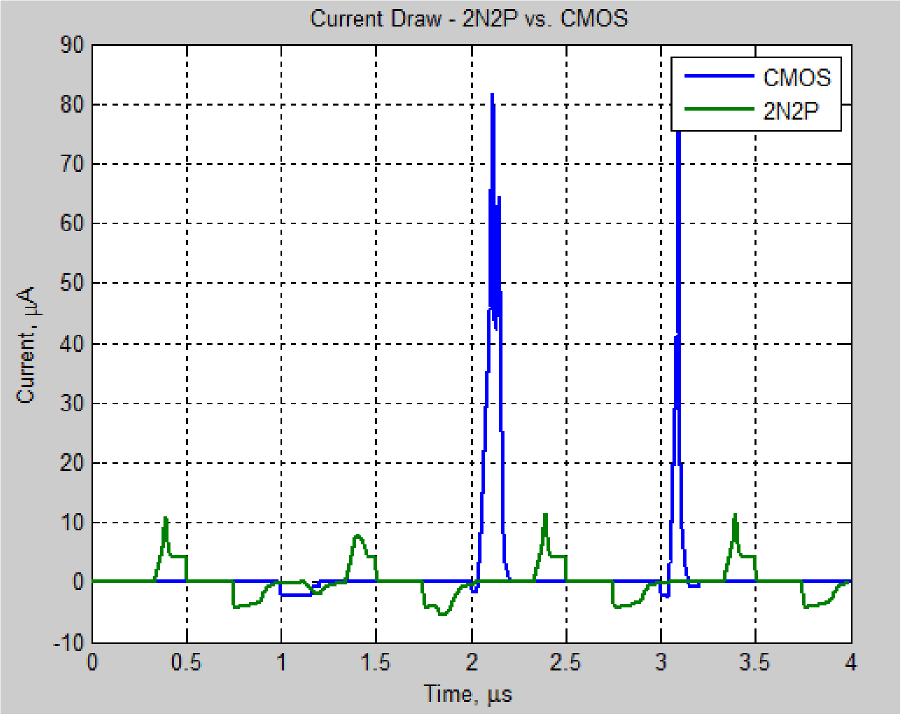


Figure 27: Current Waveforms – 2N2P vs. CMOS

This is done to illustrate the energy analysis results. By visual inspection, it is easy to see that the 2N2P current waveform is significantly lower in magnitude than the CMOS model. Because, as seen in Equation 6, the current plays a prominent role in the energy consumption and the input voltages are similar, it would be fair to assume that the 2N2P model will consume less energy than the CMOS model. This also has a smaller standard deviation, providing an upgrade to side-channel attack resistance over CMOS. Again, these points will be further explored in Section 6 of this chapter, but it is important to note that 2N2P is helpful in this regard.

III.5 IDPAL AND/NAND GATE ANALYSIS

The current draw, much like the 2N2P model, appears to be more constant than the CMOS. If clocked properly, there are no spikes or drastic differences in the magnitude of the current draw waveform. In order to determine this implementation's resistance to side-channel attacks, the current trace analysis and the energy analysis were applied to the current draw waveform found in Figure 19. The results of the current trace analysis were quite promising. The script properly detected the correct input combination (one guess, two guess) at rates of 21.875% and 50.000%, respectively. This is significantly better than that of the CMOS model, which was predicting the correct input combinations at much higher rates. As for the energy analysis results, the results supplemented the theory that IDPAL provides a resistance to side-channel attacks. The resulting energy statistics for the IDPAL AND/NAND gate can be found in Table 4.

Table 4: IDPAL AND/NAND Gate – Energy Analysis

Statistic	Min (J)	Max (J)	Mean (J)	Std Dev (J)	NED (%)	NSD (%)
Value	-5.97E-11	7.63E-10	1.89E-10	2.05E-10	4.31E-04	4.36E-04

As with the 2N2P model, it is possible to directly infer information from the current draw waveforms themselves. In Figure 28 shows the direct comparison between the CMOS and IDPAL current draw waveforms.

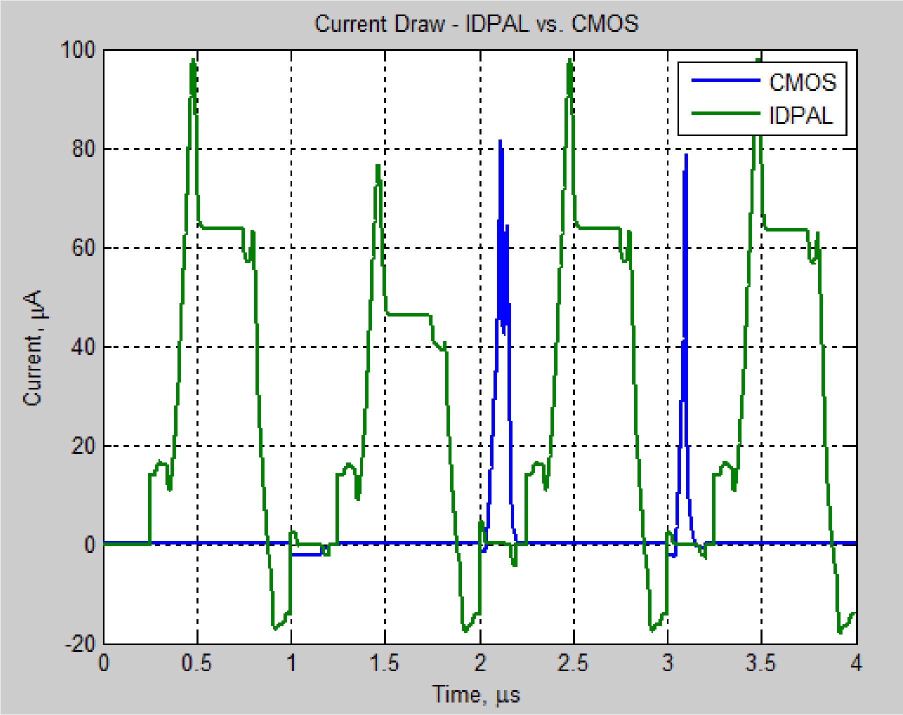


Figure 28: Current Draw Waveforms – CMOS vs. IDPAL

While the IDPAL current draw shows that unused energy is being fed back into the power supply (shown as negative current values), the magnitude of the current draw is directly comparable to that of the CMOS current draw waveform. This does not agree with previous research. There is one key characteristic that IDPAL appears to deal with better than CMOS. Upon inspecting Figure 28, the current draw in the CMOS model shows significant spikes when the output has to be charged. This is not the case in the IDPAL model, as the current drawn during each clock cycle is very close in magnitude to each other. A full breakdown of each model's current trace and energy analysis results can be found in the next section.

III.6 COMPARISON OF AND/NAND GATE MODELS

As seen in the previous sections, there were two major tests used to determine the strength of cybersecurity in each design: 1) the current trace analysis, and 2) the energy analysis. The first to be addressed will be the current trace analysis. This was performed in an exhaustive manner, containing four main trials, one for each possible input combination in the circuit. Tables 5–8 illustrate the results of each trial, followed by an explanation of the tables.

Table 5: Current Trace Analysis – Trial 1

Trial 1 (00)			
Inputs	CMOS	2N2P	IDPAL
00	Yes	Yes	Yes
01	Yes	Yes	No
00	2nd Choice	No	2nd Choice
10	Yes	No	2nd Choice
10	Yes	Yes	2nd Choice
00	Yes	Yes	2nd Choice
11	Yes	No	No
00	No	Yes	Yes

Table 6: Current Trace Analysis – Trial 2

Trial 2 (01)			
Inputs	CMOS	2N2P	IDPAL
01	Yes	2nd Choice	No
00	No	No	Yes
01	2nd Choice	Yes	No
10	2nd Choice	Yes	2nd Choice
10	Yes	Yes	Yes
01	No	Yes	No
11	Yes	No	No
01	Yes	Yes	No

Table 7: Current Trace Analysis – Trial 3

Trial 3 (10)			
Inputs	CMOS	2N2P	IDPAL
10	Yes	Yes	2nd Choice
00	Yes	Yes	2nd Choice
10	No	No	Yes
01	Yes	Yes	No
01	Yes	2nd Choice	No
10	Yes	Yes	2nd Choice
11	Yes	2nd Choice	No
10	Yes	Yes	2nd Choice

Table 8: Current Trace Analysis – Trial 4

Trial 4 (11)			
Inputs	CMOS	2N2P	IDPAL
11	Yes	2nd Choice	No
00	No	Yes	Yes
11	Yes	No	No
01	Yes	Yes	No
01	Yes	2nd Choice	No
11	Yes	2nd Choice	No
10	Yes	Yes	Yes
11	Yes	2nd Choice	No

The options in each of the tables are listed as “Yes”, “2nd Choice”, and “No”. If the program detects the correct input combination on the first attempt, “Yes” is selected. If the second attempt produces the proper input combination, “2nd Choice” is chosen. If both of the options are incorrect, the program returns “No”. What is important with this analysis is that the input combinations proceed through every possible transition available. As an example, assume the input combination is ‘00’. From this set of inputs, it can go to one of four input combinations: ‘00’, ‘01’, ‘10’, and ‘11’. This follows for each input combination, meaning there is a minimum of sixteen possible transitions in a 2-input AND/NAND gate. By using four tests with one primary input combination, all of the necessary transitions could be observed. As a result of this analysis, Table 5 shows the performance of each design in the current analysis trace.

Table 9: AND/NAND Gate – Current Trace Analysis Results

Implementation	One Guess	Two Guesses	Error
CMOS	75.000%	90.625%	9.375%
2N2P	56.250%	78.125%	21.875%
IDPAL	21.875%	50.000%	50.000%

Recall that the MATLAB current trace program output the two most likely candidates for the input combination. In most cases, the difference between options were as little as 10^{-13}, which make the level of precision for this test quite high. If both input combination candidates were incorrect, the tracing test failed, and the appropriate entry is listed under the “Error” column. With this in mind, the table can be reviewed. The CMOS implementation forms the baseline for this analysis. Its inputs were determined within two guesses roughly 90% of the time. The 2N2P implementation improved upon that number, with its inputs being determined about 78% of the time. Finally, the IDPAL implementation saw a significant decrease in its traceability. With two guesses, its inputs were being correctly being determined at a rate of 50%. As a result of this analysis, the following order of security, from most secure to least, can be derived from these results: IDPAL, 2N2P, and CMOS.

The other test for each implementation was an energy analysis. The analysis assumes the simulation time of all designs are equal. If this was not the case, the power would be calculated, as it can be related to both energy and time as shown previously. Table 6 shows the energy values computed, followed by a discussion on the analysis itself.

Table 10: AND/NAND Gate – Energy Analysis Results

Implementation	CMOS	2N2P	IDPAL
Min (J)	-6.47E-11	-4.73E-11	-5.97E-11
Max (J)	9.41E-10	4.93E-11	7.63E-10
Mean (J)	1.55E-10	9.17E-13	1.89E-10
Std Dev (J)	2.79E-10	1.65E-11	2.05E-10
NED (%)	4.28E-04	7.84E-04	4.31E-04
NSD (%)	7.18E-04	7.20E-03	4.36E-04

In this analysis, the primary statistics that illustrate each implementation's performance are the NED and the NSD. As stated in a previous section, the number of data points is non-uniform in each simulation. The NED and NSD show variance and deviance in the energy consumed, which Cutitaru suggested shows the resilience to power analysis attacks.

In terms of energy savings, 2N2P is the best implementation. By following the mean energy consumed, it saves a significant amount as compared to CMOS. IDPAL, in its current form, consumes slightly more energy than CMOS (following the average energy). This is believed to be directly attributed to the modifications made to the design (addition of draining MOSFETs), but a thorough reanalysis will take place to determine the true cause. The energy consumption, while important to consider, is not the focal point of this research.

III.7 AND/NAND GATE CONCLUSIONS

As seen in the previous section, there were two tests to compare the performance of each implementation. With the current trace analysis, the IDPAL implementation had the highest error rate, signifying a more significant resistance to side-channel attacks. This was followed by 2N2P and CMOS, respectively. The other test was an energy analysis. In this analysis, IDPAL produced one of the two desired values. As a result of this small scale analysis, it would be fair to assume that IDPAL, overall, provides a resistance to side-channel attacks.

The previous conclusion is made knowing that this is a small scale experiment (one logic gate). This conclusion will look for confirmation with the analysis of a larger system (the Kogge-Stone adders in Chapter 4).

CHAPTER IV

DESIGN OF 4-BIT KOGGE-STONE ADDER

In the previous chapter, three separate implementations of the AND/NAND gate were analyzed to observe each model's resistance to side-channel attacks. This served as a small scale experiment. In this chapter, a larger scale experiment will be used to confirm the previous findings. This chapter uses 4-bit Kogge-Stone adders as the larger scale mode for analysis.

IV.1 CHAPTER OVERVIEW

This section serves as an overview for the chapter. To begin, Section 2 will cover the testing metrics used for the Kogge-Stone adder models. Sections 3 through 5 cover the CMOS, 2N2P, and IDPAL 4-bit Kogge-Stone adder models, respectively. In Section 6, each model is compared and related to the results found in the previous chapter to ensure continuity. Section 7 presents the conclusions derived from Section 6.

IV.2 POWER/ENERGY ANALYSIS TEST

In order to ensure the results in Chapter 3 were not erroneous, it is important to hold tests for the digital systems in this chapter. The energy analysis will be used again in this chapter, measuring the following statistics: minimum, maximum, mean, standard deviation, NSD, and NED. There is one drawback to this analysis, however. The simulation times of the CMOS, 2N2P, and IDPAL circuits are not equivalent. Because of this, the energy consumption is not directly comparable. To correct this, the power consumption and its statistics will be computed. This removes time from the equation, leaving the supply voltage and the system's current draw as the values that determine how one implementation performs against the others.

There will be one other difference in this chapter's analysis techniques. Unlike Chapter 3's experimental setup, only the power and energy analysis will be performed. As stated in the second chapter, scaling up the MATLAB simulation presents a challenge, as it would require a large number of training data sets. Recall Equation 7:

$$\#\, Training\ Data\ Sets = \ 2^{(\#\, true\ input\ bits+1)}. \tag{7}$$

Because there are nine true input bits (4 for input A, 4 for input B, and a carry in), a total of 2^{10} or 1,024 training data sets would be required to make this script file work as expected. This does not account for the storage of each training data set as well as adjusting their respective array lengths in MATLAB. In this chapter, the power and energy statistics will be utilized to show some characteristics regarding the resistance to side-channel attacks. According to the results of the previous chapter, IDPAL produced one of the two desired statistics while producing the 2nd lowest in the other statistic. Recall Cutitaru's initial claim with the NSD and NED. By minimizing these statistics, a side-channel attack is more difficult to carry out. Thus, the model that produces the lowest NSD and NED should, in theory, produce the best resistance to side-channel attacks. Thus, the results of Chapter 3 can be confirmed if the statistical values for the NSD and NED are lowest with the IDPAL 4-bit Kogge-Stone adder model.

IV.3 CMOS 4-BIT KOGGE-STONE ADDER ANALYSIS

Being the fastest known adder variation, the Kogge-Stone adder is a parallel prefix form carry look-ahead adder. They are made up of three groups of blocks. The first block type creates two signals: generate (G) and propagate (P). The second block type updates these two signals based on the previous bit values for them. The final block type solidifies the signals, allowing them to be read as outputs [23]. Figure 29 shows what a 4-bit Kogge-Stone adder looks like in a broad sense.

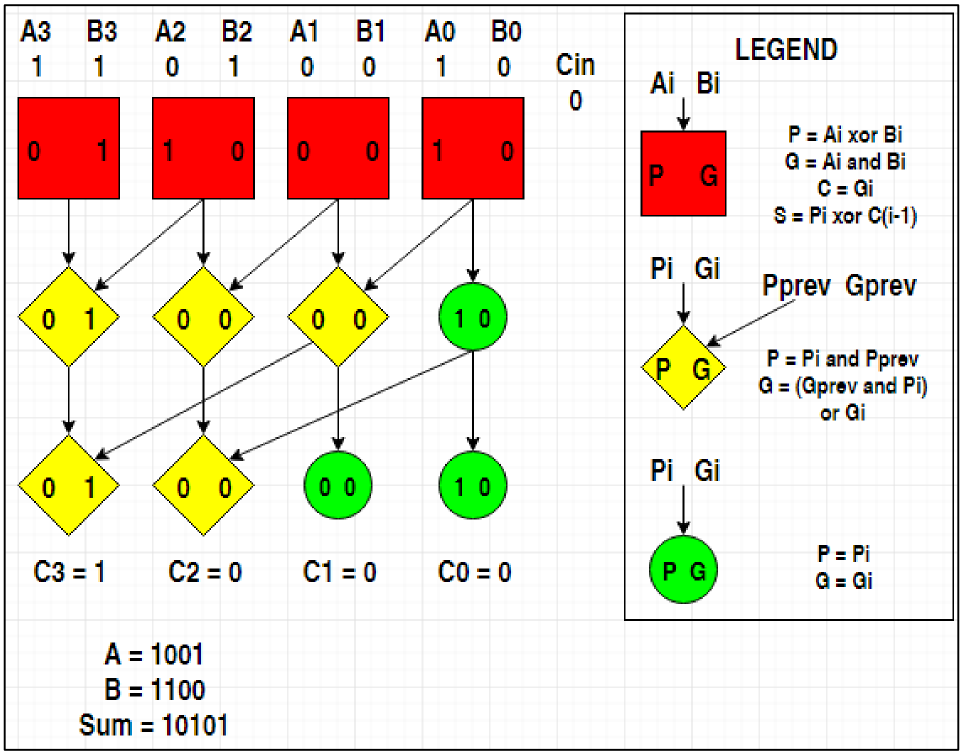


Figure 29: 4-Bit Kogge-Stone Adder Structure

The red block receives the inputs A and B, and it outputs P, G, C, and S. The yellow circle receives the current inputs (P and G) and the previous bit's inputs (P and G), and it outputs an updated set of signals (P and G). The green circle simply passed the current P and G signals further through the circuit. While this work only addresses 4-bit Kogge-Stone adders, this concept can be scaled to larger adder designs.

In order to proceed with this section of the work, a quick summary of the program used to generate the Kogge-Stone adder models would be appropriate. Known as VHSIC (Very High Speed Integrated Circuits) Hardware Description Language, VHDL is a program that allows for a digital system to be described at various levels of abstraction. In short, the program builds descriptions for hardware components in software. The program can display inputs and outputs both numerically (in a table) and graphically (as a waveform chart). Using advanced VHDL commands and programming techniques, it can also produce a netlist of the digital system. This netlist can be imported into LTspice® and simulated similarly to the LTspice® schematics [19].

In this chapter, there will be a couple of differences as to how the Kogge-Stone adders will be analyzed as compared to the AND/NAND gate implementations. As shown in Section 2 of this chapter, there will be no current trace analysis. Additionally, circuit schematics are not available for this chapter. As previously stated, these adder models were created in VHDL by Dr. Lee Belfore. His VHDL code produced a netlist model that LTSpice® can read and simulate. Because it is a netlist model, however, a circuit schematic was not present. The figure above illustrates the overall idea of the 4-bit Kogge-Stone adder, with the foundational components being logic gates.

The first of the three configurations to be implemented is CMOS. With the circuit's layout covered in the previous section, this section will cover the energy analysis of the system. The power supply's voltage and current draw will be observed in the same manner as the AND/NAND gate. In each of the LTspice® simulations, the simulation time obscures the model's behavior without zooming in. Further, the simulation time differs in each implementation due to the number of input signals required to operate each Kogge-Stone adder. To begin, Figure 30 shows the current draw waveform, followed by a zoomed in view in Figure 31.

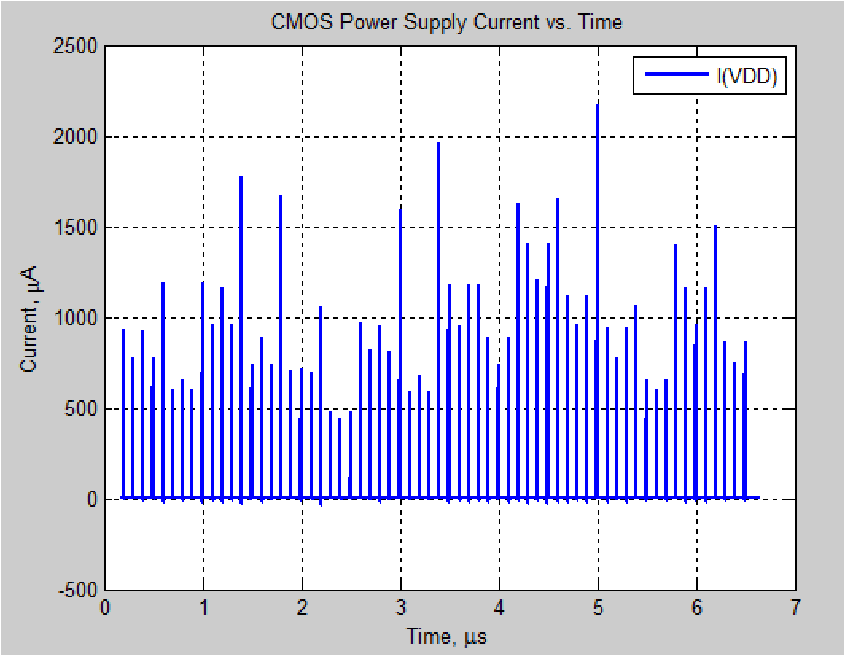


Figure 30: CMOS Kogge-Stone Adder Current Draw

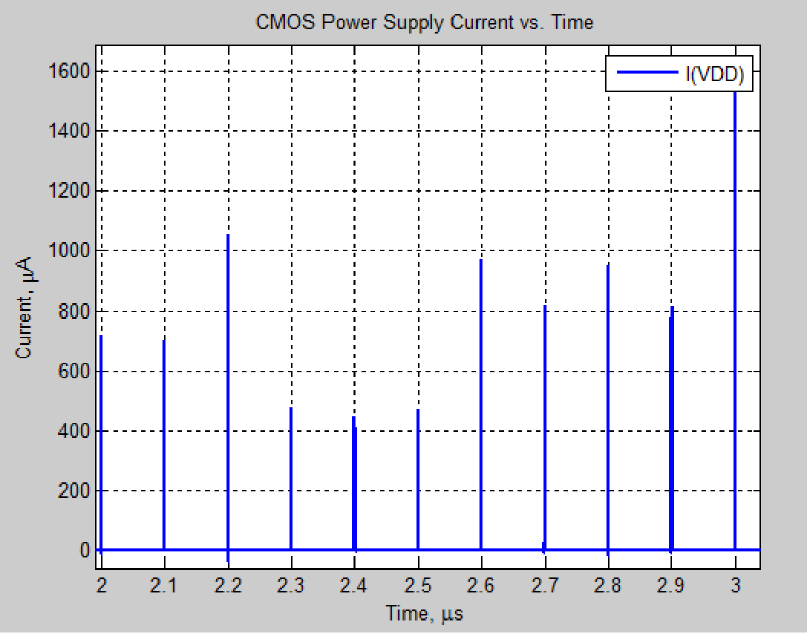


Figure 31: CMOS Kogge-Stone Adder Current Draw – Zoomed

In order to determine the benchmark for the Kogge-Stone adder models in terms of side-channel attack resistance, Tables 11 and 12 show the power analysis and energy analysis results.

Table 11: CMOS Kogge-Stone Adder Power Analysis

Statistic	Min (W)	Max (W)	Mean (W)	Std Dev (W)	NED (%)	NSD (%)
Value	-1.18E-04	6.50E-03	7.45E-04	1.19E-03	1.02E+02	1.59E+02

Table 12: CMOS Kogge-Stone Adder Energy Analysis

Statistic	Min (J)	Max (J)	Mean (J)	Std Dev (J)	NED (%)	NSD (%)
Value	-7.84E-10	4.32E-08	4.96E-09	7.89E-09	6.77E-04	1.06E-03

The values in both Tables 11 and 12 will serve as the baseline for the comparisons in this chapter. It is important to note that, although the energy values for each implementation are being included, the power analysis values are going to be relied upon, as they are not dependent on the simulation time.

IV.4 2N2P 4-BIT KOGGE-STONE ADDER ANALYSIS

The second 4-bit Kogge-Stone adder implementation will be performed with the 2N2P logic family. With the adiabatic implementations, there is a need for more than one power clock. In particular, the design of the adiabatic Kogge-Stone adder utilizes one power clock for each layer of logic. In this implementation as well as the IDPAL model, there are a total of four power clocks utilized. Some of the data is not clearly visible due to the magnitude of one or more of the power clock's current draws. Figures 32 and 33 display the 2N2P Kogge-Stone adder's current draw at its normal view and a zoomed perspective, respectively.

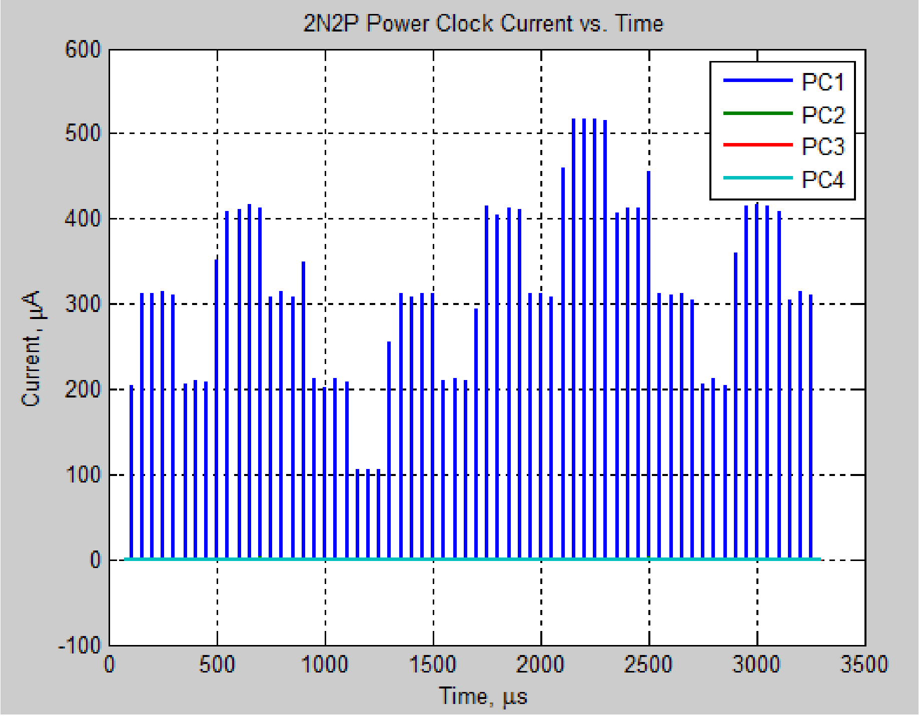


Figure 32: 2N2P Kogge-Stone Adder Current Draws

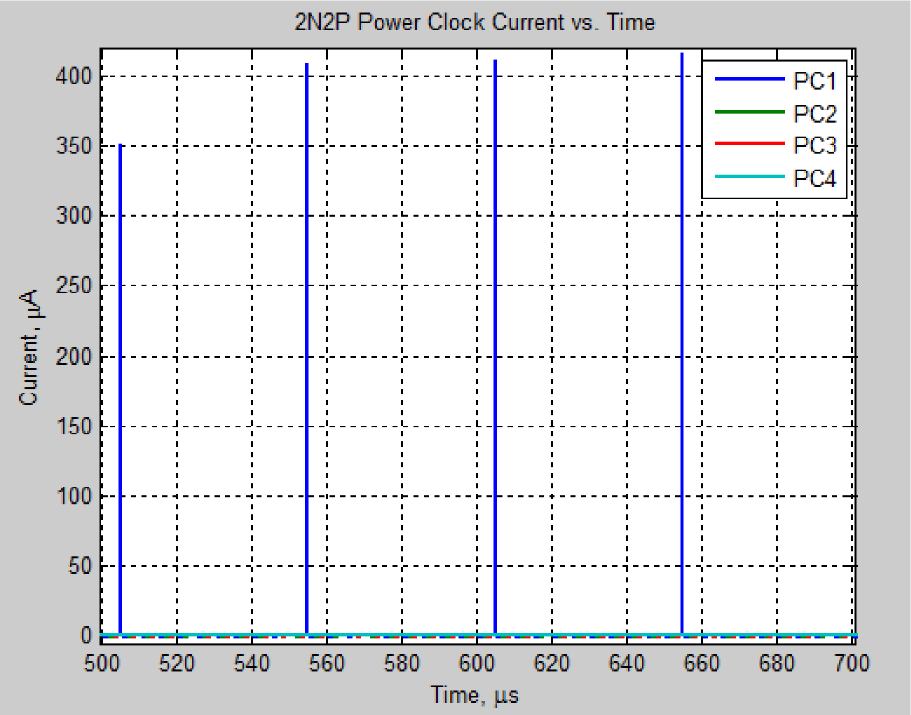


Figure 33: 2N2P Kogge-Stone Adder Current Draws – Zoomed

As with the CMOS model, the 2N2P Kogge-Stone adder was analyzed with the power and energy analysis tests. Tables 13 and 14 are the results of the respective tests.

Table 13: 2N2P Kogge-Stone Adder Power Analysis

Statistic	Min (W)	Max (W)	Mean (W)	Std Dev (W)	NED (%)	NSD (%)
Value	-8.13E-07	7.83E-04	9.99E-06	6.38E-05	1.00E+02	6.39E+02

Table 14: 2N2P Kogge-Stone Adder Energy Analysis

Statistic	Min (J)	Max (J)	Mean (J)	Std Dev (J)	NED (%)	NSD (%)
Value	-2.68E-09	2.58E-06	3.30E-08	2.10E-07	3.30E-01	2.11E+00

IV.5 IDPAL 4-BIT KOGGE-STONE ADDER ANALYSIS

The final implementation for the 4-bit Kogge-Stone adder will be done with the IDPAL logic family. In order to verify the results of the previous chapter, this model should show a similar value in standard deviation. It should also be clear that it has the lowest NSD and NED values. To begin analyzing this model, Figures 34 and 35 illustrate the current draw waveforms at normal and zoomed in perspectives, respectively. Tables 15 and 16 follow with the respective power and energy analysis results.

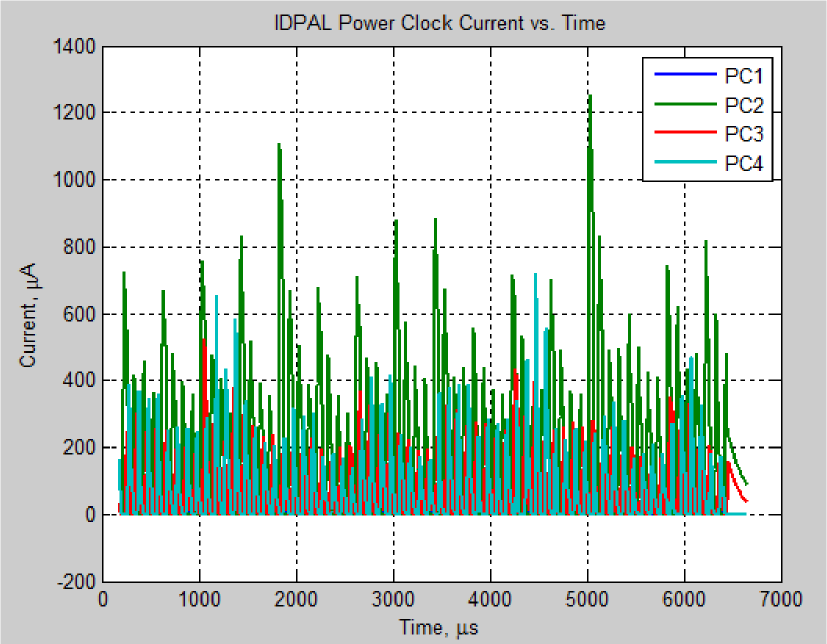


Figure 34: IDPAL Kogge-Stone Adder Current Draws

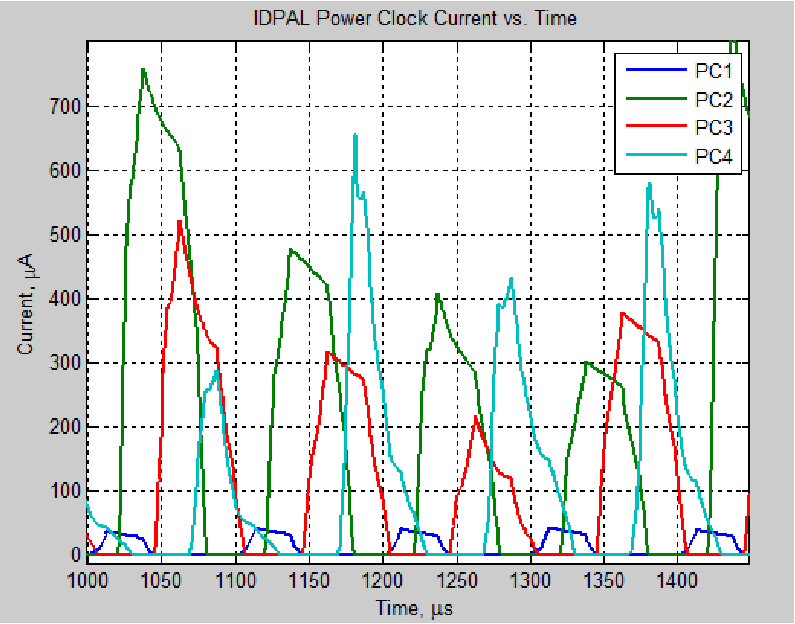


Figure 35: IDPAL Kogge-Stone Adder Current Draws – Zoomed

Table 15: IDPAL Kogge-Stone Adder Power Analysis

Statistic	Min (W)	Max (W)	Mean (W)	Std Dev (W)	NED (%)	NSD (%)
Value	2.65E-05	3.84E-03	9.97E-04	6.53E-04	9.93E+01	6.55E+01

Table 16: IDPAL Kogge-Stone Adder Energy Analysis

Statistic	Min (J)	Max (J)	Mean (J)	Std Dev (J)	NED (%)	NSD (%)
Value	1.75E-07	2.55E-05	6.63E-06	4.34E-06	6.60E-01	4.35E-01

IV.6 COMPARISON OF 4-BIT KOGGE-STONE ADDER MODELS

In order to properly interpret the results found in this chapter, the expectations should be known prior to this chapter’s comparison of results. In the previous chapter, it was concluded that the IDPAL model provided the most significant resistance to side-channel attacks, with both the current trace analysis and energy analysis reflecting that result. In the energy analysis, the IDPAL model provided the lowest value in one of the desired statistics (being NSD and NED). Therefore, the IDPAL would confirm the previous chapter’s results if it can produce at least one of the two lowest values for those metrics. If the IDPAL model produces both of the lowest values, this would further exaggerate that the IDPAL model is more resilient against side-channel attacks. With that in mind, Table 17 illustrates each model’s performance when compared to each other.

Table 17: Kogge-Stone Adder Power Analysis Results

Implementation	CMOS	2N2P	IDPAL
Min (W)	-1.18E-04	-8.13E-07	2.65E-05
Max (W)	6.50E-03	7.83E-04	3.84E-03
Mean (W)	7.45E-04	9.99E-06	9.97E-04
Std Dev (W)	1.19E-03	6.38E-05	6.53E-04
NED (%)	1.02E+02	1.00E+02	9.93E+01
NSD (%)	1.59E+02	6.39E+02	6.55E+01

Finally, for the sake of completion and comparing all of the collected data, Table 18 shows the energy analysis results for all Kogge-Stone adder implementations.

Table 18: Kogge-Stone Adder Energy Analysis Results

Implementation	CMOS	2N2P	IDPAL
Min (J)	-7.84E-10	-2.68E-09	1.75E-07
Max (J)	4.32E-08	2.58E-06	2.55E-05
Mean (J)	4.96E-09	3.30E-08	6.63E-06
Std Dev (J)	7.89E-09	2.10E-07	4.34E-06
NED (%)	6.77E-04	3.30E-01	6.60E-01
NSD (%)	1.06E-03	2.11E+00	4.35E-01

Once again, it is important to note that, for the energy analysis, the simulation times were all different (6.65μs for CMOS, 3.3ms for 2N2P, and 6.6ms for IDPAL). Thus, Table 18 is merely shown for the sake of completion. Recall that each of the simulation times for each model in Chapter 3 were all run for the same amount of time. Because power can be calculated without directly utilizing time, this will more accurately illustrate the performance of one Kogge-Stone adder model against another.

By inspection of Table 17, it can be seen that IDPAL produced the lowest values in both the NSD and NED. As stated previously, only one of the two values was lowest with IDPAL in the small scale experiment. This means that the results derived from analyzing the AND/NAND gate is confirmed, as IDPAL seems to display a significant resistance to side-channel attacks. As with the small scale experiments, the average power consumption is larger than expected, but this will be rectified in the future. It is also important to note that the 2N2P and IDPAL models incorporate a total of four power clocks in order to operate. All four of these power clocks are factored into the power and energy analysis calculations, while the CMOS model only has one voltage supply that is factored in. At this time, it is unclear how significant the impact of each power clock on both models (2N2P and IDPAL) is.

IV.7 4-BIT KOGGE-STONE ADDER CONCLUSIONS

In this chapter, three 4-bit Kogge-Stone adder models were simulated and run through a power and energy analysis. IDPAL produced the lowest values for the NSD and NED, which were the desired statistics. Because of this, the results of the small scale experiment (single

AND/NAND gate) were confirmed; IDPAL seems to provide a more significant resistance to side-channel attacks than CMOS and 2N2P.

CHAPTER V

FUTURE WORK

IDPAL has shown promise in proving a resistance to side-channel attacks. In light of this, there are a variety of possible future research possibilities for this technology. The first is in the technique used to collect current draw information to predict logic gate inputs. It is entirely possible that this technique could be used in larger designs, such as the Kogge-Stone adders used in this work. The second is in the energy consumption of IDPAL. In the future, the design might be further developed to save energy. The third and fourth relate to the scale of implementation (i.e., encryption scheme) and the cryptographic attack (i.e., differential power analysis). The fifth and final is in the adiabatic logic families used as comparison. By comparing IDPAL to more logic families, it would further show IDPAL's performance against other logic families.

Utilizing the current draw of a digital system to determine the input values is a primary contribution for this work. In the future, this will be researched at a deeper level to determine if this scales to larger designs, such as the Kogge-Stone adders used in this work. Assuming it does scale appropriately, the next step would be to increase the scale of the circuit observed in order to determine if this technique can be utilized at larger scales.

The research and development of IDPAL was one of the focal points of this work. In performing the current trace and energy analyses, it became apparent that this logic family may not be fully optimized in terms of energy consumed. Further, the energy consumption of IDPAL does not agree with previous work. The following comments were made throughout those chapters: the IDPAL model had seen structural changes between previous work and this work, the power/energy consumption of the IDPAL model did not match previous work, and the number of power supplies required to power the Kogge-Stone adder models are not identical. These are all things that are to be explored in depth in the near future.

In addition, there were other concepts and testing methods that were discussed. One of the initial hopes would have been to design a digital encryption scheme for each technology covered in this work. If that could have been achieved, an intrusion attempt of higher complexity, most likely being a differential power analysis (DPA) attack, would have been designed and implemented against the encryption scheme implementations. That would have been the ultimate test for each technology, as it would more accurately emulate a real world application.

Although the number of files required to perform each simulation and analysis in this work was not excessive, their quality is something that should be preserved and built upon. If there was more time, a full-scale current trace analysis would have been present. This would have allowed simulations, regardless of their size, to be fully analyzed with MATLAB. Alternatively, more exhaustive energy calculations and analyses would have been explored.

On a larger scale, adiabatic logic circuits might be further explored for their application in secure computing. The analyses performed in this work has shown that there is plausibility to this claim, as both 2N2P and IDPAL performed well against today’s standard technology. There are other adiabatic logic families that could be analyzed in this manner.

CHAPTER VI

CONCLUSIONS

This work has laid out a great deal of information about the performance of all three logic families. The first technical chapter illustrated their performance in direct comparison to each other. In that chapter, a proof of concept AND/NAND gate was built using CMOS, 2N2P, and IDPAL. By utilizing the designed MATLAB tool, two tests were implemented: a simulated side-channel attack and an energy analysis. In the side-channel attack, the higher error meant a higher resistance to the attack. IDPAL produced the largest error at 50.000%, which was significantly better than CMOS and 2N2P. In the energy analysis, the NSD and NED were the two directly observed statistics. The lower these values, the harder it would be to carry out a successful side-channel attack. IDPAL produced one of these two values in this small scale experiment. The initial conclusion from this is that IDPAL provides a meaningful resistance to side-channel attacks.

The final chapter with circuit implementations covered the design and operation of 4-bit Kogge-Stone adders. This served as the larger scale experiment. The energy analysis from the previous chapter was reused and modified to produce power values for each statistic. Using these power values to confirm the previous chapter's initial conclusion, IDPAL produced both the lowest NSD and NED values. Thus, the results found in this section agree with the small scale AND/NAND gate experiment. This means that IDPAL is producing a significant resistance to side-channel attacks. It should be noted that the current draw magnitude for IDPAL does not match expected values, but this will be revisited in the near future.

As seen in the work, each logic family can be utilized in secure computing applications, all with varying degrees of success. IDPAL shows the most promise, as each experiment showed the improvement in side-channel attack resistance. Because IDPAL is still being actively researched and developed, this is a good sign for this technology. While this is the case for IDPAL, the other technologies covered in this work are not bad alternatives. 2N2P also improved upon the values that CMOS produced. Further, it was shown to save a significant amount of energy over CMOS, which is desirable in specific applications. It saved more energy than IDPAL as well, but that, again, will be revisited in the near future. Ultimately, it would depend on the specific application. If low energy consumption is the primary focus, 2N2P would be a good choice at this time. If protecting against side-channel attacks and reducing leaked information is the primary goal of the design, IDPAL would be a logical choice.

REFERENCES

[1] University of Cincinnati. (N/A). “Introduction to Side Channel Attacks”. Available: http://gauss.ececs.uc.edu/Courses/c653/lectures/SideC/intro.pdf

[2] Z. Martinasek, V. Clupek, T. Krisztina. “General Scheme of Differential Power Analysis”. *36th International Conference on Telecommunications and Signal Processing (TSP)*, p. 1, (2013).

[3] P. Kocher, J. Jaffe, B. Jun. (1998). “Introduction to Differential Power Analysis and Related Attacks”. Available: http://42xtjqm0qj0382ac91ye9exr.wpengine.netdna-cdn.com/wp-content/uploads/2015/08/DPATechInfo.pdf, p. 3.

[4] National Institution of Standards and Technology. (N/A) “Side-Channel Attacks: Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing”. Available: http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-3/physec/papers/physecpaper19.pdf, p. 15.

[5] P. Kocher, J. Jaffe, B. Jun. (N/A). “Differential Power Analysis”. Available: https://www.cis.upenn.edu/~nadiah/courses/cis800-02-f13/readings/kocher-jaffe-jun.pdf

[6] A. Sedra and K. Smith. “MOS Field-Effect Transistors (MOSFETs)” in *Microelectronic Circuits*, 6th ed. New York, NY: Oxford, 2010, ch. 5, sec. 0, pp. 231-232.

[7] A. Sedra et al. “CMOS Digital Logic Circuits” in *Microelectronic Circuits*, 6th ed. New York, NY: Oxford: 2010, ch. 13, sec. 4, pp. 1111-1115.

[8] A. Sedra et al. “MOS Field-Effect Transistors (MOSFETs)” in Microelectronic Circuits, 6th ed. New York, NY: Oxford, 2010, ch. 5, sec. 1, pp. 235-243.

[9] Fairchild Semiconductor. (1983). “CMOS, the Ideal Logic Family” Available: https://www.fairchildsemi.com/application-notes/AN/AN-77.pdf

[10] Georgia State University. (2012). “Adiabatic Processes”. Available: http://hyperphysics.phy-astr.gsu.edu/hbase/thermo/adiab.html

[11] University of California, Los Angeles. (N/A). “Adiabatic Logic”. Available: http://nanocad.ee.ucla.edu/pub/Main/SnippetTutorial/AdiabaticLogic.pdf

[12] 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*, p. 3, 2013.

[13] M. Cutitaru, L. A. Belfore, II. “IDPAL – A Partially-Adiabatic Energy-Efficient Logic Family: Theory and Applications to Secure Computing”. Ph.D. dissertation. Dept. Elect. and Comp. Eng., Old Dominion University, Norfolk, VA, 2014, pp. 35-37, 63.

[14] M. Cutitaru, L.A. Belfore, II. “New Single Phase Adiabatic Logic Family”. *Proc. IEEE Intl Midwest Symp. Circuits and Systems*, p. 5.

[15] R. Rise, S. Cho, D. Kaylor. (N/A). “RC4 Encryption”. Available: https://www.math.washington.edu/~nichifor/310_2008_Spring/Pres_RC4%20Encryption.pdf

[16] Purdue University. (2005). “The RC4 Stream Cipher”. Available: https://www.cs.purdue.edu/homes/ninghui/courses/Fall05/lectures/355_Fall05_lect13.pdf

[17] National Institute of Standards and Technology. (2001). “Announcing the Advanced Encryption Standard (AES)”. Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[18] A. Kak. (2016). “Lecture 8: AES: The Advanced Encryption Standard”. Available: https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture8.pdf

[19] C. Roth and L. John. “Computed-Aided Design” in *Digital Systems Design Using VHDL*, 2nd ed. Stamford, CT: Cengage Learning: 2008, ch. 2, sec. 1, pp. 52-53.

[20] L. G. Heller, W. R. Griffin, J. W. Davis, N. G. Thome. “Cascode Voltage Switch Logic: A Differential CMOS Logic Family”. *1984 Digest of Technical Papers, IEEE International Solid-State Circuits Conference*. (1984).

[21] L. Belfore, private communication, April 2016.

[22] H. Abdi. (2010). “Coefficient of Variation”. Available: https://www.utdallas.edu/~herve/abdi-cv2010-pretty.pdf

[23] Stanford University. (2006). “Lecture 4 – Adders”. Available: http://web.stanford.edu/class/archive/ee/ee371/ee371.1066/lectures/lect_04.2up.pdf

VITA

Matthew Edward McAllister

Department of Electrical and Computer Engineering

Old Dominion University

Norfolk, VA 23529

EDUCATION

- B.S. Electrical Engineering, Old Dominion University, Norfolk, VA, December 2014.

AWARDS

- Dean’s List, Frank Batten College of Engineering and Technology, Old Dominion University, Norfolk, VA, Spring 2014.

EXPERIENCE

- Peer Educator Program Tutor, Old Dominion University, Norfolk, VA, October-December 2014.
- Teaching Assistant, Old Dominion University Department of Electrical and Computer Engineering, Norfolk, VA, January-May 2015.

COMPUTER PROFICIENCY

- C/C++, MATLAB, Arduino, VHDL
- Microsoft Office (Word, Excel, PowerPoint)