Air Force Institute of Technology
AFIT Scholar

Theses and Dissertations | Student Graduate Works

6-16-2011

Combinational Circuit Obfuscation through Power Signature Manipulation

Hyunchul Ko

Follow this and additional works at: https://scholar.afit.edu/etd

Part of the Programming Languages and Compilers Commons, and the Software Engineering Commons

Recommended Citation
Ko, Hyunchul, "Combinational Circuit Obfuscation through Power Signature Manipulation" (2011). *Theses and Dissertations*. 1402.
https://scholar.afit.edu/etd/1402

This Thesis is brought to you for free and open access by the Student Graduate Works at AFIT Scholar. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of AFIT Scholar. For more information, please contact richard.mansfield@afit.edu.

COMBINATIONAL CIRCUIT OBFUSCATION THROUGH POWER SIGNATURE MANIPULATION

THESIS

Hyunchul, Ko, Captain, ROKA

AFIT/GCS/ENG/11-05

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, the Department of Defense or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT/GCS/ENG/11-05

COMBINATIONAL CIRCUIT OBFUSCATION THROUGH POWER SIGNATURE MANIPULATION

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science

Hyunchul, Ko, B.S.C.S

Captain, ROKA

JUNE 2011

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT/GCS/ENG/11-05

COMBINATIONAL CIRCUIT OBFUSCATION THROUGH POWER SIGNATURE MANIPULATION

Hyunchul, Ko, B.S.C.S

Captain, ROKA

Approved:

/signed/	1 June, 2011
Yong C. Kim, Ph.D., Chairman	date
/signed/	1 June, 2011
Michael R. Grimaila, Ph.D., CISM, CISSP, Member	date
/signed/	1 June, 2011
Major Jeffrey M. Hemmes, Ph.D., Member	date

Abstract

Reverse engineering is a current threat to both the military and the commercial sector of system. One method for protecting the system against the threat of reverse engineering is obfuscation. The obfuscation could transform an internal logic of system into an equivalent one that is harder to reverse-engineer. In a recent research, Yasinsac and McDonald proposed the Random Program Model (RPM) which consists of randomly selecting sub-circuits from original circuits and replacing these sub-circuits with randomly selected, semantically equivalent new one for a circuit obfuscation and implemented such obfuscation techniques in Java, which is referred to Circuit Obfuscation via Randomization of Graphs Iteratively (CORGI) [6].

The previous obfuscation methods in CORGI are mainly focusing on hiding the structural and functional information on a circuit against white-box analysis. In other words, the final variants from the previous version of CORGI are likely to leak the significant information through the side-channel such as time, power consumption, and electro-magnetic emission. For this reason, this research primarily focuses on hiding the side-channel information rather than the internal structure of circuit. This is the first known work that focuses on generalized side-channel signature characterization and provides the tool to minimize the signature leakage.

In this effort, the proposed research is conducted in four steps, namely the circuit signature estimation, characterization, signature manipulation, and signature validation. During the signature estimation and characterization phase, the power signature of the circuit is estimated both statically and dynamically via probabilistic signature estimation, then verified with dynamic simulation with HSPICE. Once the signature is estimated by both static and dynamic techniques, the signature of the circuit is characterized and classified as one of the four predefined power signature. After characterization phase, a power signature manipulation method is applied to

alter its original power signature to a different class of power signature. Finally, the altered power signature is verified via both static and dynamic signature analysis. In addition, the proposed signature manipulation method is applied on RSA circuit on Xilinx Virtex5 FPGA against adversarial power analysis. Ultimately, this research expects that the new signature manipulation method in this research can give more burdens to adversarial to compromise critical systems.

Acknowledgements

First and foremost, I owe a large debt of gratitude to my wife for her encouragement and support over the last 20 months. I especially want to thank her for enduring the life away from our home country. You have been a magnificent part of my life and my academic achievements. I could not have done any of this without you.

I also owe a large debt of gratitude to my thesis advisor, Dr. Yong C. Kim, for his patience, availability and enthusiasm. I have become interested in topics and made decisions that I would never have. To the remainder of my committee, Maj. Jeffrey M. Hemmes and Dr. Michael R. Grimaila, thank you for your suggestions and advice. Lastly, I would like to express my appreciation to members of Program Encryption Group and VLSI Group who have shared many difficulties for this research. Especially, Daniel Koranek who helped me to overcome obstacles and get back on track more times than I can count. Thank You.

Hyunchul, Ko

Table of Contents

Page

Page

List of Figures

List of Tables

List of Abbreviations

Abbreviation		Page
RPM	Random Program Model	iv
PEG	Program Encryption Group	3
SID	Signature IDentification	4
SM	Signature Manipulator	4
TA	Timing analysis Attack	12
FA	Fault injection Attack	12
PA	Power analysis Attack	12
MOSFET	Metal Oxide Semiconductor Field Effect Transistor	14
SPA	Simple Power Analysis	16
DPA	Differential Power Analysis	16
SPICE	Simulation Program with Integrated Circuit Emphasis	26
PRBS	Pseudo Random Bit Generator Source	47

COMBINATIONAL CIRCUIT OBFUSCATION THROUGH POWER SIGNATURE MANIPULATION

I. Introduction

In our modern world, protecting one's own intellectual property is important. Likewise, because of increases in the importance of the intellectual property contained in software and hardware, protecting properties from adversarial actions is of great interest to both military and civilian communities. Traditionally the conceptual distinction between hardware and software has been apparently divided in that the hardware is usually referred to as logic circuit designs on the physical technology but the software is referred to as a set of codes or instructions [19]. With the increasing use of hardware description languages, however, circuits are increasingly implemented in a software-like manner.

In this context, there will soon cease to be a distinction between logic circuit designs for implementation in physical hardware and logic circuit designs implemented as software. In other words, any circuit can easily be mapped to a piece of software which computes the same functions. A general software, in turn, could be translated to logic circuits. Based on this assumption, rather than viewing the two as distinguished categories, hardware protection and software protection both will be regarded as a circuit protection in our research.

1.1 Motivation

Protecting a smart card is one of the good examples in the domain of this research. Such a smart card can be applied in many applications such as mobile communications, banking, and electronic signatures. Since 2008, the Dutch public transit system has used special plastic cards with an embedded smart chip that allows passengers to use all transportation throughout the entire country without having to

buy individual tickets [21]. But, those cards were successfully attacked by students and hackers. One of attackers, Karsten Nohl [15], released his approach of deducing functionality from circuit images and applying it to the internal circuitry of the transit card at the Black Hat 2008 security conference. Breaking the Mifare Classic chip using Nohl's approach makes it possible for anyone to use all transportation in Holland at no cost. As a result, the Dutch government invested about $2 billion in their new transit system to cover its weakness from adversarial reverse-engineering attacks.

Unlike Nohl's approach in the Dutch public transit system, there is another method for revealing the secrets of smart cards. This method have been proposed that use side-channel information such as timing measurements, power consumption, electromagnetic emissions and faulty hardware [22]. Of all the sources of side-channel information, power measurements are the most difficult to control [8]. Current technological constraints result in different power consumptions when manipulating a logical one compared to manipulating a logical zero [22]. An attacker of a smartcard can monitor such power differences and obtain useful side-channel information. Kocher et al. [8] claim one can monitor the actions of a single transistor within a smartcard using such a power analysis. In [8], the authors outline a specific power analysis attack against smartcards running the encryption algorithm.

From these lessons learned, it is motivated that circuit protections are extremely significant for preventing these types of attacks on both critical software and hardware. With these motivation, our research group has considered circuit protection for embedded systems and developed several methods to prevent from reverse-engineering. This research assume that one of the known methods for protection from reverse-engineering is obfuscation which means the ability to efficiently rewrite a program so that an adversary gains no advantage beyond having observable program with input and output behavior.

1.2 Problem Statement

Program Encryption Group (PEG) had developed obfuscation algorithms and techniques for achieving white-box protection by hiding functional and structural information in circuits. However, the previous white-box obfuscation technique does not check nor provide protection against Side-Channel Analysis (SCA) attacks. The circuit variants generated by CORGI might be particularly susceptible to SCA attacks which is performed by the information such as timing, power consumption, electromagnetic radiation, heat, noise, and more. With these side-channel information, an adversary can gain access to the internal of a circuit and break an encryption technique on a circuit without having the internal structural and functional information. In many cases, attacker can combine side-channel information with the observed structural or functional information of a circuit to exploit it [1]. Therefore, it is reasonable for this research to start focusing on the circuit protection against side-channel analysis.

1.3 Research Objectives and Contributions

The objective of this research is to detect and characterize power signature which is the abstract form of power consumption during operation, and to manipulate such a signature to increase the level of obfuscation. To achieve the objective, this research is broken down into four different sections with an ultimate goal of manipulating the side-channel information.

1. **Estimation and Simulation**
2. **Characterization and Classification**
3. **Implementation and Manipulation**
4. **Evaluation and Validation**

1.3.1 Estimation and Simulation. The power signature estimation is performed by the two approaches, namely static and dynamic approach. The static ap-

proach is achieved in probabilistic and statistical way without dynamic factors such as input patterns. Additionally, it is implemented on CORGI, which is referred to as Signature IDentification (SID) system. Dynamic approach, on the other hand, is performed by simulation process using an external tool considering input patterns. This dynamic simulation technique provides better accuracy compared to static one since it operates at the transistor level and its output is varied depending on input patterns.

1.3.2 Characterization and Classification. The estimated circuit's signatures from the previous section need to be characterized and classified. This is in order to set a baseline and a metric for evaluating how much the signature is changed after applying manipulation process.

1.3.3 Implementation and Manipulation. To achieve the primary goal of this research, power signature manipulation technique is designed and implemented on CORGI, which is called a Signature Manipulator(SM). SM is capable of providing a suitable transformation of the power signature of the circuit by manipulating switching activity of a circuit.

1.3.4 Evaluation and Validation. In order to evaluate the final circuit variant generated by SM, dynamic and static estimation approaches are used again to measure how much the signature is manipulated from the original one. Lastly, it is needed to validate the final variant obtained by the power signature manipulation method whether it provides the protection of the secret key from adversarial power analysis. The proposed signature manipulation method is applied on RSA circuit on Xilinx Virtex5 FPGA against adversarial power analysis.

1.4 Organization

The remainder of the thesis is organized as follows. Chapter II provides an overview of background information in the realms of software obfuscation and side-

channel analysis. Chapter III defines our methodology which includes a technique that measures power signature in dynamic and static ways, characterizes the signature, manipulates the circuit's signature using new manipulation algorithm, evaluates the proposed method using static and dynamic analysis, and validates the final variant using the FPGA based Encryption system test-bed. Chapter IV presents the results of experiments using power signature estimation and manipulation techniques, and Chapter V gives our conclusions along with our contributions and discussion of future work for the power signature manipulation method.

II. Literature Review

This chapter presents a background review of the literature pertaining to a circuit obfuscation technique against SCA attack. We organize this chapter in the following manner. Section 2.1 describes the concept of reverse engineering, Section 2.2 provides a brief introduction to obfuscation principles and previous circuit obfuscation techniques including the overview of *CORGI*, Section 2.3 introduces the side-channel attacks, and Section 2.4 covers the concepts of power dissipation model and the different types of power analysis.

2.1 Reverse Engineering

Reverse engineering is the process of analyzing a system to recognize the system's components and their interrelationships, and to create representations of the system in another form or at a higher level of abstraction [3]. There are different reasons for reverse engineering. The first one is creating the necessary documentation for maintenance, strengthen enhancement, or support replacement. A second reason is an adversarial purpose for compromising the existing technology by an industrial spy [11]. From the view of software engineering, reverse engineering is the reverse process of *forward engineering* [2]. Forward engineering consists of moving from high-level abstractions and logical, implementation-independent designs to the physical implementation of a system. In other words, it is a process of taking requirements, and creating designs and an implementation from these requirements [3]. This process is shown in Figure 2.1. Whereas forward engineering ends with a software product, reverse engineering starts with the product and is going backward of forward engineering.

2.1.1 Black-Box Analysis. The main purpose of black box analysis is to predict the intent of a system based on only its inputs and outputs without any information related to as internal structure of system. In the case of logic circuits, black-box analysis is the simplest way to reverse engineer a circuit in a brute force approach [12]. Identifying the overall function requires enumerating all possible input

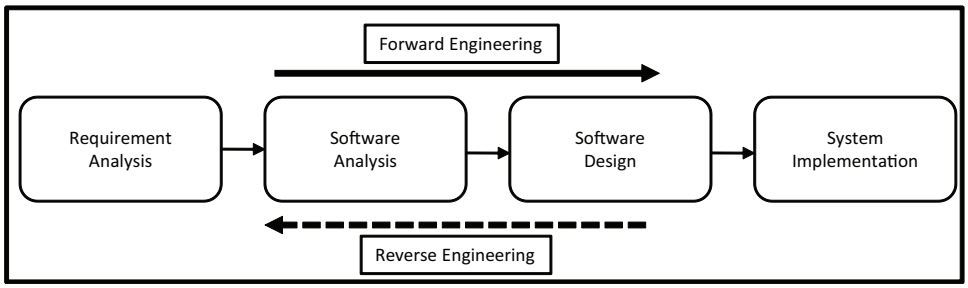


Figure 2.1: The Concept of Reverse/Forward Engineering [2]

Figure 2.2: Black-Box Analysis [12]

combinations and evaluating the circuit's output. This is impractical for a circuit with a large number of inputs due to the requirement of a high computing power and large search space.

2.1.2 White-Box Analysis. Unlike black-box analysis, white-box analysis focuses on the internal structure of a system. This approach provides an adversary with a better functional understanding than black-box analysis because it can be performed without having enumerating all possible input combinations. Adversaries can directly access the underlying white-box structure of a circuit in the real world. Therefore, protecting against white-box analysis is much more challenging than black-box protection. White-box protection has been the primary focus in the PEG research group.

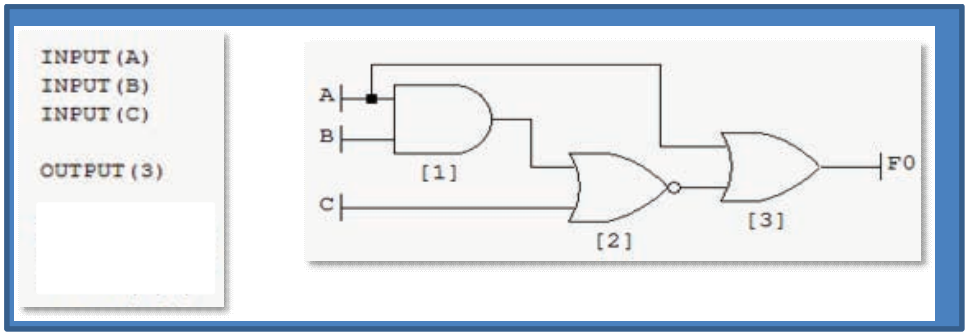


Figure 2.3: White-Box Analysis [12]

2.2 Circuit Obfuscation

This research assume that general programs are themselves abstractions of Boolean logic because they have collections of straight-line logic. Thus, a program logic can be represented in a software as Boolean logic circuits [6]. In this regard, there is a connection between protecting original functions of circuit and protecting a software, either directly or indirectly. For this reason, protecting circuits is the primary focus of PEG's research to make secure software. To hide programmatic logic or original functions of circuit, obfuscation is one of known protections against reverse engineering. Obfuscation is a process that produces a semantically equivalent variant of a program. Thus, the obfuscated program still has the same number of inputs and outputs and performs the same logic function but has a different white-box structure [6]. This obfuscation technique based on the white-box structure of a program attempts to confuse a reverse engineer.

2.2.1 CORGI. To achieve the main purpose of the program encryption group, McDonald and Kim developed Java-based a circuit obfuscator known as Circuit Obfuscation via Randomization of Graphs Iteratively *CORGI* at the Air Force Institute of Technology (AFIT). *CORGI* mainly provides the feature of representation and obfuscation of a combinational circuit. As seen in Figure 2.4, The first version of *CORGI*, called CORGI 1.0, provides a random based obfuscation technique, namely random sub-circuit selection and replacement. Since then, algorithms

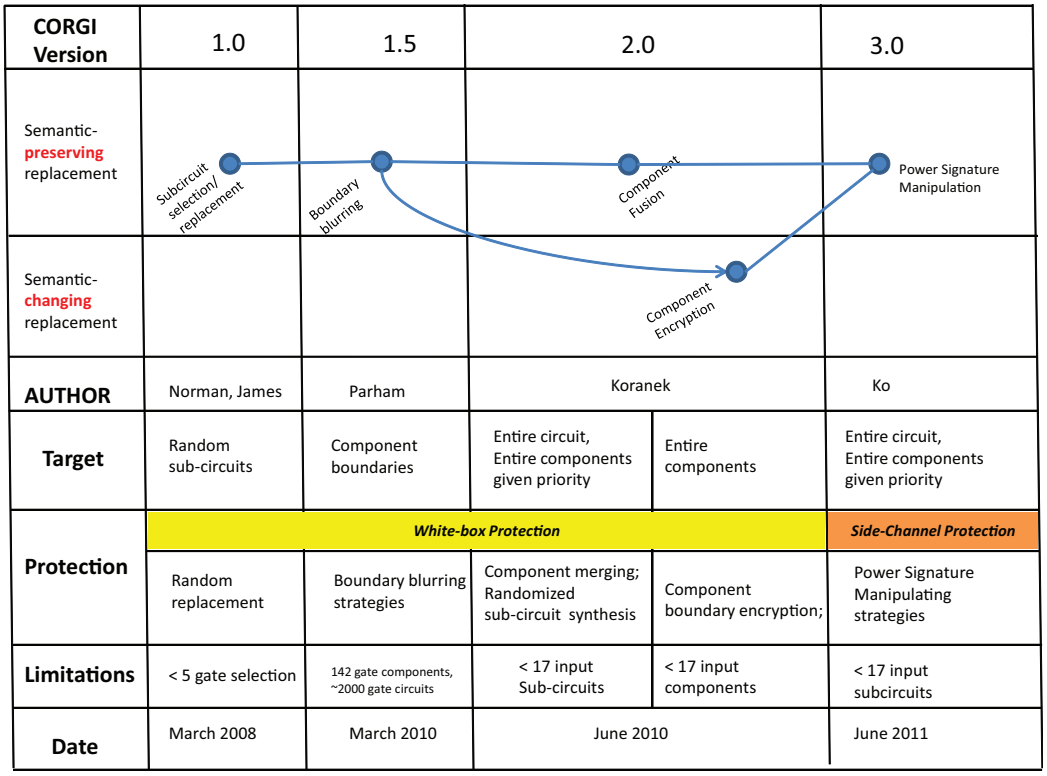


Figure 2.4: Revision History and Proposed Features in CORGI

and techniques for circuit obfuscation have been improved. In 2010, CORGI 2.0 was designed in a deterministic approach for obfuscation method. Additionally, CORGI 2.0 provides component-level obfuscation technique rather than a single gate-level technique against a white-box analysis attack. Now, this research, on the other hand, is toward to version 3.0 in order to achieve the signature protection against an adversarial side-channel analysis attack.

2.2.2 Random Sub-Circuit Selection and Replacement. Random SSR Method selects a sub-circuit in the entire original circuit at random and then replaces it by requesting a sub-circuit to be replaced from the circuit library(CXL). Then, CXL provides a random, semantically equivalent sub-circuits. From this action, the original sub-circuit is removed from the circuit, and the sub-circuit obtained from CXL is inserted into the original place. This is not a single step, but an iterative process.

But such a random SSR does not guarantee that the final circuit variant from the large number of iterations will be more secure.

2.2.3 Component Identification. Parham [16] focused on larger set of gates than a single gate. Such a set of gates is called a component. Component consists of one or more gates which has an own function such as a multiplexer and an adder. Therefore, it is possible for an adversary to focus their attention on such a component after they failed for the reverse-engineering by knowing a gate level structure alone. For this reason, Parham designed a component identification tool based on the assumption that adversaries may use component identification techniques for determining the overall function of the system [16]. The component ID enumerates a sub-circuit and compares it with the known pre-defined components in a library. If the matched sub-circuit were found, the component was identified. This algorithm makes it possible to automatically identify the existing component without visual searching process. But, the pre-defined library modules are limited for applying possible components and circuits. For example, if there is a circuit having components which does not exist in a library, it is not possible to identify the components in the circuit.

2.2.4 Component Fusion. In order to overcome the limitation on random SSR in CORGI 1.0, Koranek developed new obfuscation method in CORGI 2.0, which is called *component fusion* [13]. Component Fusion provides a deterministic selection strategy instead of random selection. With the new obfuscation method, the obfuscator selects both components identified by Parham's component ID tool and the sub-circuits connecting to the identified components for replacement. Next, component fusion synthesize all selected gates together to eliminate the border of components. This is accomplished by using the ESPRESSO two-level optimizer [13]. As a result, component fusion contributes to prevent from discovering components in a circuit.

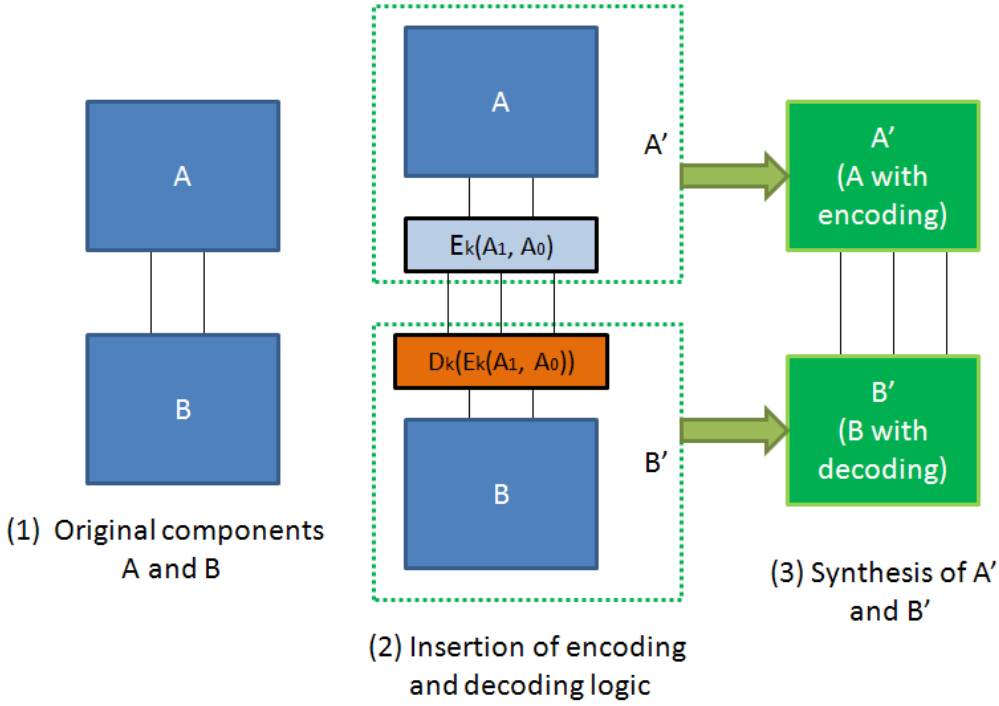


Figure 2.5: Component Encryption [13]

2.2.5 Component Encryption. Component fusion guarantees to hide the existing components in a circuit, but it might still remain the original signals in a circuit. To overcome this limitation on component fusion, Koranek [13] developed another deterministic method for obfuscating the existing signals in a circuit - *Component Encryption*. As seen in Figure 2.5, component encryption is implemented by selecting the signals between components, generating encryption and decryption logic on the connected components, and then synthesizing the generated logic and the each connected component. With component encryption, the number of signals and the semantics between components are successfully changed.

2.3 Side-Channel Attacks

Even if the circuitry is protected from the reverse engineering attack based on white-box analysis, the circuitry might still leak significant information related to the circuit's function. An attacks on a circuit using leaked side information is called a side-channel attack. Side-channel attacks are mainly divided into the categories

of Timing Analysis Attacks(TA), Fault Injection Attacks(FA), and Power Analysis Attacks(PA) [12].

1. **Timing Analysis**: *Timing Analysis is the method which focuses on the variation of the timing in processing by modifying the speed of the circuit clock, either speeding it up or slowing it down to achieve a desired information about the circuit implementation.*
2. **Fault Injection**: *The purpose of fault injection is to cause the targeted digital system or circuity to be malfunction to reveal useful information for additional attacks. Fault injections include a variety of attacks:raising voltage glitches and clock glitches, modifying the temperature, and so on.*
3. **Power Consumption Analysis**: *By observing the power usage across a circuit, an attacker can gain insight into what signals are changing the most in the circuit. Particularly, if a circuit is being used to compute some mathematical function, more power will be used in the area of the circuit dedicated to this function.*

Of course, there are other types of side channel attacks. However, the previous three techniques are the main technologies known as the side-channel attacks on digital systems. This research tries to prevent combinational circuits from power consumption analysis as well as white box analysis.

2.4 Power Analysis

In general, power analysis techniques are performed by obtaining the power transient directly from physical circuitry with physical equipments as shown in Figure 2.6. However it is very costly to set up the all physical environments for power analysis. As an alternative, power analysis for CMOS digital circuits can be done at several levels of abstractions [20]. The techniques can be divided into from low-level technique to statistical and probabilistic technique by trading off between accuracy and speed.

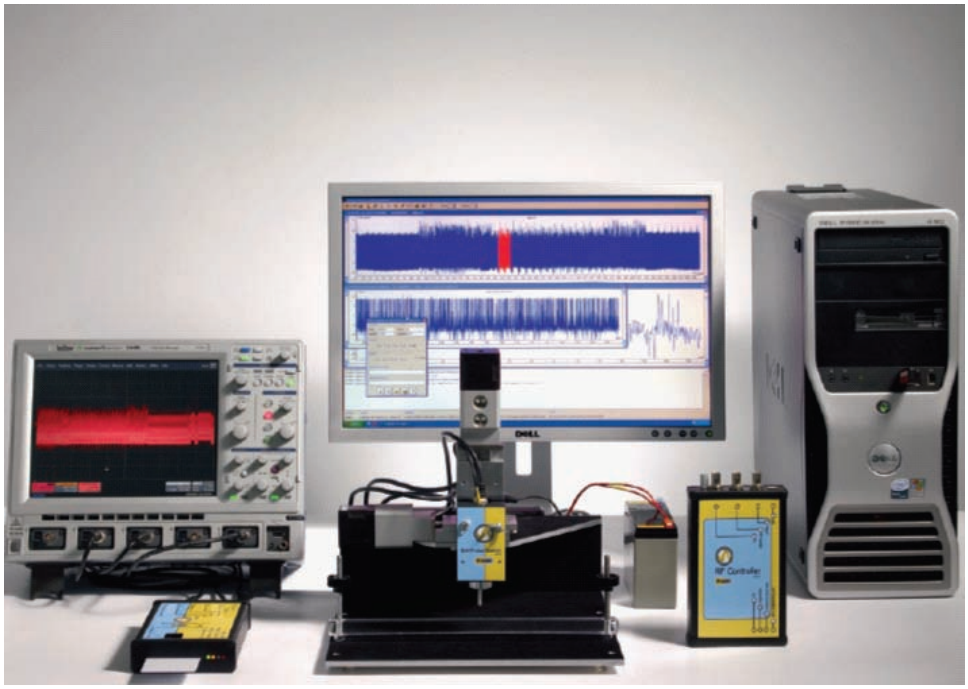


Figure 2.6: The integrated equipments for power analysis(http://www.riscure.com)

2.4.1 Power Dissipation Model . In general, power is a secondary consideration behind speed and area for many chips. As chip density and operating frequency increases, power consumption skyrockets and becomes the primary concern in terms of high performance and life-time for CMOS design [14]. Various methods [10] to estimate power consumption have been proposed for low-power design. Even if the general purpose of power estimation is different from the purpose of this research, power analysis attacks (and more generally side-channel attacks) are also performed by using the power dissipation model. For this reason, the fundamental power consumption model will be discussed in this section. The power consumption in a digital system is given by the following expression [14]:

$$P_{total} = P_{dynamic} + P_{short} + P_{static} \tag{2.1}$$

where:
P_{total} : Total power dissipation;
$P_{dynamic}$: Dynamic circuit power dissipation;

P_{short} : Short circuit power dissipation;
P_{static} : Static circuit power dissipation

The three terms in the order written represent the dynamic, short, and static dissipation respectively of the circuit [14] [18]. The second term, short dissipation refers to the power consumed due to the small current from source to ground appearing at the output of a CMOS gate during switching from one logic level to another, at the moment that both output transistors drive current [7]. In more technical terms, current can only flow while both pMOS and nMOS networks are partially ON.

$$P_{short} = V_{DD} \sum_{i} I_{short} \tag{2.2}$$

Where:
V_{DD} : supply voltage;
I_{short} : short circuit current

The third term, static dissipation, P_{static}, does not depend on the circuit activity and is determined by the target technology, undesired short circuits and (MOSFET) leakage current. This static power dissipation is the product of total leakage current and the supply voltage.

$$P_{static} = V_{DD} I_{static} \tag{2.3}$$

Where:
V_{DD} : supply voltage;
I_{static} : leakage current

The most important term among the three terms is the first one, namely the dynamic dissipation, since it is consumed by charging and discharging the parasitic capacitors present in all circuit nodes. Additionally, it is known that the dynamic power usually accounts for about 70% of the power consumed in a combinational circuit [7]. Thus, the power in a digital CMOS circuit can be determined using the

following formula for dynamic power:

$$P_{dynamic} = \sum_{i} V_{DD}^2 C_{load} \alpha_{0\to1} f \tag{2.4}$$

V_{DD} is the supply voltage at node i, and C_{load} is the load capacitance at node i. $\alpha_{0\to1}$ is the power consuming switching activity at node i, and f is the frequency of operation of the circuit [18].

To make more sense of the previous expression, let us begin by reviewing more definitions for deriving the average power dissipation over some time interval T.

$$P_{dynamic} = \frac{E}{T} = \frac{1}{T}\int_{o}^{T} P(t)dt = \frac{1}{T}\int_{o}^{T} i_{DD}(t)V_{DD}dt \tag{2.5}$$

Where:
E : energy consumed over the interval ;
T : time interval;
$P(t)$: instantaneous power on time t;
i_{DD} : supply current;
V_{DD} : supply voltage;

Based on the previous expression, the average dynamic power can be written as:

$$P_{avg-dynamic} = \frac{1}{T}\int_{o}^{T} i_{DD}(t)V_{DD}dt = V_{DD}^2 C_{load} f_{switching} \tag{2.6}$$

Since most gates do not switch every clock cycle, it is often more convenient to express switching frequency $f_{switching}$ as an activity factor $\alpha_{0\to1}$ times the clock frequency f. Now the average power dissipation may be rewritten as:

$$P_{dynamic} = \alpha_{0\to1} V_{DD}^2 C_{load} f \tag{2.7}$$

Where:
$\alpha_{0\to1}$: the probability of a $0 \to 1$ power consuming transition.

By reviewing the power dissipation model so far, it is learned that the power consumption in CMOS devices is proportional to the switching activity. Thus, $\alpha_{0\to1}$ is the principal parameter for the domain in an adversarial power analysis as well as general low-power circuit design.

2.4.2 Simple Power Analysis (SPA). SPA is one of the attacking method for a cryptography system. It is implemented by directly interpreting power consumption during cryptographic operations to recovering the main key or logic in a system [8]. By observing the power consumption on a smart card, SPA can directly attack the secret key information embedded in the card since the amount of power consumed varies depending on the microprocessor instruction performed. For example, SPA can be used for breaking RSA or DES implementations by revealing differences during the cryptographic operations. It is known that many current smart cards are vulnerable to SPA [9].

2.4.3 Differential Power Analysis (DPA). DPA has a much more complicated process, and is much more difficult to prevent than SPA. SPA is more likely to observe the power consumption, on the other hand, DPA uses the technique for statistical analysis and error correction to extract the exact correlated to secret keys. DPA has largely two steps: Data collection and data analysis [9]. Data collection is implemented by sampling the power consumption for a smart card during the cryptographic operations. Data analysis is performed by reducing the unnecessary signals and applying digital signal interpretation and statistical technique to conduct attacks.

III. Methodology

3.1 Problem Definition

The fundamental goal in the program encryption group is to provide combinational circuit protections by proposing more secure and advanced methods for circuit obfuscation in order to increase the cost of reverse engineering. To meet this goal, PEG has proposed various methods for obfuscating a combinational circuit based on hiding the structural and functional information of the original circuit against a white-box analysis. The structural information normally means the information about the number and the type of gates, signals, components, and so on. However, this research focuses more on dealing with the side-channel information than on the structural information for obfuscating a combinational circuit.

As discussed in Chapter II, observing power consumption over a circuit is the most well-known method among the wide variety of side-channel attacks. The main functionality or the encryption key in a circuit could be easily detected by using the power analysis attacks. Therefore, our method in this research takes into account manipulating the power signature of the original circuit.

The process for this research is divided into four steps as follows:

1. **Estimation and Simulation**

   - Estimating power signature using an internal tool in static approach and simulating a circuit using an external tool in the dynamic approach

2. **Characterization and Classification**

   - Characterizing the original power signature and classifying it by the pre-defined four types of power signature patterns

3. **Implementation and Manipulation**

   - Implementing Signature Manipulator (SM) in CORGI and manipulating the original power signature by using SM

4. **Evaluation and Validation**

   - Evaluating the SM using static and dynamic analysis and validating the pro-

posed method applying on Xilinx Virtex5 FPGA against an adversarial power analysis.

The first step is an estimation and simulation to measure power signature for combinational circuit using an internal and external tool. For power estimation, two approaches are executed in this research, namely static and dynamic technique. Static approach is performed in probabilistic and statistical way by using an internal tool, which is referred to as SID. In static approach, SID can provides the predictable power signature from the switching activity estimation technique without considering input pattern. In dynamic approach, simulation process is executed by using SPICE simulator, which is one of the well-known circuit simulators to simulate power consumption of a circuit at the transistor level under dynamic input patterns. Second, characterization and classification are performed. This process mainly focuses on characterizing the original power signature and classifying it as one of the four predefined power signature patterns. Third, implementation and manipulation processes are executed for the primary goal of this research which is to transform the class of the original power signature. Signature Manipulator (SM) is designed by manipulating the amount of switching activity. Lastly, evaluation and validation processes will be performed to quantify how much power signature is transformed by the proposed signature manipulation techniques and to verify the ability to protect an encryption key in the circuit against adversarial power analysis.

3.2 Estimation and Simulation

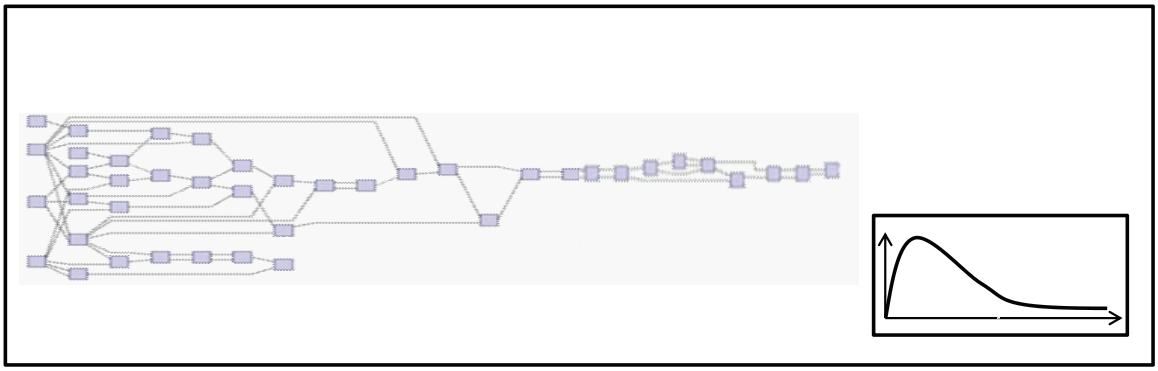
As mentioned previous, this section discusses both static and dynamic approaches for estimating power signature in combinational circuits. The static technique is performed without the circuit's input pattern. It can provide an power signature in statistical and probabilistic way based on the assumption that the circuit's inputs are uniformly distributed. For this reason, this technique is referred as static. This simplified no simulation-based approach is implemented in CORGI, which is called SID. In dynamic technique, on the other hand, the final result can be varied

depending on what type of the input patterns. Thus, this technique is called dynamic. The dynamic approach is performed by simulation process given pseudo-random or user-defined input vectors per clock cycle.

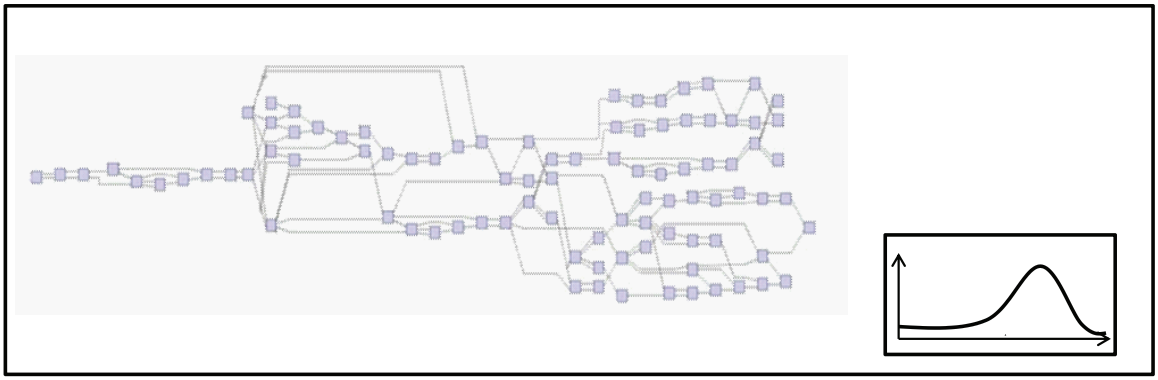
3.2.1 Static Approach . These types of techniques usually attempt to estimate an average power dissipation in a probabilistic and statistical way without considering dynamic factors such as an input pattern. These no simulation-based techniques are referred to as static approach in this research. The easiest way to predict power signature in this static approach can be performed by just looking at the overall structure of circuit. For instance, Figure 3.1 provides two examples of a circuit's structure. In Figure 3.1(a), front-loaded shape of power signature can be predicted, because most gates are located in the front part of circuit. Likewise, in the Figure 3.1(b), rear-loaded shape of power signature can be expected by the overall structure of a circuit. But, it is not easy for this technique to apply every other cases. In order to achieve more accurate result, this research examines the previous works for static approach. From the previous works, it is found that most techniques in this approach focus on the dynamic circuit power dissipation among three factors of Eq.2.1, power dissipation model, as seen in Chapter II, because the dynamic circuit power dissipation usually accounts for about 70% of the power consumed in a combinational circuit [18]. Thus, most methods aimed at optimizing this term will have the highest effect on static power signature estimation and manipulation.

$$P_{dynamic} = \alpha_{0\to1} V_{DD}^2 C_{load} f \tag{3.1}$$

As seen in Eq.3.1, the parameters of the dynamic power expression consist of V_{DD}, C_{load}, f, and $\alpha_{0\to1}$ as explained in chapter II. At the logic-level, the switching activity of a circuit, $\alpha_{0\to1}$ is the prime factor out of four terms for static power estimation and manipulation in this research. It is widely accepted that the switching activity is strongly related to power consumption in combinational logic circuits since significant power is consumed only during logic transitions. Therefore, the switching



(a)



(b)

Figure 3.1: Power Signature Example associated with Circuit's Structure
(a) The Front loaded Circuit Example
(b) The Rear Loaded Circuit Example

Table 3.1: The Truth Table for Two Input NAND Gate

A	B	O
0	0	1
0	1	1
1	0	1
1	1	0

- Minterms of NAND gate = $Z(X,Y) = \overline{XY} + \overline{X}Y + X\overline{Y} = \sum m(0,1,2)$
- Maxterms of NAND gate = $\overline{Z}(X,Y) = XY = \prod M(3)$

activity will be employed to both estimate and manipulate the power signature in static approach.

In static approach, Menon et al. [17] proposed a method to estimate an switching activity at an each gate and calculate the total switching activities for a combinational circuit in their research. The switching activity definition given in [17] assumed that the switching activity for a logic gate is the average rate at which its output switches from 0 to 1 or 1 to 0. The procedure proposed in [17] is given by the following expression:

$$P_0 \times P_1 = \frac{|R|}{|R| + |F|} \times \frac{|F|}{|R| + |F|} \tag{3.2}$$

where:
P_0 : The probability of Zero;
P_1 : The probability of One;
$|F|$: The cardinality of the set of minterms of the logic function;
$|R|$: The cardinality of the set of maxterms of the logic function

For example, consider a two input NAND gate whose inputs are statistically independent and uniformly distributed. It means that the four possible states for inputs A and B (00, 01, 10, 11) are equally likely. F is a set with minterms corresponding to the 1's of the function, and R is a set corresponding to the 0's of the function. Thus, $|F| = 3$ and $|R| = 1$ are given by the Table 3.1.

Then, $P_0 = \frac{1}{4}$ and $P_1 = \frac{3}{4}$. Calculating by the Eq.3.2, $P_{0\to1} = \frac{1}{4} \times \frac{3}{4} = \frac{3}{16}$ and $P_{1\to0} = \frac{3}{4} \times \frac{1}{4} = \frac{3}{16}$, where $P_{0\to1}$ and $P_{1\to0}$ are the transition probabilities of

the outputs switching from $0 \rightarrow 1$ and $1 \rightarrow 0$ respectively. Since power is drawn from the battery source only when the output changes from 0 to 1, the discussion on switching activity in [18] accounted only the power consuming transition of $0 \rightarrow 1$, so the switching activity of the two input NAND gate equals to $\frac{3}{16}$ [17]. As a result, the switching activity probability can be written as:

$$\alpha_{0\rightarrow1} = P_0 \times P_1 = P_0 \times (1 - P_0) \tag{3.3}$$

For an another example of an inverter, on the other hand, the switching activity probability with uniformly distributed inputs is given by $P_0 \times (1 - P_0) = \frac{1}{2}(1 - \frac{1}{2}) = \frac{1}{4}$.

The following example illustrates using the switching activity estimation technique on a circuit with more gates. The custom circuit in Figure 3.2 consists of three inputs, two outputs, eight gates, and three levels. In order to apply the Menon's technique to this circuit, a truth table is generated for the circuit. Then, the switching activity at each gate can be also obtained as shown in Figure 3.3 and Figure 3.4. But, what this research mainly want to measure is the power signature of a circuit rather than the individual or total switching activity for a circuit. For this reason, this switching activity estimation technique needs to be improved for predicting the power signature. So, it is revised by comparing the total switching activities by the level. Figure 3.5 shows the variation of the switching activities by the level of the circuit. Finally, the switching activity variation can be predicted by converting Figure 3.5 to Figure 3.6, which is comparable to the power signature simulated by SPICE in dynamic approach. Of course, this technique is limited in terms of the accuracy compared to dynamic simulation. However, this technique is much faster and easier to use than dynamic approach. Additionally, it is accurate enough for this research to make a decision for selection strategy of power signature manipulation algorithm later. Even if some limitations exist in this fashion, it has been widely applied in the problem of optimization of total power consumption as well as the domain of power estimation.

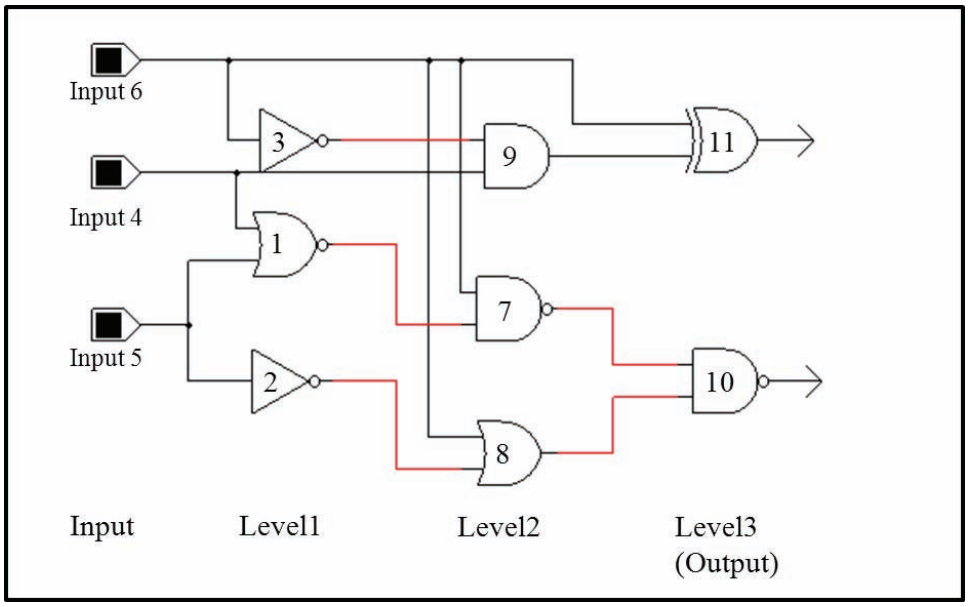


Figure 3.2: Custom Circuit c(3-2-8)

In6	In4	In5	3	1	2	9	7	8	Out 11	Out10
			NOT(6)	NOR(4,5)	NOT(5)	AND(3,4)	NAND(1,6)	OR(6,2)	XOR(9,6)	NAND(8,7)
0	0	0	1	1	1	0	1	1	0	0
0	0	1	1	0	0	0	1	0	0	1
0	1	0	1	0	1	1	1	1	1	0
0	1	1	1	0	0	1	1	0	1	1
1	0	0	0	1	1	0	0	1	1	1
1	0	1	0	0	0	0	1	1	1	0
1	1	0	0	0	1	0	1	1	1	0
1	1	1	0	0	0	0	1	1	1	0
Switching Activity (189/64)		Node	1/2	3/8	1/2	3/8	7/32	3/8	3/8	15/64
		Level	11/8			31/32			39/64	

Figure 3.3: Truth Table with Switching Activity for c(3-2-8)

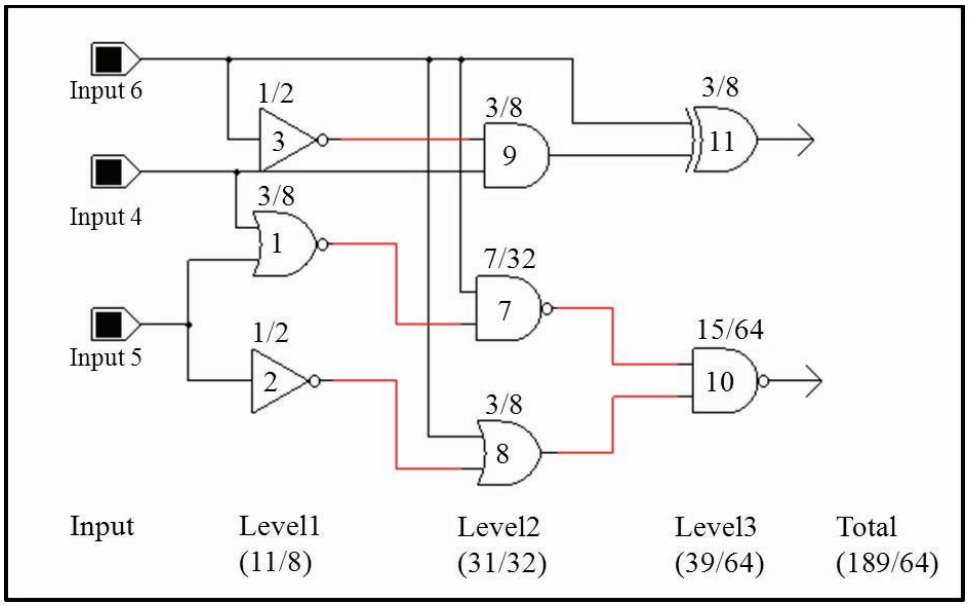


Figure 3.4: Custom Circuit c(3-2-8) with Switching Activity

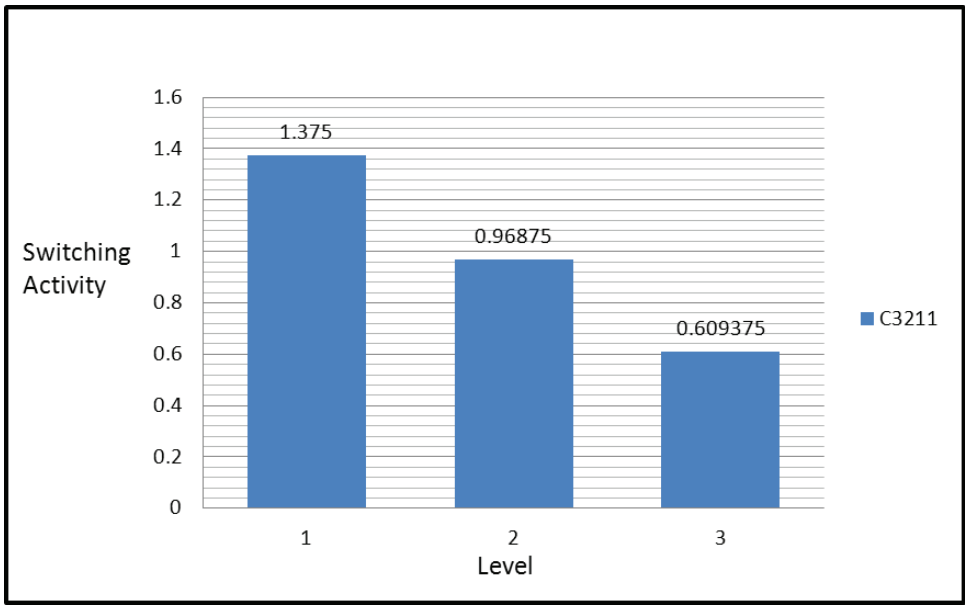


Figure 3.5: Comparison for Switching Activity by Level

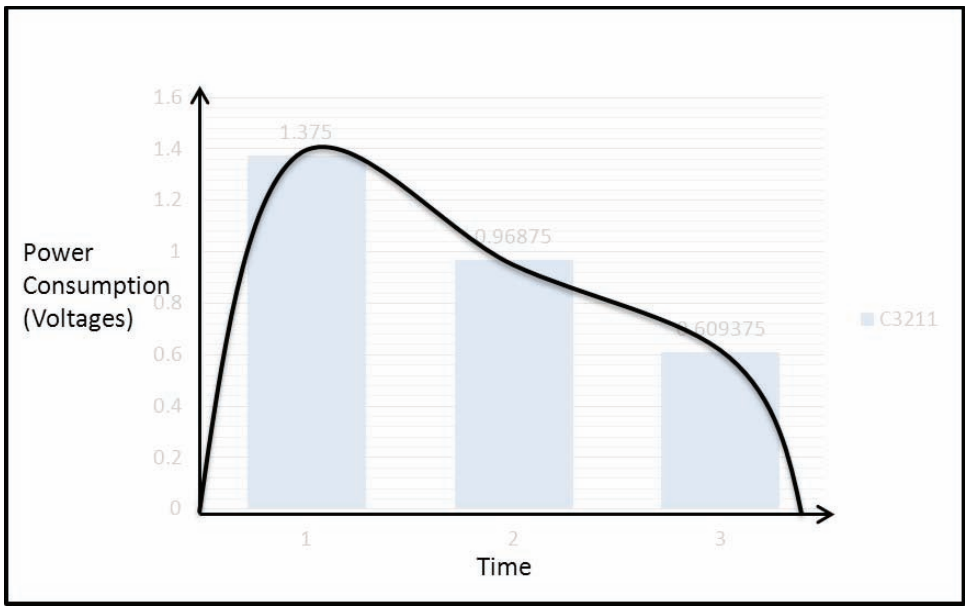


Figure 3.6: The Predictable Power Signature For c(3-2-8)

3.2.2 Signature Identification System. The SID is a tool designed for making power signature measurement an easier task based on the static approach. Although there is a tool to support the dynamic technique like SPICE simulator, an automatic tool for supporting the static approach does not exist. Thus, the automatic system is implemented using CORGI in this research. Figure 3.7 provides an overview of the SID system.

The previous implementations of CORGI do not support signature detection nor manipulation. Now, the SID is implemented based on the components of CORGI to support the statistical power estimation technique. In general, hardware description files such as VHDL, SPICE netlist are needed for the SPICE-like simulation tools; however, the SID just needs a simple circuit netlist as an input, and then the total switching activities and the predictable power signature can be automatically generated after computing the switching activity at each node and level.

3.2.3 Constraints on Static Approach. Static estimation is more simple than dynamic simulation using SPICE in terms of easy of use and speed for estimation. But the results obtained with the SID is limited on accuracy. Therefore, The results

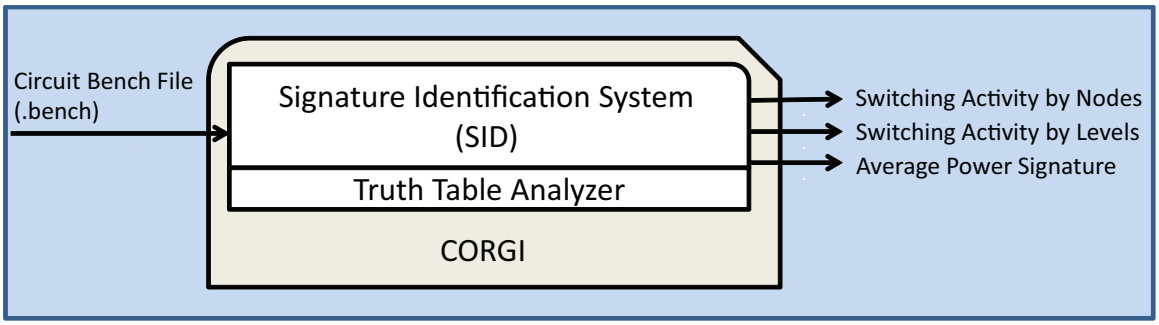


Figure 3.7: The System Overview of SID

from the static signature predictor SID should be validated with the results obtained with the dynamic simulation tool, such as HSPICE.

3.2.4 Dynamic Approach. Most techniques in this area operate mainly at the transistor level to estimate power consumption. These techniques are sometimes referred to as "low-level technique". Additionally, the reason why it is called dynamic approach is that the output will be varied depending on dynamic vectors such as input pattern and time interval. For this reason, this approach is sometimes called a pattern-dependent technique in other researches [10]. A good example of this approach is a simulation using SPICE(Simulation Program with Integrated Circuit Emphasis), which is an electronic circuit simulator. The SPICE is one of the well-known simulation tool which models the behavior of a circuit containing digital and analog devices. With the SPICE-like simulators, using dynamic technique can test and measure the power consumption of a circuit before touching the physical equipments. Accordingly, simulating the circuit with SPICE is the industry-standard way to verify circuit's operation at the transistor level before committing to manufacturing an integrated circuit in these days.

Originally, SPICE was developed at the Electronics Research Laboratory of the University of California, Berkeley in the early 1970s. Since then, SPICE has been widely distributed and used so far. ELDO is one of the commercial versions of SPICE simulator which will be used in this research for measuring power signature for a circuit as dynamic approach.

In order to carry out this simulation, the circuit must be described in SPICE format. This format is called the SPICE netlist, which has basically a transistor level description. Before running this simulation, it is required to know an element of description for a gate with nMOS and pMOS transistors using MOSFET Model. For this SPICE simulation, the SPICE netlist needs more efforts to describe the information from scratch such as the all types of gates in a circuit, how they are connected each other, what types of input pattern will be using, and how much amount of time will be assigned for simulation. Figure 3.8(b) shows the SPICE Netlist description of c17, which is one of the ISCAS 85 benchmark circuits [5]. Additionally, Figure 3.9 describes the two input NAND gate in C17 in a SPICE netlist format and schematic representation of the NAND gate in MOSFET Model generated by open software, Netlist Viewer.

3.2.4.1 Analysis using SPICE. The type of analysis in SPICE are largely divided into three types. These types of analysis are summarized as follows [23]:

1. **DC Analysis**

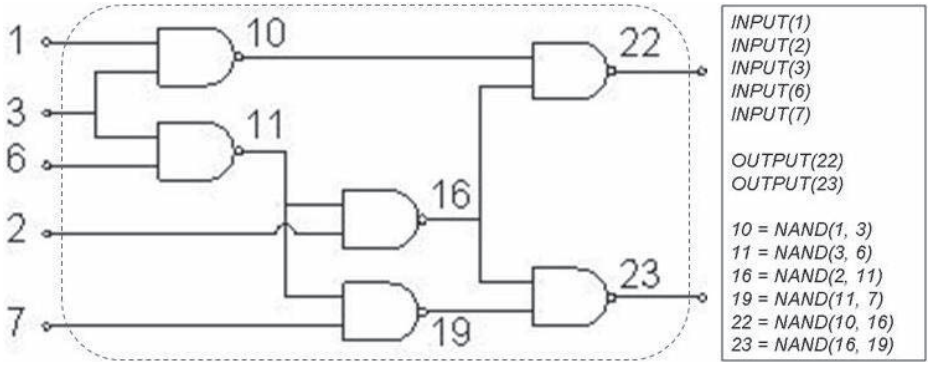
   :*A DC-level-based analysis that provides data to predict the DC response of an output to a DC voltage at the input*

2. **AC Analysis**

   :*A frequency-based analysis that predicts the output amplitude or phase shift through a circuit as a function of a fixed amplitude frequency applied to the input. This input frequency is over a specified range and thus frequency is over a specified range and thus frequency versus output response date is provided*

3. **Transient Analysis**

   :*A time-based analysis that provides an oscilloscopic display of an output. The output display typically is a result of an input stimulus to the circuit and is in the form of time versus voltage or current.*



(a)

```
* Define Benchmark Circuit C17
* Six NAND Implementation
* AFIT/ENG Hyunchul Ko

.include ami05_models

X10 1 3 10 NAND
X11 3 6 11 NAND
X16 11 2 16 NAND
X19 11 7 19 NAND
X22 10 16 22 NAND
X23 16 19 23 NAND

.subckt NAND2 top_in bot_in out
M1 out top_in 1 1 p L=0.6u w=1.2u AD=2.88p AS=2.88p PD=7.2u PS=7.2u
M2 out bot_in 1 1 p L=0.6u w=1.2u AD=2.88p AS=2.88p PD=7.2u PS=7.2u
M3 out top_in 2 0 n L=0.6u w=1.2u AD=2.88p AS=2.88p PD=7.2u PS=7.2u
M4 2 bot_in 0 0 n L=0.6u w=1.2u AD=2.88p AS=2.88p PD=7.2u PS=7.2u
.ends NAND2

VDD 1 0 DC 5

VIN1 1 0 pwl(0ns 0v 15ns 0v 15.1ns 5v 30ns 5v 30.1ns 0v 35ns 0v r)
VIN2 3 0 pwl(0ns 0v 10ns 0v 10.1ns 5v 15ns 5v 15.1ns 0v 35ns 0v r)
VIN3 6 0 pwl(0ns 0v 15ns 0v 15.1ns 5v 30ns 5v 30.1ns 0v 35ns 0v r)
VIN4 2 0 pwl(0ns 0v 10ns 0v 10.1ns 5v 15ns 5v 15.1ns 0v 35ns 0v r)
VIN5 7 0 pwl(0ns 0v 15ns 0v 15.1ns 5v 30ns 5v 30.1ns 0v 35ns 0v r)

.TRAN .01ns 35ns
.PLOT TRAN V(*) I(*)
.OPTION PROBE
.END
```

Define information of nodes including a type of gate and their connections

Define sub-circuit (NAND gate)

Define power and load source

Define input patterns

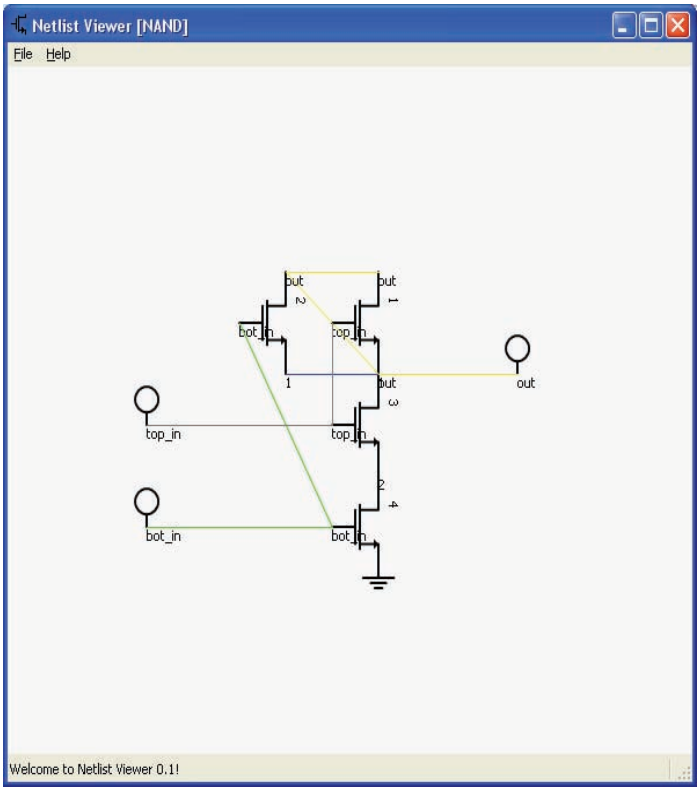
Define time interval and display option

(b)

Figure 3.8:
(a) c17 Benchmark Circuit
(b) SPICE Netlist expression of c17

```
.subckt NAND2 top_in bot_in out
M1 out top_in 1 1 p L=0.6u w=1.2u AD=2.88p AS=2.88p PD=7.2u PS=7.2u
M2 out bot_in 1 1 p L=0.6u w=1.2u AD=2.88p AS=2.88p PD=7.2u PS=7.2u
M3 out top_in 2 0 n L=0.6u w=1.2u AD=2.88p AS=2.88p PD=7.2u PS=7.2u
M4 2 bot_in 0 0 n L=0.6u w=1.2u AD=2.88p AS=2.88p PD=7.2u PS=7.2u
.ends NAND2
```

(a)



(b)

Figure 3.9:
(a) SPICE Netlist expression of a two input NAND gate
(b) Schematic representation of (a) generated by Netlist Viewer

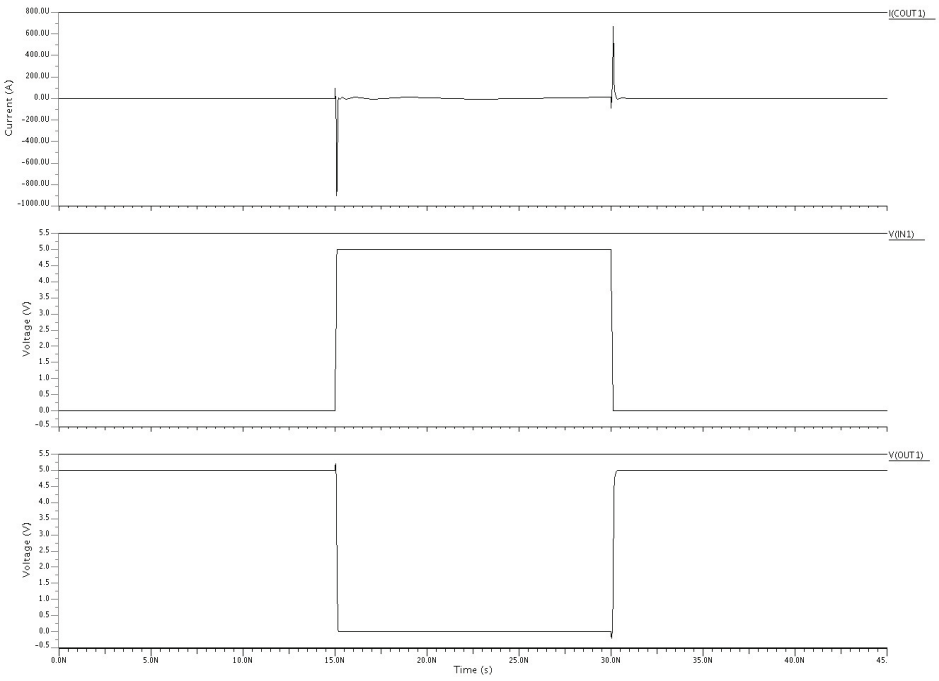
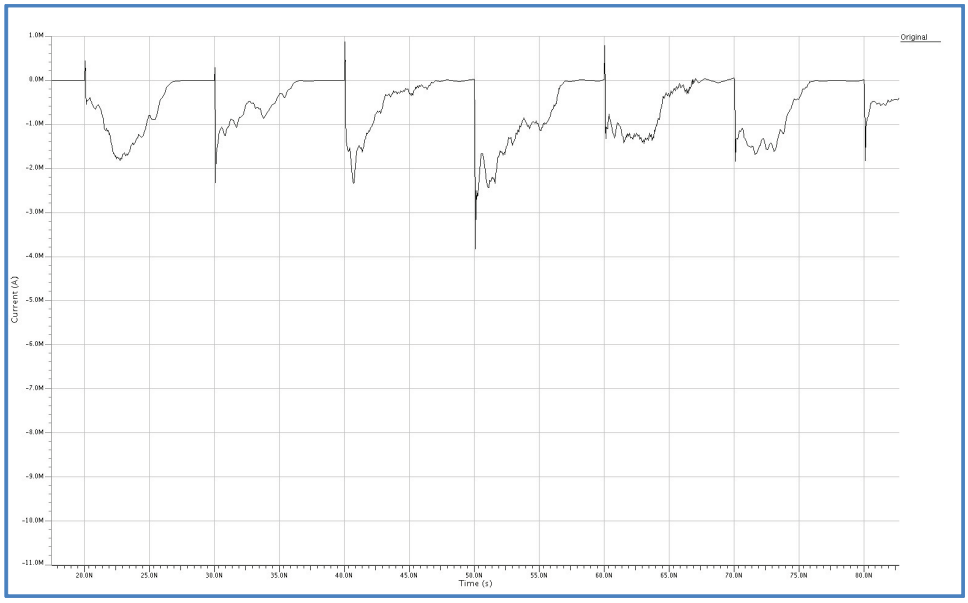


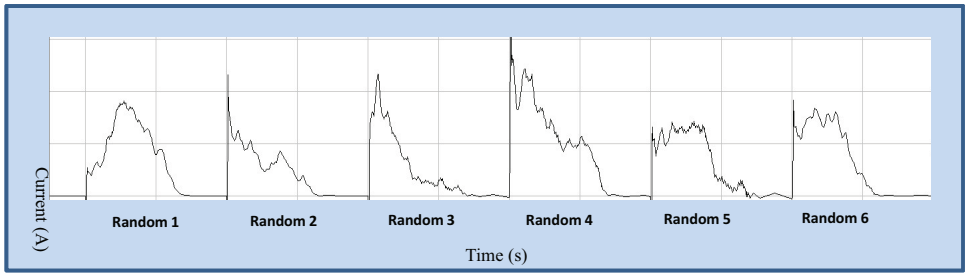
Figure 3.10: The Waveform of Transient Analysis In SPICE

Among the three types of analysis, the transient analysis is the primary type of analysis used in this research. It allows to measure power signature of a circuit by simulating instantaneous voltages and currents consumption. It also support a various set of inputs, both random and user-defined inputs. Figure 3.10 shows the waveform of voltages and currents over a specific time interval for an inverter generated by the transient analysis.

The waveform of voltages provides the logic transitions over the time, but the current waveform shows the power signature which the power analysis is mainly looking for. Based on the power dissipation model, namely $P = V_{DD}I$ as mentioned in Chapter II, SPICE has an option to generate the power variation calculated by the expression during the time interval. However, the overall shape of current (I) curve is virtually identical to the shape of power signature, because voltage (V_{DD}) does not changed significantly during the simulation process. The current (I) is the primary variable factor in the power dissipation model. Thus, by looking at the current wave-



(a)



(b)

Figure 3.11:
(a) Six Cycles of Current Waveform of c264 generated by ELDO spice simulator
(b) Six Power Signatures of c264

form, the abstract form of power signature can be readily obtained. For instance, Figure 3.11(a) shows the six cycles of the current waveform of c264 generated by ELDO simulator. By making an each cycle of wave form downside-up, each power signature can be obtained as shown in Figure 3.11(b).

3.2.5 Improving Dynamic Simulation Process using CORGI. In general, a set of task must be completed to perform the SPICE simulation. By reviewing the summary of process the dynamic simulation, this research found a similar work which

can be pre-defined or automatically generated using CORGI. Figure 3.12 provides the improved procedure for SPICE simulation using CORGI. Firstly, it is needed to import a circuit bench file, which describes information related a circuit structure such as the type of gates and their relation. All types of gates used in CORGI and ISCAS85 benchmark circuits can be pre-defined and stored in the standard gate library shown in Figure 3.12. Additionally, As part of this research, the SPICE netlist exporter was implemented as shown in Figure 3.13. All circuits generated CORGI can be automatically exported to SPICE netlist format. Lastly, it is just needed to modify the analysis options such as input patterns and time interval for simulation on the exported SPICE netlist file. This new procedure for SPICE simulation can be significantly improved in terms of efforts and time for this research.

3.2.6 Constraints on Dynamic Approach. Dynamic approach using SPICE simulation is accurate enough to be used as the estimation of power consumption for a circuit, but there are some disadvantages. First, it is not easy to handle and set up the parameters for the simulation. For example, in order to measure the power dissipation, one would have to know a dedicated SPICE Netlist description as an input, and also apply a large number of input patterns and keep track of the current waveforms. Secondly, It is extremely difficult to make accurate estimation for power signature of a circuit since the number of possible input sequences is exponential [10]. Lastly, it is too slow to simulate a large circuit. Even if it were possible, the question of what input patterns to apply would have to be considered.

3.3 Characterization and Classification

Regarding characterizing the power signature, it is needed to define the general power signature types to make easier to know the original signature and to compare it with the manipulated signature. Thus, the types of power signature are generalized into four types such as front-loaded, middle-loaded, rear-loaded, and front rear loaded signature. Figure 3.14 shows each possible signature in a circuit. These types are

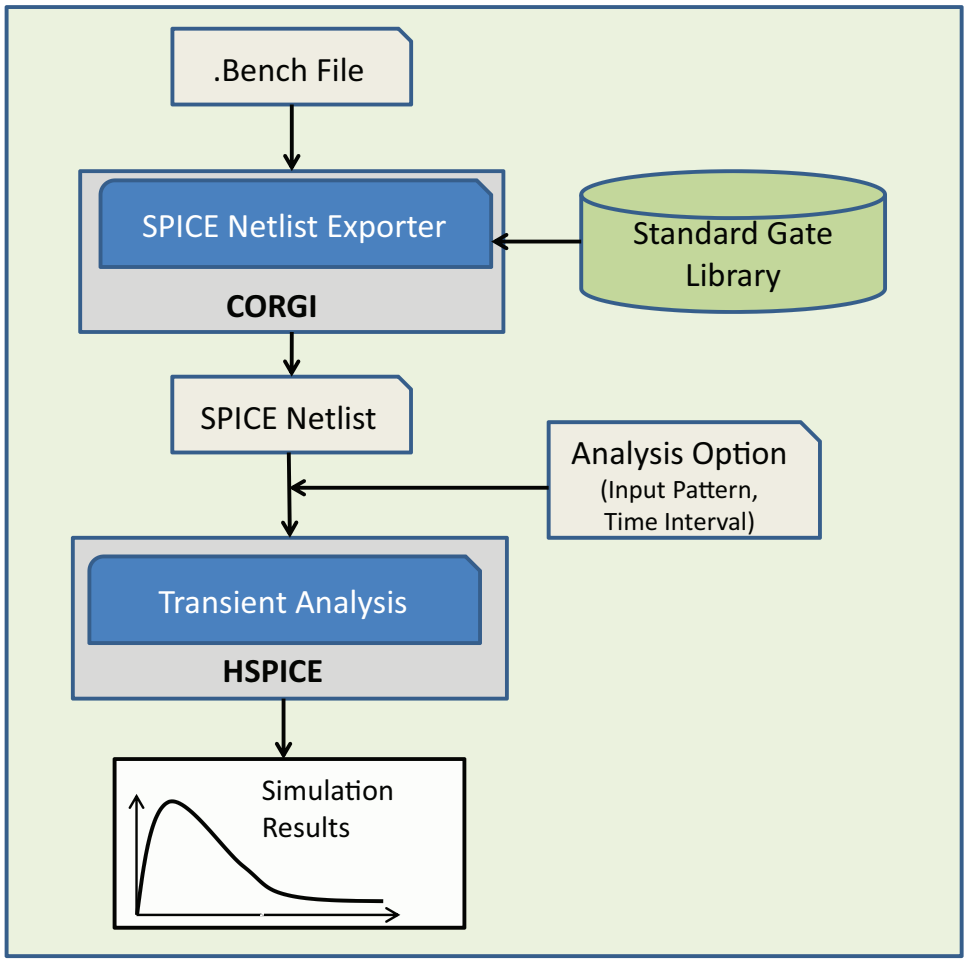


Figure 3.12: The Summary of Procedure for SPICE Simulation

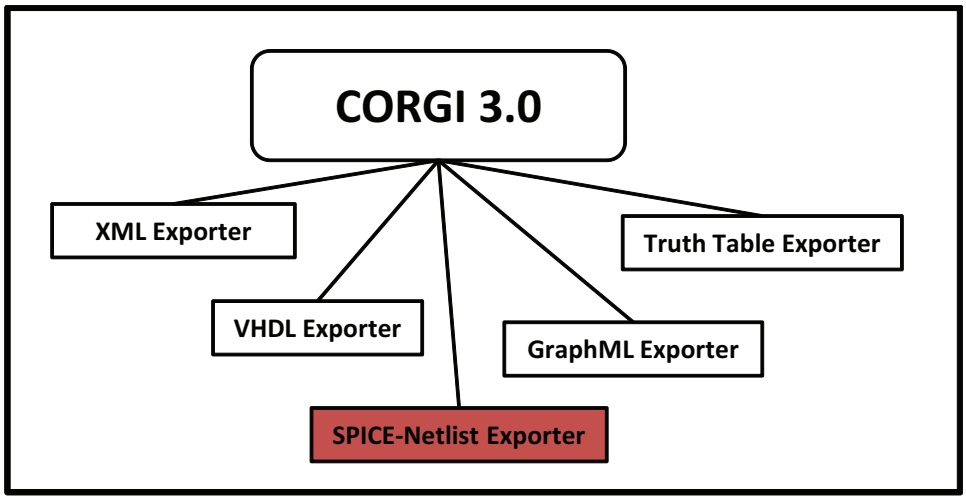


Figure 3.13: Implementing SPICE-Netlist Exporter in CORGI

referred to as Type I, II, III, and IV respectively. Figure 3.15 describes the circuit's structure associated with the type of power signature. As seen in Figure 3.15, the type of power signature can be predicted by the gates' distribution in the four different sample circuit. To be more accurate, such a type of signature can be recognized by a simple mathematical way. SID provides a suitable type of power signature among four types after comparing the region of three parts in a graph. For example, given a sample signature generated by SID as shown in Figure 3.16(a). It is divided by three parts by the number of levels. Figure 3.16(b) shows the signature with L1, L2, and L3 which is the interval of an each part, called F (front), M (middle), and R (rear). This process can be easily done by comparing the three parts of the signed area of the region by the graph of signature. As a result of this comparison, this example shows F > M > R, which indicates the front-loaded shape. In dynamic technique, the characterization process is the same as the static one. Figure 3.17(a) shows an example of power signature generated by dynamic approach with SPICE simulation. It is also divided into three parts by the time interval like in Figure 3.17(b). Then, the regions of an each part are compared each other by the following expression:

$$\text{F(front) vs M(middle) vs R(rear)}$$

$$= [\int_{T1}^{T2} S(x)dt] vs [\int_{T2}^{T3} S(x)dt] vs [\int_{T3}^{T4} S(x)dt]$$

3.4 Implementation and Manipulation

3.4.1 Signature Manipulator (SM). The primary goal of this research is to manipulate the original power signature. Thus, Signature Manipulator (SM) is devised for the purpose of transform the class of the original power signature. The power signature is manipulated by using switching activity since the power consumption for a circuit is highly related to the switching activity as mentioned in the Section 3.2.1. Figure 3.18 shows the overview of the SM system. Intentionally manipulating the amount of switching activities at a designated part of a circuit based on CORGI is

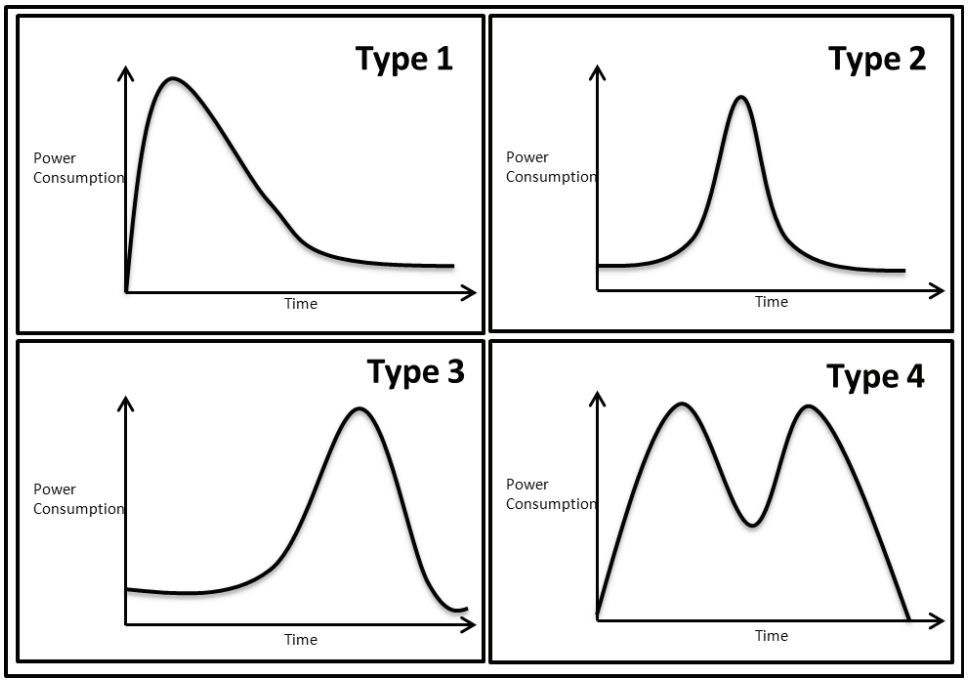


Figure 3.14: Four Types of General Power Signature

the main function of this system. For example, as seen in Figure 3.18, a given circuit P with Type I power signature pattern will be transformed to the variant P′ with Type II or III power signature type.

3.4.2 Power Signature Obfuscation Methods in SM. To manipulate power signature of a circuit, this research needs to find the primary factor for affecting switching activity. The first one is the number of gates and levels in a circuit. Simply, increasing the number of levels in a circuit may increase the delay of a circuit. Such a delay is mainly caused by switching activity. Thus, the gate's topology in a circuit can have a strong influence on the overall switching activities of the circuit. Second, the number of signals is also one of the main factors for affecting switching activity. Like the number of circuits, increasing the number of signals between gates is highly possible to increase the logic transitions. Third, the type of gate can be a factor. As you can see the Menon's switching activity estimation algorithm in Section 3.2.1, the switching activity is different depending on the gate type, the number of inputs, and fan-outs. For instance, the switching activity of a two input NAND gate is $\frac{3}{16}$, but

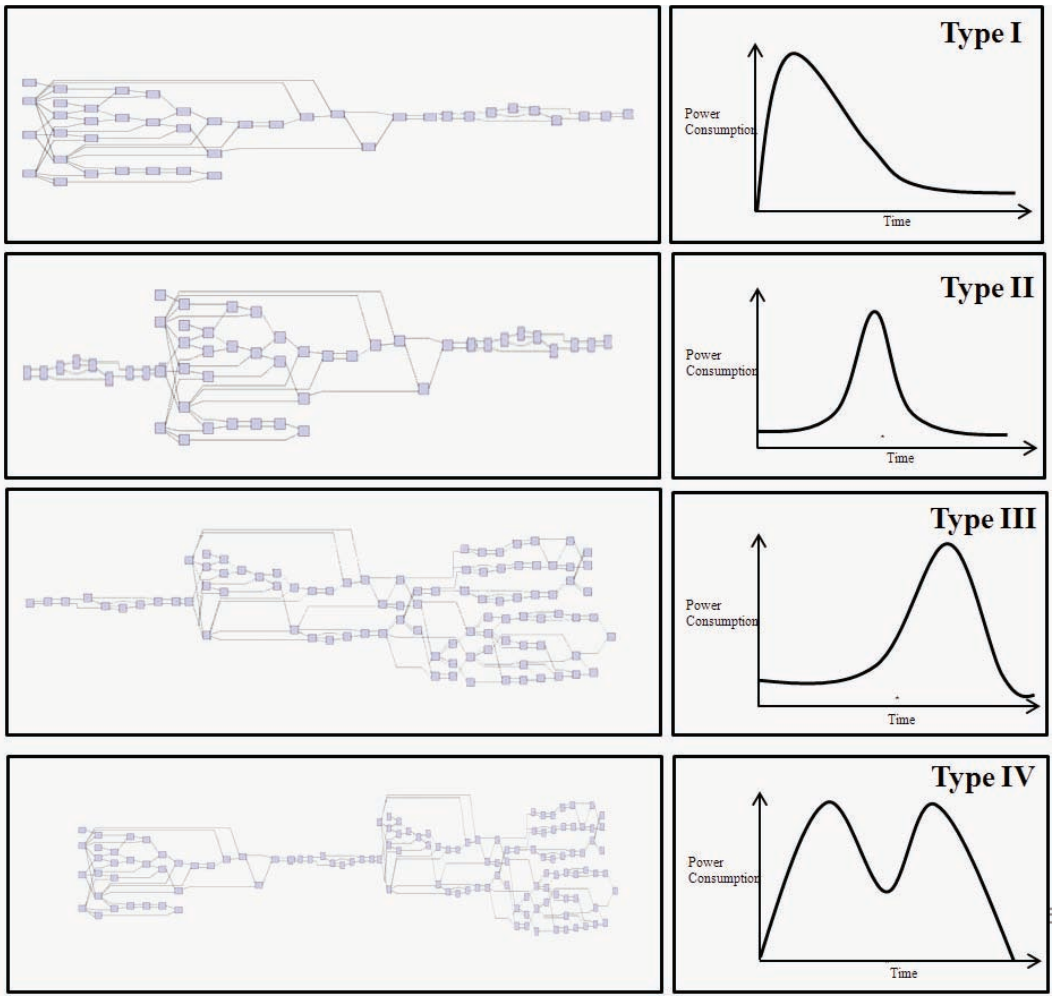
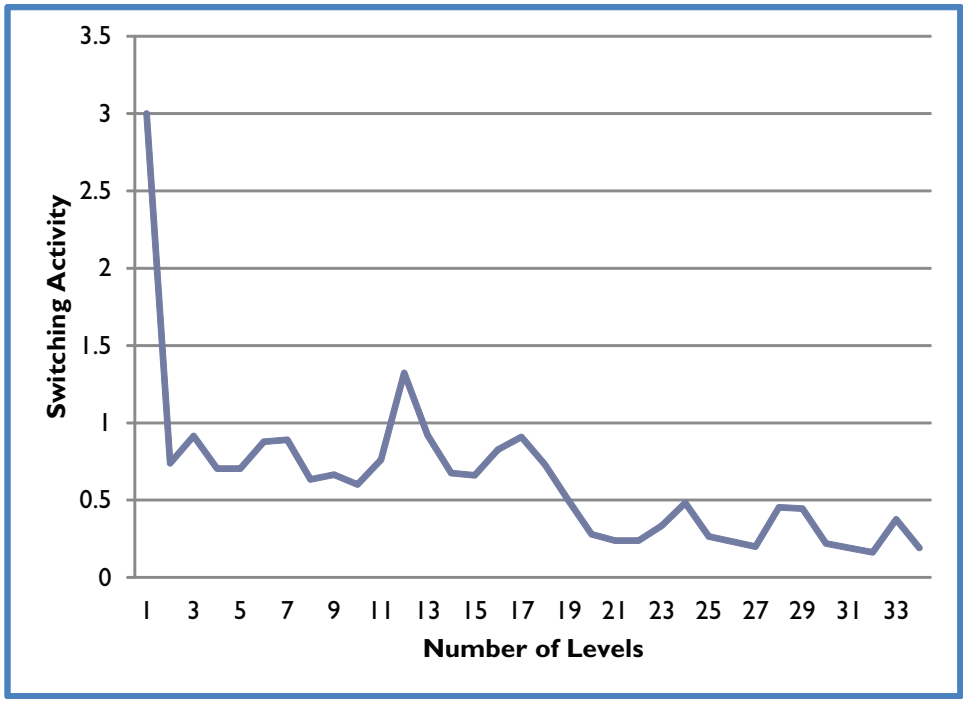
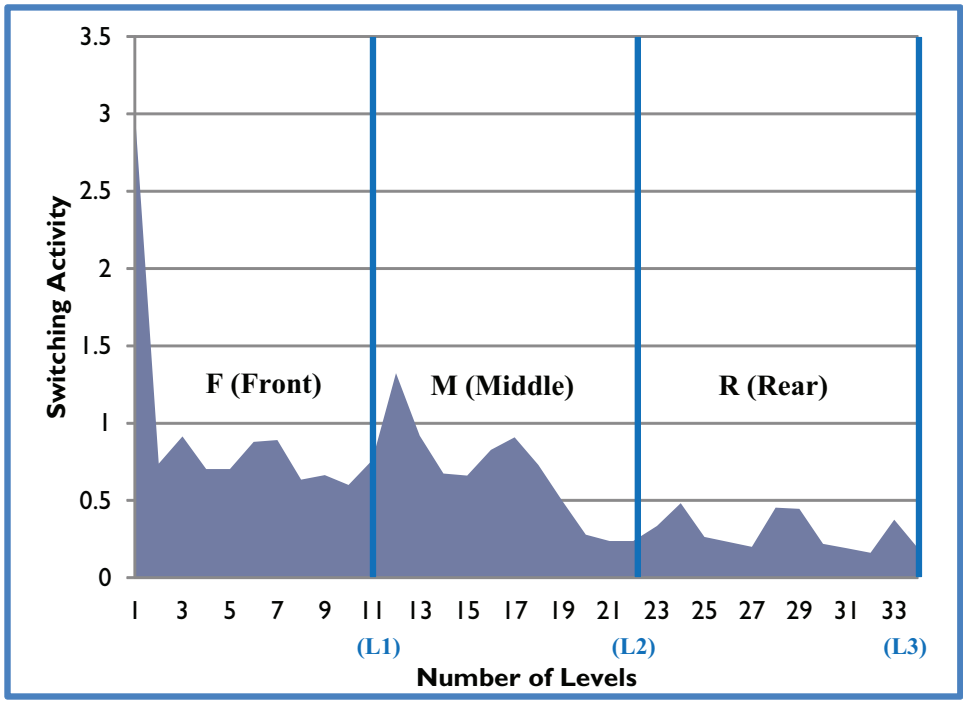


Figure 3.15: Four Types of Power Signature according to a structure of a circuit

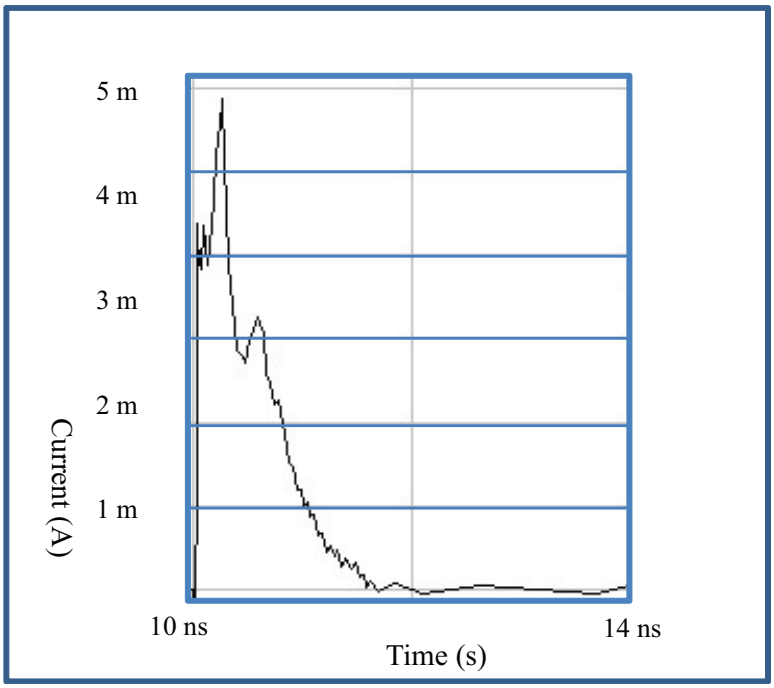


(a)

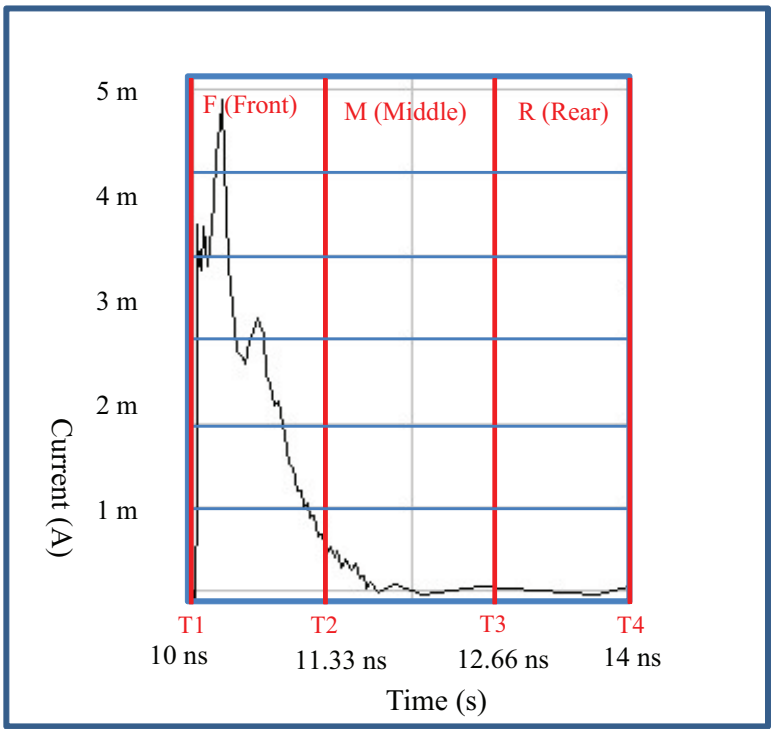


(b)

Figure 3.16:
(a) A sample circuit signature generated by SID
(b) A sample circuit signature divided into three parts by the number of levels



(a)



(b)

Figure 3.17:
(a) A sample circuit signature generated by SPICE
(b) A sample circuit signature divided into three parts by the time

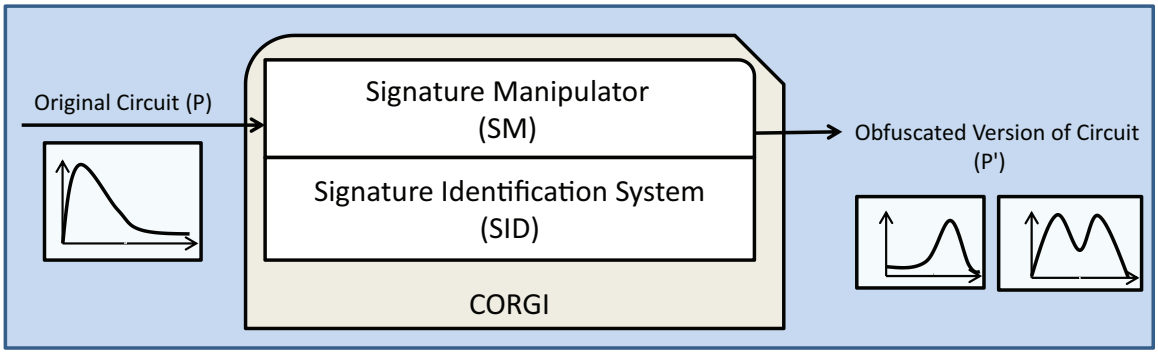


Figure 3.18: The System Overview of SAM

the one of an inverter is $\frac{1}{4}$. From this example, it is noticed that switching activity is minimum, when the difference between $|F|$ and $|R|$ of the function is maximum, and switching activity is maximum when $|F| = |R|$. Thus, it is realized that inverter has the highest switching activity since inverter has equal number of 1s and 0s in its function.

Based on these foundation of the factors for affecting switching activity, SM system provides new two algorithms, which are called smart selection and replacement method (Smart SSR) and Smart Component Encryption (Smart CE) method respectively. Regarding the Smart SSR method, it is implemented based on the random SSR in CORGI 1.0. Only difference between random and smart SSR is to apply smart selection strategy instead of random selection as shown in the Table 3.2. Such a circuit selection is performed depending on the user-defined part of a circuit such as front, middle, and rear. For example, when performing the obfuscation with FrontLevelTwoGates option, the two gates in the 30% of front level of circuit are selected and then replace semantically equivalent three gates from the CORGI library at random as iteratively. As a result of the iteration, it is expected to increase the number of gates at the front part of a circuit toward power signature with Type I. The other options in Table 3.2 are also performed as the same manner.

If the expected waveform were not obtained after applying the smart SSR, the second method will be used. It is implemented by adding more signals using component encryption method, which is developed by Koranek [13]. This new method

Table 3.2: Smart Selection Strategies For Smart SSR

Selection Algorithm	**Description**
FrontLevelTwoGates	Selects a gate in 30% of front level at random, and replacement to two gates with high switching activity selected at random (± 1 level)
MiddleLevelTwoGates	Same as FrontLevelTwoGates except the the gate in 30% of middle level is selected
RearLevelTwoGates	Same as MiddleLevelTwoGates except the the gate in 30% of rear level is selected
FrontRearLevelTwoGates	Same as RearLevelTwoGates except the the gate in 30% of both front and rear level is selected

is referred to as smart Component Encryption (Smart CE). Its difference is also to add the selection strategy like the smart SSR. The original version of component encryption method covers the all identified components to encrypt, but smart CE can select the components depending on the selection strategies as shown the Table 3.3. With this new method, it is expected to increase the number of signals by encryption and decryption process between components.

Table 3.3: Smart Selection Strategies For Smart Component Encryption

Selection Algorithm	**Description**
FrontLevelComponents	Selects components in 30% of front level at random, and replacement to two gates with high switching activity selected at random (± 1 level)
MiddleLevelComponents	Same as FrontLevelComponents except the the components in 30% of middle level is selected
RearLevelComponents	Same as MiddleLevelComponents except the the components in 30% of rear level is selected

3.5 Evaluation and Validation

The three targets for evaluation and validation are summarized as follows:

1. **Accuracy of Signature Detection**

   :The identified signature is evaluated using static and dynamic approach

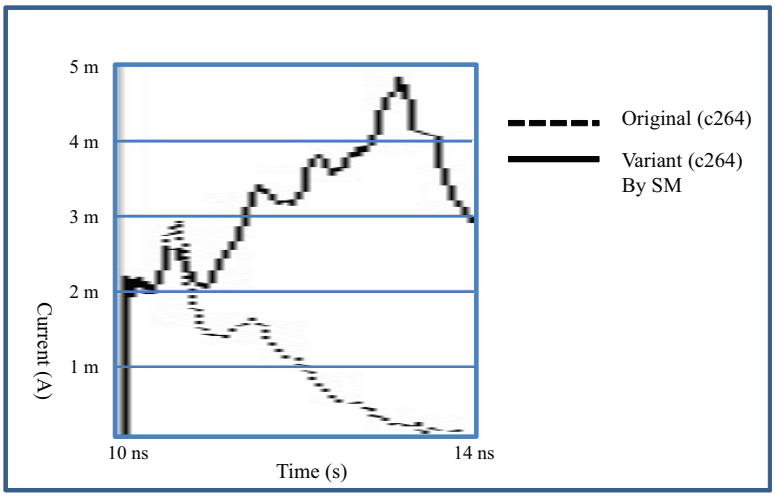
2. **Availability of Signature Manipulation**

   :The proposed signature manipulation methods is evaluated by estimating the amount of variation using static and dynamic analysis.

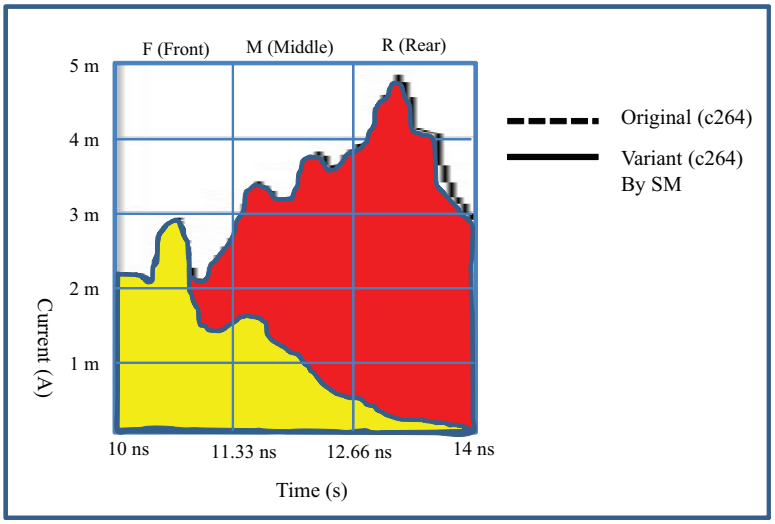
3. **Verification of the Final Circuit Variant**

   :The proposed signature manipulation method is applied on RSA circuit on Xilinx Virtex5 FPGA against adversarial power analysis.

Firstly, for evaluating accuracy of signature detection, the circuit signatures from SID is compared with the results of the dynamic simulation using SPICE under pseudo-random and user-defined inputs. Secondly, the new proposed signature manipulation method is evaluated by static and dynamic approach in terms of signature changing property and efficiency. For evaluating manipulating property, mathematical technique is used. Figure 3.19 shows an example of the evaluation technique for the circuit variant to measure how much power signature is increased by the three part of the time interval. Figure 3.19(b) describes how to divide the time interval and the increased area between the original and the variant. From the increased ratio in Figure 3.20, it is noticed that the power signature at rear part of the circuit variant is increased more than 19 times from the original one.Such efficiency level is evaluated by comparing the increased total switching activity by SID, the increased number of levels, and the increased number of circuits. Additionally, the measured signature of a variant generated by the previous algorithms is evaluated in terms of how much the signature is changed compared to original one. Lastly, the final circuit variant still need to be validated if it provides suitable transformation of power signature which an adversarial power analysis does not recognize. Thus, side-channel analysis is conducted based on the FPGA based encryption system developed by Falkinburg [4]. For installing the obfuscated version of a circuit into the test-bed, the circuit is described in VHDL format. From this implementation, it is validated if the circuit variant generated by SM provides the suitable protection of the secret key against adversarial power analysis.



(a)



(b)

Figure 3.19:
(a) Two power signatures measured by SPICE
(b) Comparing of two signatures

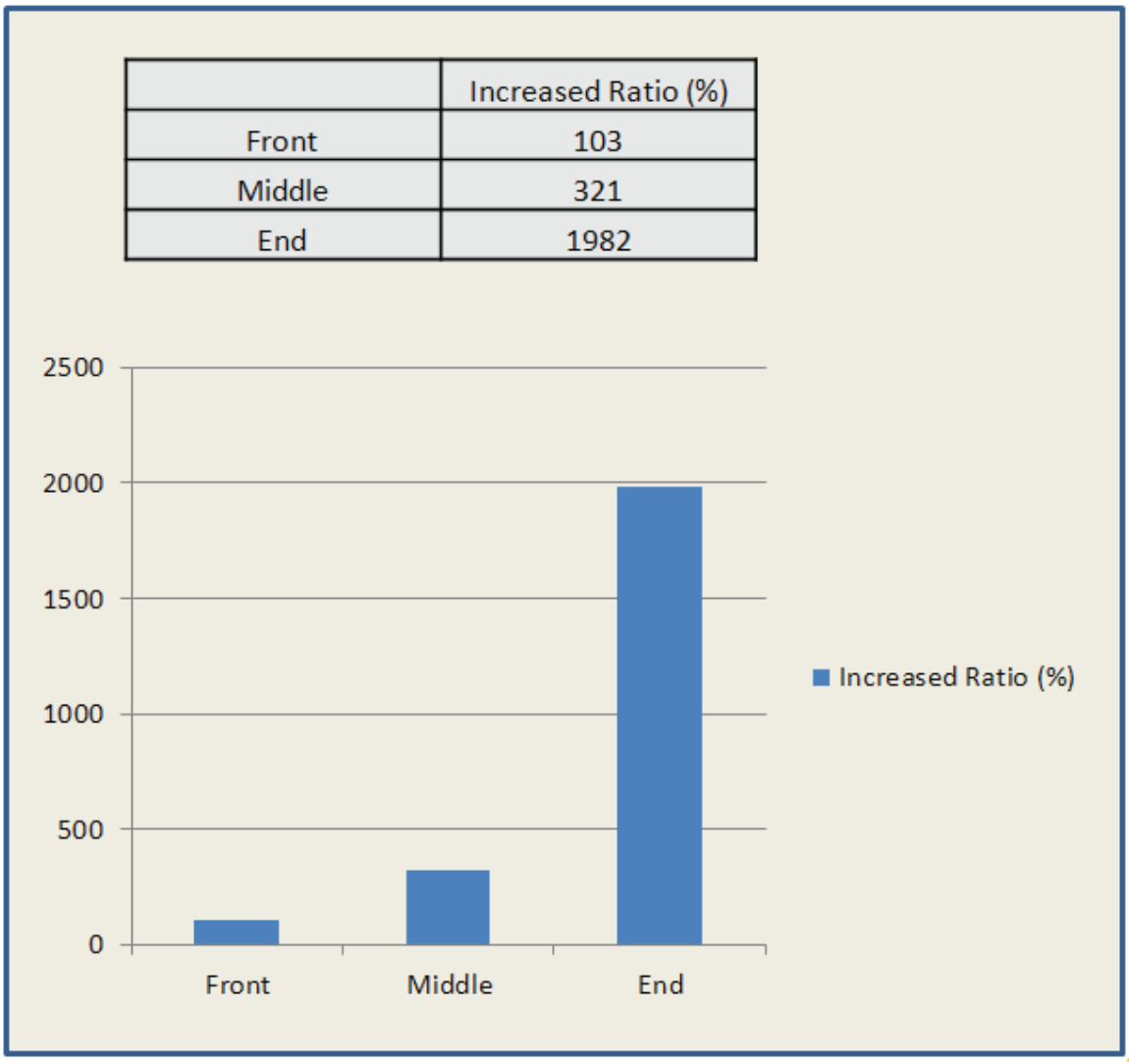


Figure 3.20: Increased power ratio by the divided time interval

3.6 Summary

This chapter defines the new methodology for power signature estimation and power signature manipulation by the four steps. Firstly, power signature estimation and simulation processes are performed by using static and dynamic approach respectively. Second process is characterization and classification. It is also performed using static and dynamic technique based on the four types of power signature patterns. Thirdly, implementation and manipulation are performed. In this process, signature manipulator (SM) is designed to transform the class of power signature. Lastly, evaluation and validation is conducted by comparing the changed signature from the original one and using FPGA based physical encryption system.

IV. Analysis and Results

This chapter first discusses the test cases for experiment and power signature estimation for the test cases using static and dynamic approach in Section 4.1 and 4.2. Section 4.3 provides the power signature chracterization for the test cases determined by static and dynamic approach, and Section 4.4 shows the changed power signature transformed by the power signature manipulation method in SM. Lastly, The final variants for an each test case are evaluated in Section 4.5 by comparing with the original signature, and one of the test cases is validated using encryption system to determines if the proposed signature method can hide the original power signature of circuits against side-channel analysis.

4.1 Test cases

4.1.1 c264 : 4-bit multiplier. The c264 circuit is a combinational 4-bit multiplier which is small version of the c6288 16-bit multiplier. It consists of four half-adder and eight full-adder components. The Figure 4.1 represents c264 using circuit logic gates. As a role of a multiplier, it can be performed as a binary multiplication.

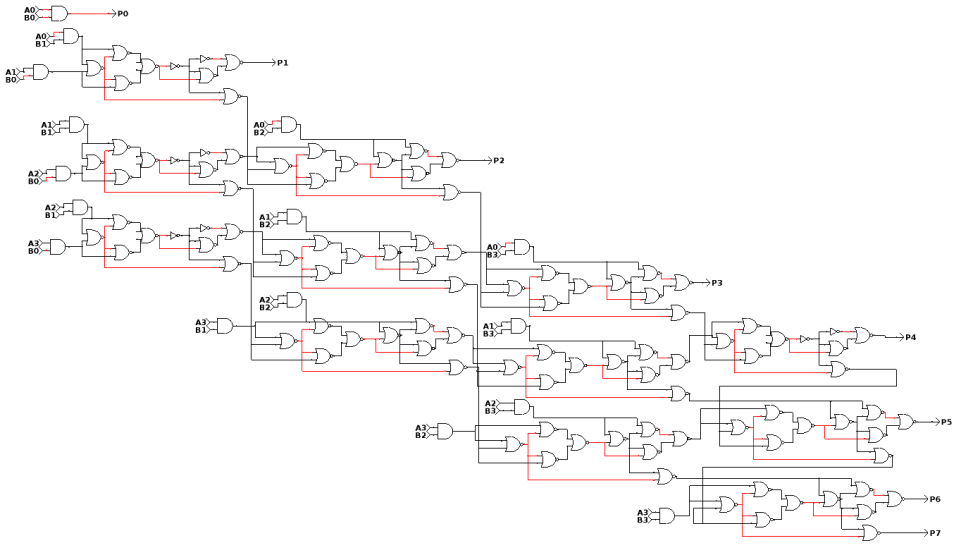
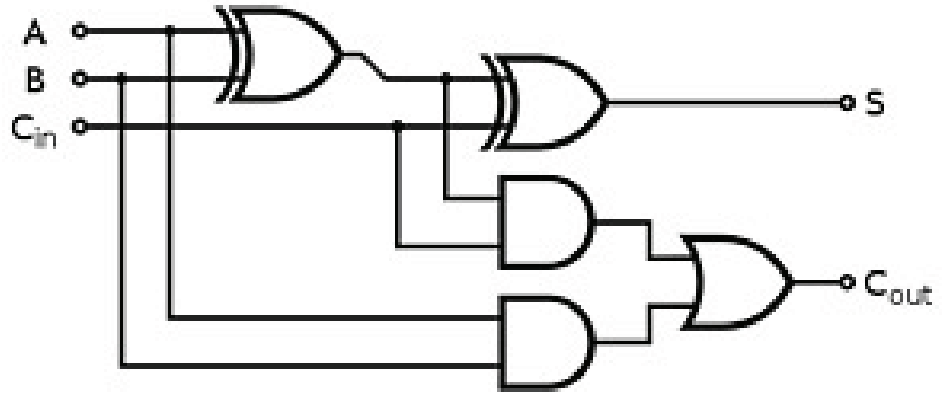
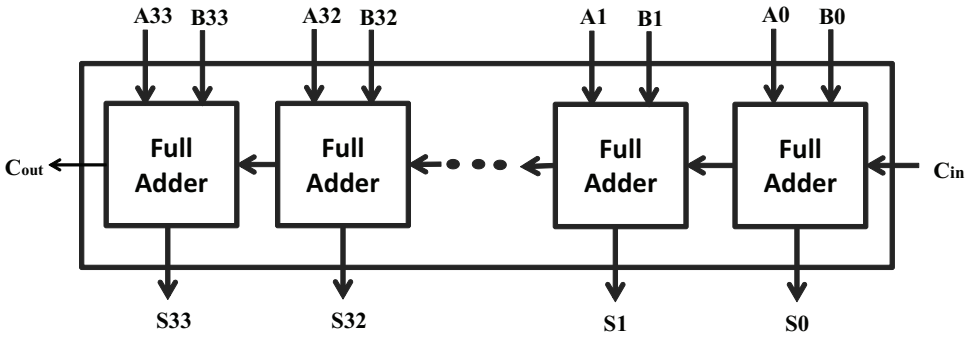


Figure 4.1: The 4-bit multiplier represented using circuit logic gates



(a)



(b)

Figure 4.2: (a) 1 bit Full Adder(b) 34 bit Ripple Carry Adder

4.1.2 34-bit Ripple Carry Adder (RCA). 34-bit RCA is builded as a cascade from 34 1-bit full-adders as shown in Figure 4.2. Such RCA is performed as a binary adder in which the carry at each stage of addition must propagate or ripple through the succeeding stages of addition in order to form the result.

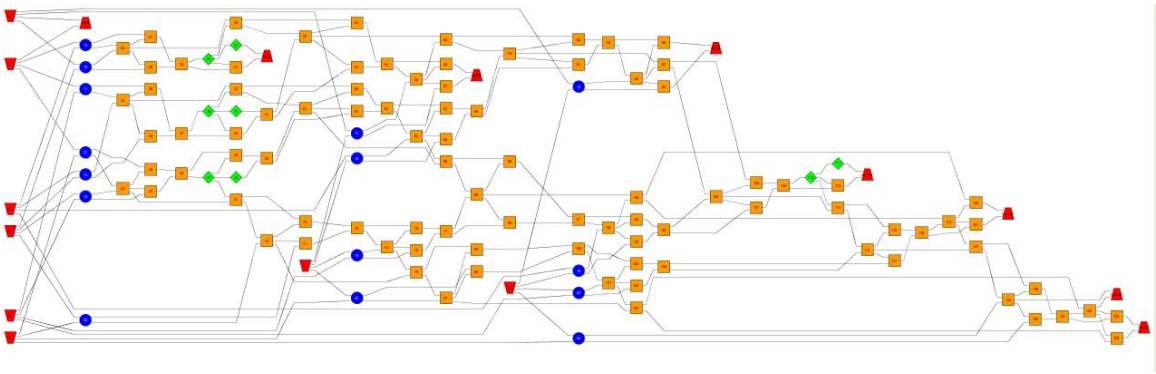
4.2 Estimation and Simulation Results

This section examines the original power signature of two test cases by using both static and dynamic estimation techniques. In static approach, SID is utilized to generate the power signature without input pattern. On the other hand, dynamic technique is required to consider input patterns. Thus, it is simulated under both pseudo random inputs and user-defined input patterns in this section.

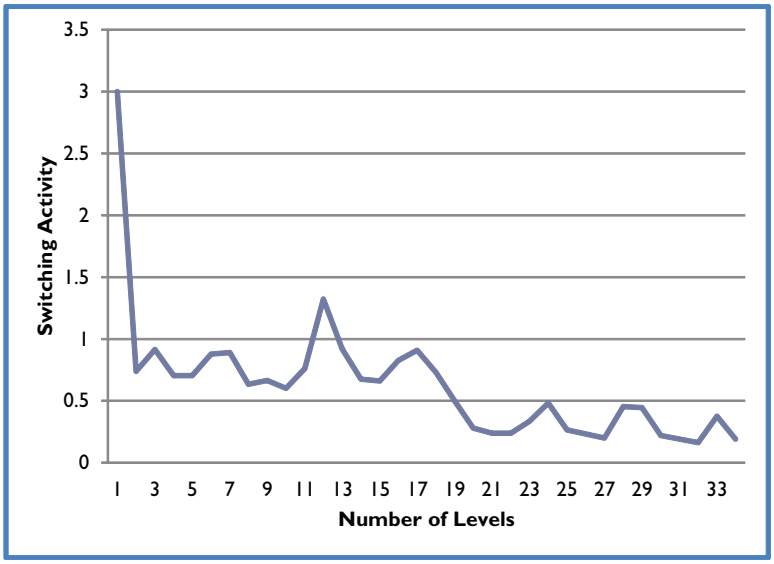
4.2.1 c264 : 4-bit multiplier. The Figure 4.3 shows a graph-based representation of the 4-bit multiplier circuit exported from CORGI and its power signature obtained with SID system. As you can see the overall structure of 4-bit multiplier in the Figure 4.3 (a), most gates are located at the front part of the circuit. Therefore, the power signature from static analysis represents front loaded shape as shown in Figure 4.3 (b).

Figure B.1 shows the power signature of 4-bit multiplier generated by ELDO simulator under pseudo random-bit sequence which is implemented in JAVA, which called PRBS (Pseudo Random Bit Generator Source). In Figure B.1, such six different power signatures are extracted from the current waveform. From the result, it is noticed that the input pattern highly influences on the shape of power signature. Thus, it is not easy to define an typical power signature for a circuit by the dynamic simulation.

Additionally, user-defined input patterns are considered to simulate 4-bit multiplier. As previously stated in Chapter III, considering all possible input variation to a circuit is not possible. Thus, this research tries to generate worst-case scenarios of power consumption for a test case. The worst-case means generating a power signature with large power consumption. It is predicted to generate a large magnitude of power signature. In order to consider the worst-case input sequences, it is needed to know how binary multiplication can be performed. Consider the Figure 4.5, binary multiplication of two positive 4-bit integer values. In the course of multiplying two binary numbers, each bit in the multiplier is multiplied with the multiplicand. Each of the four product is aligned according to the position of the bit in the multiplier that is being multiplied with the multiplicand. The four resulting products are added to form the final product. From this multiplication process, it is noticed that if the multiplier bit is a zero, then the product is zero. It means that assigning one-bit to multiplicand and multiplier instead of zero-bit is more likely to affect power consumption of a multiplier circuit. Therefore, The Table 4.1 defines the test cases for generating worst-case of 4-bit multiplier.

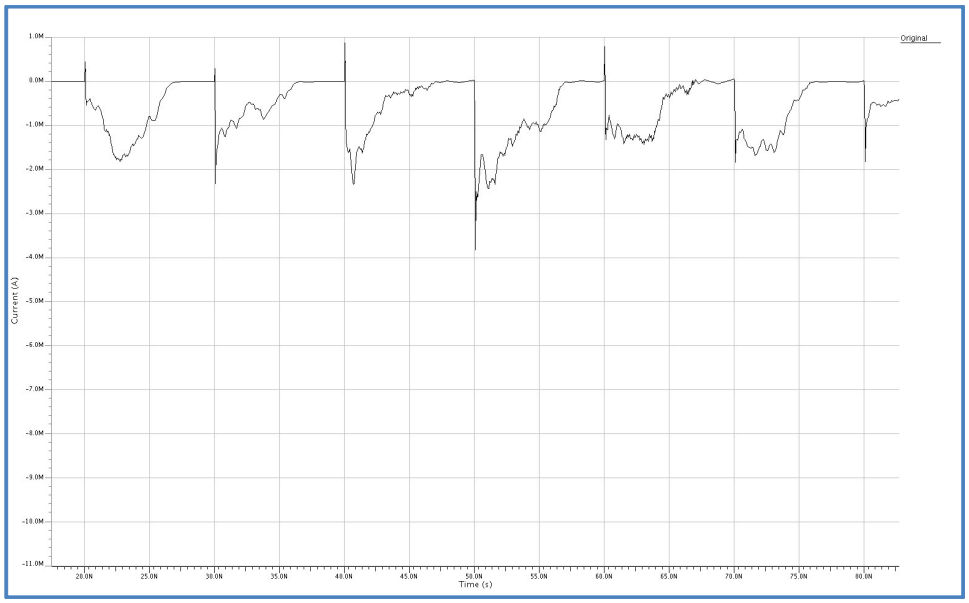


(a)

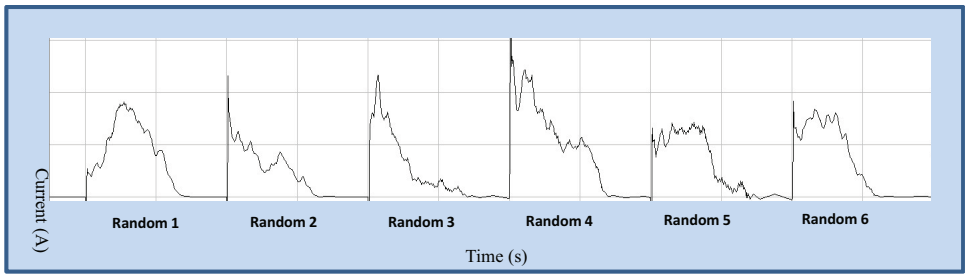


(b)

Figure 4.3: c264 Power Signature Estimation in Static Approach
(a) A graph-based representation of the 4-bit multiplier circuit
(b) A power signature of the 4-bit multiplier circuit generated by SID



(a)



(b)

Figure 4.4:
(a) Six Different Power Signatures of c264 by Pseudo-Random Input Patterns
(b) Six Power Signatures of c264

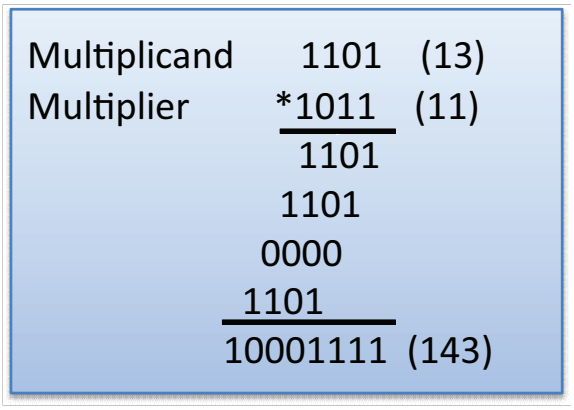


Figure 4.5: Binary multiplication of two positive 4-bit integer values

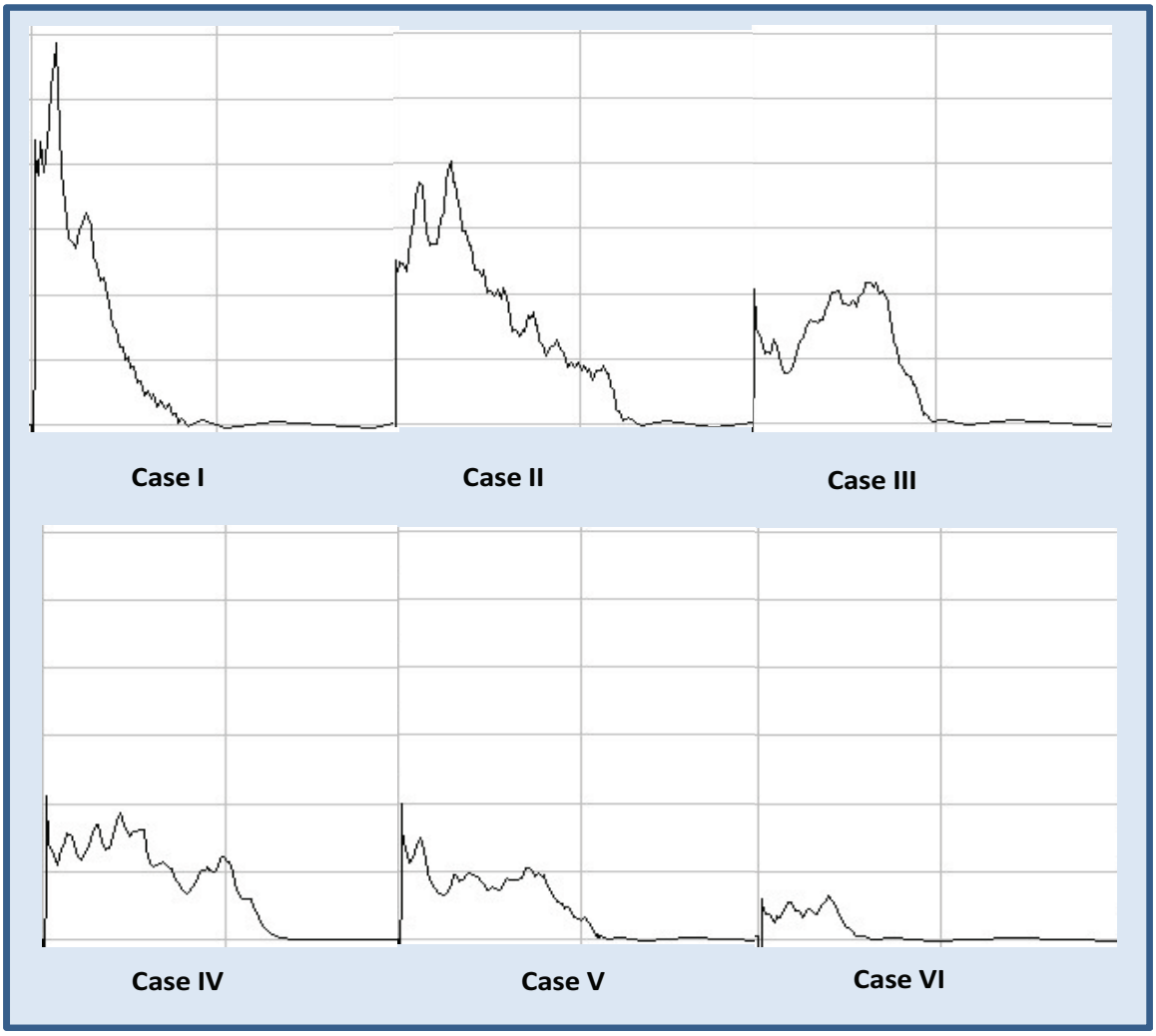


Figure 4.6: Six Different Power Signatures by User-defined Input Sequence

Table 4.1: User-defined Input Sequence for c264

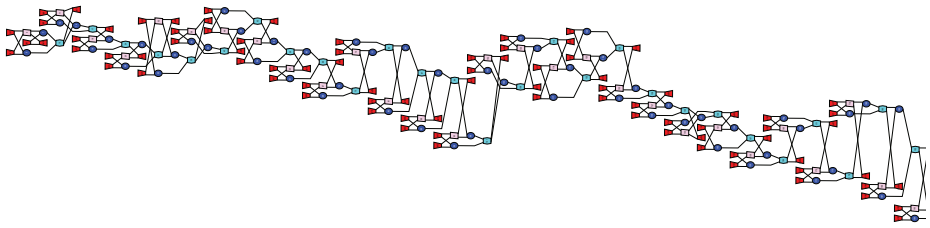
Scenarios	**Multiplicand**	**Multiplier**
Initial Case	0000	0000
Case I	1111	1111
Intermediate Case	0000	0000
Case II	1110	1110
Intermediate Case	0000	0000
Case III	1100	1100
Intermediate Case	0000	0000
Case IV	0011	0011
Intermediate Case	0000	0000
Case V	1100	0011
Intermediate Case	0000	0000
Case VI	0001	0001

Figure 4.6 shows the six different power signatures for 4-bit multiplier with pre-defined input sequences based on the Table 4.1. As expected, the case I and II show the worst-case model due to the multiplication operation of the large numbers. Based on this result, it is realized that input patterns significantly influence on the shape of the power signature.

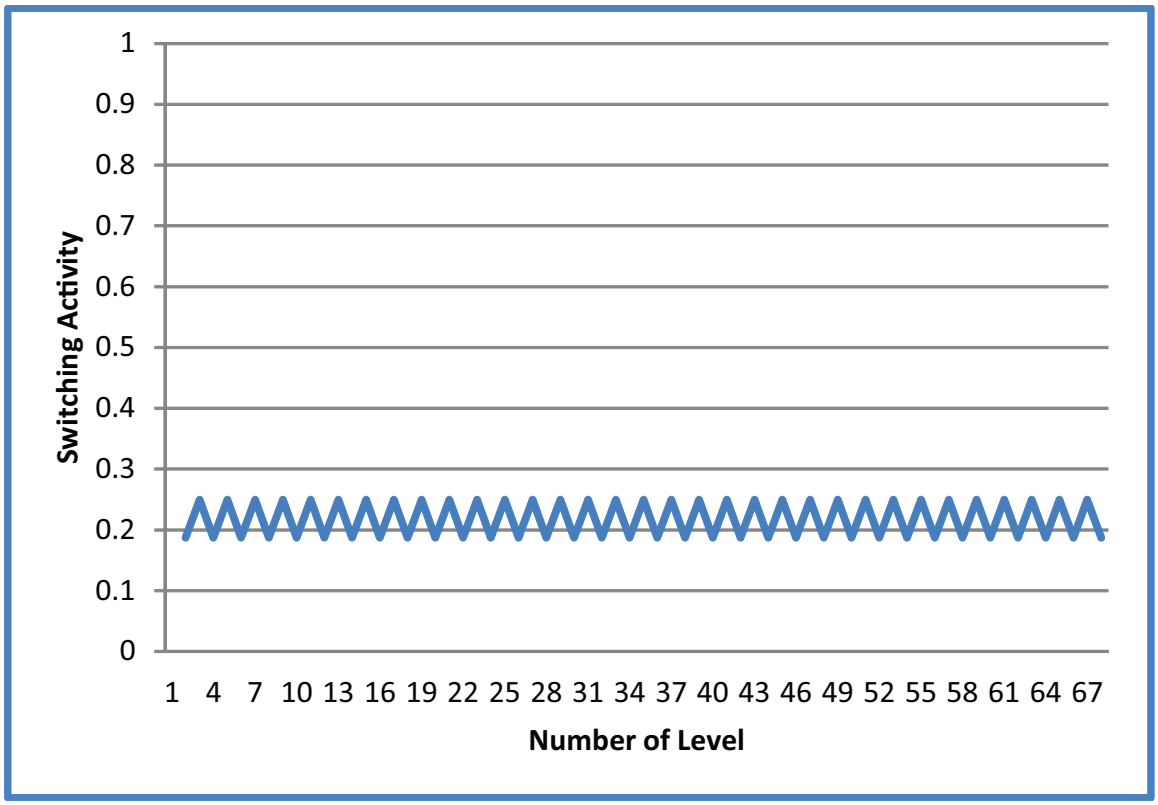
4.2.2 34-bit Ripple Carry Adder(RCA). Figure 4.7 (a) shows a graph-based representation of 34-bit RCA which is generated by CORGI. The layout of such RCA is very simple since it consists just 34 1-bit full adders in cascade. Based upon this layout, SID generates the predictable power signature as shown in Figure 4.7 (b). But, SID fails to detect the type of signature for this case since the region of three divided part in RCA are the same each other.

Dynamic simulation is performed as the same way of c264. Figure 4.8 represents six different power signatures under pseudo-random. As noticed, the results of this simulation looks different from the result of SID, because statistical estimation using SID does not consider the initial power consumption which normally needs large power consumption.

34-bit RCA has total 69 inputs which consists two 34-bit inputs and one carry-in. Basically, 2^{69} input combinations have to be considered to cover all possible cases. As mentioned many times, considering all possible input patters is impractical. Thus,

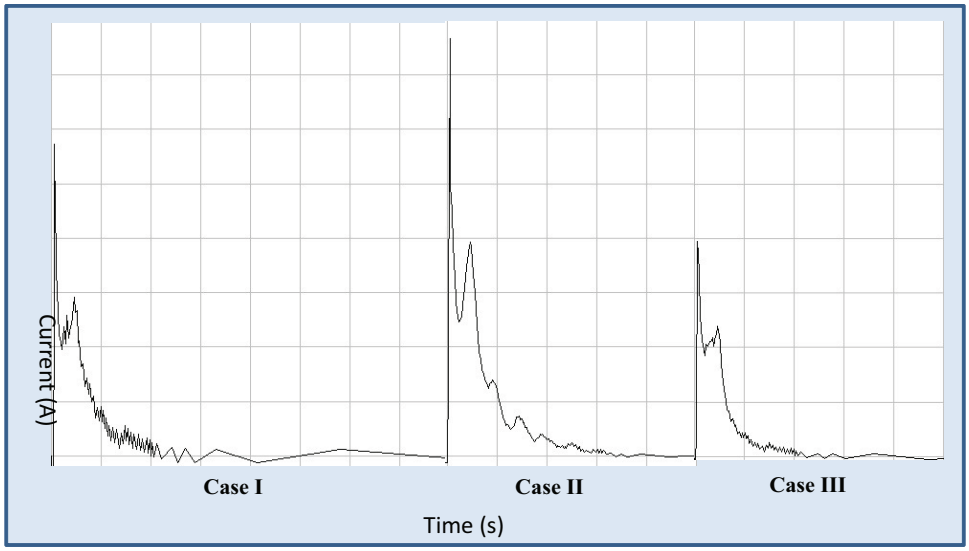


(a)

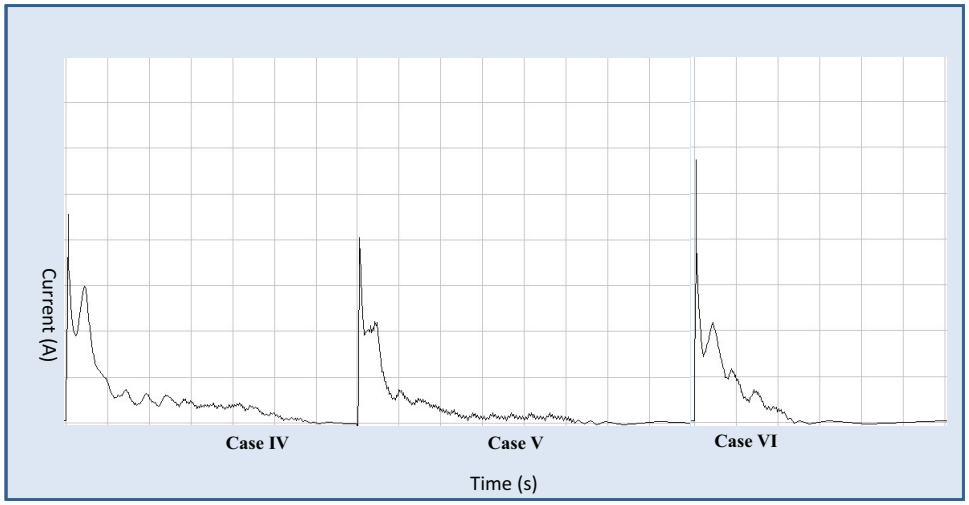


(b)

Figure 4.7: Power Signature Estimation of 34-bit RCA in Static Approach
(a) A graph-based representation of 34-bit RCA
(b) A power signature of 34-bit RCA generated by SID



(a)



(b)

Figure 4.8: Power Signatures of c264 by Pseudo-Random Input Patterns
(a) Power Signatures of 34-bit RCA by Pseudo-Random Input Patterns Case I through Case III
(b) Power Signatures of 34-bit RCA by Pseudo-Random Input Patterns Case IV through Case VI

Table 4.2: User-defined Input Sequence for 34-bit RCA

Scenarios	**Carry-in**	**A(34bits)**	**B(34bits)**
Initial Case	0	0-0	0-0
Case I	0	1-1	1-1
Intermediate Case	0	0-0	0-0
Case II	0	1-1	0-0
Intermediate Case	0	0-0	0-0
Case III	0	0-0	1-1
Intermediate Case	0	0-0	0-0
Case IV	0	0(30%) 1(70%)	0(30%) 1(70%)
Intermediate Case	0	0-0	0-0
Case V	0	0(70%) 1(30%)	0(70%) 1(30%)
Intermediate Case	0	0-0	0-0
Case VI	0	0(50%) 1(50%)	0(50%) 1(50%)

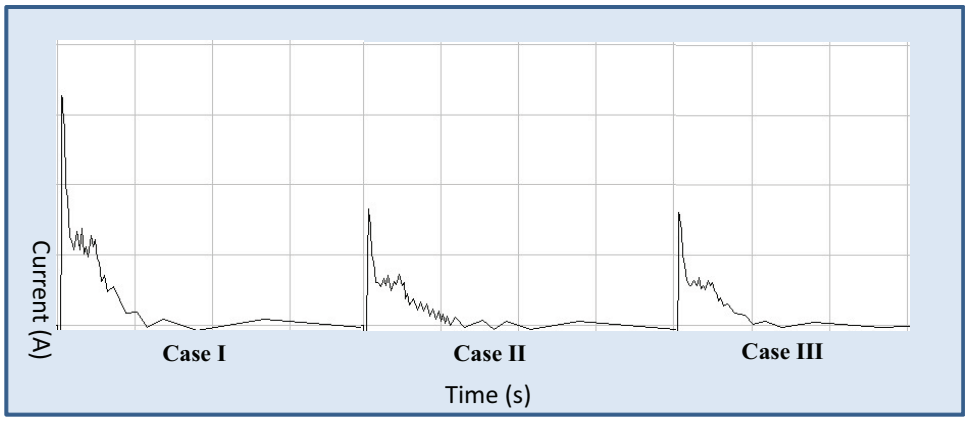
this research defines some input patterns as seen in Table 4.2. Such input patterns consists 50%/50% and 30%/70% chance of 1 and 0 input vice versa.

4.3 Characterization and Classification Results

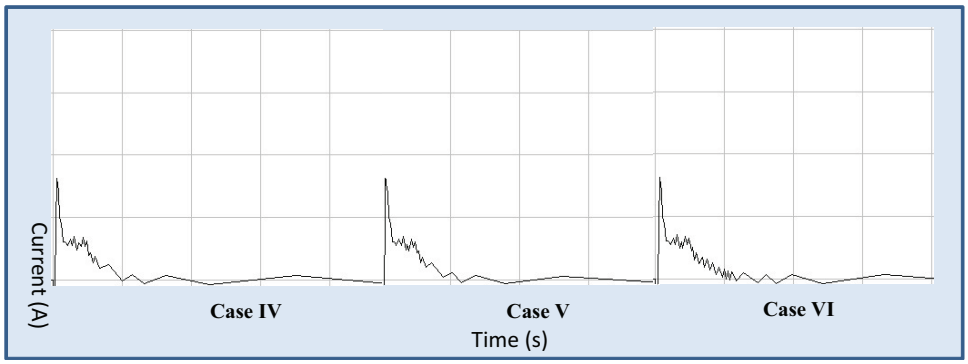
In this section, the results from the previous section is characterized by using the four types of power signature pattern.

4.3.1 c264 : 4-bit multiplier. This characterization is accomplished by calculating the region of the each part of signature and comparing them. Figure 4.10(b) shows how to divide the region from the measured signature for c264. With such a comparison method, SID provides result, F (Front) > M (Middle) > R (Rear), which indicates the power signature of c264 is Type I (Front-loaded signature).

The 12 signatures from dynamic simulation under both pseudo-random and user-defined inputs are evaluated by the method as mentioned in Chapter III. Table 4.3 shows the specific type according to each cases. Seven cases shows Type I and five cases is Type II.

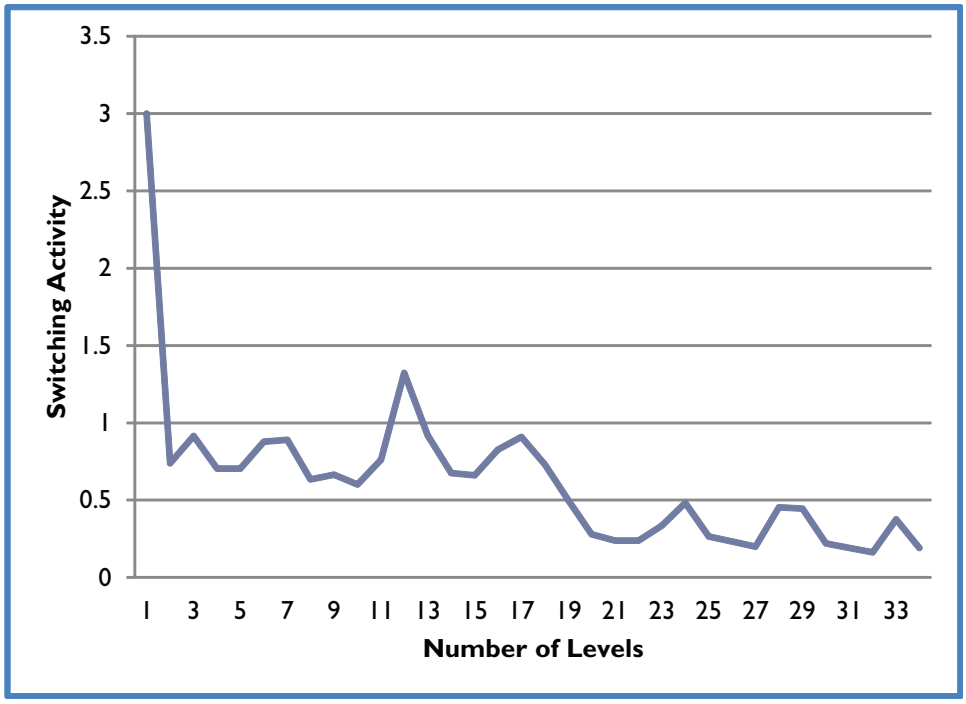


(a)

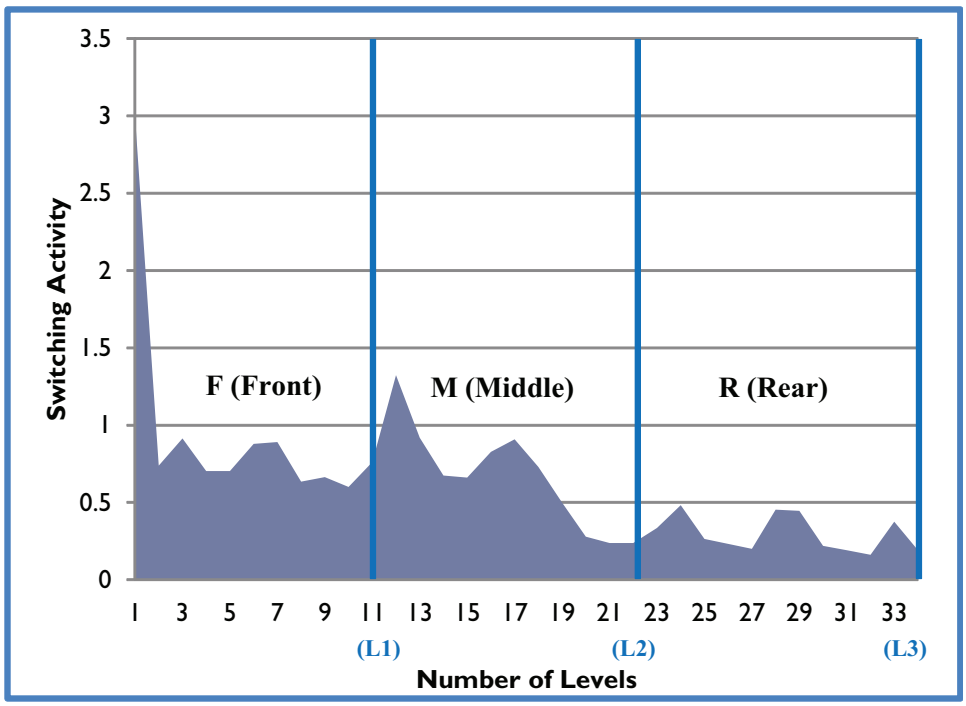


(b)

Figure 4.9: Power Signatures of c264 by User-defined Input Patterns
(a) Power Signatures of 34-bit RCA by User-defined Input Patterns Case I through Case III
(b) Power Signatures of 34-bit RCA by User-defined Input Patterns Case IV through Case VI



(a)



(b)

Figure 4.10:
(a) Power signature of 4-bit multiplier generated by SID
(b) Power signature of 4-bit multiplier divided into three parts by the number of levels

Table 4.3: Power Signature Pattern for c264

Scenarios	Signature Type	Scenarios	Signature Type
User-defined Case I	Type I	**Random Case I**	Type II
User-defined Case II	Type I	**Random Case II**	Type I
User-defined Case III	Type II	**Random Case III**	Type I
User-defined Case IV	Type I	**Random Case IV**	Type I
User-defined Case V	Type I	**Random Case V**	Type II
User-defined Case VI	Type II	**Random Case VI**	Type II

Table 4.4: Power Signature Pattern for 34-bit RCA

Scenarios	Signature Type	Scenarios	Signature Type
User-defined Case I	Type I	**Random Case I**	Type I
User-defined Case II	Type I	**Random Case II**	Type I
User-defined Case III	Type I	**Random Case III**	Type I
User-defined Case IV	Type I	**Random Case IV**	Type I
User-defined Case V	Type I	**Random Case V**	Type I
User-defined Case VI	Type I	**Random Case VI**	Type I

4.3.2 34-bit Ripple Carry Adder (RCA). For 34-bit RCA, it is noticed that applying static technique using SID is impractical since it provides the same value in each part of the circuit. It means the shape of power signature from SID is flat line. In this case, analyzing results of dynamic simulation provides better accuracy for recognizing the type of power signature. Table 4.4 shows results. All cases for 34-bit RCA show Type I.

4.4 Implementation Results

This section mainly shows the results of the new power signature manipulation method.

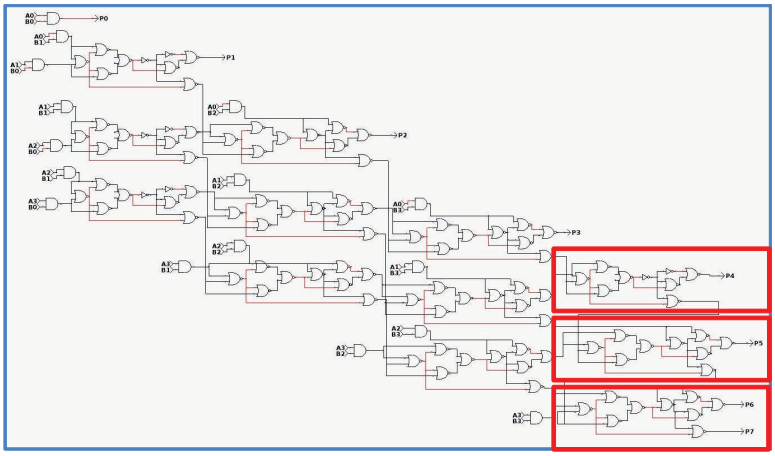
4.4.1 c264 : 4-bit multiplier. Figure 4.11 and 4.12 shows one of the manipulation cases for c264. In order for the original power signature to Type III, smart CE and smart SSR are applied to increase both internal signals and gates at the rear levels of a circuit. With smart component encryption, three components at the rear part of circuit are encrypted as shown in Figure 4.12 (a). And then, the encrypted circuit variant again is obfuscated by smart SSR algorithm with rear-level two gates option to increase the number gates iteratively as shown in Figure 4.12 (b).

4.4.2 34-bit Ripple Carry Adder (RCA). Figure 4.13 shows the result of 34-bit RCA obfuscated by smart SSR toward Type II and Type III respectively.

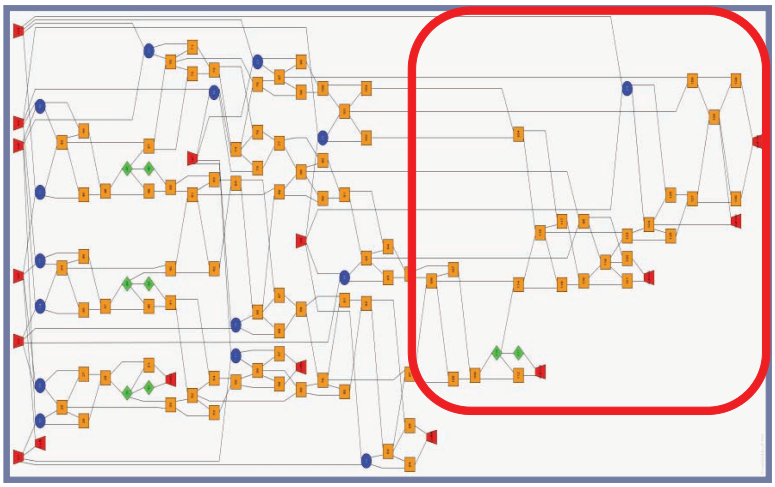
4.5 Evaluation and Validation Results

This section discusses about evaluation and validation results in terms of three criteria as mentioned in Chapter III.

4.5.1 Accuracy of Signature Detection. In this section, signature detection is validated in terms of accuracy. It is accomplished by comparing the results from between the static estimation and the dynamic simulation. With comparing the results between static and dynamic approach, it can be validated how much SID system can be applied in this domain of research. If it were accurate enough for measuring power signature of a circuit, it could significantly save time and effort without having all possible pattern-dependent simulation processes. According to the two test-cases in this research, it is concluded that static approach using SID does not provide high accuracy for measuring power signature. But, it is realized that such a statistical approach is accurate enough to detect the high switching activities, which generates the peak point of power trace from dynamic simulation. Therefore, even if SID is limited on accuracy of power signature estimation, SID system can be utilized as a metric to evaluate how much switching activity is changed between the original and the variant circuit without exhaustive simulation using dynamic technique.

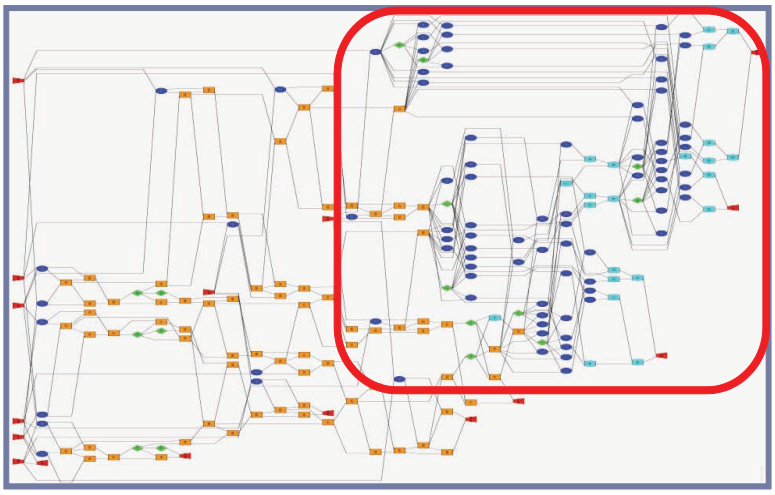


(a)

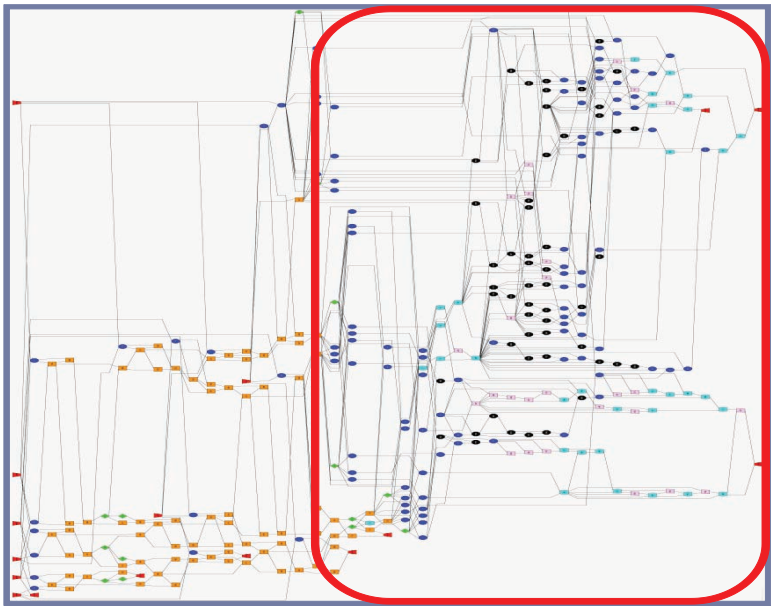


(b)

Figure 4.11: Smart Component Encryption and Smart SSR for c264
(a) The 4-bit multiplier represented using circuit logic gates with selecting three components
(b) A graph-based representation of the 4-bit multiplier circuit with selecting three component

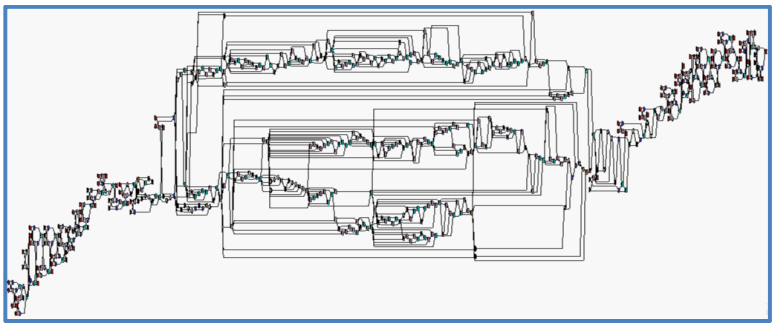


(a)

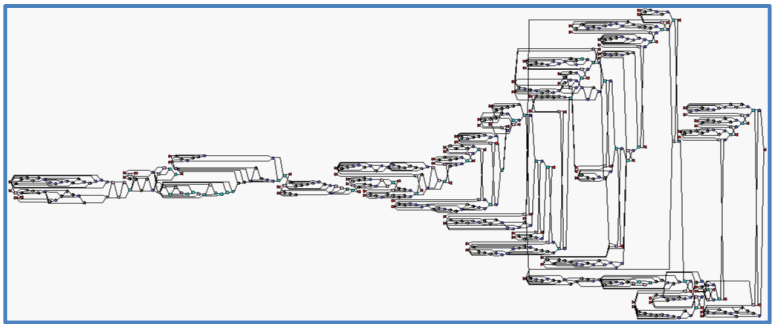


(b)

Figure 4.12: Smart Component Encryption and Smart SSR for c264
(a) A graph-based representation of the 4-bit multiplier circuit manipulated by SmartCE
(b) A graph-based representation of the 4-bit multiplier circuit manipulated by SmartSSR in 500 iterations



(a)



(b)

Figure 4.13: Smart SSR for 34-bit RCA
(a) A graph-based representation of the 34-bit RCA manipulated by SmartSSR with MiddleLevelTwoGates selection in 500 iterations
(b) A graph-based representation of the 34-bit RCA manipulated by SmartSSR with RearLevelTwoGates selection in 500 iterations

4.5.2 Availability of Signature Manipulation. To evaluate the manipulation techniques in CORGI 2.0 and CORGI 3.0 in terms of ability of changing the type of power signature, static technique is used in this research since it is confident that static approach can be utilized for measuring the total variation of switching activity. Table 4.5 provides the summary of experiments for c264. Even though security is a great concern of this research, efficiency of a circuit could not be underestimated in a real world. The feature size of circuit and total switching activity could be a good metric in terms of the efficiency of the final circuit variant. Excessive power consumption by the large switching activity is undesirable since high power dissipation causes overheating, which degrades performance and reduces chip lifetime. In this regards, Figure 4.14 through 4.16 shows total switching activity, level count, and gate count per algorithm respectively.

Table 4.5: Metrics of c264 variants

	Original	**Random SSR**					**Component Fusion**		**Component Encryption**		**Smart Component Encryption**		**Smart SSR**	
Metrics	**1**	**100**	**500**	**1000**	**2000**	**3000**	**1**	**2**	**1**	**2**	**1**	**2**	**100**	**500**
Trials	1	1					2		2		2		1	
Power Signature	TypeI	TypeI	TypeI	TypeI	TypeI	TypeI	TypeI	TypeI	TypeI	TypeII	TypeII	TypeII	TypeI	TypeII
Switching Activity increased	100%	180%	435%	701%	1190%	1637%	444%	516%	404%	516%	302%	387%	165%	430%
Inputs	8	8	8	8	8	8	8	8	8	8	8	8	8	8
Outputs	8	8	8	8	8	8	8	8	8	8	8	8	8	8
Gate count	124	260	452	569	755	881	147	164	154	150	151	162	245	430
Level count	34	61	163	256	407	554	36	37	38	55	36	38	175	232

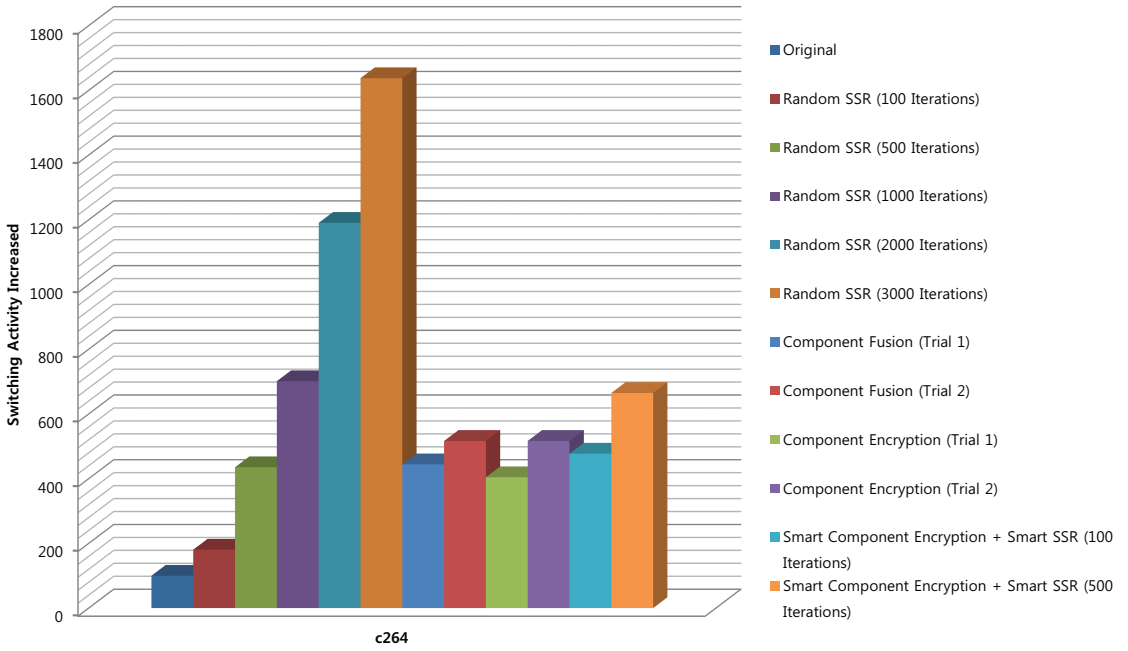


Figure 4.14: Total Switching Activities Increased per Algorithm produced by SID System

4.5.2.1 Random Sub-Selection And Replacement. As seen in Figure 4.14, random SSR can highly increase the total switching activity as increase the number of iterations. But such increased switching activities can not change the type of power signature from Type I. As seen in Figure A.1 through A.5 of Appendix A, there are no distinct peak points in the power signature by random SSR with from 100 iterations to 3000 iterations. Besides, it is noticed that random SSR significantly increases the number of level as well as the number of gates. In other words, it needs considerably high power compared to original one. The power signatures with more than 1000 iterations shows more than seven times large total switching activity, the number of level, and gates compared to the original circuit. From this observation, this research decides that more than 1000 iterations of SSR algorithm cannot be applied in new obfuscation algorithm. As seen in Table 4.5, however, it is realized that random SSR does not provide a suitable variant with changed the type of power

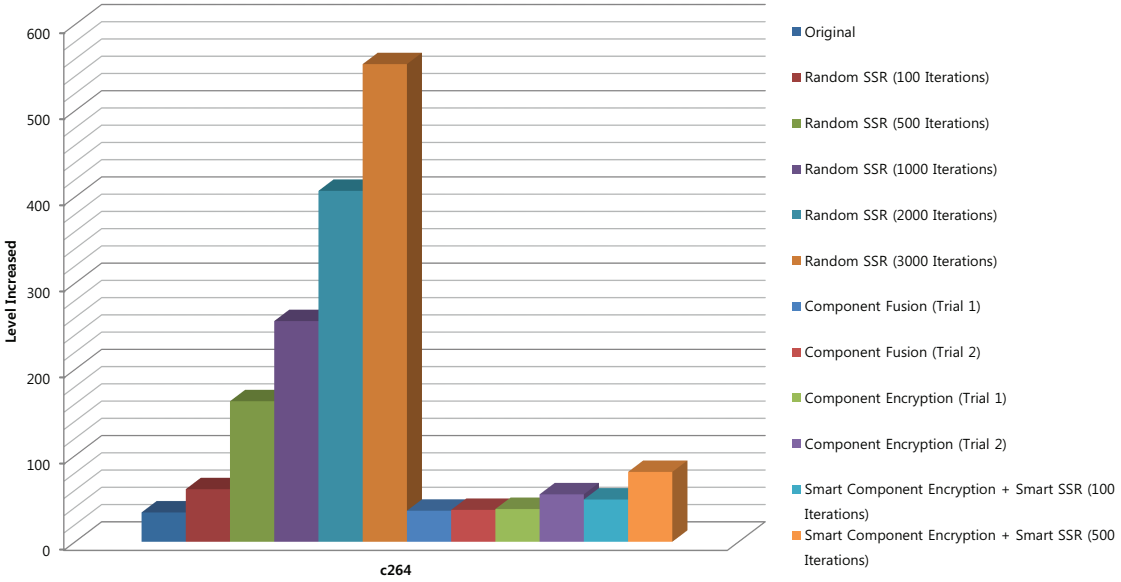


Figure 4.15: Level Increased per Algorithm

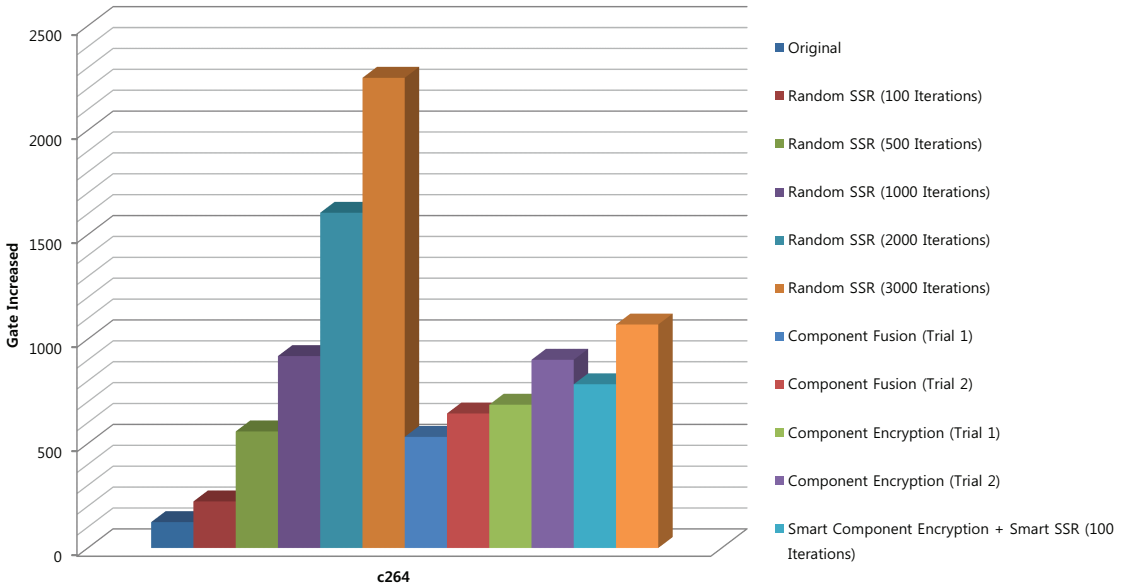


Figure 4.16: Gate Increased per Algorithm

signature from the original signature (Type I), and it does shows low efficiency due to the properties of highly increasing the number of gates and levels.

4.5.2.2 Component Fusion. As you can see the Figure A.6 and A.7 of Appendix A, the two trials shows additional peak points compared to the power signature generated by random SSR. Additionally, it is noticed that they provides a better efficiency because they do not highly increase the number of gates and levels from the original one due to the circuit synthesis algorithm. But, component fusion does also preserve the original power signature, Type I.

4.5.2.3 Component Encryption. From two trials of component encryption, the second trial finally provides the changed type of signature, Type II. As seen in Figure A.9 of Appendix A, overall shape of power signature is significantly changed compared to the original one. Besides, component encryption does not also significantly increase the number of levels and gates like component fusion. From this observation, it is decided that component encryption is the best candidate to be applied into the new power signature obfuscation method in this research.

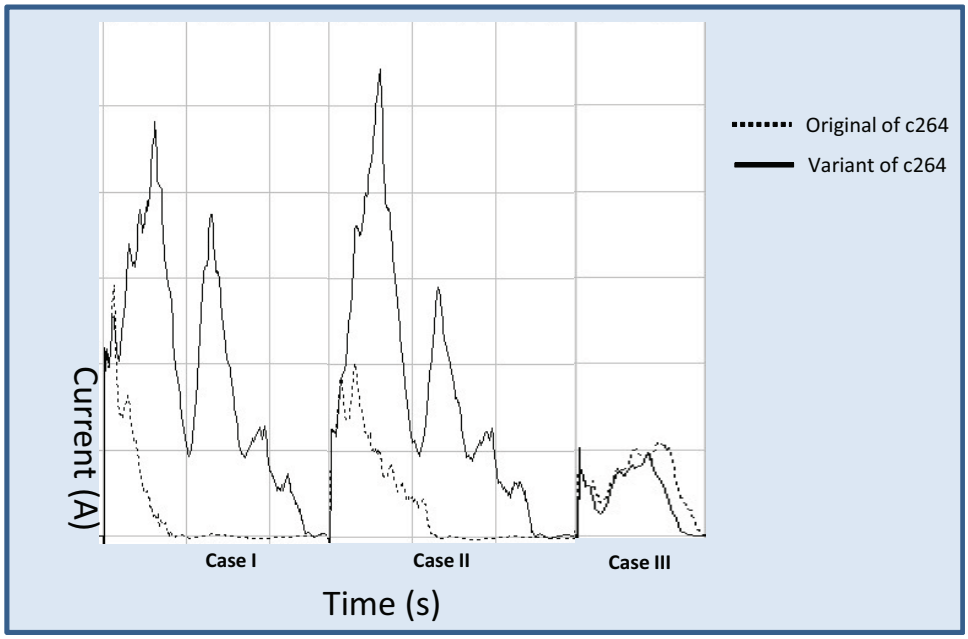
4.5.2.4 Signature Manipulator (SM). This section mainly discusses about how much the proposed methods can change the power signature from the original one for test cases. Figure 4.18 shows the comparison between the original signature and the final variant generated by new method for c264 under six different input patterns as defined in Section 4.3. For evaluating each cases, mathematical method is applied as mentioned in Chapter III. Figure 4.18 and 4.19 shows the comparison method and result for the case I of c264. According to the result in Figure 4.19, it is noticed that the power signature in the rear part is 19 times increased, which shows Type III. This case does successfully change the original power signature from Type I to Type III. Such an evaluation process proves that SM has an ability to intentionally manipulate the shape of original power signature. As the same process with the Case I of c264, evaluation is performed for c264 and 34-bit RCA. As seen in Table 4.6, five

Table 4.6: Evaluation of Changing Power Signature with Rear-level Selection Strategy for c264

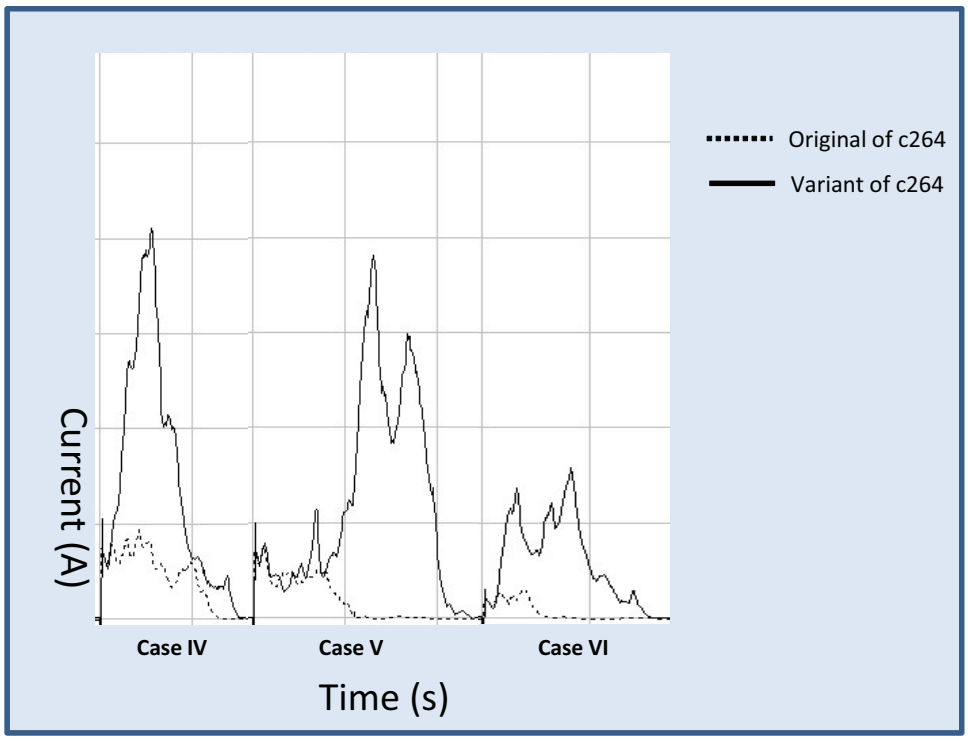
Scenarios	**Original Signature**	**Changed Signature**	**Front**	**Middle**	**Rear**
Case I	Type I	Type III	103%	321%	1982%
Case II	Type I	Type III	144%	417%	426%
Case III	Type II	Type II	171%	467%	198%
Case IV	Type I	Type II	173%	467%	199%
Case V	Type I	Type III	99%	134%	732%
Case VI	Type II	Type III	96%	378%	579%

cases out of six do successfully change the type of power signature. Only the case III does not provide suitable transformation of power signature from Type II. Table 4.7 and 4.8 provide the summary of evaluation for 34-bit RCA variant generated by smart SSR algorithm with MiddleLevelTwoGates and RearLevelTwoGates option respectively. For the cases of 34-bit RCA, SM system can completely change the type of power signature from Type I. Even if all cases transform the original power signature, the changed type is not consistent with the predicted one. For example, in Figure 4.6, although the smart SSR with MiddleLevelTwoGates option is performed toward Type II, all results provides Type III as unexpected. It is noted that there are three reasons for this unexpected results. Firstly, the changed type of power signature can be varied based on time interval. The evaluated results in this section are extracted based on the time interval of the original circuit. The second reason is that smart CE algorithm is not applied in this case of 34-bit RCA, because the components in RCA cannot be identified using CORGI. This issue is because Component ID system in CORGI does not have the components in the RCA. Before solving this issue, smart SSR is applied alone for this test-case. The third reason is that it is not easy to remove or decrease the initial peak point in the power signature since the proposed method only focuses on more increasing the switching activity than decreasing the one. Thus, the initial peak point is still remained after manipulation process.

4.5.3 Verification of the Final Circuit Variant. Lastly, the proposed power signature manipulation method still need to be validated if it provides suitable trans-

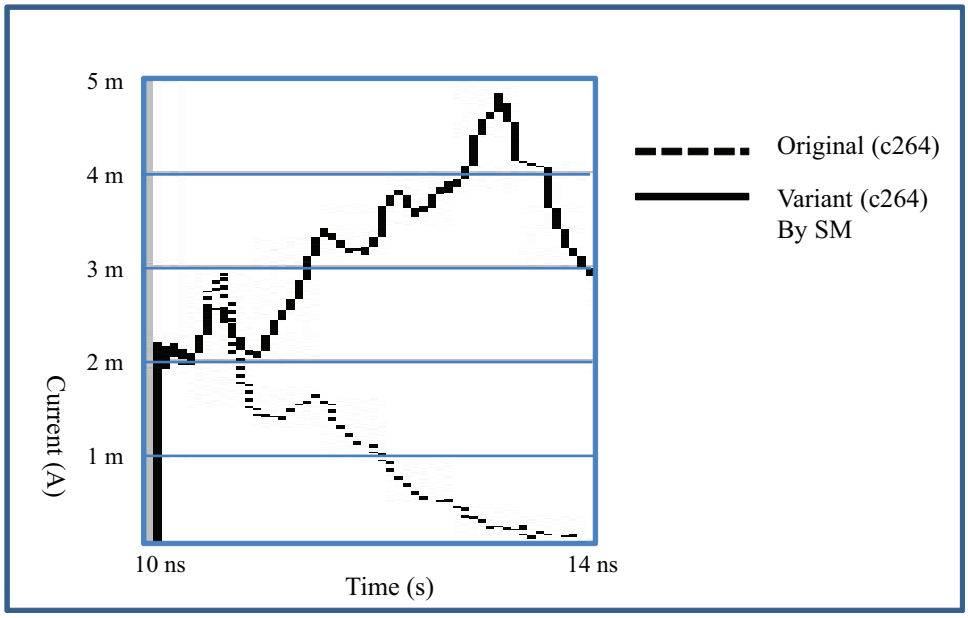


(a)

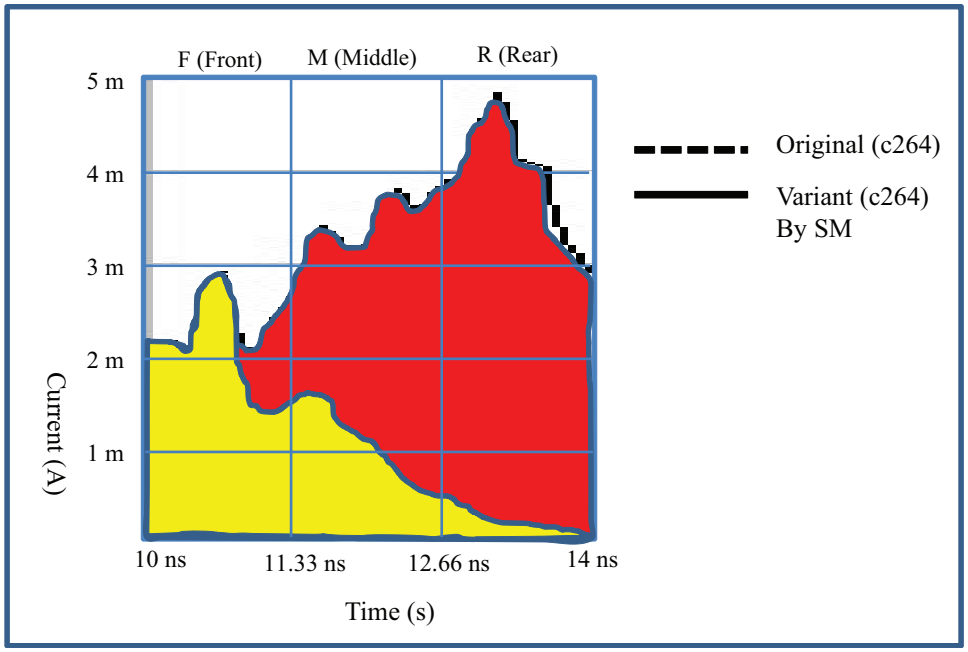


(b)

Figure 4.17: Comparison Between Original c264 and Obfuscated version of c264 by Smart Component Encryption and Smart SSR with Rear-level selection strategy
(a) Case I through Case III
(b) Case IV through Case VI



(a)



(b)

Figure 4.18: Comparison of Power Signature for case I between the original of c264 and the variant generated by Smart Component Encryption and Smart SSR with Rear-level selection strategy under user-defined inputs
(a) Two Power Signatures in Case I
(b) Comparion of the Amount of the Current with Color

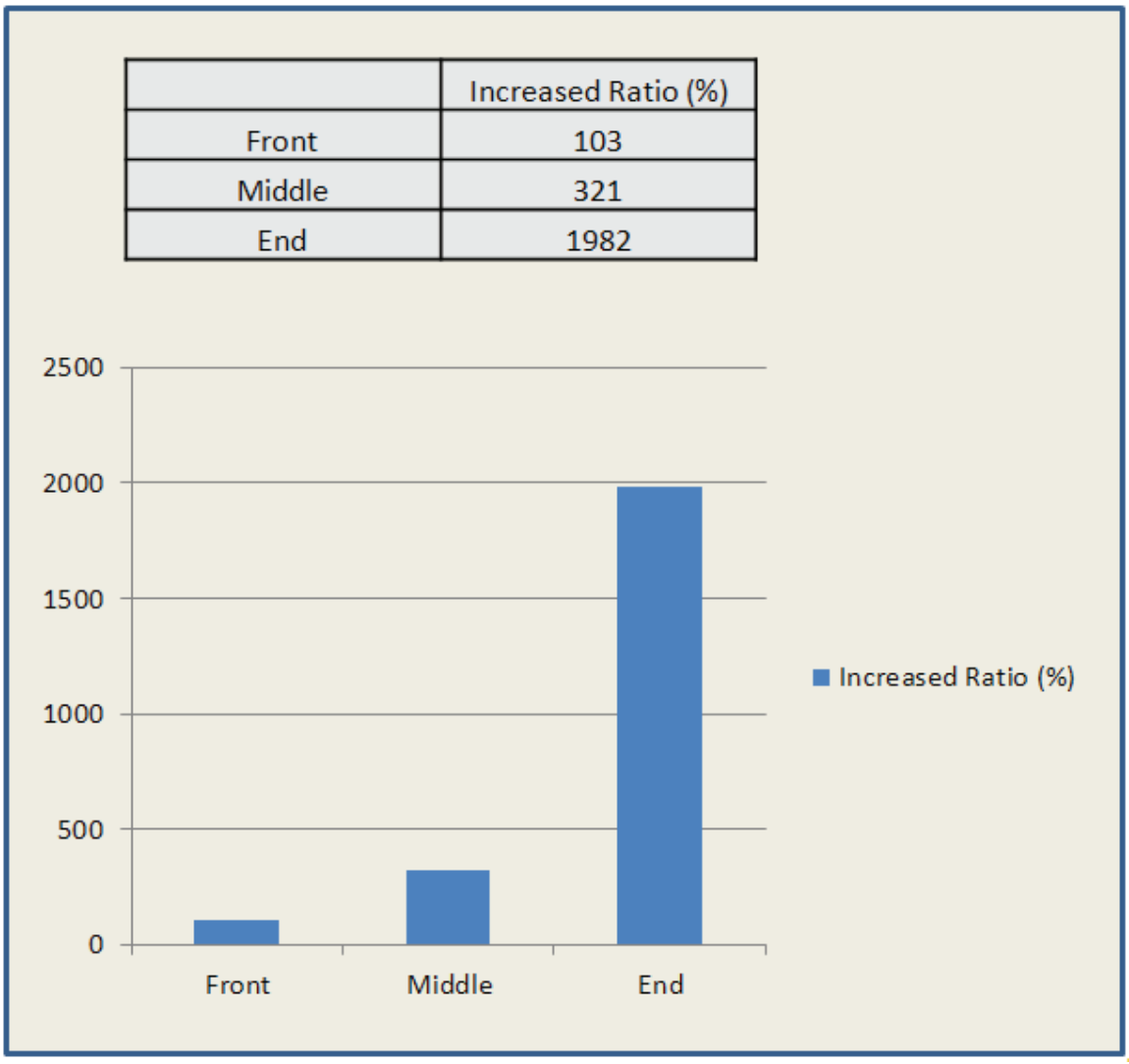
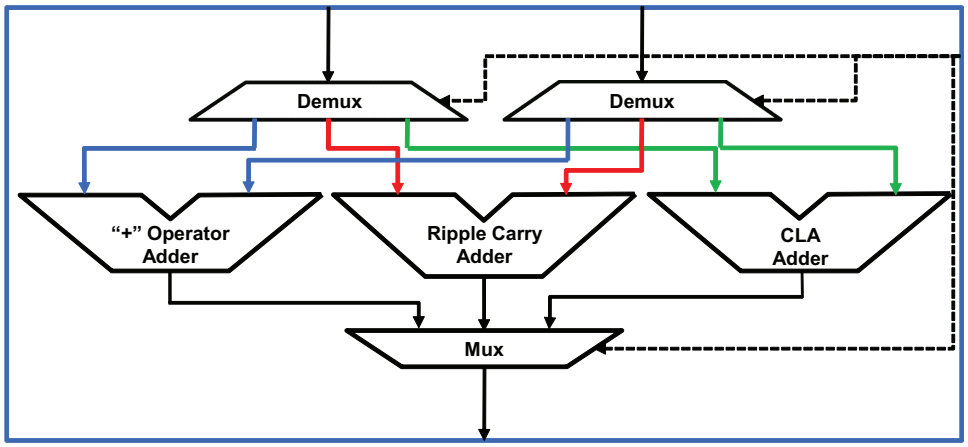
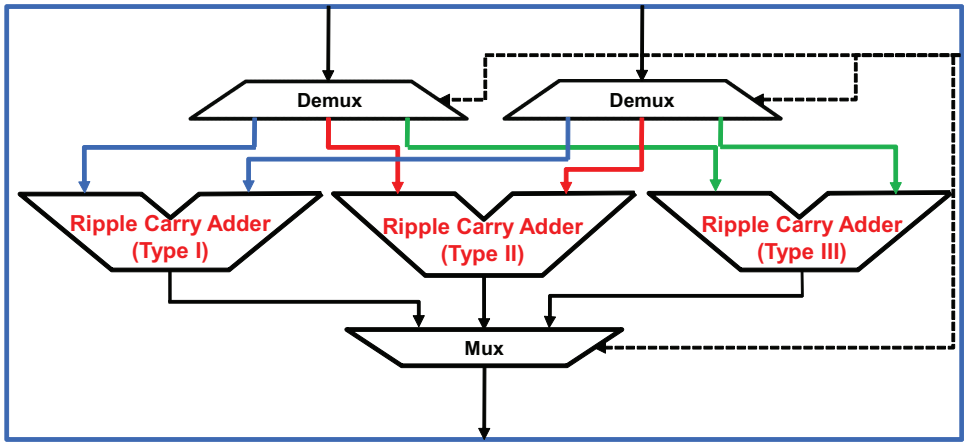


Figure 4.19: Evaluation for Case I of c264



(a)



(b)

Figure 4.20: Conducting side-channel analysis using FPGA based Encryption System designed by Falkinburg [4]
(a) A Polymorphic Circuit using Three Different Adders designed by Falkinburg
(b) A Polymorphic Circuit using Three RCA Adders with Type I, Type II, and Type III generated by Power Signatuer Manipulation Method

Table 4.7: Evaluation of Changing Power Signature with Middle-level Selection Strategy for 34-bit RCA

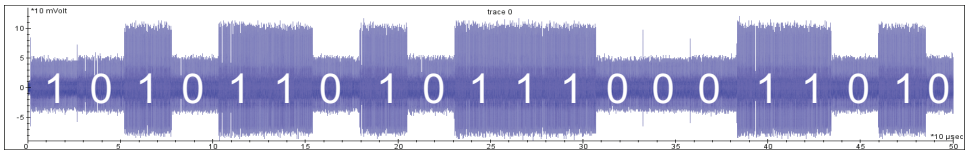
Scenarios	**Original Signature**	**Changed Signature**	**Front**	**Middle**	**Rear**
Case I	Type I	Type III	103%	135%	478%
Case II	Type I	Type III	81%	110%	200%
Case III	Type I	Type III	122%	168%	504%
Case IV	Type I	Type III	77%	107%	206%
Case V	Type I	Type III	98%	123%	326%
Case VI	Type I	Type III	68%	119%	152%

Table 4.8: Evaluation of Changing Power Signature with Rear-level Selection Strategy for 34-bit RCA

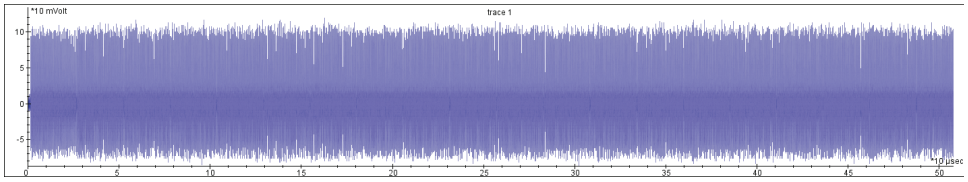
Scenarios	**Original Signature**	**Changed Signature**	**Front**	**Middle**	**Rear**
Case I	Type I	Type IV	267%	185%	565%
Case II	Type I	Type III	159%	307%	472%
Case III	Type I	Type III	259%	239%	1253%
Case IV	Type I	Type III	165%	302%	1103%
Case V	Type I	Type IV	283%	244%	487%
Case VI	Type I	Type III	141%	304%	468%

formation of power signature which an adversarial power analysis does not recognize. Thus, side-channel analysis is conducted based on the FPGA based encryption system developed by Falkinburg [4]. For this validation process, 34-bit RCA is utilized. Thus, 34-bit RCA is manipulated by the signature manipulation algorithm toward Type I, Type II, and Type III and generated in VHDL format to apply on the polymorphic circuit designed by Falkinburg [4] as seen in Figure 4.21(a). Figure 4.21(b) describes that the three adders in Figure 4.21(a) are replaced with the generated three types of 34-bit RCA based on the purpose to randomize the power signature.

Figure 4.21 shows Falkinburg's results and comparison between the baseline and the randomized signature obtained by his polymorphic circuit design. As seen in Figure 4.21(b), the power signature was successfully randomized to protect secret key from the power trace [4]. In order for the redesigned circuit to achieve such a protection, the circuit is applied on Xilinx Virtex5 FPGA, and simulation is performed. Figure 4.22 shows the result from the simulation. Visually, it is also not easy to detect the key from the generated trace. But, the power traces does not be analyzed



(a)



(b)

Figure 4.21: Randomizing Power Signature developed by Falkinburg [4]
(a) Falkinburg's original power signature of 512-bit RSA(baseline)
(b) The power signature of 512-bit RSA Obtained with Falkinburg's Randomized Power Signature Method

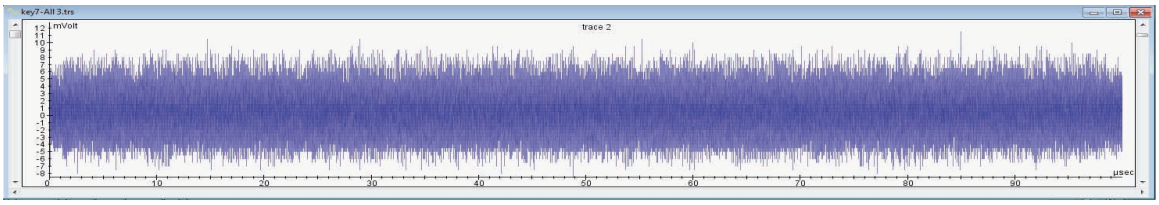


Figure 4.22: Randomized Signature obtained with manipulated 34-bit RCA (Type I, Type II, and Type III)

using detailed signal process to be validated whether or not it provides the suitable protection of the secret key from adversarial power analysis.

4.6 Summary

This chapter shows the results associated with the four steps as defined in Chapter III. Firstly, power signature estimation and simulation processes shows the original power signature for c264 and 34-bit RCA estimated and simulated by static and dynamic approach. Secondly, characterization and classification process provides the type of power signature based on the four types of power signature patterns. Thirdly, implementation and manipulation step provides the variant of c264 and 34-bit RCA by the Smart CE and Smart SSR methods. Lastly, the final process provide

three categories which need to be validated or evaluated. The results for this step are summarized as follows:

1. **Accuracy of Signature Detection**

   :Static approach using SID is limited to accurately estimate the power signature, but it is possible to compare the total switching activity and its variation. Dynamic approach using SPICE simulation provides better accuracy, but the result varies depending on the input patterns.

2. **Availability of Signature Manipulation**

   :Only component encryption method in CORGI 2.0 shows an ability to transform the type of the original power signature. The other methods are limited in terms of signature changing property and efficiency. On the other hand, new proposed signature manipulation method based on the component encryption provides better signature manipulating property and efficiency.

3. **Verification of the Final Circuit Variant**

   :The final circuit variant of 34-bit RCA is evaluated using the RSA encryption system implementing on Xilinx Virtex5 FPGA. The result visually demonstrates that the key is no longer visible in plain sight and appears more random than Falkinburg's results.

The next chapter will provide conclusions from this thesis and present suggestions for future work.

V. Conclusions

This research effort has determined that a combinational circuit can be obfuscated to hide the original power signnature information against side-channel analysis. This chapter summarizes the research effort by summarizing the objectives and conclusions, discusses contribution to the field of study, and establish the proposal for future research.

5.1 Conclusions

5.1.1 Provided power signature detection and characterization. Power signature estimation is performed by using two different techniques in static and dynamic approach. In static approach, it is found that using static technique for power signature estimation is limited in terms of accuracy of measuring power signature of a circuit due to not considering dynamic input variation, but it is useful to provide a quick estimate on the power signature without the cost of dynamic approach. Dynamic simulation, on the other hands, provides better accuracy for measuring power signature than static estimation. But, it must contain dynamic factors such as input pattern and time interval. From experiments using static and dynamic approach, it is shown that static approach provides 75% accuracy for 4-bit multiplier, but is limited to smaller circuit for an accurate result. The accuracy of the simulation results varies based on many facts, such as input patterns, complexity of the gate connections, gate types, and operational temperature.

5.1.2 Provided power signature manipulation method. In order to achieve the primary goal of this research, smart SSR and smart CE methods are developed for a circuit obfuscation which is implemented by increasing the number of circuits and the number of signals at the designated part of a circuit to change switching activity at such a selected portion. With these methods, 4-bit multiplier and 34-bit RCA are successfully changed in terms of their overall structure intentionally (85% for 4-bit multiplier 100% for 34-bit RCA). Thus, the proposed signature manipulation method provides 90% availability for transformation of the type of power signature

5.1.3 Provided visually randomized power signature against side-channel analysis. Firstly, an accuracy of signature detection is evaluated by using static and dynamic approach. From this evaluation, the limitation of static estimation is found by comparing its results with the results of dynamic simulation. But, it is noticed that the high switching activity estimated by static estimation reflects the peak point in the power trace obtained by dynamic simulation. Therefore, SID can be utilized as a metric to recognize the increased or decreased switching activity in the circuit variant. Secondly, the proposed power signature manipulation provides the best ability to alter the class of power signature among the previous obfuscation algorithm. Lastly, the proposed signature manipulation method is applied on RSA circuit on Xilinx Virtex5 FPGA against adversarial power analysis. As a result, the proposed method provides the ability to randomize power signature against side-channel analysis.

5.2 Contributions

1. **Implementing Signature Identification System in CORGI**
   Developed SAID system in CORGI based on the Menon's switching activity estimation method to be used for measuring an average power signature.
2. **Implementing SPICE Netlist Exporter**
   Created the standard cell library which stores the pre-defined gate information in SPICE-Netlist format, and implemented the SPICE-Netlist exporter in CORGI to generate SPICE-Netlist description by combining pre-defined gates from the cell library.
3. **Designing tool to manipulate power signature**
   Designed new power signature manipulation technique based on the Random SSR and Component Encryption in CORGI 2.0.

5.3 Future Work

The proposed signature manipulation method in this thesis increases the total power consumption of a circuit since the number of gates and signals in a circuit

are generally increased during manipulation process. Additionally, it is realized that it is not easy to eliminate or reduce the initial peak points in the original power signature. Thus, a future research can focus on the algorithm using switching activity minimization technique to transform other types of power signature. Additionally, the signature manipulation method does not use the special types of gates which statistically provides higher switching activity than other circuits such as an inverter. Thus, developing smart replacement of a circuit with high switching activity can be considered to more increase the switching activity at the selected part of circuit for the future work.

Bibliography

1. Bonneau, Joseph. "Side-Channel Cryptanalysis", May 4, 2010. University of Cambridge Computer Lab.

2. Chih-Hung Chang, William C.Chu, Chih-Wei Lu. "Reverse Engineering". *Handbook of Software Engineering and Knowledge Engineering*, Vol2, 2002.

3. Chikofsky, E.J. and II Cross, J.H. "Reverse engineering and design recovery: a taxonomy". *Software, IEEE*, 7(1):13–17, Jan 1990. ISSN 0740-7459.

4. Falkinburg, Jeffrey L. *POLYMORPHIC RECONFIGURATION TO EFFECTIVELY "CLOAK" A CIRCUIT'S FUNCTION*. Master's thesis, Air Force Institute of Technology (AFIT), March 2011.

5. Hansen, Mark C., Hakan Yalcin, and John P. Hayes. "Unveiling the ISCAS-85 benchmarks: a case study in reverse engineering". *IEEE Design and Test of Computers*, 16(3):72 – 80, 1999. ISSN 07407475. URL `http://dx.doi.org/10.1109/54.785838`. Carry look ahead;Error correcting circuits;Register transfer;.

6. Jeffrey T. McDonald, Yong C. Kim, Kenneth Norman. "Introducing CORGI: A Framework for Whitebox Circuit Obfuscation", November 2008.

7. Joao M. S. Alcantara, Federico Galvez-Durand, Antonio C. C. Vieria. "A Methodology For Dynamic Power Consumption Estimation using VHDL Descriptions". The 15th Symposium on Integrated Circuits and Systems Design, 2002.

8. Jun, Paul Kocher Joshua Jaffe Benjamin. "Differential Power Analysis". Cryptography Research, Inc. http://www.cryptography.com.

9. Jun, Paul Kocher Joshua Jaffe Benjamin. "Introduction to Differential Power Analysis and Related Attacks". Cryptography Research, Inc. http://www.cryptography.com.

10. Keutzer, Srinivas Devadas Kurt and Jacob White. "Estimation of Power Dissipation in CMOS Combinational Circuits Using Boolean Function Manipulation". *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems Vol. II No. 3*, 1992.

11. Kim, Hanseok. *Removing Redundant Logic Pathways in Polymorphic Circuits*. Master's thesis, Graduate School of Engineering, Air Force Institute of Technology (AETC), Wright-Patterson AFB OH, March 2009. AFIT/GCS/ENG/09-03.

12. Kim, Yong C. and Lt. Col. J. Todd McDonald. "Considering Software Protection for Embedded Systems". *Crosstalk The Journal of Defense Software Engineering*, 22(6):4–8, 2009.

13. Koranek, Daniel. *Deterministic, Efficient Variation of Circuit Components To Improve Resistance To Reverse Engineering*. Master's thesis, Air Force Institute of Technology, June 2010.

14. NEIL H.E. WESTE, DAVID HARRIS. *CMOS VLSI DESIGN (A Circuits and Systems Perspective) Third Edition*. Addison Wesley, 2005.

15. Nohl, Karsten, David Evans, Starbug Starbug, and Henryk Plötz. "Reverse-engineering a cryptographic RFID tag". *SS'08: Proceedings of the 17th conference on Security symposium*, 185–193. USENIX Association, Berkeley, CA, USA, 2008.

16. Parham, James D. *Component Hiding Using Identification and Boundary Blurring Techniques*. Master's thesis, Air Force Institute of Technology, March 2010.

17. Radhakrishnam, R. V. Menon S. Chennupati N. K. Samala D. and B. Izadi. "Power Optimized Combinational Logic Design". *ESA '03 International Conference*. 2004.

18. Radhakrishnam, R. V. Menon S. Chennupati N. K. Samala D. and B. Izadi. "Switching Activity Minimization in Combinational Logic Design". *International Conference on Embedded Systems and Applications*, 2004.

19. Rugaber, Spencer. "The use of domain knowledge in program understanding". *Ann. Softw. Eng.*, 9(1-4):143–192, 2000. ISSN 1022-7091.

20. Shenoy, Sandeep P. *Switching Activity in CMOS Digital Circuit*. Master's thesis, McGill University, Montreal, 1996.

21. Tanenbaum, Andrew. "News Summary of Broken Dutch Transit Card". World Wide Web, 2008. URL `http://www.cs.vu.nl/ ast/ov-chip-card/`.

22. Thomas S. Messerges, Robert H. Sloan, Ezzy A. Dabbish. "Investigations of Power Analysis Attacks on Smartcards", 1999. Motorola Labs, University of Illinois at Chicago.

23. THORPE, THOMAS W. *Computerized Circuit Analysis With SPICE*. A Wiley-Interscience Publication, 1994.

A.1 Power Signature for c264 Circuit Variant per Algorithm produced by SID

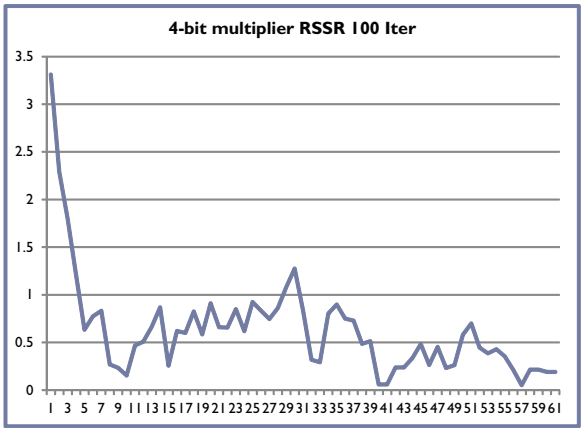


Figure A.1: Power Signature of Obfuscated Circuit Variant after applying Random SSR with 100 iterations

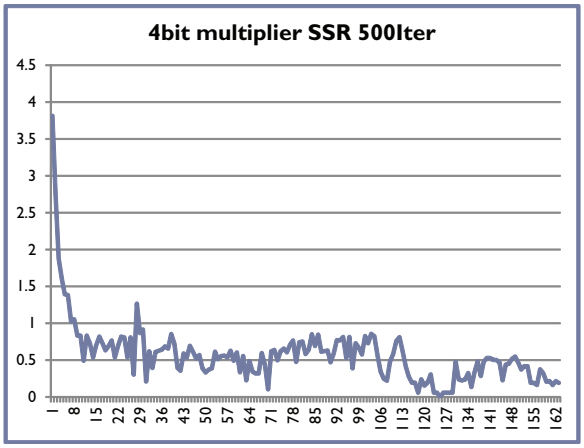


Figure A.2: Power Signature of Obfuscated Circuit Variant after applying Random SSR with 500 iterations

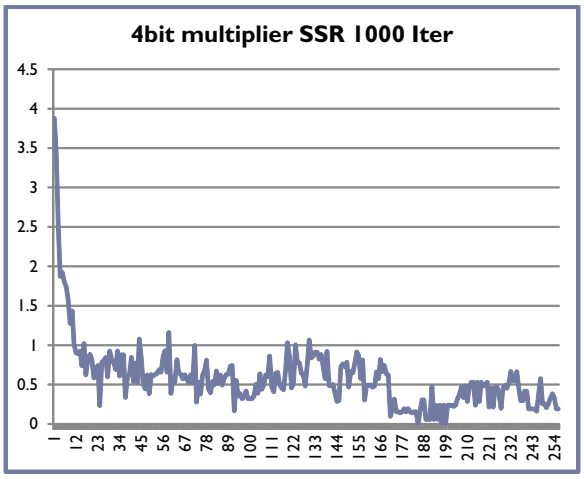


Figure A.3: Power Signature of Obfuscated Circuit Variant after applying Random SSR with 1000 iterations

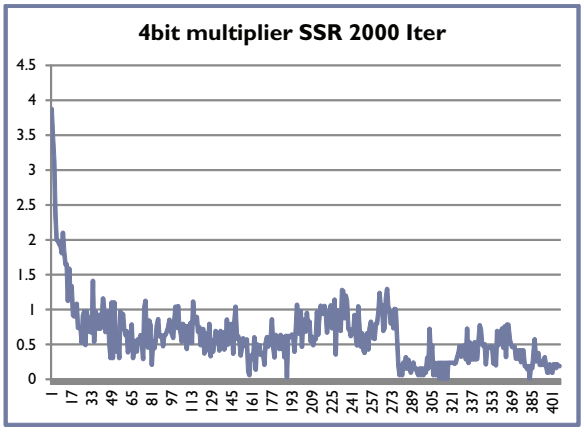


Figure A.4: Power Signature of Obfuscated Circuit Variant after applying Random SSR with 2000 iterations

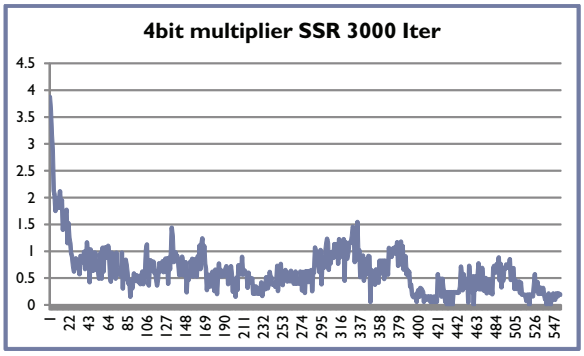


Figure A.5: Power Signature of Obfuscated Circuit Variant after applying Random SSR with 3000 iterations

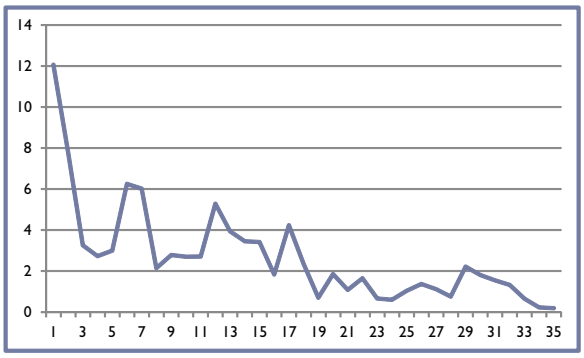


Figure A.6: Power Signature of Obfuscated Circuit Variant after applying Component Fusion Trial 1

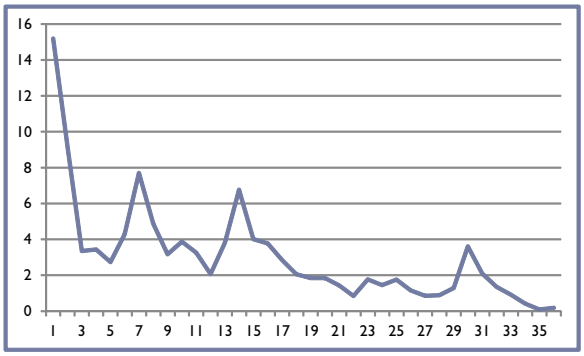


Figure A.7: Power Signature of Obfuscated Circuit Variant after applying Component Fusion Trial 2

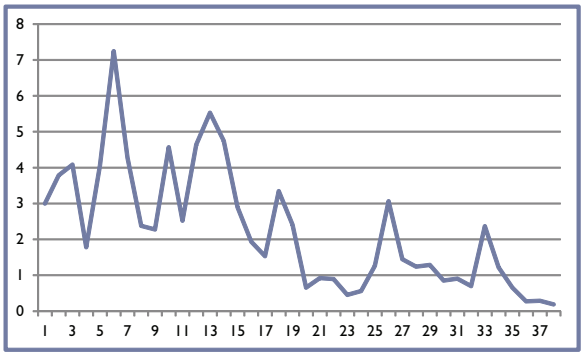


Figure A.8: Power Signature of Obfuscated Circuit Variant after applying Component Encryption Trial 1

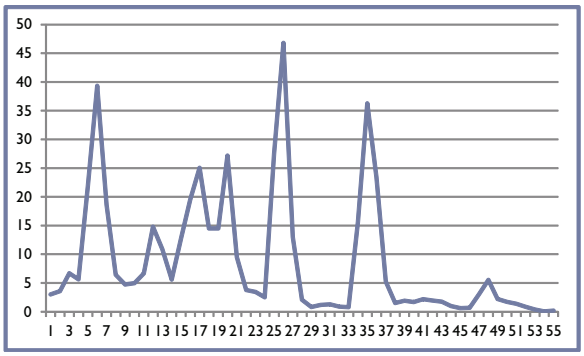


Figure A.9: Power Signature of Obfuscated Circuit Variant after applying Component Encryption Trial 2

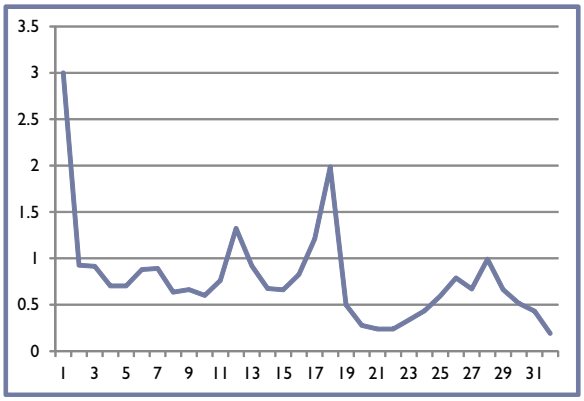


Figure A.10: Power Signature of Obfuscated Circuit Variant after applying Smart Component Encryption selecting Rear Level Components Trial 1

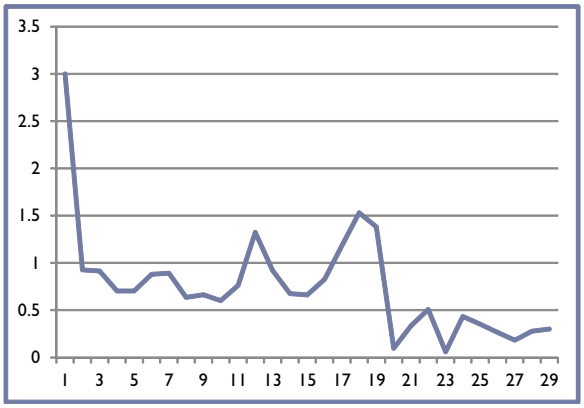


Figure A.11: Power Signature of Obfuscated Circuit Variant after applying Smart Component Encryption selecting Rear Level Components Trial 2

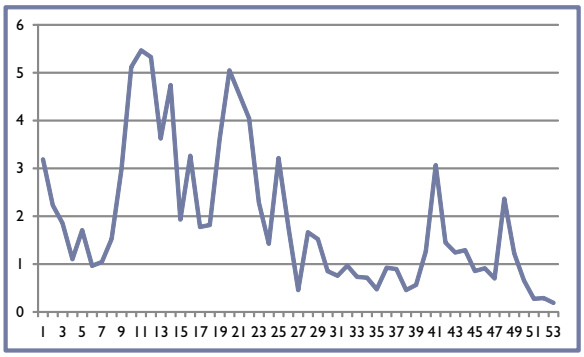


Figure A.12: Power Signature of Obfuscated Circuit Variant after applying Smart Component Encryption and Smart SSR selecting rear level components and gates with 100 iterations

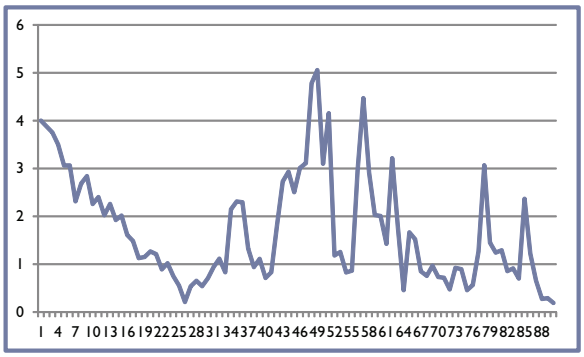


Figure A.13: Power Signature of Obfuscated Circuit Variant after applying Smart Component Encryption and Smart SSR selecting rear level components and gates with 500 iterations

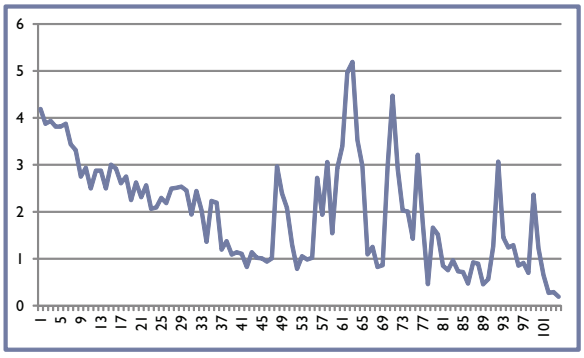


Figure A.14: Power Signature of Obfuscated Circuit Variant after applying Smart Component Encryption and Smart SSR selecting rear level components and gates with 1000 iterations

B.1 Power Signature for c264 Circuit Variant per Algorithm produced by SPICE Simulation

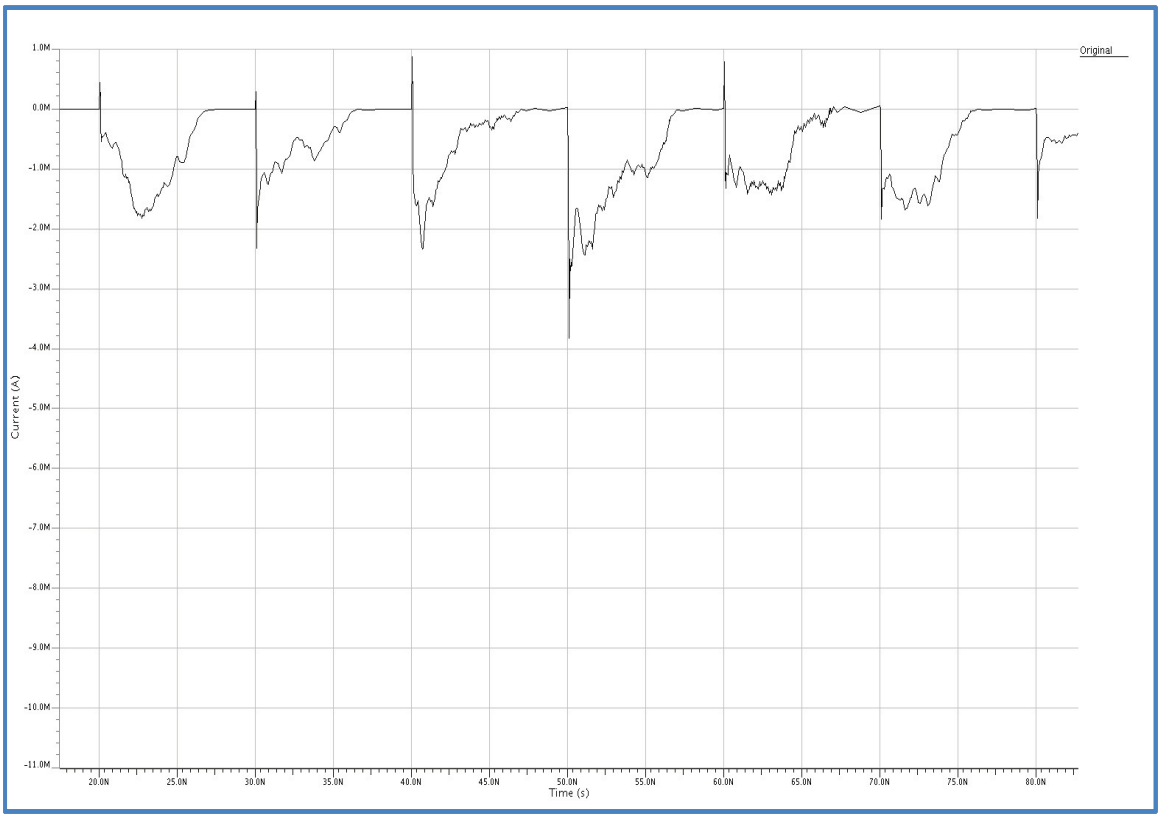


Figure B.1: Power Signature for c264 By Random Sequence

B.2 Power Signature for c5355 and c499 Circuit Variant per Algorithm produced by SPICE Simulation

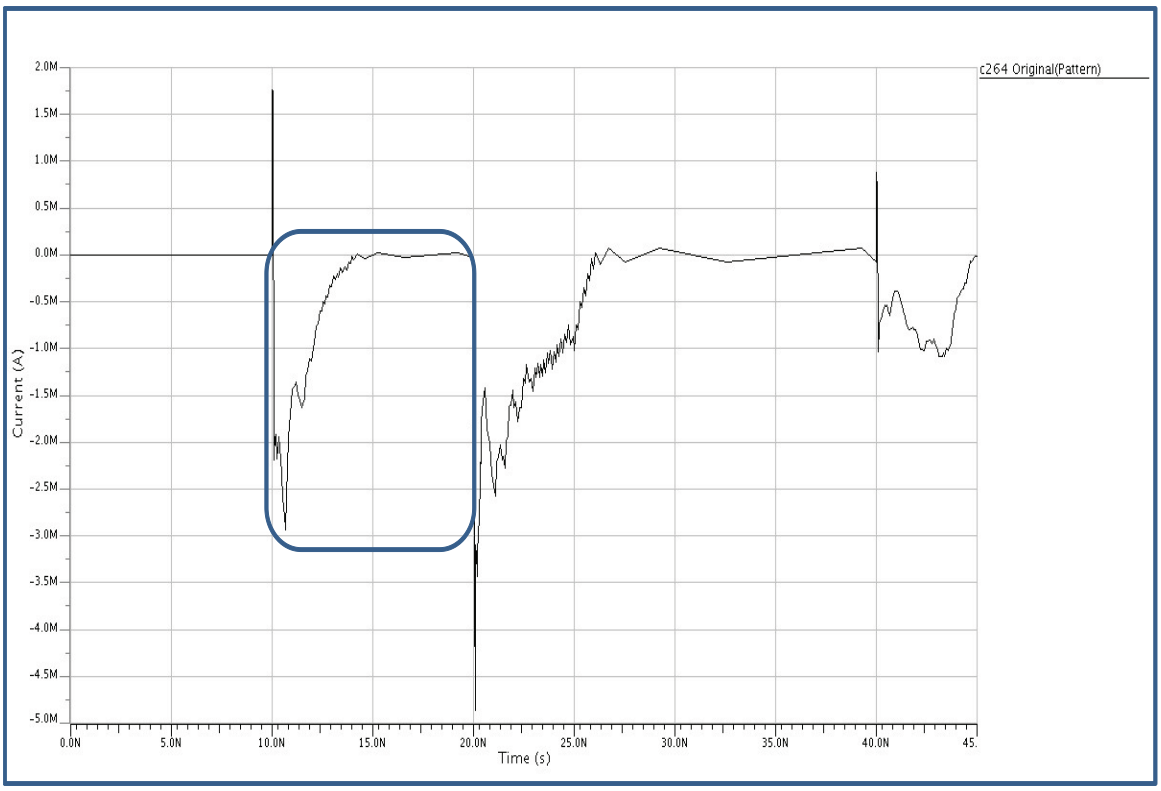


Figure B.2: Power Signature for c264 By User-defined Input(case1)

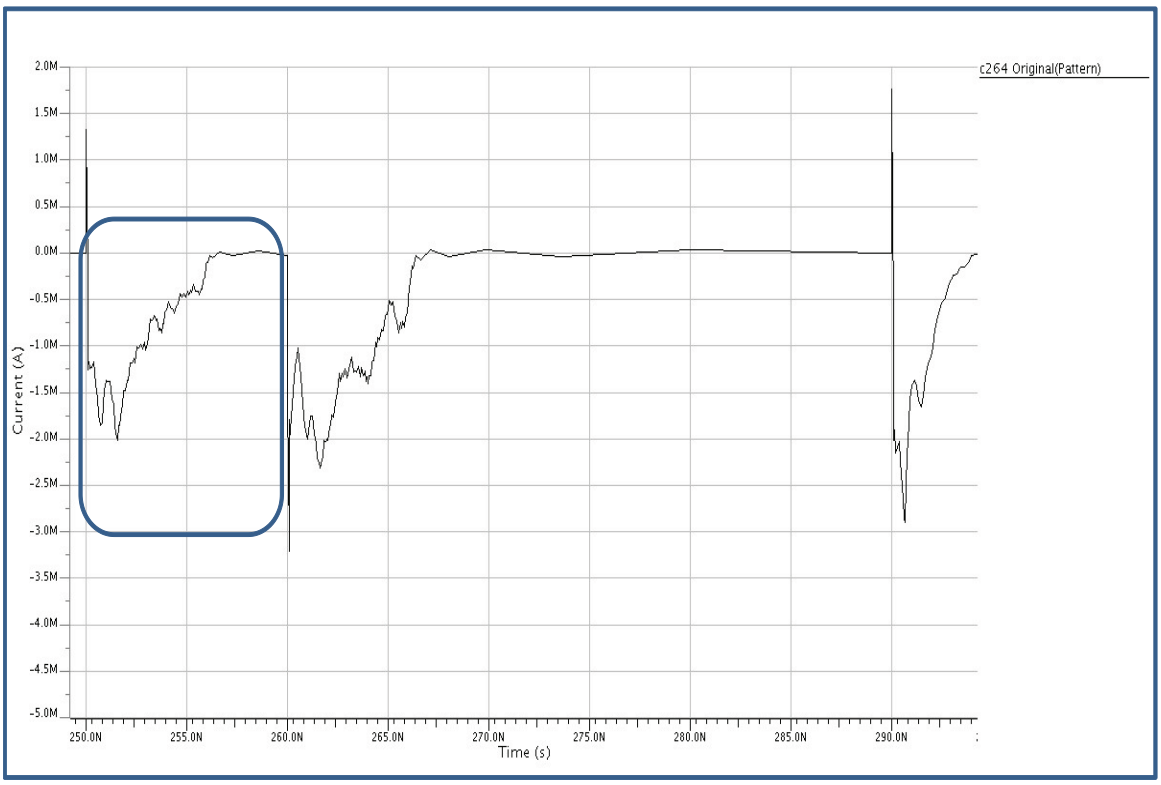


Figure B.3: Power Signature for c264 By User-defined Input(case2)

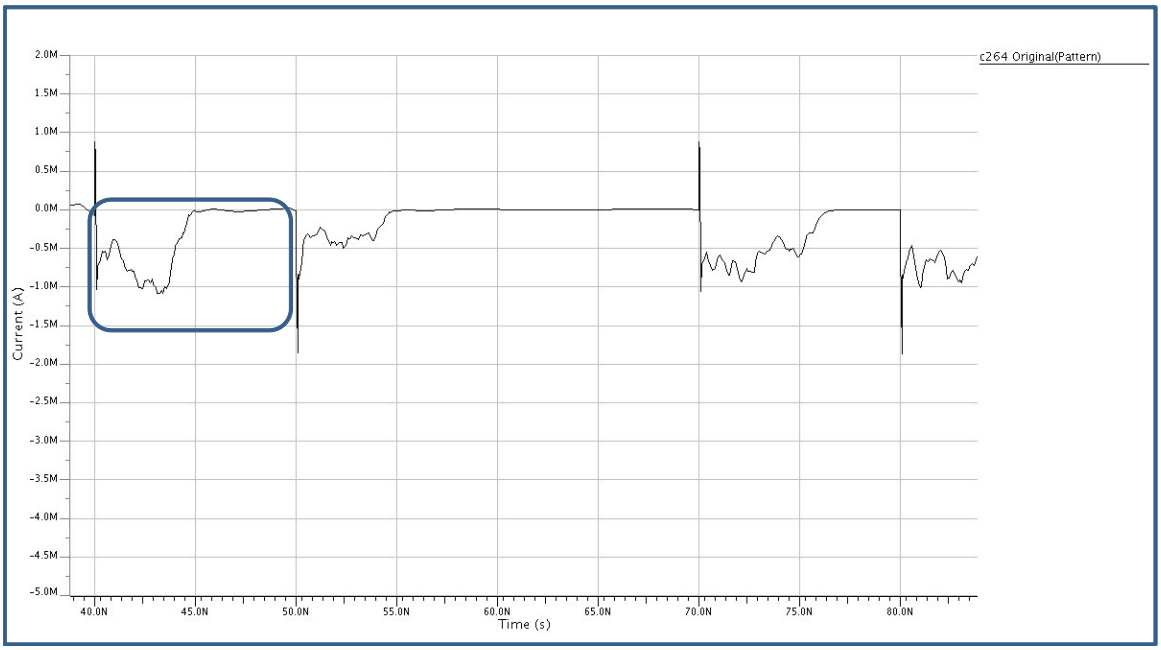


Figure B.4: Power Signature for c264 By User-defined Input(case3)

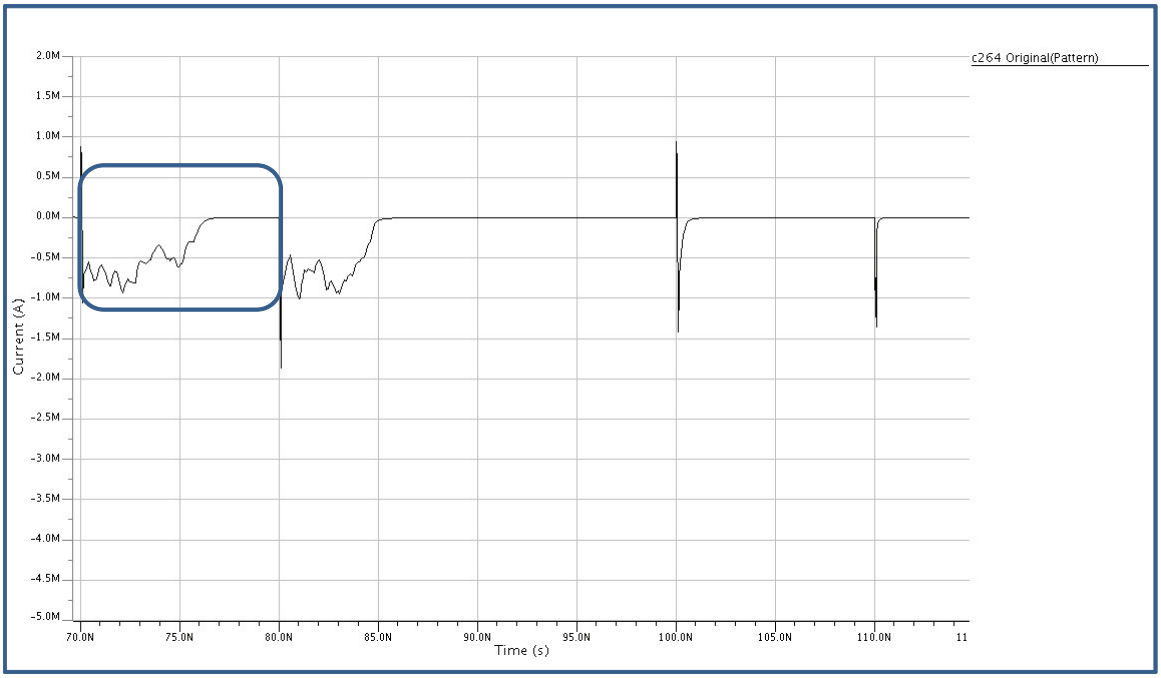


Figure B.5: Power Signature for c264 By User-defined Input(case4)

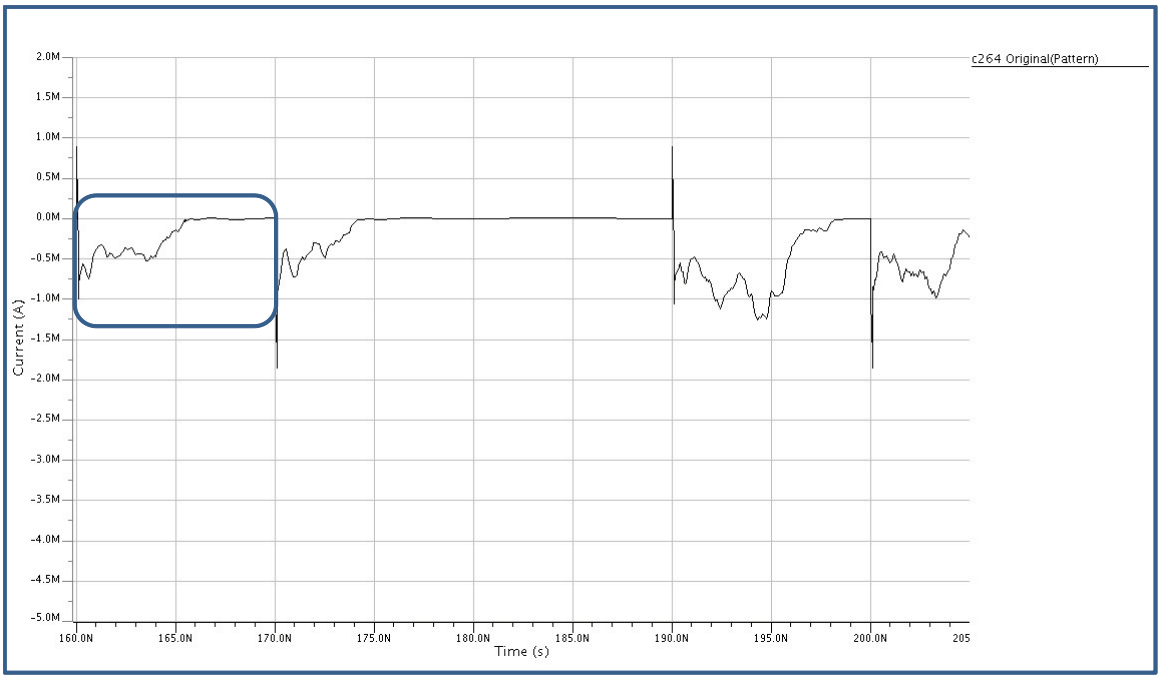


Figure B.6: Power Signature for c264 By User-defined Input(case5)

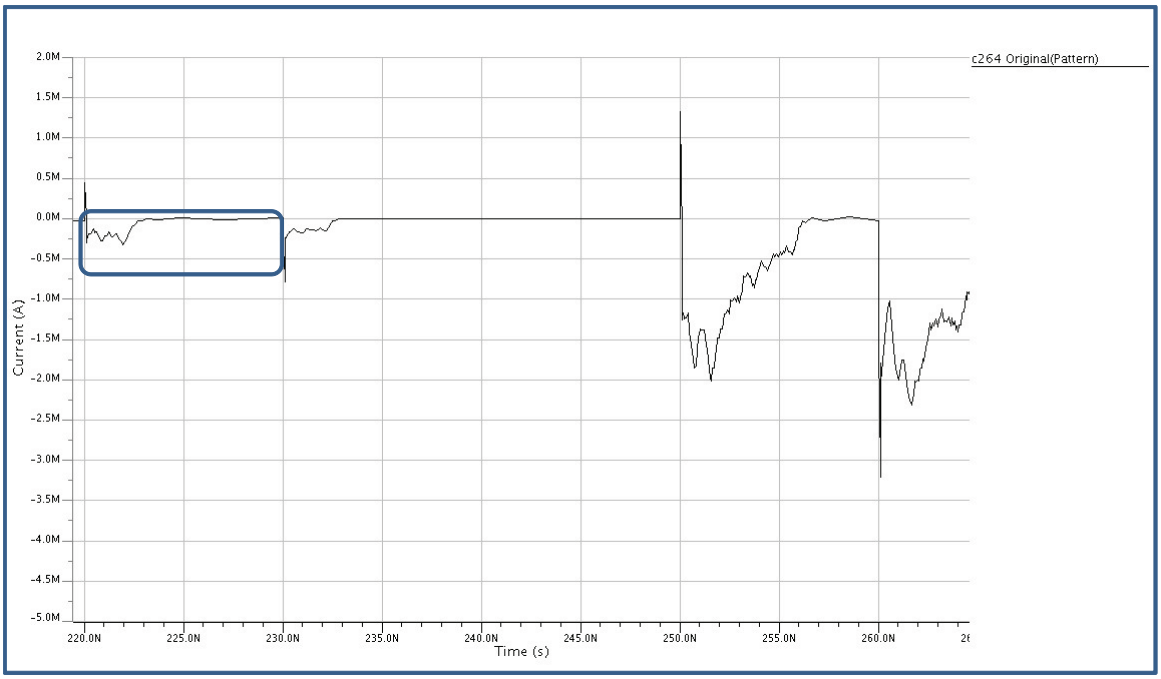


Figure B.7: Power Signature for c264 By User-defined Input(case6)

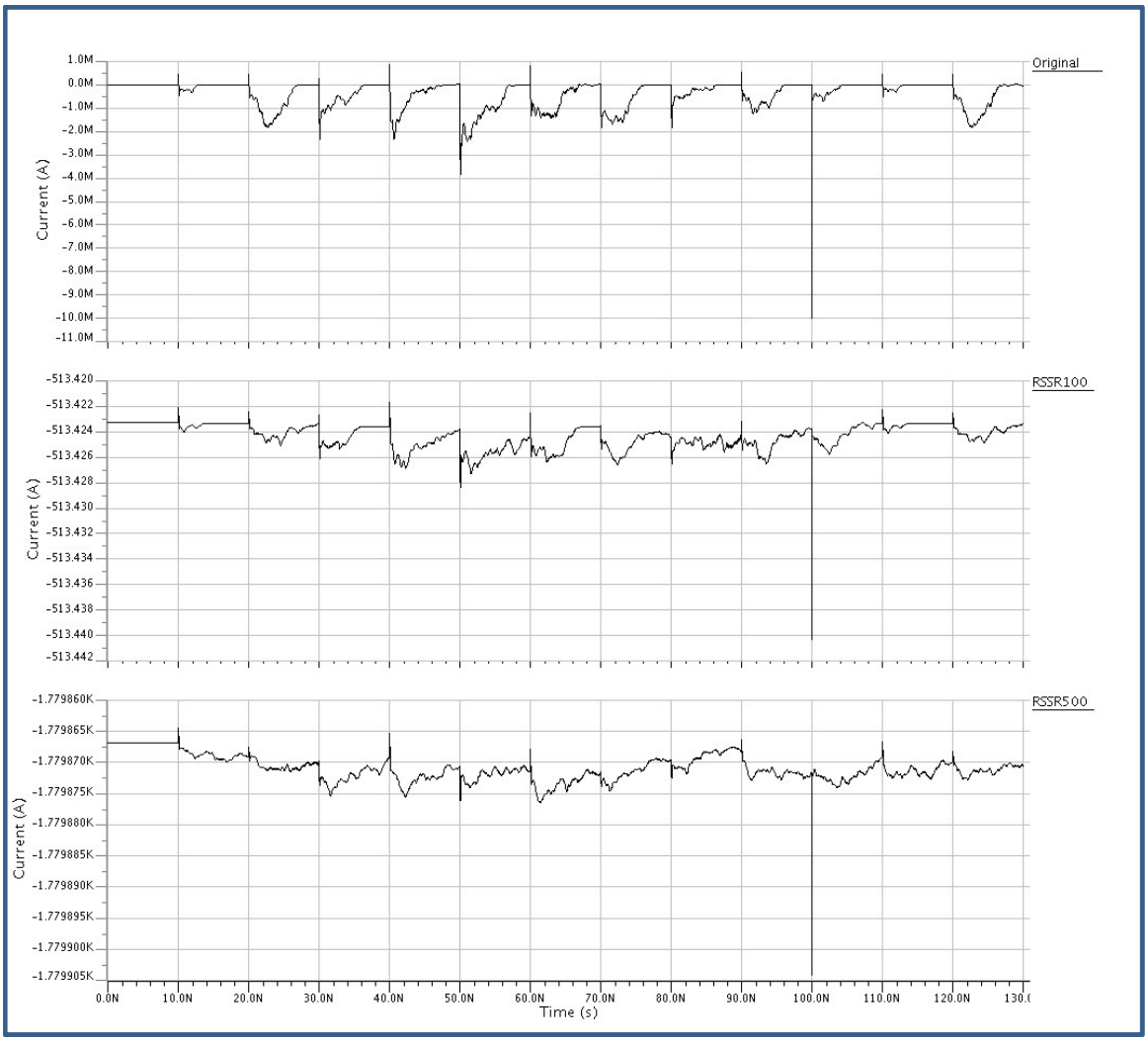


Figure B.8: Comparing Power Signature with Random SSR

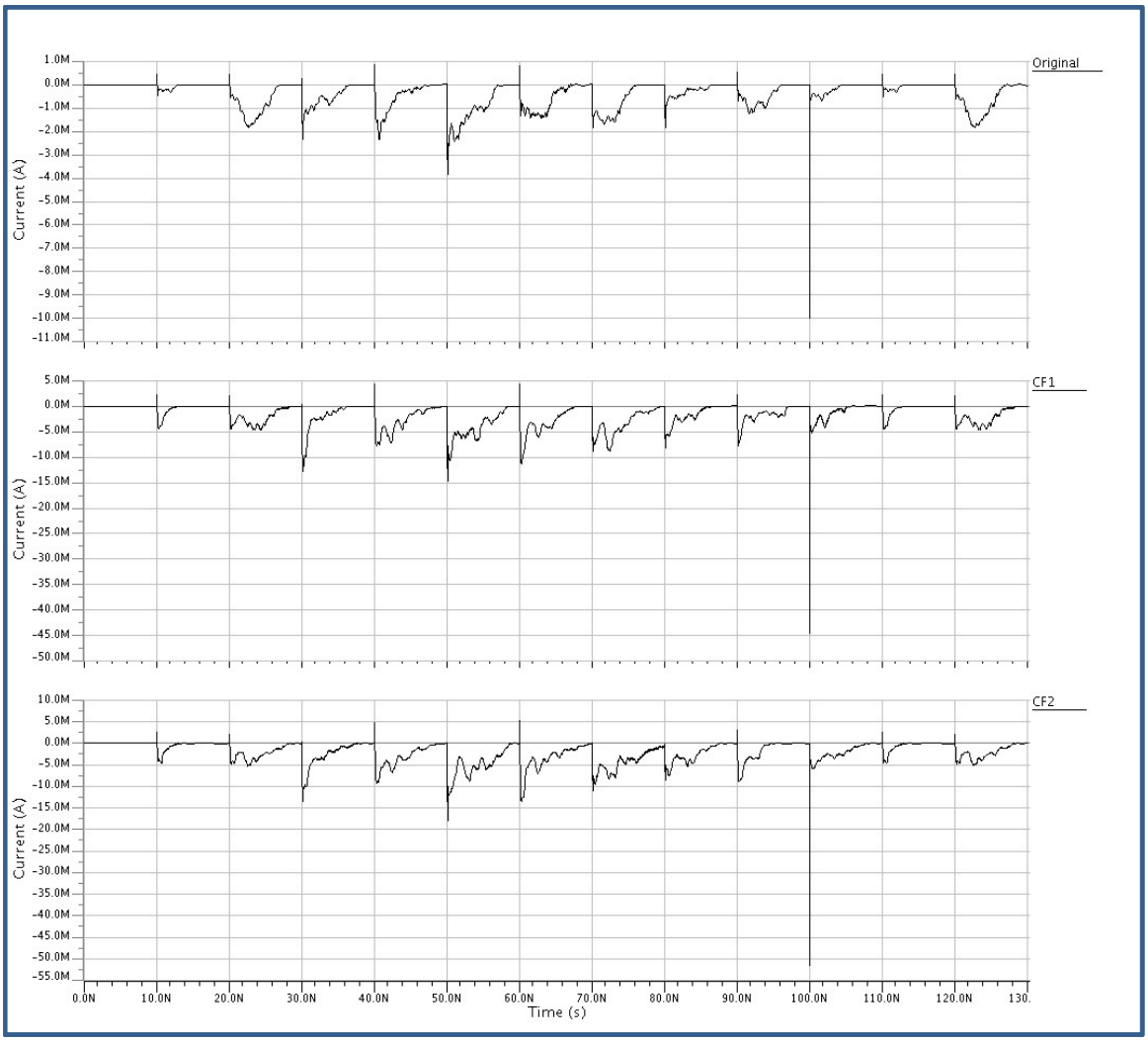


Figure B.9: Comparing Power Signature with Component Fusion

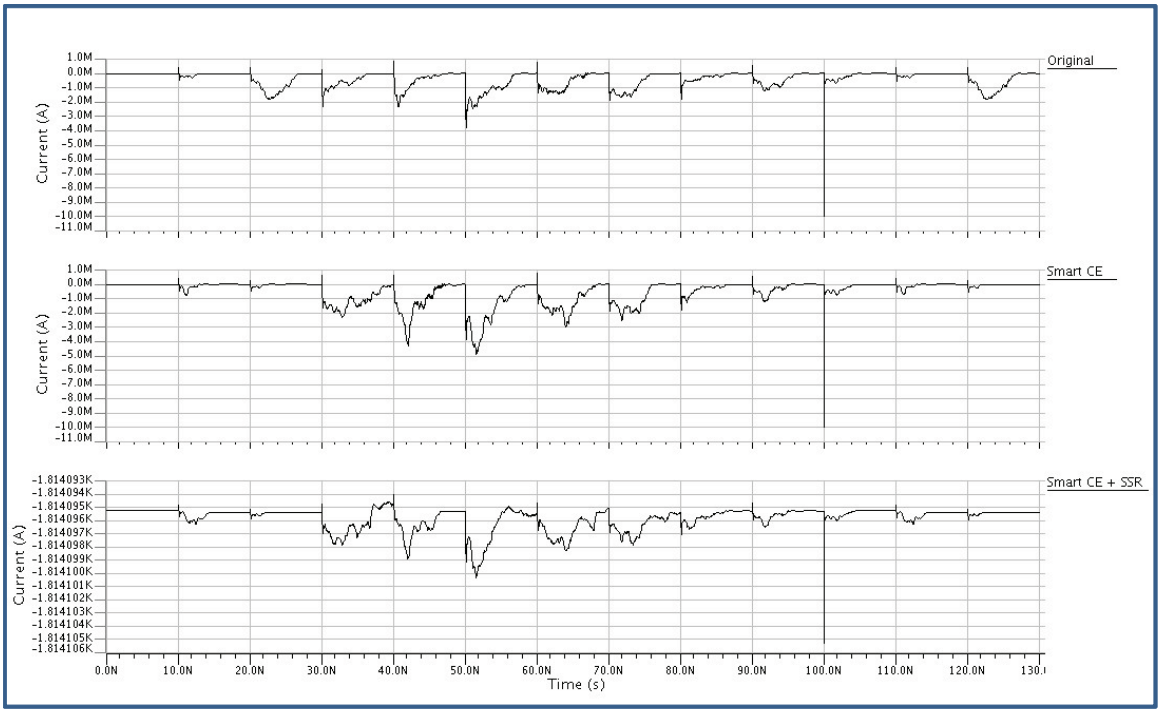


Figure B.10: Comparing Power Signature with Smart Component Encryption + Smart SSR

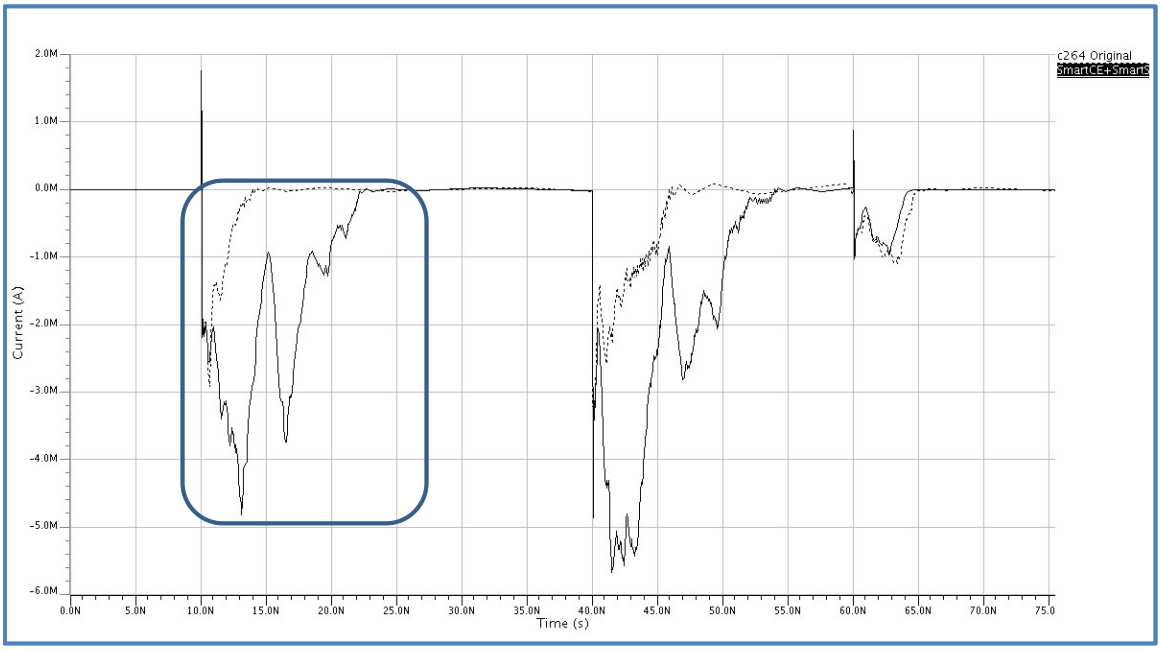


Figure B.11: Comparing Power Signature with Smart Component Encryption + Smart SSR (Case1)

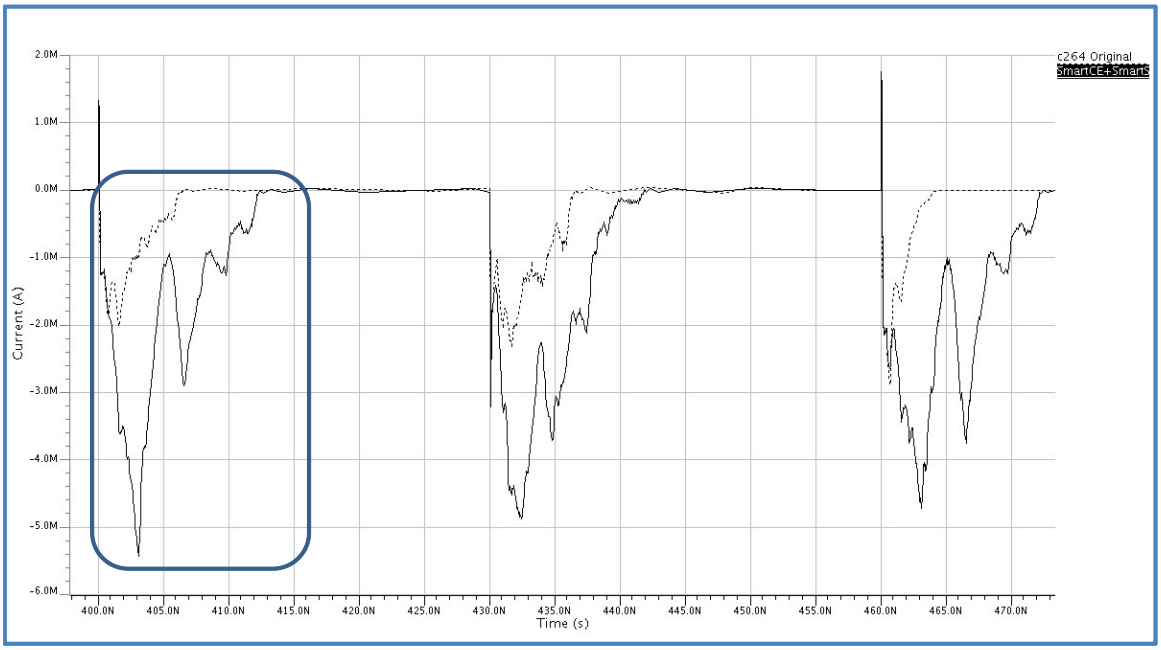


Figure B.12: Comparing Power Signature with Smart Component Encryption + Smart SSR (Case2)

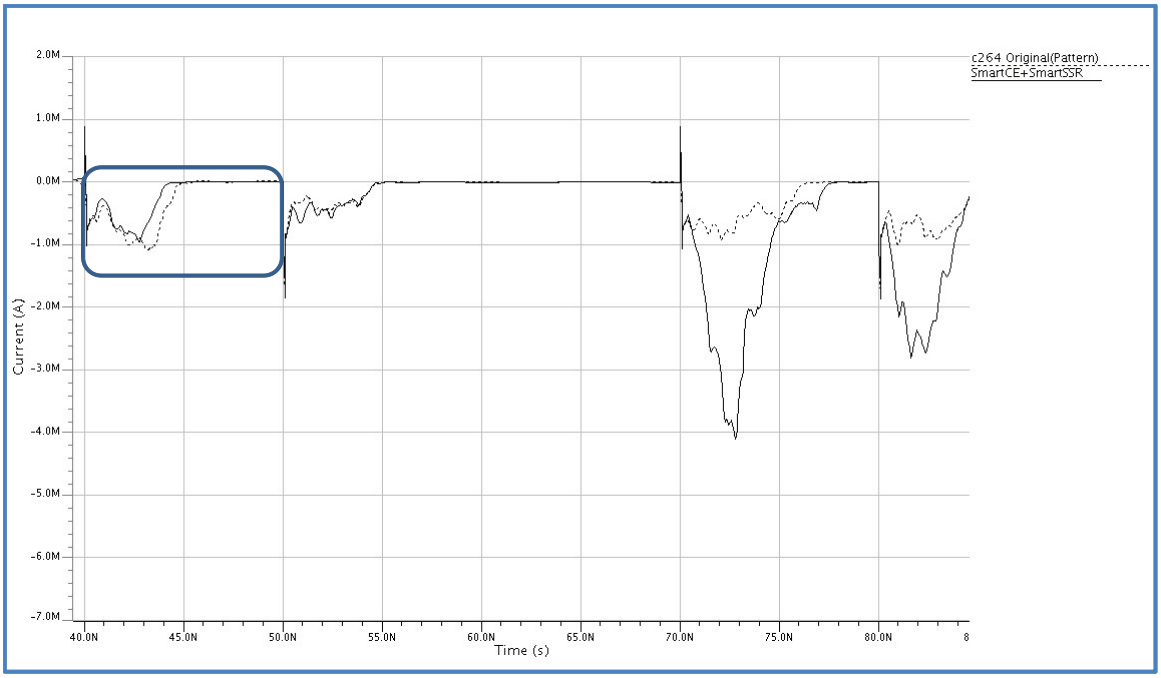


Figure B.13: Comparing Power Signature with Smart Component Encryption + Smart SSR (Case3)

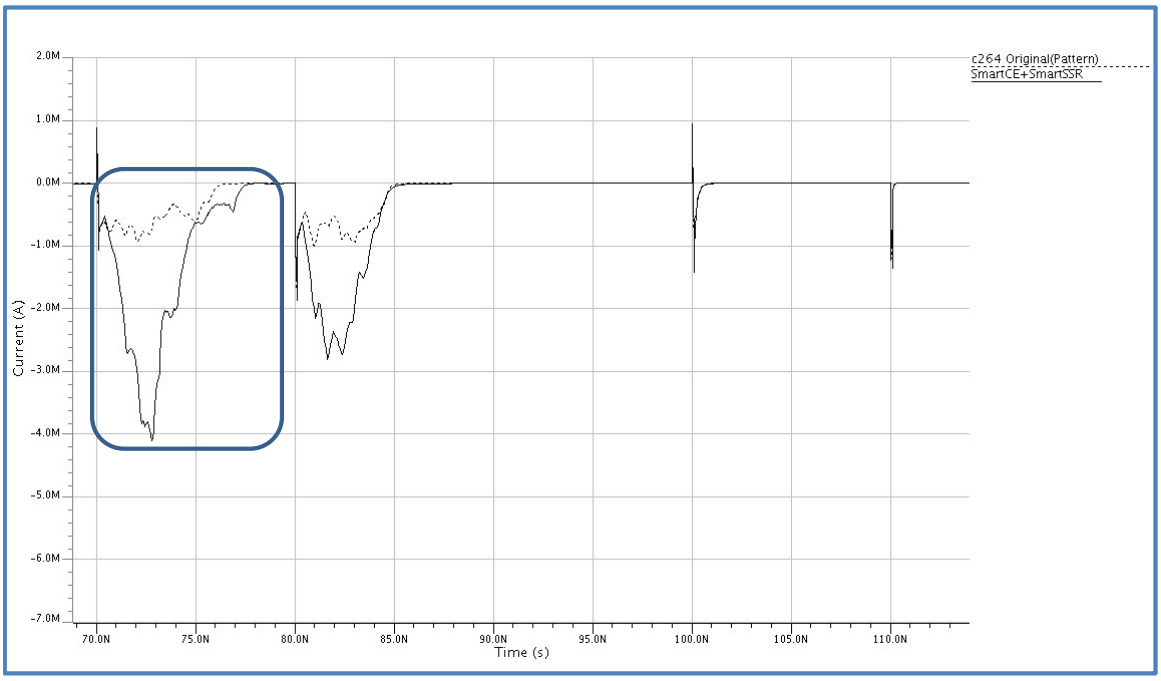


Figure B.14: Comparing Power Signature with Smart Component Encryption + Smart SSR (Case4)

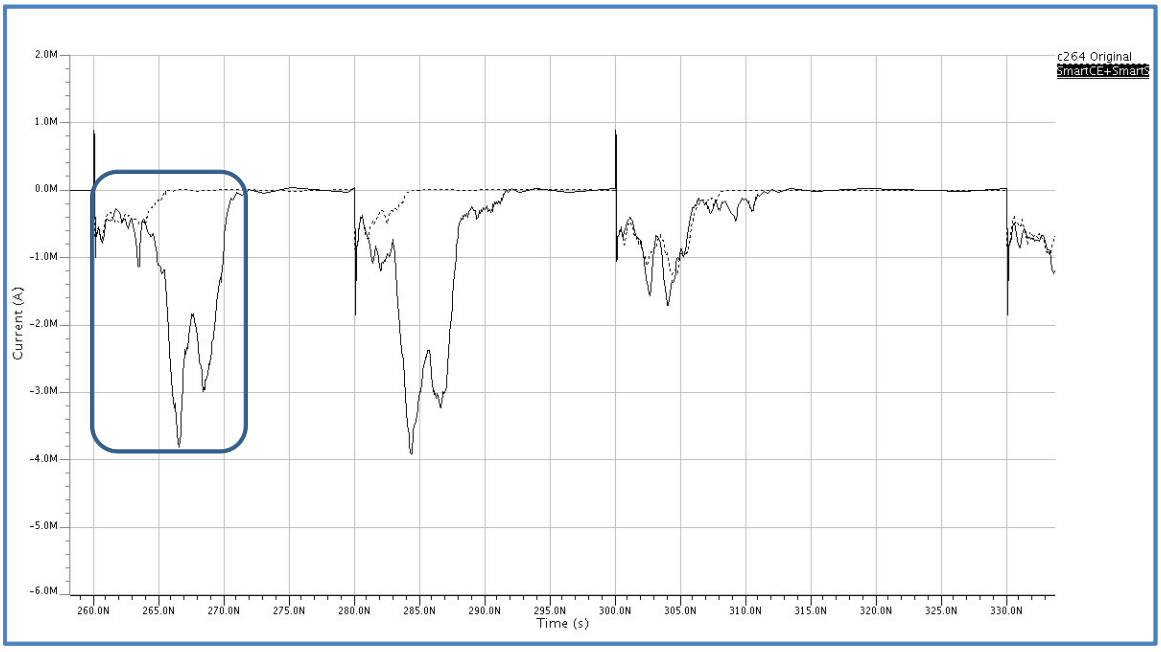


Figure B.15: Comparing Power Signature with Smart Component Encryption + Smart SSR (Case5)

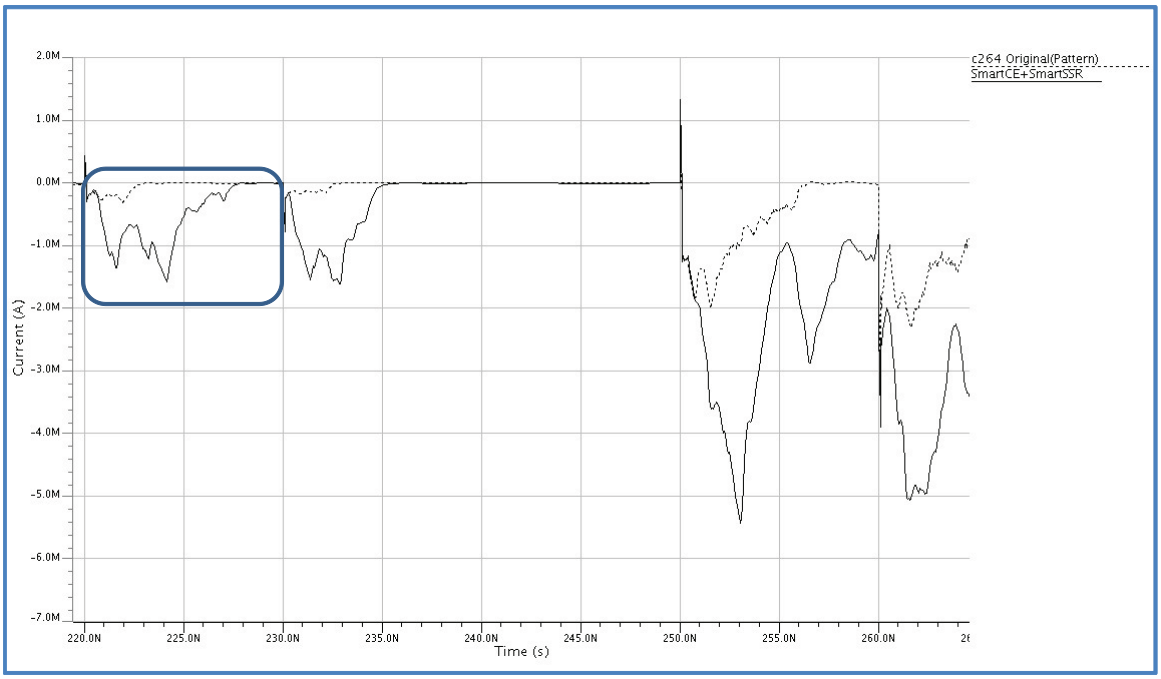


Figure B.16: Comparing Power Signature with Smart Component Encryption + Smart SSR (Case6)

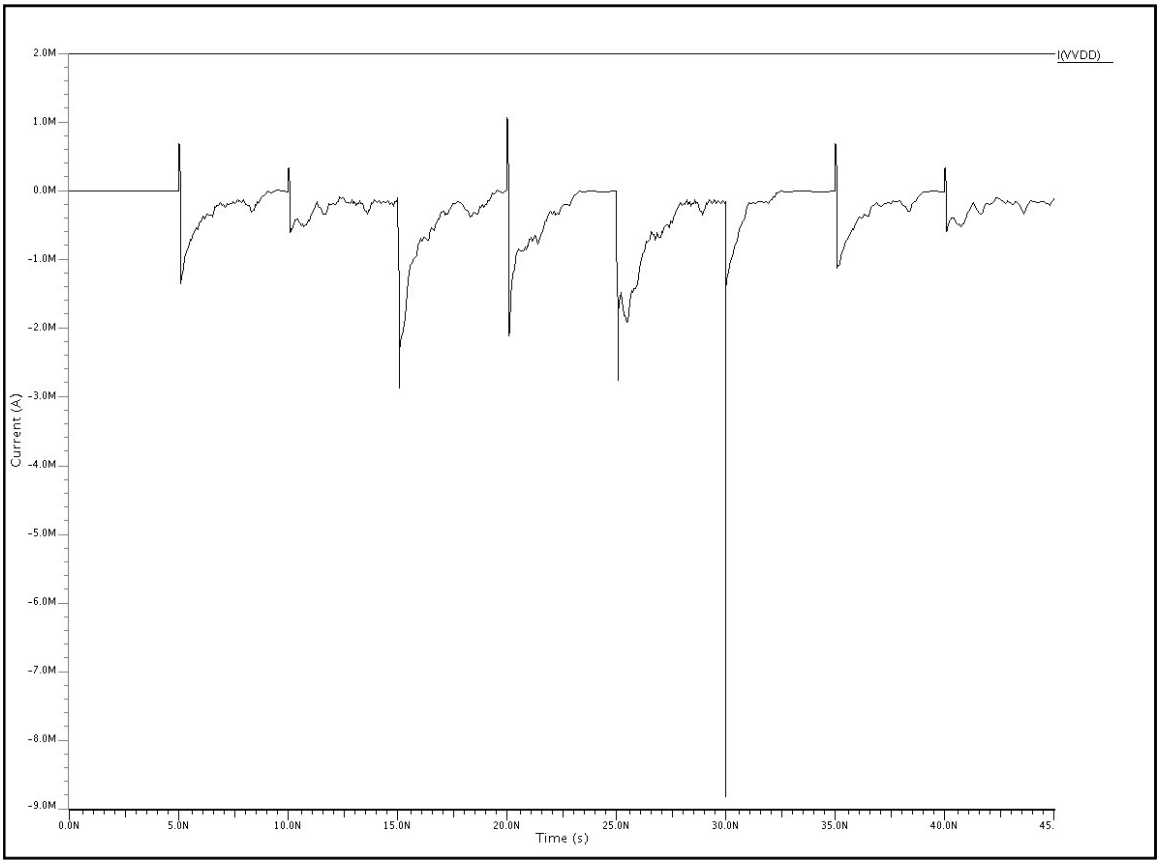


Figure B.17: Power Signature for c5355 By Random Sequence

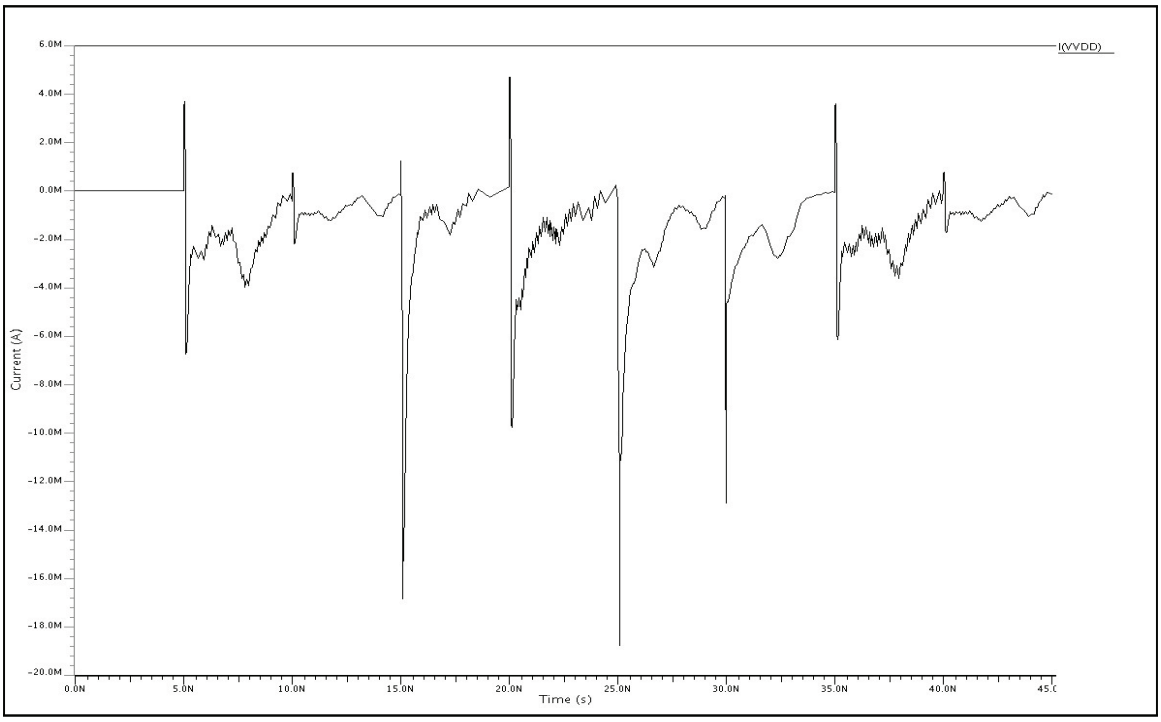


Figure B.18: Power Signature for c499 By Random Sequence

C.1 Power Signature for 34-bit RCA Circuit Variant per Algorithm produced by SPICE Simulation

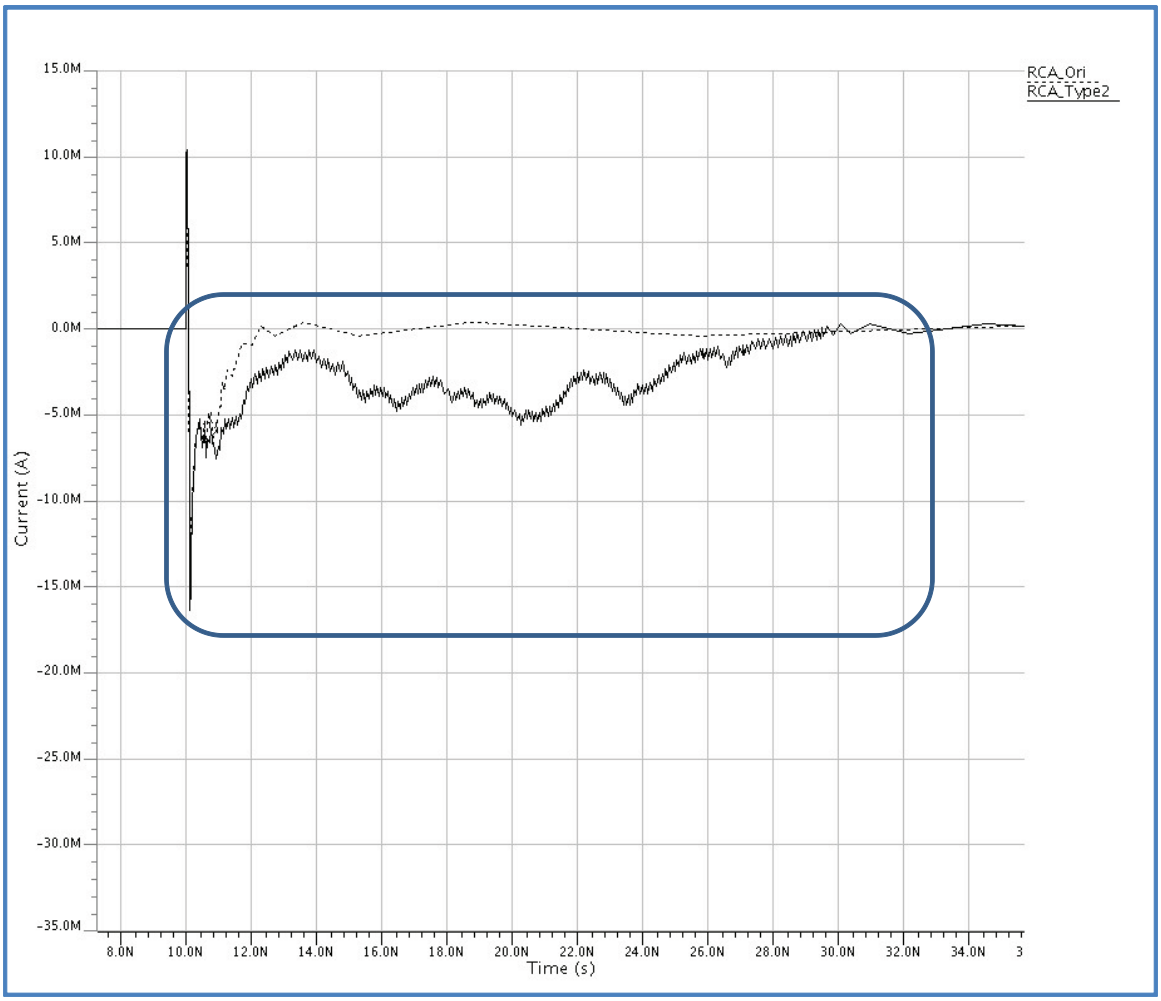


Figure C.1: Comparing Power Signature of 34-bit RCA with Smart Smart SSR toward Type II (Case1)

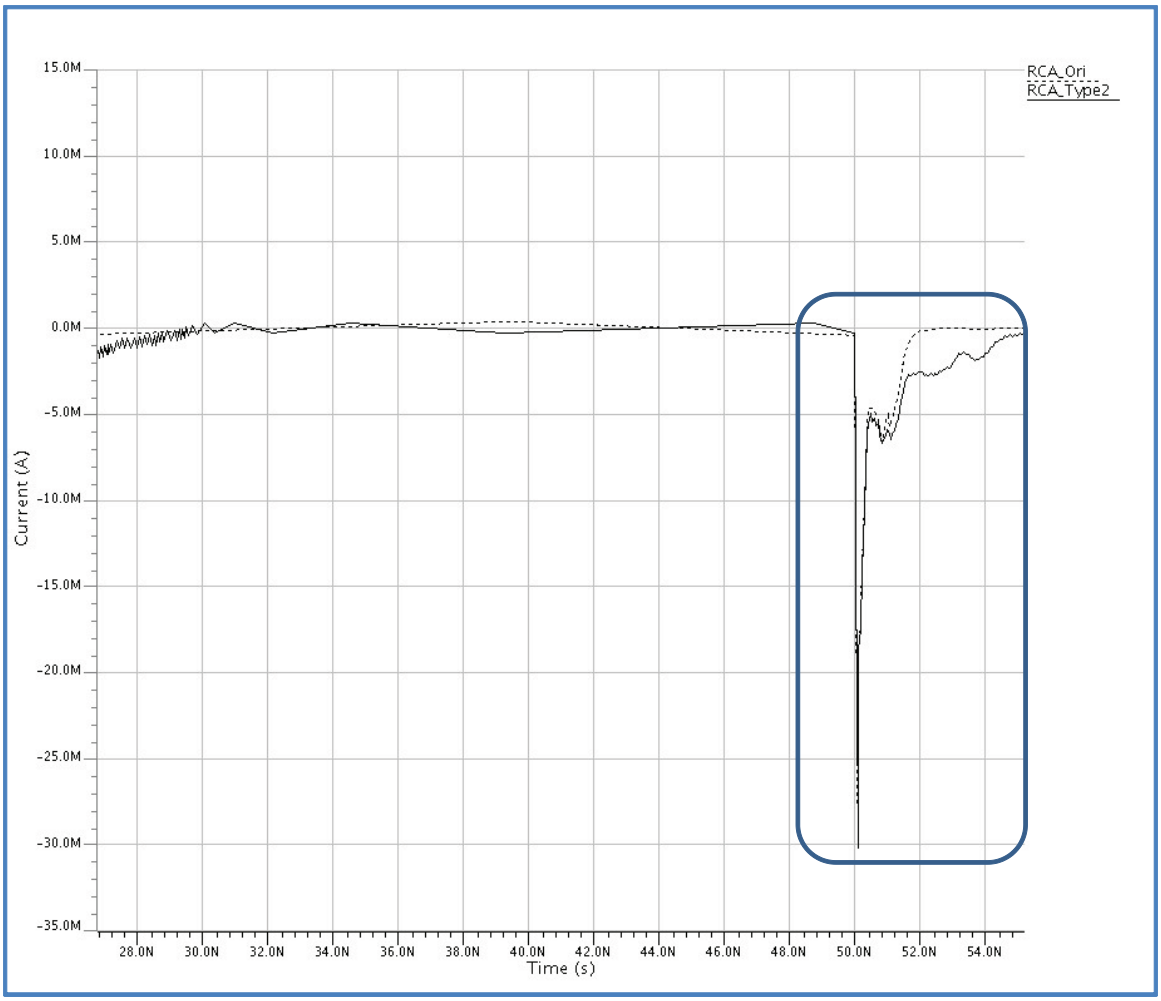


Figure C.2: Comparing Power Signature of 34-bit RCA with Smart Smart SSR toward Type II (Case2)

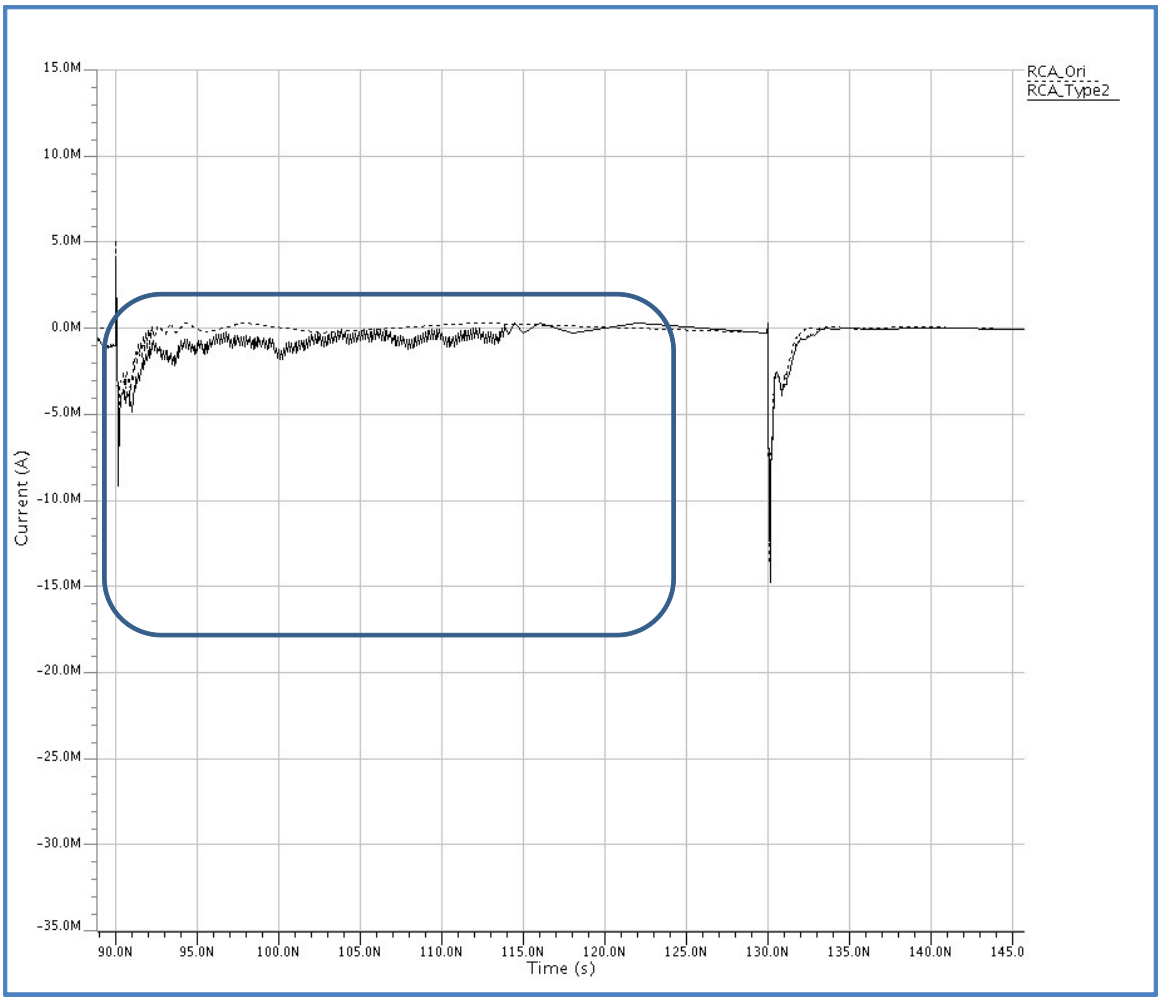


Figure C.3: Comparing Power Signature of 34-bit RCA with Smart Smart SSR toward Type II (Case3)

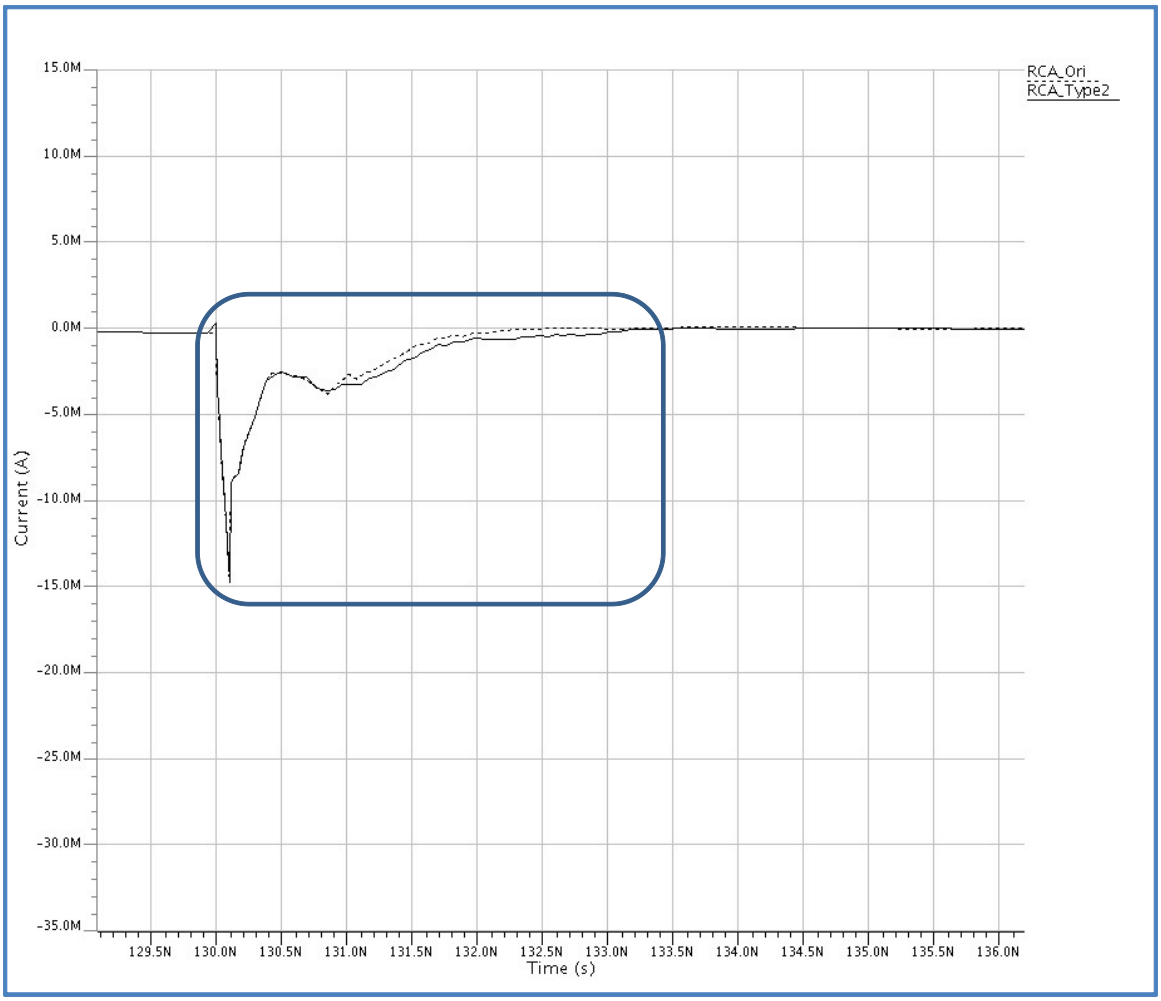


Figure C.4: Comparing Power Signature of 34-bit RCA with Smart Smart SSR toward Type II (Case4)

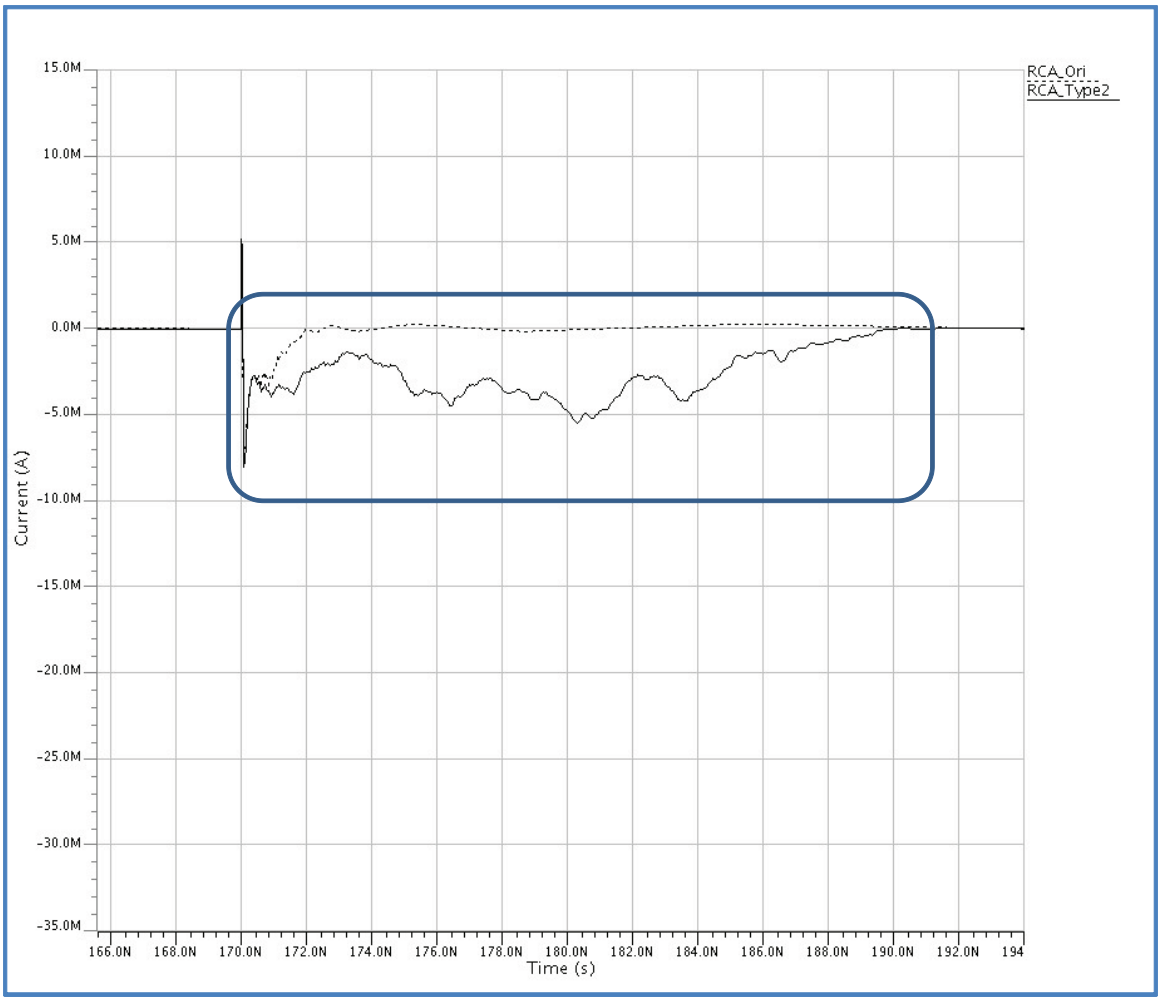


Figure C.5: Comparing Power Signature of 34-bit RCA with Smart Smart SSR toward Type II (Case5)

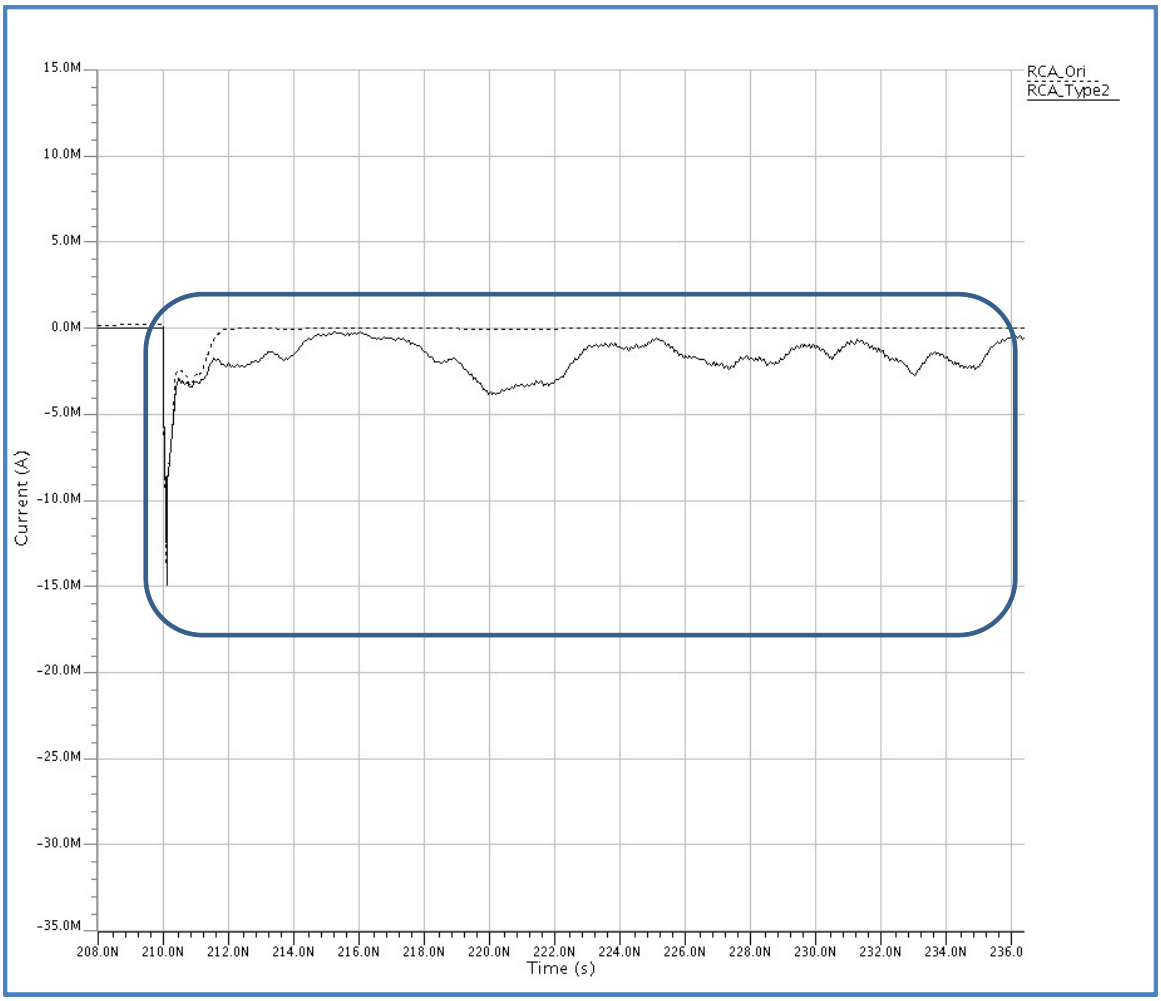


Figure C.6: Comparing Power Signature of 34-bit RCA with Smart Smart SSR toward Type II (Case6)

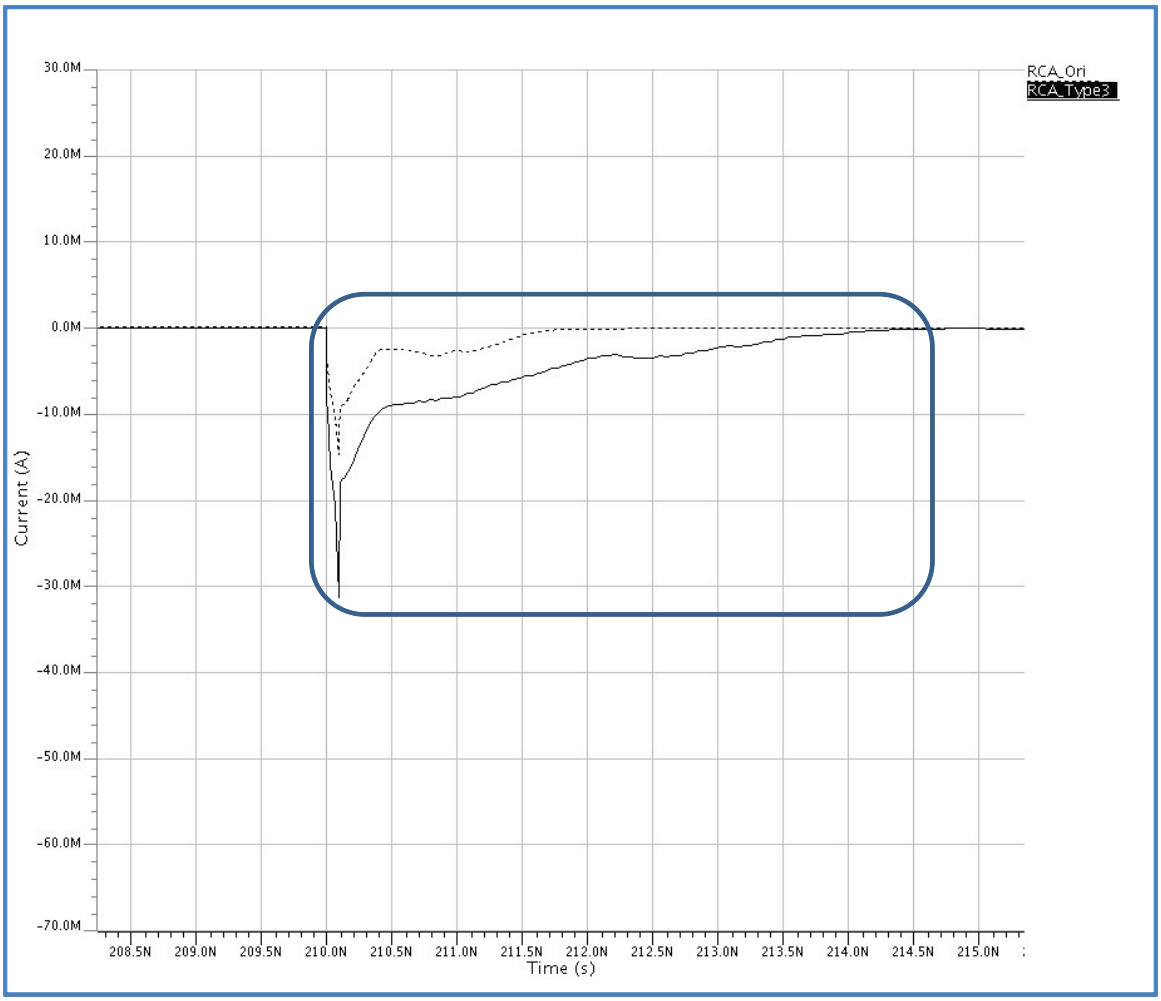


Figure C.7: Comparing Power Signature of 34-bit RCA with Smart Smart SSR toward Type III (Case1)

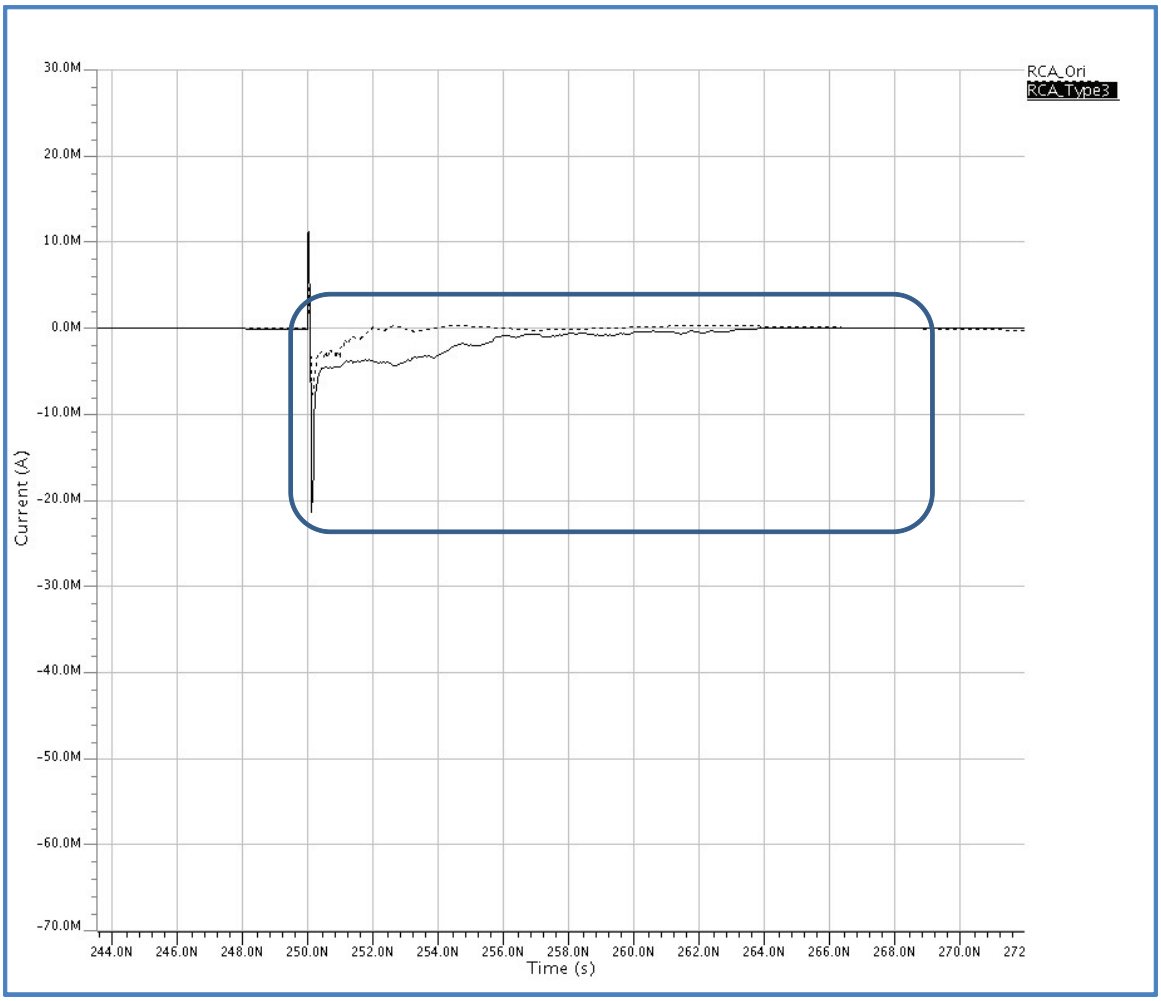


Figure C.8: Comparing Power Signature of 34-bit RCA with Smart Smart SSR toward Type III (Case2)

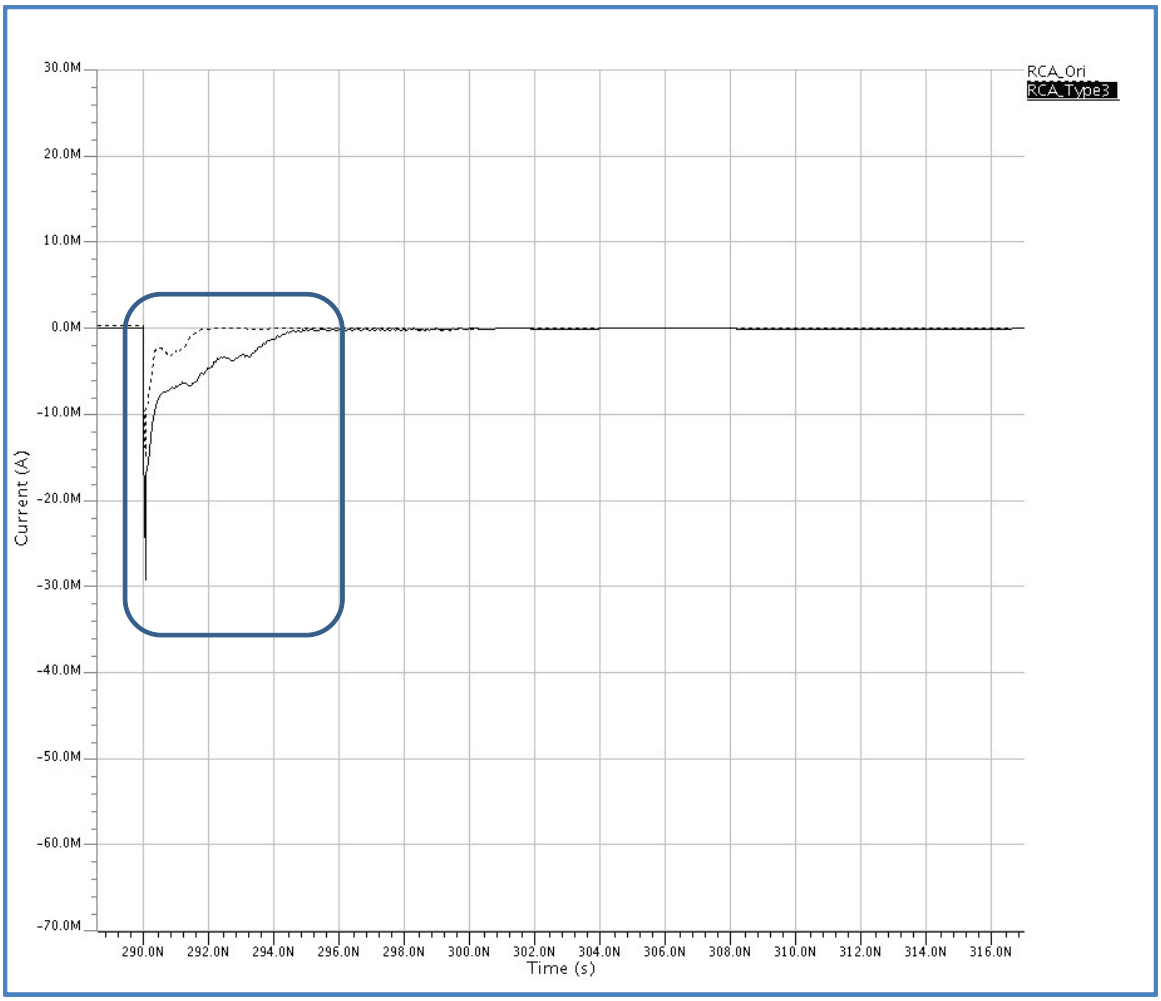


Figure C.9: Comparing Power Signature of 34-bit RCA with Smart Smart SSR toward Type III (Case3)

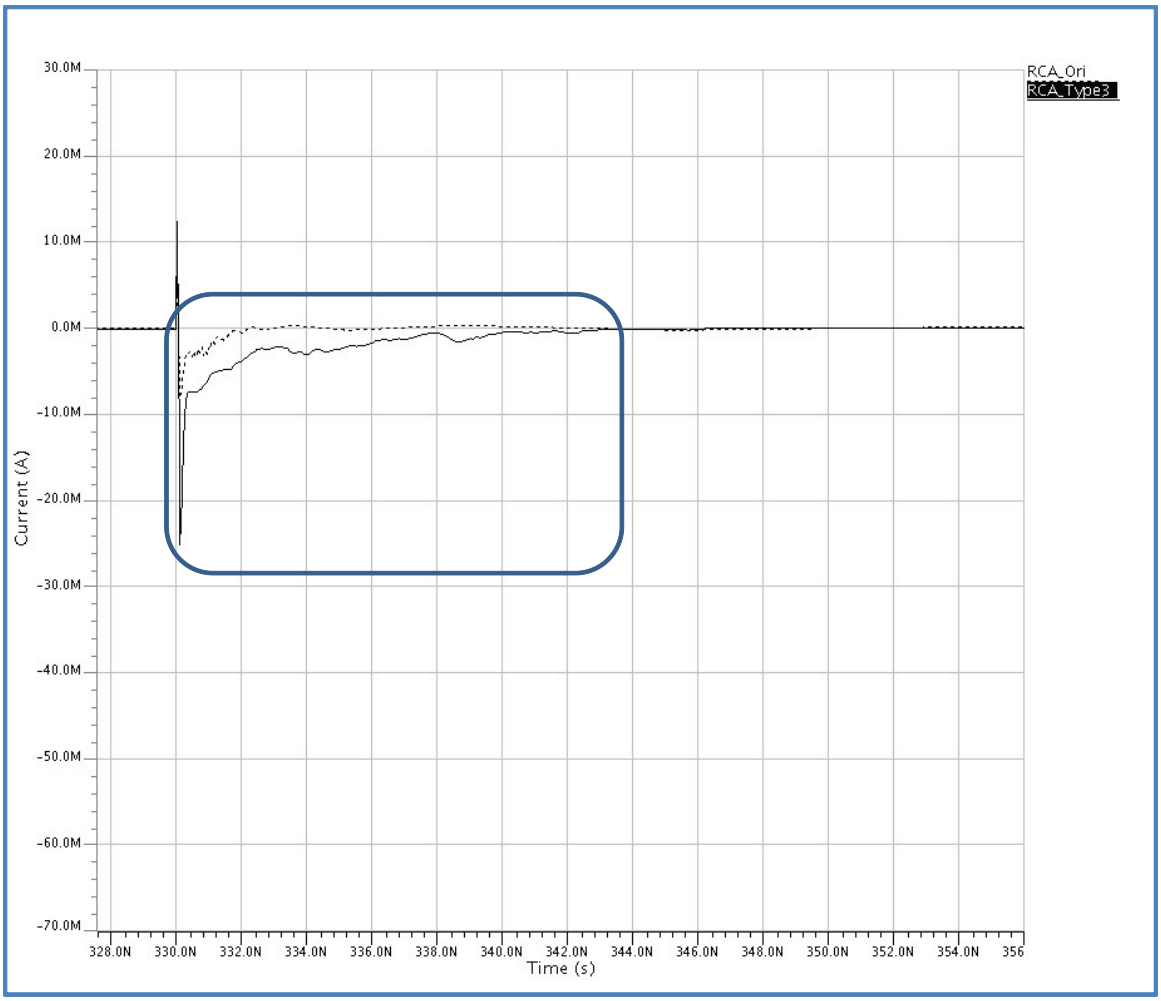


Figure C.10: Comparing Power Signature of 34-bit RCA with Smart Smart SSR toward Type III (Case4)

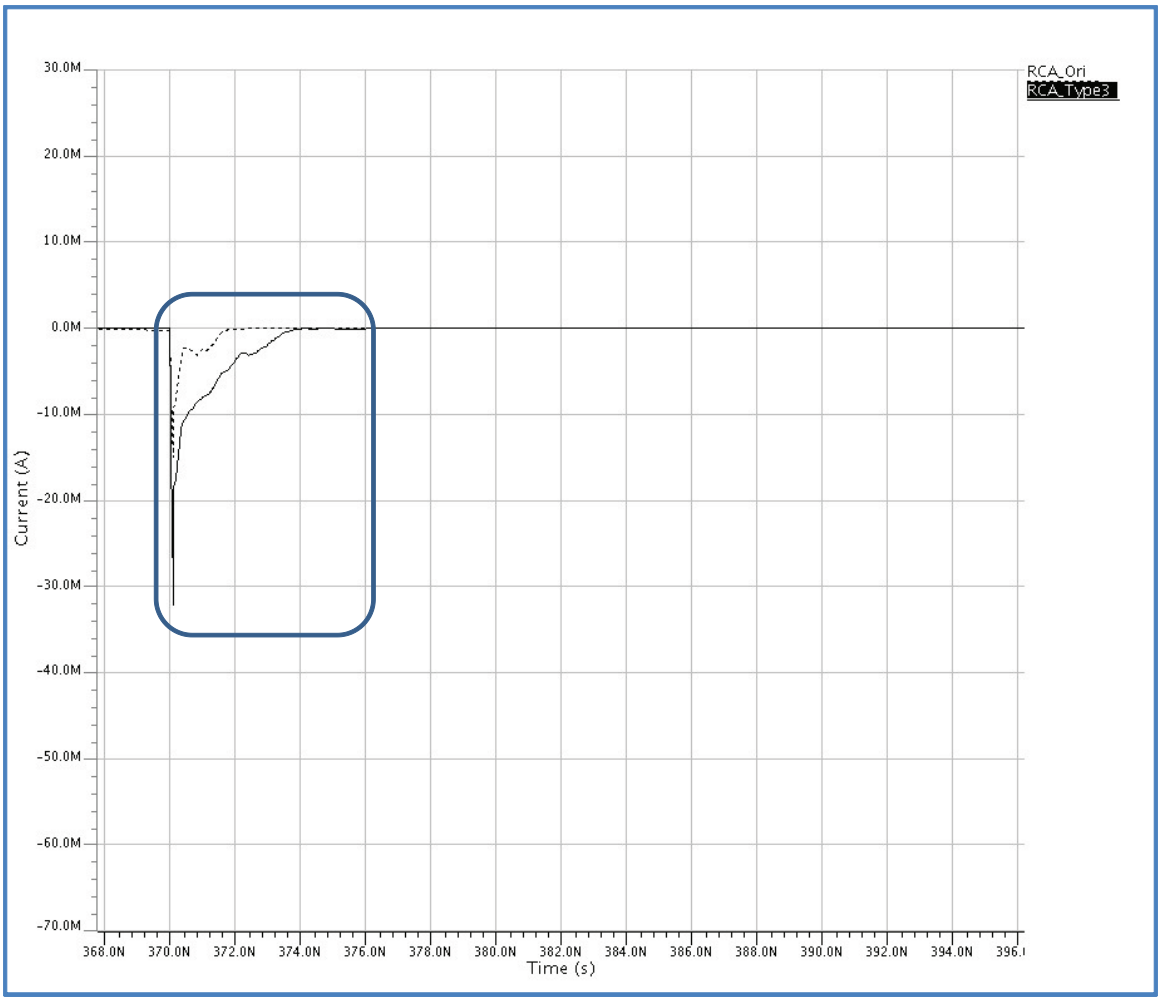


Figure C.11: Comparing Power Signature of 34-bit RCA with Smart Smart SSR toward Type III (Case5)

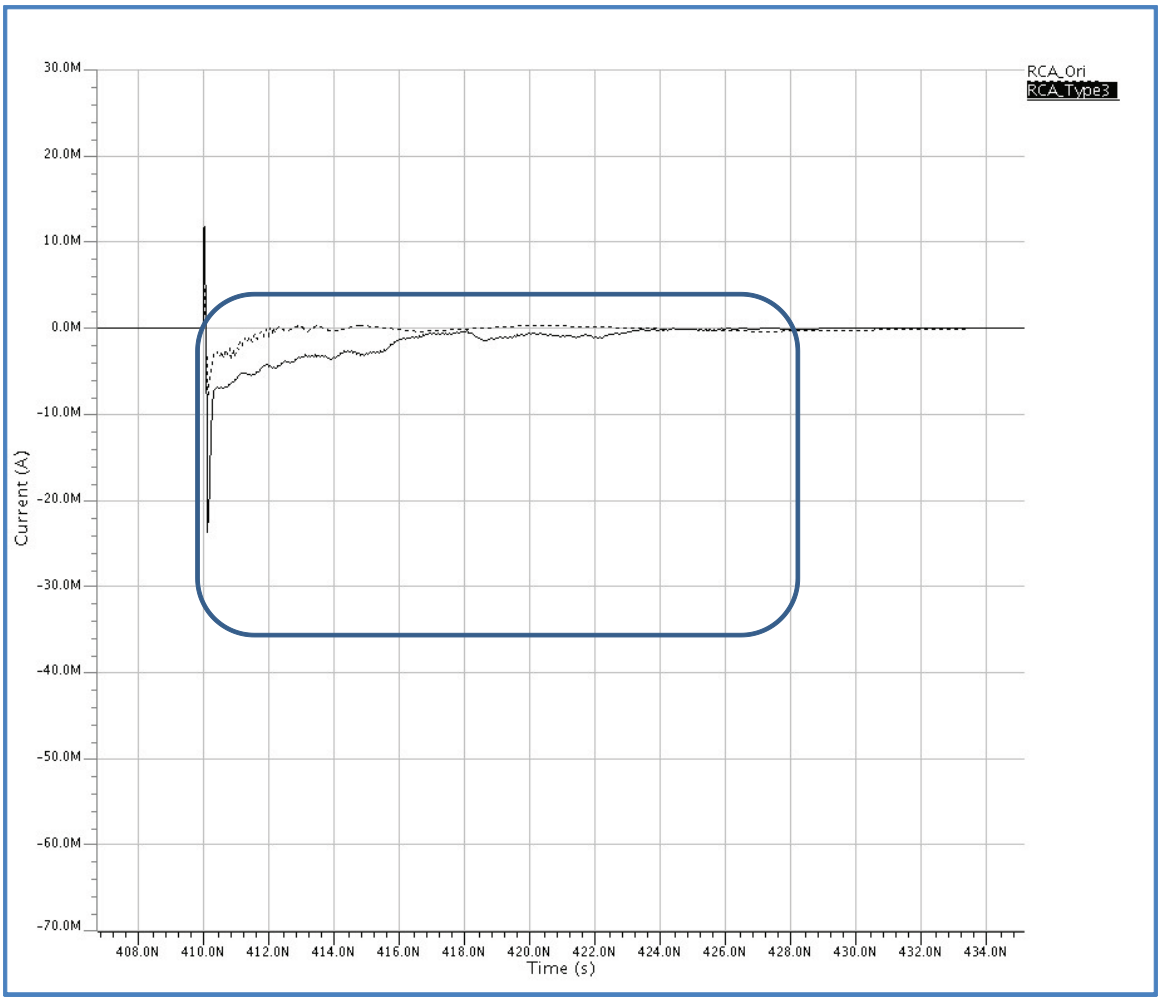


Figure C.12: Comparing Power Signature of 34-bit RCA with Smart Smart SSR toward Type III (Case6)

Vita

Captain Hyunchul Ko was commissioned through Officer Training School in Yeongcheon-si, Gyeongbuk, Korea in 2004. Captain Ko was first assigned to the Army Signal School in Yuseong-gu, Daejeon, South Korea for the Computer Officer Basic Course Training. Before leaving the Army Signal School, he was honored for the most exceptional academic achievement by the Korean Army Chief of Staff. Capt Ko was next assigned to Army Computer Center in Army Headquarters, Gyeryong-si, Chungnam, South Korea as a software QA officer. He earned the gold medal for the competition of military software development in 2006.

He was transferred to National Defence Computer Center in Ministry of National Defence, Seoul, South Korea in October 2007 as a network management officer. In 2009, Capt Ko entered the Graduate School of Engineering and Management, Air Force Institute of Technology. Upon graduation, he will be assigned to the Army Signal School in Yuseong-gu, Daejeon, Korea for the Officer Advanced Course Training.

Permanent address: Hite Mansion Ga-Dong 101-Ho
Sosa-Dong 33-1, Wonmi-Gu, Bucheon-Si, Gyunggi-Do
Republic of Korea

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704–0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
16–06–2011	Master's Thesis	Aug 2009 — June 2011

4. TITLE AND SUBTITLE

Combinational Circuit Obfuscation
Through Power Signature Manipulation

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Hyunchul, Ko, Captain, ROKA; green7767@gmail.com

5d. PROJECT NUMBER

ENG 10-326

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GCS/ENG/11-05

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Dr. Robert L. Herklotz
Program Manager - Information Operations and Security
Air Force Office of Scientific Research(AFOSR/RSL)
875 N. Randolph Street, Suite 325, Room 3112
Arlington VA 22203-1768
703-696-6565 (DSN: 426) robert.herklotz@afosr.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

AFOSR/RSL

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED
This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Today's military systems are composed of hardware and software systems, many of which are critical technologies, and must be protected to ensure our adversaries cannot gain any information from a various analysis attacks. Side Channel Analysis (SCA) attacks allow an attacker to gain the significant information from the measured signatures leaked by side-channels such as power consumption, and electro-magnetic emission. In this research the focus on detecting, characterizing, and manipulating the power signature by designing a power signature estimation and manipulation method. This research has determined that the proposed method capable of characterizing and altering the type of power signature can provide a protection against adversarial SCA attacks.

15. SUBJECT TERMS

software protection, side-channel analysis, obfuscation, power signature manipulation

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Yong C. Kim
U	U	U	U	127	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636, x4620; yong.kim@afit.edu