

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GÜÇ TÜKETİM TABANLI YAN
KANAL ANALİZİ KULLANILARAK
DONANIMSAL TRUVA ATI TESPİTİ

MEHMET KOCABAŞ
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

GEBZE
2021

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GÜÇ TÜKETİM TABANLI YAN
KANAL ANALİZİ KULLANILARAK
DONANIMSAL TRUVA ATI TESPİTİ

MEHMET KOCABAŞ
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMANI
DR. ÖĞR. ÜYESİ ALP ARSLAN BAYRAKÇI

GEBZE
2021

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

**HARDWARE TROJAN HORSE
DETECTION WITH SIDE CHANNEL
ANALYSIS BASED ON POWER
CONSUMPTION**

MEHMET KOCABAŞ
**A THESIS SUBMITTED FOR THE DEGREE OF
MASTER OF SCIENCE
DEPARTMENT OF COMPUTER ENGINEERING**

THESIS SUPERVISOR
ASSIST. PROF. DR. ALP ARSLAN BAYRAKÇI

GEBZE
2021

GEBZE TEKNİK ÜNİVERSİTESİ	YÜKSEK LİSANS JÜRİ ONAY FORMU
----------------------------------	--------------------------------------

GTÜ Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 29/01/2021 tarih ve 2021/05 sayılı kararıyla oluşturulan jüri tarafından 18/02/2021 tarihinde tez savunma sınavı yapılan Mehmet KOCABAŞ'ın tez çalışması Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI) : DR. ÖĞR. ÜYESİ ALP ARSLAN BAYRAKÇI

ÜYE

: DOÇ. DR. MEHMET GÖKTÜRK

ÜYE

: DOÇ. DR. TUFAN COŞKUN KARALAR

ONAY

Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun

...../...../..... tarih ve/..... sayılı kararı.

İMZA/MÜHÜR

ÖZET

Teknolojinin hızlı gelişimi, gelişmiş ülkelerin çoğu için bile yerli üretimi sürdürmenin imkânsız olduğu artan tümleşik devre fabrikasyon maliyetlerini beraberinde getirir. Buna paralel olarak, maliyetlerin minimum olduğu ve daha düşük ücretlerle kalifiye iş gücünün mevcut olduğu Çin ve Tayvan gibi uzak doğu ülkeleri tarafından tümleşik devrelerin imalatı daha çok ele alınmaktadır. Bu nedenle, kötü amaçlı devrelerin tümleşik devrelerin içine kasıtlı olarak eklenmesiyle ilgili endişeler, donanım Truva Atı algılama sorununa olan ilgiyi gittikçe arttırmaktadır. Donanım Truva atlarının tespiti için önerilen birden fazla yöntem mevcuttur. Yan kanal analizine dayalı algılama yöntemi, bunlardan en popüler olanlardan birisidir. Fakat bu yöntemin bazı büyük kusurları mevcuttur. Bunlardan en önemlisi, özellikle donanımsal Truva atının küçük olması durumunda, etkisinin kolayca gizlenebildiği üretim süreci varyasyonlarıdır. Bu tezde, öncelikle devreye, doğru ve gerçekçi bir varyasyon modeli uyguluyoruz ve ardından dinamik akım tabanlı yan kanal analizi yöntemini kullanarak küçük donanımsal Truva atlarını tespit etmenin zorluğunu ölçüyoruz. En hassas transistor seviyesine sahip Spice simülasyonlarını, doğru sonuçları elde etmek için kullanılmaktadır. Sonuçlar, Truva atı algılamanın belirli bir girişe, farklı güç kaynaklarının sayısına ve devrenin boyutuna duyarlılığını göstermektedir. Bu tezde, arzu edilen güç kaynağı sayısı ne olursa olsun, orijinal devreye yalnızca iki ek pim ekleyerek güç kaynaklarının ayrılmasını fiilen gerçekleştiren küçük bir devre öneriyoruz. Bu nedenle, donanım Truva atının etkisi, süreç varyasyonlarının etkisi altında bile ortaya çıkana kadar farklı dinamik akım ölçümlerinin sayısı artırılabilir.

Anahtar Kelimeler: Tümleşik Devreler, Kötücül Amaçlı Devreler, Donanımsal, Varyasyonlar, Güç Truva Atı, Yan Kanal Analizi, Spice Simülasyonları, Devreyi Izgaralara Bölme.

SUMMARY

The rapid development of technology comes with increasing integrated circuit fab costs impossible to carry on domestic manufacturing even for most of the developed countries. In parallel to this, the manufacturing of integrated circuits handled mostly by far-east countries like China and Taiwan, where the costs are minimal and skilled labor with lower wages exists. Hence, the concern about the intentional insertion of malicious circuits inside the integrated circuits draws an increasing interest towards hardware Trojan detection issue. There is multiple method suggested for the detection of hardware Trojan horses. Side channel analysis based detection is one of the most popular ones. But it has some major flaws. The most important of these is the manufacturing process variations that can easily hide the effect of the hardware Trojan, especially when it is small. In this thesis, we first apply an accurate and realistic variation model to the circuit and then measure the difficulty of detecting small hardware Trojans, using dynamic current based side channel analysis. The most precise transistor level Spice simulations are utilized for accurate results. The results demonstrate the sensitivity of Trojan detection to the particular input, to the number of distinct power supplies and to the size of the circuit. In this thesis, we also propose a small circuit to actually realize the separation of power supplies by only adding two additional pins to the original circuit whatever the number of desired power supplies is. Thus, the number of distinct dynamic current measurements can be increased until the effect of the hardware Trojan reveals even under the effect of process variations.

Key Words: Integrated Circuits (IC), Malicious Circuits, Hardware, Variations, Power, Trojan (HT), Side Channel Analysis (SCA), Spice Simulations, Dividing the Circuit Into Many Grids.

ACKNOWLEDGEMENTS

My advisor Asst. Prof. Dr. Alp Arslan BAYRAKÇI, who guided me in every stage of my thesis work correctly, who constantly controls me in the process of fulfilling every task he has given, ensures that my thesis progresses in a healthy way, and who has a great share in bringing my thesis to this stage by always making me feel his breath on my neck.

To Assoc. Prof. Dr. Mehmet GÖKTÜRK and Assoc. Prof. Dr. Tufan Coşkun KARALAR for their acceptance of being a jury member in my defense and their valuable comments.

My only wife, Gülay KOCABAŞ, who has always been by my side during this difficult process and who has the biggest share in my graduation by constantly supporting me.

In addition to making me the happiest father in the world when she was born, to my only daughter Deniz KOCABAŞ, who took my stress by entertaining me in my thesis with her unwitting cuteness.

To Celil MUTLU, Hasan BARIŞ and Yusuf ERENTÜRK, whom I love like my brother, who has always been with me since my childhood and who supported me during my education and had a great share in my coming here.

To my aunt Teslim KOCABAŞ and my uncle Ergün KOCABAŞ, who always treated me like their own children and taught me what family is.

All of my teachers whose names I remember or not who contributed to me to come this far.

First of all, I owe a debt of gratitude to my only grandmother, Bahriye KOCABAŞ, who I cannot pay for the labor that made me be me.

TABLE OF CONTENTS

	<u>Page</u>
ÖZET	v
SUMMARY	vi
ACKNOWLEDGEMENTS	vii
TABLE OF CONTENTS	viii
LIST OF ABBREVIATIONS AND ACRONYMS	x
LIST OF FIGURES	xi
LIST OF TABLES	xiii
1. INTRODUCTION	1
2. PRELIMINARIES AND LITERATURE	4
2.1. Instantaneous Power	4
2.2. Hardware Trojan	6
2.3. Trojan Model, Instances, And Classification	7
2.3.1. Trojan Models and Examples	7
2.3.2. Trojan Taxonomy	10
2.4. Countermeasures	12
2.4.1. Challenges	12
2.4.2. Different Approaches for Solving HTH Problem	13
2.5. Hardware Trojan Detection	14
2.5.1. Logic Testing	15
2.5.2. Side-Channel Analysis	16
3. POWER BASED HT DETECTION USING SEPARATION TO GRIDS	18
3.1. Motivation	18
3.2. Modeling Parameter Variations	19
3.3. The Trojan Horse Used in Experiments	20
3.4. Effect of Variations on Conventional Power Based SCA	21
3.5. Separation to Grids	23
3.6. I/O Issue of Separated Grids Approach	25
4. RESULTS	28
4.1. Experimental Setup	28
4.2. Results for Separated Power Supplies	29

4.3. Dependence on the Input	30
4.4. Dependence on the Grids	31
4.5. Dependence on the Circuit Size	32
5. CONCLUSION	34
REFERENCES	36
BIOGRAPHY	38

LIST of ABBREVIATIONS and ACRONYMS

<u>Abbreviations and Acronyms</u>	<u>Explanations</u>
C	: Capacitor
$i()$	: Current
E	: Energy
f	: Frequency
G	: Grid
i	: Input
L	: Length
t	: Time
P	: Power
V	: Voltage
CMOS	: Complementary Metal Oxide Semiconductor
CPU	: Central Processor Unit
DAC	: Design Automation Conference
DFS	: Design for Security
EM	: Electromagnetic
HOST	: Hardware Oriented Security and Trust
HTH	: Hardware Trojan Horse
IC	: Integrated Circuit
ICCAD	: International Conference on Control, Automation and Diagnosis
IP	: Intellectual Property
ISCAS	: International Symposium on Circuits and Systems
ITRS	: International Technology Roadmap for Semiconductors
MCU	: Micro Controller Unit
PMOS	: P-channel Metal-Oxide Semiconductor
SCA	: Side Channel Analysis
SNR	: Signal-to-Noise Ratio
TCR	: Trojan-to-Circuit Ratio
VLSI	: Very Large-Scale Integration

LIST of FIGURES

<u>Figure No:</u>	<u>Page</u>
2.1: CMOS Inverter.	4
2.2: CMOS Parasitic Capacitance.	5
2.3: Inverter Very-Large-Scale Integration (VLSI) Design.	6
2.4: Basic HTH Gate Design.	6
2.5: Combinational and Sequential Trojans General Model.	8
2.6: Detailed Taxonomy Showing Physical, Activation, and Action Characteristics of Trojans.	10
2.7: Overview of different Protection Approaches Against Hardware Trojan Attacks.	13
2.8: Logic testing approach for hardware Trojan detection based on multiple excitation of rare conditions (a) for combinational Trojan (b) for sequential Trojan.	15
3.1: Added Hardware Trojan Horse.	20
3.2: Conventional power-based HT detection with single power supply for 48th input (64-bit adder): (a) Power (in Watt) vs. time (b) Energy (in Joule) vs. sample no.	22
3.3: A symbolic chip divided into grids.	23
3.4: Circuit to Provide Individual Power Supply.	25
3.5: Further decrease in additional pins for supply separation.	26
4.1: Power vs. time and energy vs. sample no plots for 64-bit adder (same x, y axis and scale as in Figure 4.2): $P(t)$ for (a1) 2×2 grid (a2) 4×4 grid (a3) 8×8 grid and $E(tr_{48})$ for (b1) 2×2 grid (b2) 4×4 grid (b3) 8×8 grid.	29
4.2: Power vs. time and energy vs. sample plots for 14 th input for grid (6, 7) (64-bit adder): (a) $P(t)$ (in Watt), 8×8 grid (b) $E(tr_{14})$ (in Joule), 8×8 grid.	31
4.3: Power vs. time and energy vs. sample no plots for 48 th input for grid (5, 6) (64-bit adder): (a) $P(t)$ (in Watt), 8×8 grid (b) $E(tr_{48})$ (in Joule), 8×8 grid.	31

4.4.: Power vs. time and energy vs. sample no plots for 16-bit adder (same 32
x, y axis and scale as in Figure 4.2): $P(t)$ for (a1) 2×2 grid (a2)
 4×4 grid (a3) 8×8 grid and $E(tr_{82})$ for (b1) 2×2 grid (b2)
 4×4 grid (b3) 8×8 grid.

LIST of TABLES

<u>Table No:</u>	<u>Page</u>
2.1: Comparison of Logic Testing and Side-Channel-Analysis-Based Trojan Detection.	14
4.1: Truth Table.	20

1. INTRODUCTION

Parallel to the development of technology, provision of information security is vital. If this security cannot be provided, the damage that may occur is not only materially limited but may also result in vital consequences. Information security here refers not only to the security of a software, but also to the security of a hardware or even an integrated circuit (IC).

As it is known, the hardware is used from the remote control where we use our home, to the mobile phone in our pockets, to the plane, tank etc. that are used in the military field. Given today's technology, we cannot imagine any electronic product that does not contain hardware. This hardware includes many components which are from less complex (Adder, Multiplexer, Flip-Flop etc.) to more complex (CPU, MCU, etc.). Many of these ICs are not designed in our country. Even if they can be designed in domestic companies, they are manufactured in mostly far east countries. At this point, there is a distrust in such ICs. The main reason for this is that these ICs can be deliberately modified by an adversary during manufacturing and cause the hardware to behave differently than desired one. The most important part here is that the designed integrated circuit will behave differently than desired when it is only in a specific state. We call such intentional malicious modifications in the circuit as "hardware Trojan horses (HTH)" [1].

In the literature, HTHs are the malicious altering of hardware specification or implementation in such a way that its functionality is altered under a set of conditions defined by the attacker [2]. There are many methods for detecting such changes. An obvious solution is to physically open the ICs with destructive methods, and then to determine with sensitive devices whether interior design is different or not than the desired one. But the biggest problem with destructive methods is that the investigated ICs are destructed and become useless and also it does not guarantee that other ICs do not have a Trojan. Because the Trojan horse can be selectively inserted only to a portion of manufactured ICs. Therefore, this method is not the preferred for detecting HTH. So, different methods have been developed to detect HTHs. One of them is power based HTH detection which is used in our thesis.

Adding a Trojan to an IC means interfering with the physical conditions of this IC. This physical intervention may sometimes be thought of as adding an extra

transistor, adding extra paths, or adding another ICs. Any changes made to physical conditions will cause deviations in the measurements obtained during the operation of this IC. In power based detection the aim is to detect the deviation at the currents that the IC will draw under various operating conditions. There are 2 main working conditions here. One of them is static current in which the presence of a Trojan circuit will be reflected in the current drawn from the power supply, even if no switching occurs in the Trojan circuit. The other one is the transient current. The goal of transient current (I_{DDT}) analysis is to detect switching activity inside a Trojan circuit [3].

In our thesis the main approach of side channel analysis based on power/current measurements is investigated and a solution to HTH detection problem based on this approach is aimed. Side channel analysis methods are superior in the sense that they are both non-destructive and easier to implement. The research also considers the realistic process parameter variation models as well as the spatial correlations between circuit parameters.

There are already many projects in our country especially in the military field. Most of the ICs which are to be used on existing projects or on the hardware to be used in new projects are supplied from abroad. In this reason, this is a serious security issue for our country. To remove this security problem as much as possible, it is necessary to apply such tests on ICs. Therefore, hardware Trojan detection will be one of the most important research topics in the future, especially in our country. There are currently large conferences with HTH detection and hardware security topics worldwide. The most famous ones are Hardware Oriented Security and Trust (HOST), The International Conference on Control, Automation and Diagnosis (ICCAD) and Design Automation Conference (DAC) conferences.

In this thesis, we first apply a realistic variation model explained in Section 3.4. For all results in the thesis, the precise transistor level Spice simulations in a Monte Carlo fashion are performed. This thesis aims to quantify the effect of Trojan on power, the effect of parameter variations on HTH detection, the detection performance using single vs. multiple power supplies with layout separation to grids, the required level of grid separation for a successful detection without a need for golden model, the effect of circuit inputs on detection. In Section 3.6 we also devise a circuit that decreases the number of additional input pins to make the grid separation applicable to actual circuits. Results on our test circuit synthesized with

45nm cell library, reveal the importance of the separation level of the circuit layout and the negative effect of variations on avoiding the power based HTH detection.

Section 2 gives the preliminaries and the literature review to clarify power based HTH detection. Section 3 explains the utilized variation model, separation the circuit to grids and the proposed circuitry to decrease the number of additional inputs. Section 4 gives all demonstrating the effect of grid size, inputs and circuit size on power based HTH detection. Last section concludes the thesis.

2. PRELIMINARIES AND LITERATURE

2.1. Instantaneous Power

There are more than one side channel analysis methods in the literature. The main ones are based on analysis of quantifiable quantities such as electromagnetic, path delay, temperature and power consumption. Below is the general formula of instant power consumption shown in (2.1). According to the formula, the power consumed instantly varies depending on the voltage value (V_{supp}) supplied by the source and the current ($i(t)$) drawn from the source.

$$P(t) = i(t) \times V_{supp} \quad (2.1)$$

The current in the equation depends on the transistor gate length and threshold voltage value, which are highly affected from manufacturing process variations [4]. The change in instantaneous current can hide the effect of insertion to the circuit for malicious purposes. This poses an obstacle to power-based Trojan detection. In the literature, the circuit is supplied from a single source in power-based hardware Trojan detecting normally. Different samples (chips) can be explored to detect the presence of any hardware Trojan horse by measuring the current value drawn from the source and making power and energy calculations. The hardware Trojan horse can be considered a small circuit. If we are to reduce this circuit to the simplest, it can be thought of as a structure formed by the combination of multiple transistors. If we think at the transistor level, the simplest structure of a CMOS is as shown below.

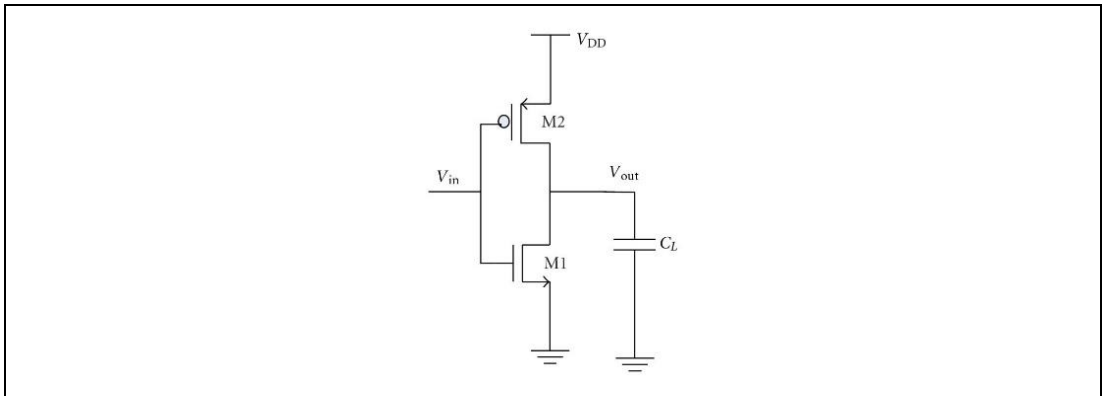


Figure 2.1: CMOS Inverter.

The following general formula (2.2) is used to measure the average dynamic power consumed by the CMOS circuit shown in the Figure 2.1[5]

$$P_{avg} = C_L \times V_{DD}^2 \times f \quad (2.2)$$

In formula, C_L represents the total capacitance, V_{DD} is the power supply and f is the frequency of the signal transition.

As can be seen from the formula (2.2), the current consumed is directly related to the load capacitance at the output of the CMOS circuit. So, if we consider our current circuit, every extra insertion we made means that an extra load inserted to our circuit. Since the current drawn from the source will increase due to this extra load, the total power consumption by the circuit will also increase. By making use of the changes that occur in this power, it is possible to detect hardware Trojan horses using the power-based detection method.

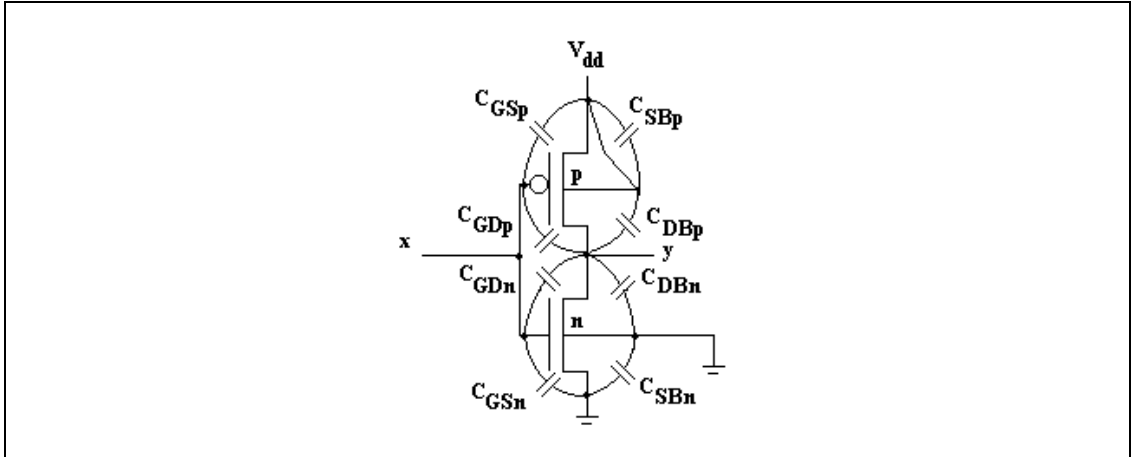


Figure 2.2: CMOS Parasitic Capacitance.

If we examine the more detailed CMOS circuit above, the total capacitance at the CMOS output is equal to the sum of the capacitances contained in the Figure 2.2 [6], and the states of these capacitors are directly related to the order of inputs and inputs given. In other words, the total capacitance at the output varies depending on the type and order of the given input. Since changing the total capacitance will also change the power consumed, the input given, and the order of the inputs are important in determining the hardware Trojan horses. In Section 4 we will talk about this in more detail. Another important issue is variations.

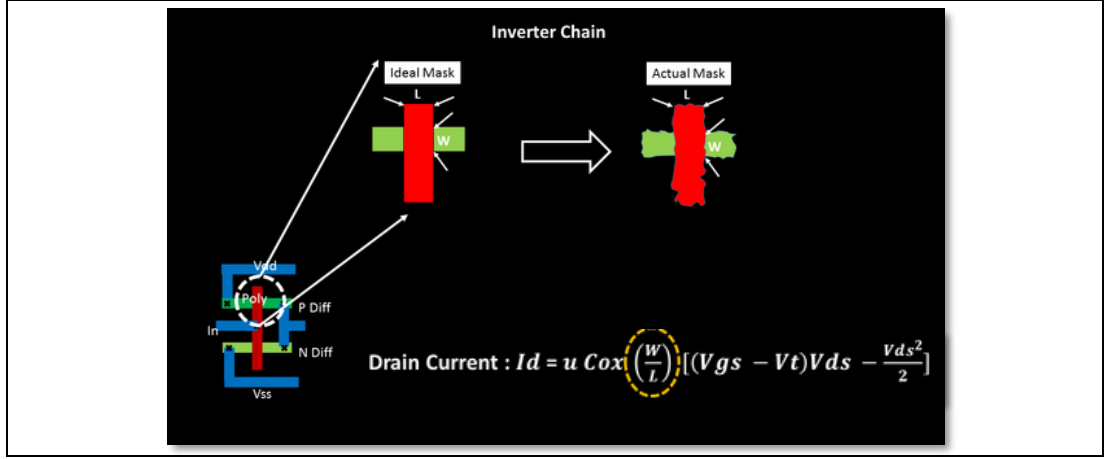


Figure 2.3: Inverter Very-Large-Scale Integration (VLSI) Design.

In its simplest form, a transistor design is made like the ideal mask seen in the Figure 2.3[7]. However, no matter how smooth the transistor is designed, after the fabrication step, the actual mask is produced in a deformed form, as will be seen in the Figure 2.3. Changes that occur in this fabrication step are called variations. If we examine the formula in the Figure 2.3, the current drawn from the source is also related to the length and width of the gate channel. Therefore, the total power consumed varies with the effect of variations occurring during the fabrication step. These changes make it difficult for us to detect hardware Trojan horses. Therefore, a variation model should be developed. We will talk about the variation model we have developed in Section 4 in more detail.

2.2. Hardware Trojan

There are many hardware Trojans that have been projected so far. Even a taxonomy has been created for this and studies have been made on it [1]. Hardware Trojan horses generally consist of two subsections. These are payload and trigger part as in Figure 2.4[8].

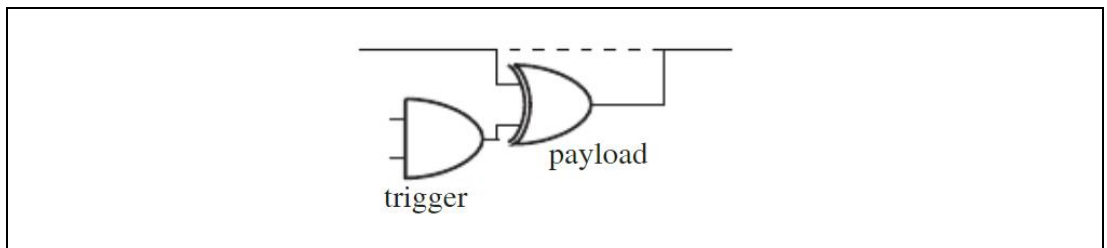


Figure 2.4: Basic HTH Gate Design.

The trigger part is responsible for activating the Hardware Trojan horse, while the payload part is the part that operates the malicious activity in the circuit after it is triggered. Essentially, the HTH is malicious modification in the circuit, and since these changes are minor changes, it becomes difficult to distinguish the hardware Trojan horse from process variations. This is the biggest factor that complicates the work of the side channel analysis methods. To protect the functionality of the circuit, one XOR gate has been identified as the payload portion of the hardware Trojan. The purpose of this is that the added XOR gate causes a change in the circuit for malicious purposes as we want and can remain passive until it is activated with a certain input.

This makes it impossible to detect Hardware Trojan horses using conventional functional tests. Therefore, it is more convenient to use side channel analyses to detect hardware Trojan horses. If one input pin of the XOR gate is kept at logical zero, the output of the XOR gate is the same as the other input value. Since this does not change the functionality of the circuit, the effect of the hardware Trojan horse cannot be seen in the circuit. But if the input that is held as logical zero is pulled to logical one, this time the output of the XOR gate will be the opposite of the input, and the hardware Trojan horse will become active and cause the system to operate differently from its normal behavior. These changes are called hardware Trojan horses, which are made for malicious purposes in the circuit and change the normal behavior of the system under a certain condition and make the system inoperable or cause it to operate differently from the normal. According to the literature, hardware Trojan horses (HTHs) are the malicious altering of hardware specification or implementation in such a way that its functionality is altered under a set of conditions defined by the attacker [2].

2.3. Trojan Model, Instances, And Classification

2.3.1. Trojan Models and Examples

A hardware Trojan does not need to be activated to be detected. Even the addition of the hardware Trojan horses may be sufficient to detect it. However, since they are more pronounced when they are activated, it is of great importance in establishing the conditions that will activate the hardware Trojan. The attacker who

will add the hardware Trojan horses adds the hardware Trojan horses the way making it difficult to detect in prefabricated tests [1]. Therefore, it is very likely that it cannot be detected in fabrication tests performed after the circuit is produced. That is why the Trojan horse is likely to be activated with a small probability, and long-term field tests are required to meet this possibility [9], [10].

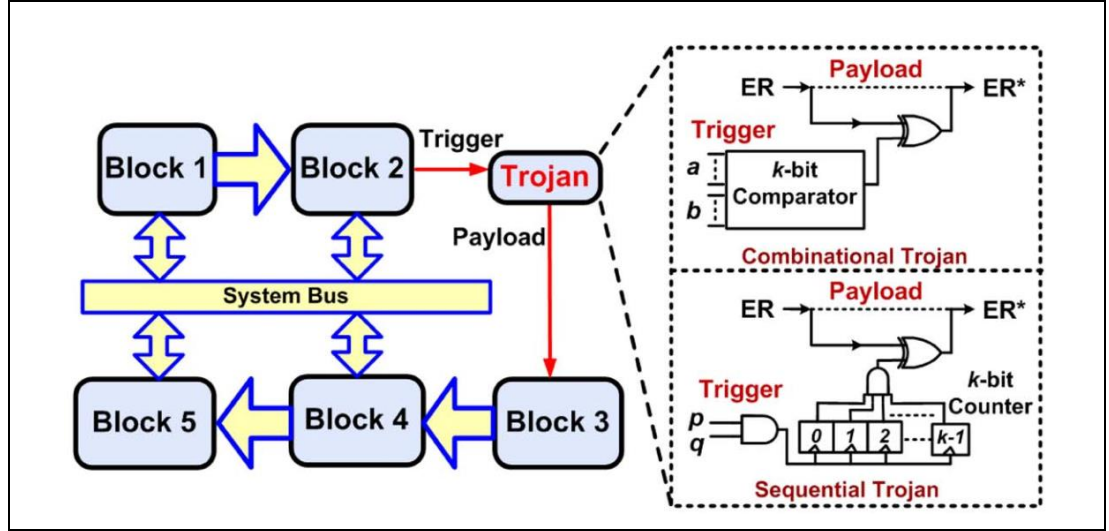


Figure 2.5: Combinational and Sequential Trojans General Model.

As can be seen from the Figure 2.5[11], there are 2 different models to be applied in order to detect the Trojan horse in field tests. One of them is a combinational model and the other is a sequential model.

Combinational models consist of logic circuits. Their output at any moment depends on the status of their current input. In other words, there is no change in time. The previous output has no effect on the next output. In order to activate the hardware Trojan horses using this model, the appropriate combination must be determined and applied to the circuit.

To give an example of the combinatorial model, let's consider a comparator comparing two m -bit numbers. Let the value given to the first input of the comparator be $\mathbf{m}_1$ and $\mathbf{m}_2$ the value given to the other input. Let's say that the attacker adds a hardware Trojan horse to be activated if the input for $\mathbf{m}_1$ is $\mathbf{x}_1$. As long as the $\mathbf{m}_1$ value is not $\mathbf{x}_1$, the system will continue normal operation since the hardware Trojan horse will not be triggered and the payload part will not be active. But if the $\mathbf{m}_1$ input is $\mathbf{x}_1$, the hardware Trojan horse will be triggered and the

payload part will be activated, so the hardware Trojan will be activated, and the system will work differently than it should be.

Sequential models, on the other hand, are the model structure in which the outputs of the circuit depend not only on the current state, but also on history of the inputs. Unlike combinational logic circuits, sequential logic circuits do not depend solely on current inputs. Sequential logic circuits have state memories. In other words, sequential logic circuits are combinational circuits carrying memory elements. Using this model, an appropriate sequence must be found and applied to the circuit in order to activate the Trojan horse.

If we give an example to the sequential model, let's assume that a Trojan horse is added that listens to the serial communication line. Let's assume that this hardware Trojan is active when there is a "10101" sequence in the bits from the serial line. If there is no "10101" sequence in the incoming bits, the system will continue normal operation since the Trojan horse will not be triggered and the payload part will not be active.

However, if a sequence that is "10101" is detected within the bits from serial communication, the payload will be active in the payload part since the Trojan horse will be triggered and the system will start working differently from its normal behavior. Here, all the other bits that come before the last bit in the sequence must be stored in a memory to determine whether the sequence that will activate the Trojan horse has come. Otherwise, it will not be possible to determine whether the sequence has come and thus the Trojan horse will never be activated. If we give an example to the sequential model, let's assume that a Trojan horse is added that listens to the serial communication line. Let's assume that this Trojan horse is active when there is a "10101" sequence in the bits from the serial line. If there is no "10101" sequence in the incoming bits, the system will continue normal operation since the Trojan horse will not be triggered and the payload part will not be active. However, if a sequence of "10101" is detected within the bits from serial communication, the payload will be active in the payload part since the Trojan horse will be triggered and the system will start working differently from its normal behavior.

2.3.2. Trojan Taxonomy

It is aimed to classify Trojans to capture the main characteristics of Trojans and identify and evaluate the capabilities of various detection strategies. In this context, Trojans are divided into three main categories according to their physical, activation and action characteristics.

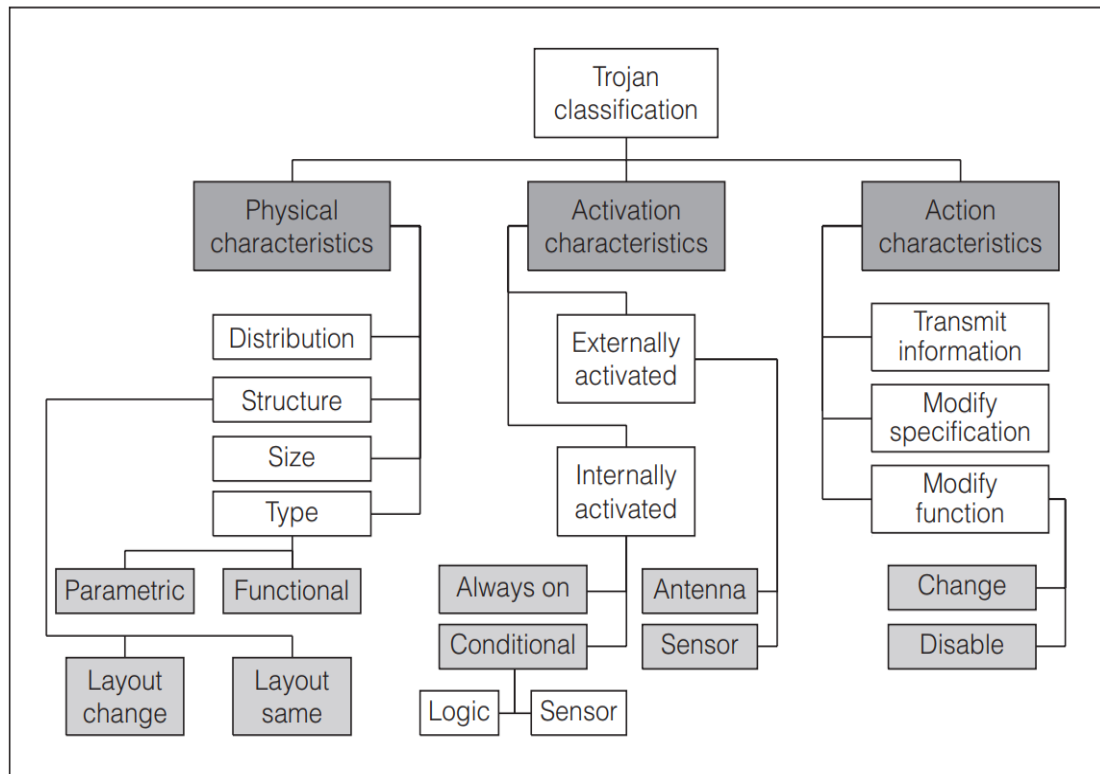


Figure 2.6: Detailed Taxonomy Showing Physical, Activation, and Action Characteristics of Trojans.

As we can see from the Figure 2.6 [12];

Physical category is divided into 4 subcategories like distribution, structure, size and type. The *distribution* category indicates the location of the hardware Trojan in the physical layout of the chip. The *structure* category is related to case when an adversary who will insert the hardware Trojan is forced to regenerate the layout during the insertion of the hardware Trojan. While regenerating the layout, in some cases it disrupts the physical structure of the chip. In some cases, it regenerates the layout without disrupting the physical structure. The *size* category represents the number of components that are added, deleted or altered from the chip. The *type*

category is divided into two subcategories too: functional and parametric. The *functional* part includes Trojans that are physically produced by adding or deleting logic gates and transistors. The *parametric* part includes tampering on existing wires and logic.

Activation category includes under which conditions a hardware Trojan horse will be active and what conditions must be met in order to perform its destructive functions. We can collect activation features under two different titles. These are *external activation* and *internal activation*. Trojans that are *always on* are always active and can disrupt the function of the integrated circuit at any time. They are usually added to nodes and paths that have a low probability. These types of hardware Trojans are mostly implemented to increase the failure sensitivity of the nodes and paths to which they are added. For the Trojans in the *conditional* activation class to be active, a certain condition must be met. They become active after this condition is met. There are several ways to activate these types of Trojans. One of them can be considered as the case that the output value of any *sensor* (temperature, humidity, voltage, height, etc.) is equal to the value required for the Trojan horse to be active. The condition to be active here can be based on the output value of a single sensor or on the output value of several sensors. Alternatively, hardware Trojans can be designed to be activated depending on an analog condition such as a sensor, or they can be designed as a *logic* circuit consisting of logic gates (flip-flops, comparator, shift register etc.). In this way, they can be activated according to a certain combination (for example, if the counter value is 0xABCDEF01) or according to the sequence of the incoming data (for example, if there is a bit sequence of 1001010 in the data from the communication line). *External* activation can be considered as designing the added hardware Trojan horse to interact with the outside world and activating it with the aid of signals (antenna, sensor) from here.

Action category can be considered as the classification of damages that Trojan horse can cause. We can collect these features under 3 main titles: modify function, modify specification and transmit information. *Modify function* is an action by adding extra logic gates to the integrated circuit or by removing some logic gates from the existing layout. *Modify specification* is the action performed by tampering on the parametric properties of the integrated circuit (tampering wire properties, tampering the structure of the transistors, etc.). *Transmit information* is the action

performed by transferring confidential information to the outside world for malicious purposes.

2.4. Countermeasures

2.4.1. Challenges

Post-manufacturing tests used today are not good at finding Trojans successfully. The main reason for this is that it tries to detect parametric defects or unacceptable changes of devices used in the manufacturing process that cause deviations in functional or parametric properties, rather than identifying various additional functions caused by deviation in circuit behavior triggered by a Trojan horse or arbitrary rare events. There are several key challenges with the reliable detection of Trojans using a post-silicon test / verification process.

First, an adversary can use multiple instances of Trojan horses of varying shapes and sizes [13]. These Trojans can also vary widely in structural and functional characteristics, including trigger conditions and payloads. This makes it difficult to create a logical model of Trojans. The number of possible Trojans has a combinational distribution based on the number of nodes in the circuit. As an example, even under the assumption of a maximum of four trigger nodes and a single payload, a relatively small ISCAS-85 benchmark circuit c880 with 451 gates could have $\sim 10^9$ triggers and $\sim 10^{11}$ different Trojan horse instances, respectively [14]. Therefore, it is not practical to determine all possible Trojan horse samples to generate test patterns or to calculate test coverage.

Second, activating arbitrary Trojan horse instances and observing the consequences they cause can be extremely challenging due to their generally inactive state. Therefore, it does not seem logical to apply deterministic and exhaustive tests.

Third, the presence of Trojans generally appears when they are active. Observing their effects on physical parameters (delay, power, process and measurement noise, electromagnetic field, etc.) is a major challenge when they are in passive state. Especially nowadays, with the shrinkage of transistor sizes, increasing process variations in advanced technologies can mask the small Trojan horse effects in the physical parameter.

2.4.2. Different Approaches for Solving HTH Problem

It is possible to classify the protections that can be taken against Trojan horse attacks under three main headings:

- i) Trojan detection approaches;
- ii) design for security (DFS) approaches;
- iii) runtime monitoring approaches.

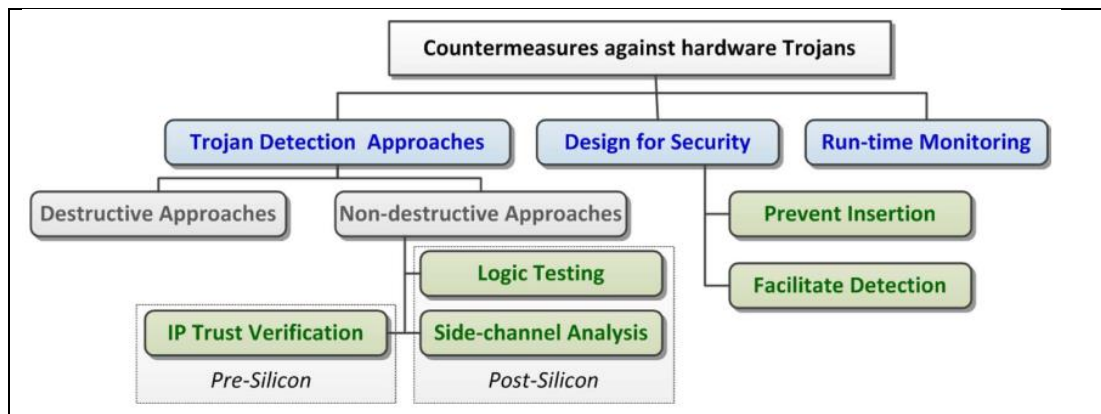


Figure 2.7: Overview of different Protection Approaches Against Hardware Trojan Attacks.

Figure 2.7 [11] shows a broad classification of measures against hardware Trojans. Typically, Trojan detection approaches can be classified into two categories as destructive and non-destructive approaches. Destructive approaches are also possible to detect the Trojan, but after using this method, it is not possible to reuse the IC again. Therefore, this approach is not preferred to detect Trojan horses. This method is used to obtain more golden models instead of detecting Trojan horses. Therefore, there is no need to take any precaution for this approach.

Non-destructive interventions can typically be subdivided into pre-silicone or post-silicone techniques.

The IP Trust Verification used in pre-silicon techniques is based on the use of a specific logic circuit previously designed by other people in the IC to be designed. Here it is necessary to verify that these logic circuits designed by others are reliable. For example, suppose that the NAND gates to be used in an adder are designed by

others. The reliability of the NAND gates used must be verified to verify the security of this adder.

Logic Testing approach can be evaluated as producing possible test models that can activate the Trojan horse. For that purpose, rare events are stimulated.

Side-channel Analysis method is an approach based on the detection of the Trojan horse by comparing the values of measurable parameters such as power, delay, temperature or electromagnetic (EM) radiation with the values previously obtained from the golden model.

Design for Security tries to prevent difficult-to-detect Trojans from being inserted to the integrated circuit or to facilitate their detection in post-silicon tests.

Run-time Monitoring is a precaution against potentially undetectable Trojans. The purpose of this method is based on monitoring the operating information of the circuit in which the IC is placed [3], [15].

2.5. Hardware Trojan Detection

There are two essential methods for detecting the Trojan horse as explained in the Table 2.1 [11]. One of them is logical tests and the other is side channel analysis.

	Logic Testing	Side – Channel Analysis
Pros	- Robust under process noise - Effective for detection ultra-small Trojans	- Effective for large Trojans - Easy to generate test vectors
Cons	- Difficult to generate test vectors - Large Trojan detection challenging	- Vulnerable to process noise - Ultra-small Trojan detection challenging

Table 2.1 Comparison of Logic Testing and Side-Channel-Analysis-Based Trojan Detection.

The purpose of the logical test is to detect the input vector that can activate the Trojan horse and observe the behavior of the system when this input vector is given. This method is not affected by the circuit parameter variations. Because giving the correct input vector is a sufficient condition to activate the Trojan horse. As a result, its other advantage is that it gives good results in detecting very small Trojans. Another reason for this is that the input vector required to activate small Trojans will be easier to find. As a disadvantage, creating input vectors is a laborious task and the length of the input vector that needs to be given to activate large Trojans will also be large. Combinations of a large input vector will also be a lot. Therefore, it will be almost impossible to find the correct input vector to activate the Trojan horse in these

combinations. Moreover, it fails especially for sequential Trojans where a sequence of correct vectors is required.

The second method is side channel analysis, which we also prefer in this thesis. These methods aim to detect the Trojan horse by making use of the different physical properties of the circuit (power, delay, electromagnetic, etc.). In our thesis, we aimed to detect the Trojan horse based on the power consumed by the circuit. Side channel analysis methods test whether the data previously obtained theoretically or from a golden model are consistent with the data obtained during the real-world testing phase after the fabrication. Physically adding a Trojan horse to the circuit means adding extra transistors to the circuit, in other words, adding an extra load. Even if the Trojan horse is not active, a few of these transistors will be under electrical charge, and this will cause deviations in the side channel signal values than a circuit without any Trojan horse would give us. Here, side channel analysis aims to detect Trojan horses by considering these deviations. The advantage of this method when compared with logical tests is that it does not require activation of the Trojan to detect it. Therefore, it is easier to generate the required test vectors. The biggest disadvantages of these methods are that they are seriously affected by variations. In other words, a circuit with a Trojan horse gives results as if there is no Trojan in the circuit, or on the contrary. Therefore, it is important to accurately model the variations while using these methods. Another disadvantage is that due to variations very small Trojans are almost impossible to detect. Hence, it is not a good method for detecting very small Trojans, as it will be almost impossible to suppress variations for small Trojans.

2.5.1. Logic Testing

To detect the presence of a Trojan horse using logic testing, it is important not only to fulfil the trigger condition, but also to propagate and observe the effect of such an event to an exit node. Due to the extremely wide range of possible Trojans, as mentioned earlier, it is not practical to enumerate all possible Trojan horse instances to create deterministic test patterns or to calculate test coverage. A statistical approach to generate test vectors may be more computationally traceable, rather than a precise approach. This motivated efforts to develop a statistical approach to Trojan horse detection.

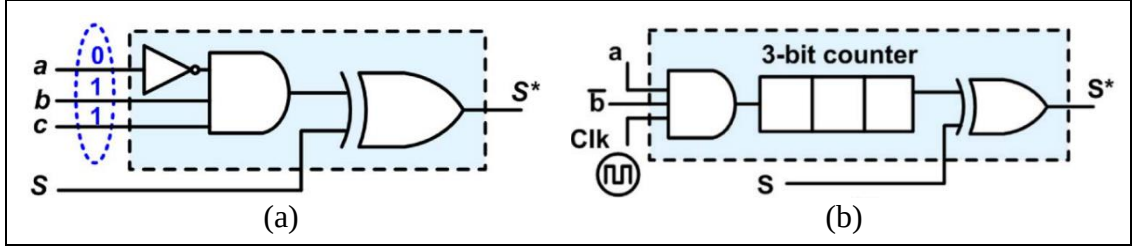


Figure 2.8: Logic testing approach for hardware Trojan detection based on multiple excitations of rare conditions (a) for combinational Trojan (b) for sequential Trojan.

Figure 2.8 [14] illustrates the concept with two examples, a combinational and a sequential Trojan horse. The combinational Trojan horse in Figure 2.8 (a) is activated when $a = 0$, $b = 1$ and $c = 1$, while the sequential Trojan horse in Figure 2.8 (b) requires occurrences of $a = 1$ and $b = 0$ in several clock cycles. Therefore, if we can create a set of test vectors that trigger these conditions individually N times on those nodes where N is large enough (e.g. > 100), then a Trojan horse with the trigger condition created by these nodes together will most likely be activated by the application of this test set. Although both the activating test vector and the number of times required for activation is unknown a priori, which makes logical testing unfeasible especially in sequential HTH case.

2.5.2. Side-Channel Analysis

The use of side channel analysis during post-silicon testing to overcome the limitations of logic testing approaches has been extensively explored [14], [16], [17]. Malicious inclusion during design or manufacturing will have an impact on the power consumed by the circuit. In most cases, it is likely that certain circuit paths also affect the delay. The effect on power is expected to be due to the addition/modification of the circuit elements used in the realization of the Trojan horse. A Trojan can cause an increase on both static (i.e. current consumed at idle state of a circuit) and transient (i.e. switching) current profile of the circuit under test.

Most Trojans constantly monitor activation conditions. The monitoring action consumes power even if its effect on consumption is minimal; actually, this would be the target of an adversary [18]. A delay effect can be caused by adding an extra level of logic to a path (as in Figure 2.8) or an increase in capacitive load on a path. Delay based side channel analysis is superior in the sense that one can test an individual

path, which will reveal any HTH existence. On the other hand, delay based analysis is difficult as the test pattern activating a particular path is difficult to obtain and also the precise delay measurements are challenging.

The influence of an unknown Trojan horse on physical parameters can be observed and compared between gold and Trojan infected circuits to determine its presence. The efficiency of Trojan horse detection by side channel analysis depends largely on two main parameters: 1) signal-to-noise ratio (SNR) and 2) Trojan-to-circuit ratio (TCR). The signal here is the influence of a Trojan horse on a side channel parameter and noise are the effect that avoids its recognition, i.e. measurement noise. Noise can also arise due to the process and environmental changes. A Trojan's influence on a side channel parameter such as current or delay can be easily masked by noise. Therefore, simple comparison techniques can lead to a lot of misperceptions. This has led to a wide variety of post-processing techniques to statistically isolate the Trojan effect from noise. On the other hand, TCR shows the gap between the original and the Trojan affected parameter. In an ideal zero noise scenario, detection sensitivity depends on the TCR. Therefore, increasing the Trojan horse effect compared to the original value helps increase confidence in identifying a Trojan horse instance.

3. POWER BASED HT DETECTION USING SEPARATION TO GRIDS

3.1. Motivation

The side channel analysis based methods has the superiority over logic based methods as they do not need the activation of HTH, which is a complex problem. However, they suffer from unavoidable parameter variations that occur during the manufacturing phase. This brings the fact that these methods have difficulty in detecting the small hardware Trojans.

Especially power based HTH detection has another problem when compared with delay based methods. That is because power of the circuit is supplied from single or a couple of sources. Therefore, all current drawn by the circuit makes it more difficult to detect any Trojan effect on the current. Especially dynamic current depends on the current activity of the circuit.

In order to detect the performance and efficiency of side channel analysis based HTH detection methods, one most important aspect is modeling parameter variations accurately. Otherwise, all reported results can be misleading. Because the main weakness of side channel analysis based methods, which is the parameter variations, must be modelled accurately to see whether the effect of HTH is bigger than the variations or not. This also determines the minimum Trojan size that can be detected by a particular detection method.

Another issue is the test vectors. Sensitivity of HTH detection depends on the input test patterns too. For instance, if the dynamic current is the side channel signal to detect the Trojan existence; then depending on the input, the HTH in the circuit can have contribution in the dynamic current or not. A test vector resulting in no propagation at HTH will not reveal Trojan existence.

As a result, in this thesis, we first apply an accurate parameter variation model, which takes into consideration both the inter and the intra die variations [19] with spatial correlations. All variation amounts are fixed according to the ITRS roadmap for semiconductors report [4]. Section 3.2 explains our variation model in detail.

After modeling variations accurately, our first aim in the thesis is to reveal the actual effect of realistic variations on HTH detection using dynamic current based side channel analysis. At this stage, the questions we seek answer for are: What

happens if we just measure the dynamic current on circuits with HTH and without HTH and can we see any difference?

Our second aim is to demonstrate the sensitivity of dynamic current based side channel analysis method for HTH detection to the circuit inputs and to the size of the circuit. The questions we seek answer are: Does the detection depend on the given input and by how much? Does the circuit vs. Trojan size affect detection and how?

Our third and most important aim is to divide the circuit into grids and apply a separate power supply for each grid. By this way we can measure the dynamic current of each grid individually. The question we seek answer for is: At what level of separation can we detect the Trojan by measuring the dynamic current at each grid?

Lastly, we want to compare HTH detection based on dynamic current measurements and based on the consumed energy. Thus, the question is: Is it better to detect an HTH observing the instantaneous power (current) or observing consumed energy?

The results of the thesis reveal the answers to above questions and provide a broad view of the problem and HTH detection approaches using dynamic current based side channel analysis.

3.2. Modeling Parameter Variations

No matter how properly designed the integrated circuits are, it is not possible to manufacture them perfectly in the fabrication phase. These undesired manufacturing process differences may cause different results when using SCA-based hardware Trojan horse detection method, even if the integrated circuit does not contain any Trojan horses. These differences will act as if there is a Trojan horse in the integrated circuit. Therefore, it is very difficult to distinguish the Trojan horse's effect on power consumption from the effect of variations. For this reason, it is very important to model the variations correctly. Otherwise, the hardware Trojan detection results will be very misleading and optimistic. In order to model process variations well, it is necessary to model the characteristics of the variations beforehand. In this context, we aimed to create a variation model based on the parameters of the integrated circuit to be designed. These parameter variations can occur both inside the same chip (intra-die) and between different chips (inter-die). The amount of variation for

the corresponding technology node is also important. Underestimating this amount could lead to misleading optimistic hardware Trojan horse detection results. Another feature of variations is spatial correlations, that is, correlations that occur due to the position of the transistors used in integrated circuit design, for example. Spatial correlations arise due to the nature of the defects that may occur during the manufacturing process of the integrated circuit and cause transistors close to each other to have similar random parameter values. Creating a model with lower variation values or any model that does not take into account the variations for the pattern or ignores the spatial correlations could lead to undesirable results in Trojan horse detection. In this thesis, we use a 4-level version of the recommended quad tree model [20], which can model the spatial correlations from which the actual variation quantities are taken from the ITRS report, as well as variations for both the inter and intra die that take into account the size of the circuit.

3.3. The Trojan Horse Used in Experiments

As the Trojan horse, as shown in the Figure 3.1, a two-input XOR gate has been preferred to form the payload part, and a two-input AND gate has been preferred to form the trigger part. When we look at the Figure 3.1, if the Trigger pin is zero, the C input will be transferred to the output ($C_{modified}$) as it is. But if the trigger pin is set to 1, the output ($C_{modified}$) will always be zero regardless of the input C . In other words, when the trigger pin is 1 and the C input is 1, the Trojan horse will be activated.

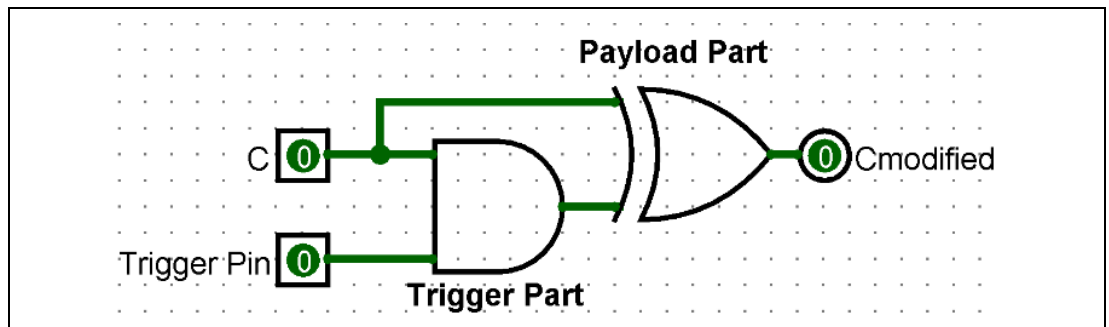


Figure 3.1: Added Hardware Trojan Horse.

Table 4.1: Truth Table.

Trigger Pin	C	$C_{modified}$
0	0	0
0	1	1
1	0	0
1	1	1

The reason for using this HTH is its minimal size. This Trojan is the smallest possible Trojan in the literature, which is normally passive but can be activated when the proper input to the trigger part is applied. Actually, this type of Trojan is not very hard to detect with logic based methods, but it is the most challenging Trojan for side channel analysis methods based HTH detection. Because of its very small size its effect on the delay or dynamic current or leakage is very minimal, which makes it harder to be detected by side channel analysis. However, it is still a hardware Trojan and an adversary can easily insert it to a circuit. Therefore, we investigate the possibility of the detection of this Trojan when the variations are modelled accurately. But before that, the next section will demonstrate the effect of variations.

3.4. Effect of Variations on Conventional Power Based SCA

This section focuses on how the variations make Trojan detection difficult and the inadequacy of conventional power tests. Conventional tests mean that the entire circuit is supply from only one or several power sources. Before talking about the results, detecting the Trojan horse that causes the power consumption depends strongly on the input vector to be given. This will be discussed experimentally in the next section. We observe the best result in our tests on the 64-bit adder when we apply the 48th input vector. However, when traditional test methods are used, giving the 48th test vector is not enough to detect the Trojan horse. Under the realistic variation model and when we apply the traditional power test in which the circuit is supplying with a single power source, the power consumption that occurs when the 48th input vector is applied is shown by (a) in Figure 3.2. For the 48th input vector, the total energy consumed during transition is shown by (b) in Figure 3.2. For this figure, 45nm Nangate open cell library [21] is utilized and Spice transistor level

circuit simulations [22] are used for precise results. Please refer to Experimental Setup Section (Section 4.1) for more details of the experimental setup.

In the plots we use in our thesis, always green circles indicate the chips without any Trojan Horse, and those with a red asterisk indicate samples (chips) with a Trojan horse. In instantaneous power plots, each asterisk or circle sign belongs to the instantaneous power for a given instance and belongs to a specific time instance. On the other hand, each asterisk or circle mark on the energy plots represents the total energy consumed for a particular sample during one transition. Therefore, the x-axis in the instantaneous power plots is the time that represents the entire transition period for an input, while the x-axis of the energy plots shows the sample index to which the asterisk or circle sign belongs. In Figure 3.2 (a), the instantaneous power for each sample is plotted on the transition time period of the 48th input vector between 48ns and 48.3 ns. This is because each input is given to the circuit with 1 ns interval and the dynamic current drain from the supply is over in 0.3ns. As the instantaneous power is computed by equation (2.1) and the supply voltage is a constant, the power plots similar to Figure 3.2 (a) show the dynamic current multiplied with a constant supply voltage. The plot in Figure 3.2 (b) represents the energy consumed for each sample during the same transition period. The x-axis in (b) indicates the sample array from 1 to 500.

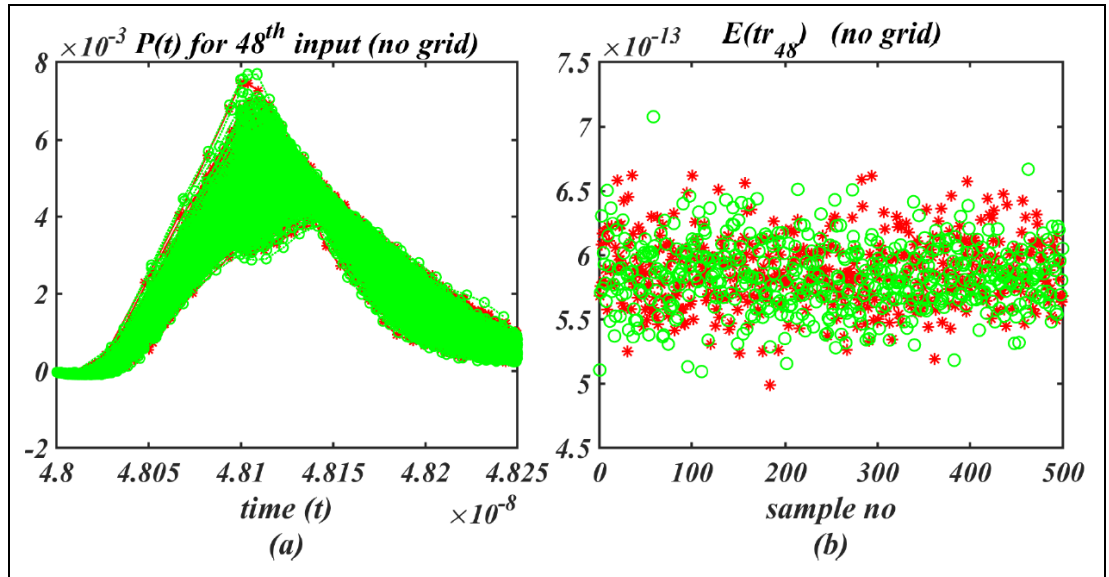


Figure 3.2: Conventional power-based HT detection with single power supply for 48th input (64-bit adder): (a) Power (in Watt) vs. time (b) Energy (in Joule) vs. sample no.

When we look at the results, it becomes clear that the samples with and without Trojan horses do not differentiate from each other when both instantaneous power consumption and energy consumed are considered. This makes it clear that even if we give the ‘correct’ input to the circuit, it is impossible to detect any Trojan horse when we supply the circuit from a single source. The ‘correctness’ of the input is elaborated in Section 4.3. This shows the crucial effect of variations on HTH detection as well as the importance of accurate variation model with realistic variation amounts. This also shows the inadequacy of using single power supply to detect HTH existence, for which we propose separation to grids as explained in the next section.

3.5. Separation to Grids

One of the main drawbacks of power-based Trojan horse detection is that all of the circuit's supply is provided from a single or a couple of pins. In addition, if we consider the effect of variations, it becomes very difficult to separate the effect of the Trojan horse from the whole circuit. In order to make this effect more pronounced, an approach as suggested in [23] but not quantified is to divide the circuit into grids and to supply each grid with a different power source to collect regional power consumption. This thesis aims to quantify the amount of required separation under realistic circuit parameter variations. For that purpose, we divide the circuit layout into $n \times n$ grids at the design phase. This brings the problem of increased number of supply pins, for which we also propose a solution in next section. By separating power supplies, we can measure the power consumption of each grid separately after manufacturing. The total power consumption can be expressed as the sum of the powers of each grid drawn from the source as formulated in (3.1).

$$P(t) = \sum_{x,y} P(x, y, t) \quad (3.1)$$

where x represents the x-axis and y represents the y-axis of the corresponding grid in the circuit. Both can take a value between 0 and $n - 1$.

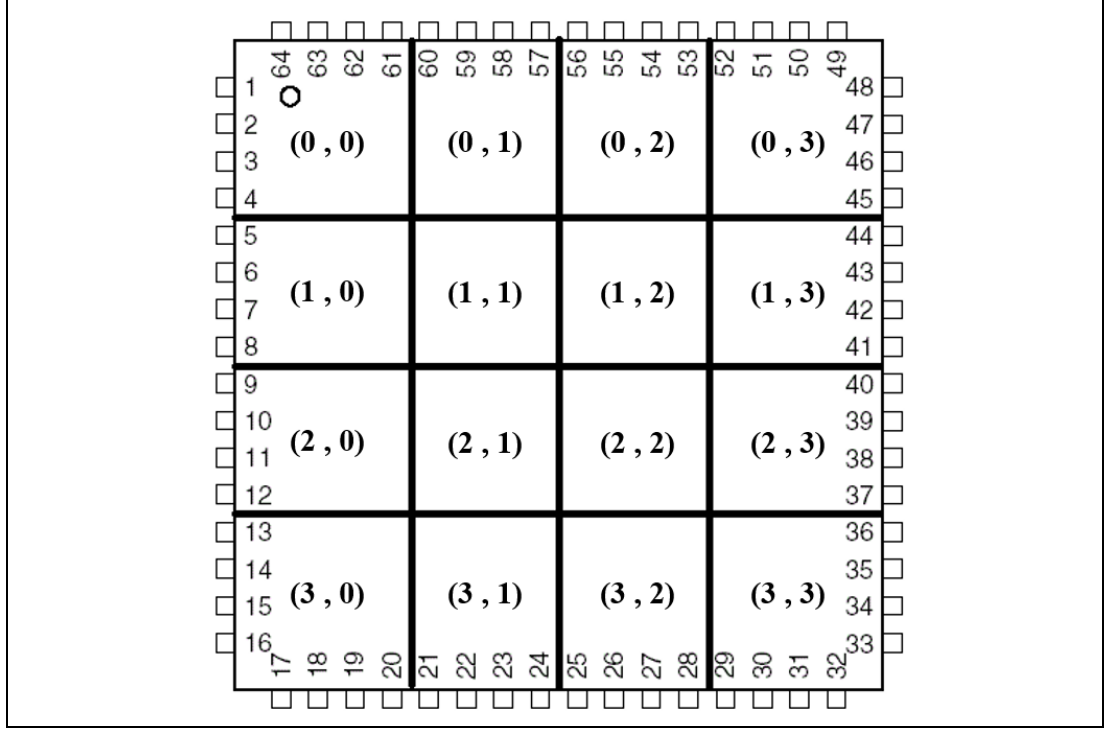


Figure 3. 3: A symbolic chip divided into grids.

As the number of n increases here, that is, as the number of grids increases, the effect of the Trojan horse becomes more evident and the detection of the Trojan horse becomes easier. For example, if we take the number n as 8, we will need a total of (8×8) 64 supply inputs, which is not feasible but can be resolved as explained in next section.

For a grid (x, y) and a sample (chip), power and energy per transition are defined as shown in (3) and (4). t in these equations is time and tr_i is the time period required for dynamic power consumption to settle for the transition of input i .

$$P(x, y, t) = i(x, y, t) \times V_{supp}(x, y) \quad (3.2)$$

$$E(x, y, tr_i) = \int_{tr_i} P(x, y, t) dt \quad (3.3)$$

The quantities in (3.2) and (3.3) are used to detect HTH in this thesis. The instantaneous power shown by (3.1) is basically the multiplication of a constant supply voltage with the dynamic current. Any HTH may consume additional dynamic current and as a result changes the instantaneous power (P). On the other

hand, the effect of the HTH can be revealed by looking at the energy (E) consumed in one total transition. Here the transition represents the time interval starting from the time instance where the input transition occurs until the time all circuit outputs settle to their final values. In this thesis we also compare the HTH detection performance when P or E is used for detection. For instance, in Figure 3.2 (a) the instantaneous power in (3.2) is plotted and in Figure 3.2 (b) total energy for the particular transition shown by (3.2) is plotted for 1000 chips where half of them has an inserted HTH.

In order to see the effect of grid separation we divide the circuits into (2×2) , (4×4) and (8×8) grids and supply the power to each grid individually to measure the isolated dynamic current drain of each grid. Then, we insert the Trojan to some samples and plot the instantaneous power and energy plots to demonstrate whether HTH detection is possible or not. Of course, the results are also compared with the case when there is no separation, shown in Figure 3.2. This figure shows that the detection of HTH is not possible when there is no separation to grids whether the instantaneous power or the energy is used.

In this thesis, we also explore whether the investigation of the instantaneous power or energy of other grids than the one having the inserted HTH can reveal the existence of Trojan as explained in Results Section.

3.6. I/O Issue of Separated Grids Approach

As we mentioned before, the higher the number of grids, the easier it will be to discriminate the effect of the Trojan horse from the effect of variations. An integrated circuit can be thought of a chip that contains multiple modules that require multiple pins for communication with outside world. In addition to the use of these many pins for communication with the circuit, the separation to grids method proposed in the previous section requires additional pins to meet the supply requirement for each grid and this will cause the need for too many physical pins in integrated circuit design.

In order to avoid that, the circuit in Figure 3.4 is proposed in this thesis to reduce the number of additional pins. This decoder-based circuit shown in Figure 3.4 can decrease the required number of additional pins exponentially.

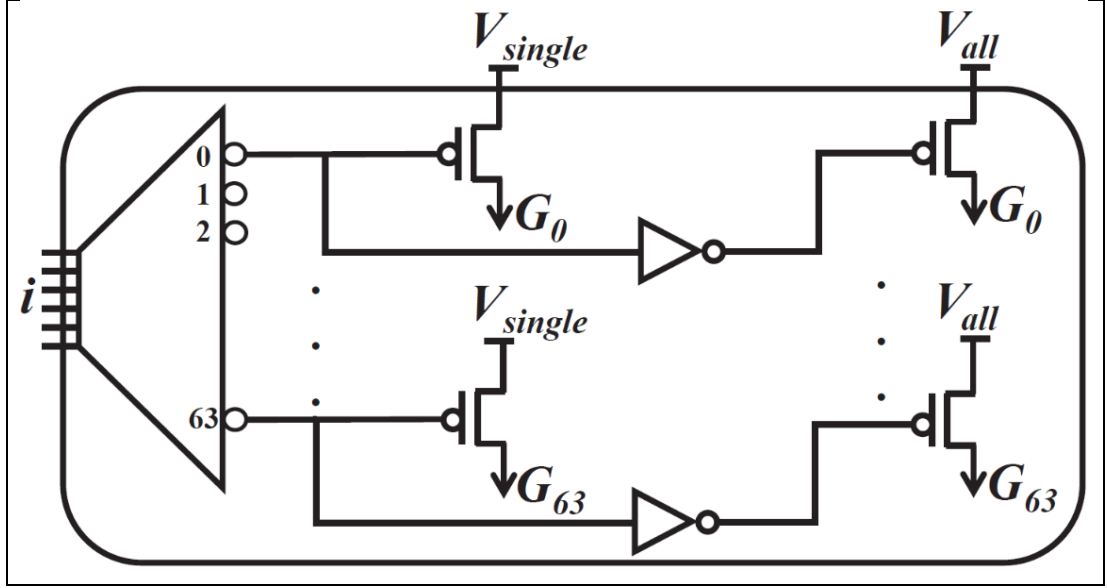


Figure 3.4: Circuit to Provide Individual Power Supply.

This circuit is made up of a 6-to-64 line inverting decoder, inverters and PMOS transistors. It illustrates the case for 64 (8×8) grids. Note that each PMOS turns on and links supply to a grid only if the gate of the PMOS is connected to low logic.

The single voltage supply (V_{single}) is used to provide voltage only for one selected grid. The grid selection is done by input i . For instance, if input i is 24 then only grid 24 (G_{24}) is supplied by V_{single} because the PMOS connected to 24th decoder output becomes logic-0, which activates the PMOS. All remaining grids are supplied power by V_{all} . Therefore, by this setup, the power consumption of each grid can be measured individually by utilizing only $\log_2(N) + 1(V_{single})$ additional pins where N is the number of total grids in the circuit. For instance, for 16×16 grid case, instead of 256 supply pins only 9 additional pins are enough. In terms of additional hardware components, the additional cost is one N -input decoder, $2N$ PMOS transistors and N inverters, each made up of two transistors. This additional area cost can be negligible for modern circuits with millions of gates.

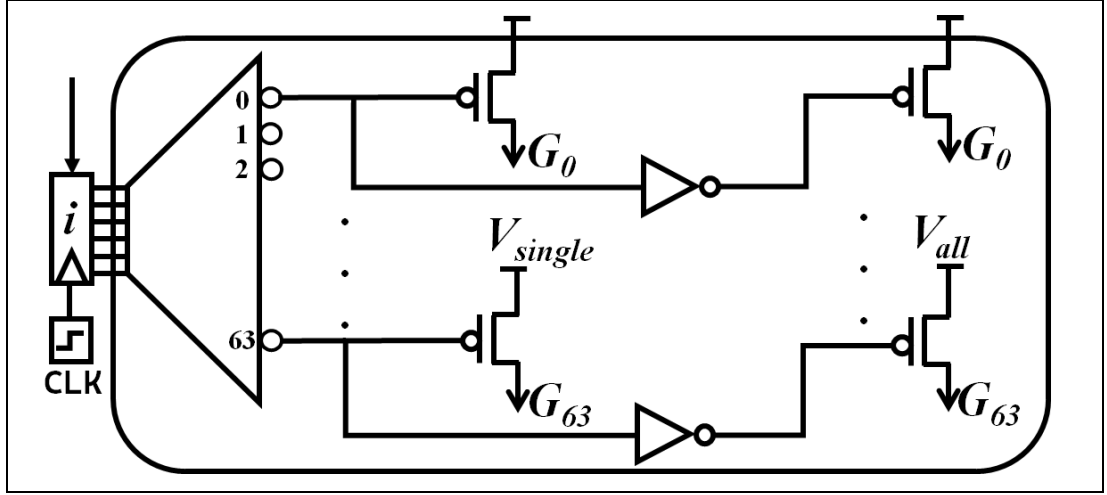


Figure 3.5: Further decrease in additional pins for supply separation.

The number of additional pins to implement the grid separation method can be decreased further to 2, independent of the number of grids in the circuit. For that purpose, a shift register can be utilized to hold i inside. The shift register at the left of Figure 3.5 is inserted for that purpose. The arrow at the top of the shift register shows the one bit input of the register. In such a case, for instance for 64 grid case in the figure, setting i would require 6 clock cycles to input 6 bits for i . But additional pins would be 2 instead of 7 (including V_{single}) and this holds whatever the number of grids is. Therefore, the choice of shift register is more meaningful when plenty of grids are required in extra-large circuits or when there is no room for additional pins in the chip packaging. Hence even a circuit that requires multiple thousands of grids can be tested for the existence of HTH and with a cost of additional 2 pins solely. Yet, this would increase the experiment delay due to the fact that i is entered to shift register serially instead of parallel shown in Figure 3.4. As this scenario is applied only at the test phase and not for the normal operation, this burden is negligible.

Another advantage of this suggestion is that it is used actively only when HTH detection tests are desired. For the normal operation of the circuit when no tests are performed the register holding the i value does not need to be modified and the circuit can work flawless without any extra delay component.

4. RESULTS

4.1. Experimental Setup

As we explained in Section 3.2, the biggest obstacle to detecting Trojan horses with the SCA method is process variations. Therefore, these variations need to be modelled. In the approach we made, we aimed to model the variations on the two most effective circuit parameters, transistor gate length(L_{eff}) and threshold voltage(V_{th}) [24]. In this context, we considered these values as random values as in manufacturing process. The quantities of variation are defined according to the ITRS report [4] stating that $3\sigma = \mu$ transistor gate length and threshold voltage ratio is 12% and 35% respectively and for 45nm process node.

We designed two different test circuits to test our approach. One of them is a 16-bit adder with 144 two-input NAND gates and a 64-bit adder with 576 two-input NAND gates. The Trojan horse we have created is connected to the carryout pin of bit 15 in the 16-bit adder circuit, and the 64-bit adder to the carry out pin of the bit 46.

An open cell library with 45 nm technology is used [21] for the NAND gates used in our circuits. 1000 different samples (copies of a chip) are created using the Monte Carlo variation model. Of these 1000 examples, 500 of them have inserted Trojan horse, and the remaining 500 samples are designed without Trojan horses. Each sample has its own Spice deck file used for Spice simulations. Each sample circuit is tested with randomly generated input vectors. The 16-bit adder circuit and the 64-bit adder circuit are tested with 128 randomly generated 32-bit test vector and 64 randomly generated 128-bit test vector respectively with a 1ns time interval between two consecutive vectors. Precise Spice power simulations with high precision (as small as **1e-6 reltol** value) are performed on both test circuits and for each of 1000 samples and each input.

In Section 3.4 the effect of realistic process variations on HT detection is demonstrated. Section 4.3 explores the separation of power sources and its impact on Trojan detection. The dependence of HT detection on the given input, measured grid and circuit size are described in Sections 4.4, 4.5 and 4.6 respectively.

4.2. Results for Separated Power Supplies

In previous chapters we consider our 64-bit adder circuit as a single piece and fed it with a single source. In this section, we divide our circuit into $n \times n$ grids where n is 2, 4 or 8. Likewise, we apply our 48^{th} test vector separately for each value of n . This test vector is correct in the sense that it can activate the edge where the HTH is inserted. The test circuit is 64-bit adder. All plots in Figure 4.1 correspond to the measurements from the grid where HTH is inserted.

The power and energy plots we obtain as a result of this process are formed as in Figure 4.1. Power plots (a1), (a2), (a3) and energy plots (b1), (b2), (b3) we have seen in Figure 4.1 are obtained for 2×2 , 4×4 and 8×8 grids, respectively. The x-axis of instantaneous power plots corresponds to time whereas for energy plots x-axis is the index showing the number of the sample (chip).

The Trojan horse is placed on grids (2, 2), (3, 4) and (6, 7) for 2×2 , 4×4 and 8×8 separation respectively and the measurements are taken from the corresponding grid. The ones marked with the red asterisk in the Figure 4.1 belong to the Trojan horse inserted circuits, whereas the ones marked with the green circle represent the circuits that do not contain the Trojan horse.

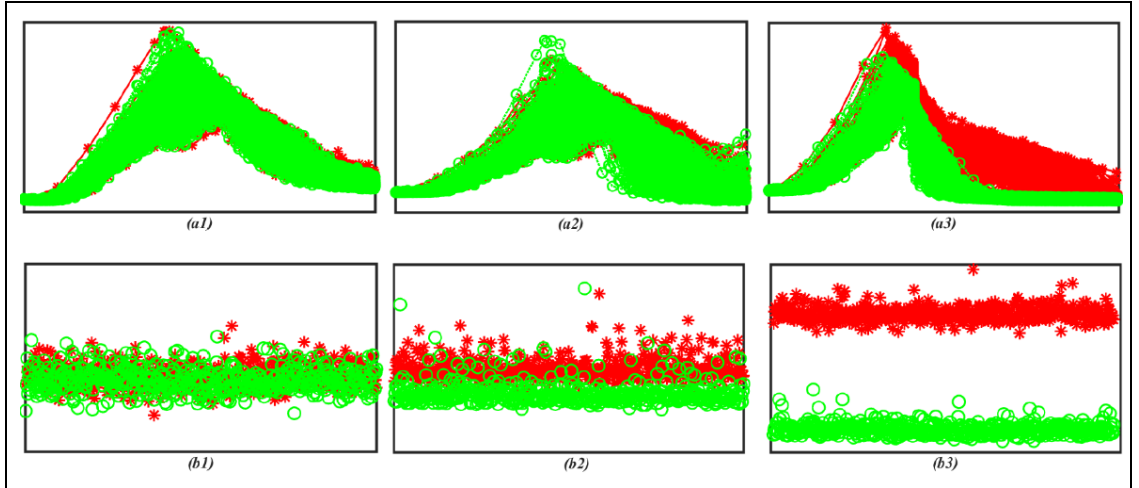


Figure 4. 1: Power vs. time and energy vs. sample no plots for 64-bit adder (same x, y axis and scale as in Figure 4.2): $P(t)$ for (a1) 2×2 grid (a2) 4×4 grid (a3) 8×8 grid and $E(tr_{48})$ for (b1) 2×2 grid (b2) 4×4 grid (b3) 8×8 grid.

When Figure 4.1 is examined, the following results appear;

- The energy drawn ((b1), (b2) and (b3)) gives better results than the instantaneous power consumption ((a1), (a2) and (a3)) in discriminating the Trojan horse effect from the variations. Even (b3) shows a clear discrimination between HTH inserted and HTH free chips.
- When we divide our circuit into 2×2 ((a1) and (b1)) or 4×4 ((a2) and (b2)) grids, it is not possible to easily differentiate the circuit with Trojan horse from the secure ones. Although there is an improvement going from 2×2 to 4×4 case. This means for a larger Trojan or for a smaller circuit 4×4 can even be enough for detection.
- When we divide the circuit into 8×8 grids, the presence of the Trojan horse becomes clear, especially in the energy plot (b3). The good point is that the Golden model is not needed to reveal the existence of the Trojan horse. As seen in the plot, the red asterisk-shaped signs representing the Trojan horse circuits are distinctly different from the green ring-shaped signs that do not contain Trojan horses.

4.3. Dependence on the Input

In our tests so far, we have chosen the **48th** Input vector, which gives the best discrimination results for all 64-bit adder samples simulated with Spice. This input vector can activate the node where the HTH is connected. In order to show the effect of a different input vector on Trojan horse detection, the input vector number 14 has been applied to our circuit as a sample case and the results shown in Figure 4.2 are obtained.

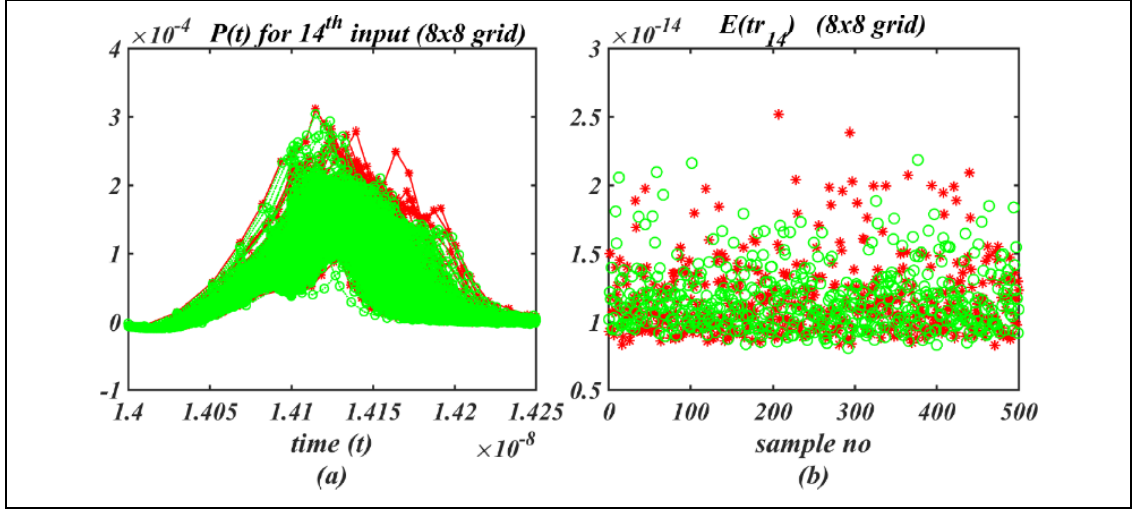


Figure 4. 2: Power vs. time and energy vs. sample plots for 14^{th} input for grid (6, 7) (64-bit adder): (a) $P(t)$ (in Watt), 8×8 grid (b) $E(tr_{14})$ (in Joule), 8×8 grid.

Even when our circuit is divided into an 8×8 grid, the importance of applying the correct input vector is revealed here. We can better understand this by comparing the energy plot in Figure 4.2 with the plot in Figure 4.1 (b3). A similar situation occurs for most of the input vectors that cannot activate the HTH location. This shows the necessity of determining the correct inputs to test the circuit for HTH existence. This is another critical point in HTH detection based on side channel analysis using power.

4.4. Dependence on the Grids

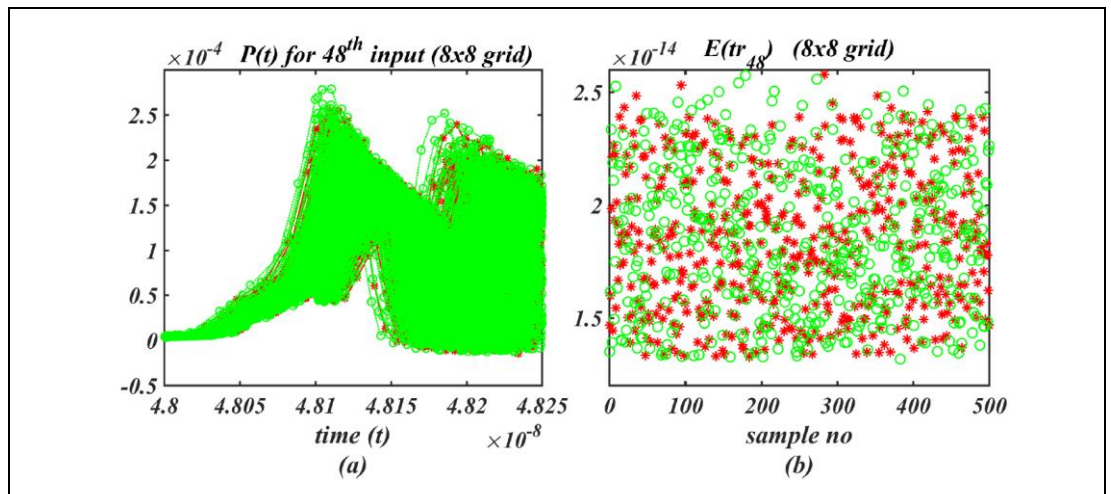


Figure 4. 3: Power vs. time and energy vs. sample no plots for 48^{th} input for grid (5, 6) (64-bit adder): (a) $P(t)$ (in Watt), 8×8 grid (b) $E(tr_{48})$ (in Joule), 8×8 grid.

In order to observe the effects of the Trojan horse on power and energy when the circuit is divided into grids, it is important to take measurements over the grid where the Trojan horse comes across. In our test circuit, we placed the Trojan horse on grid (6, 7) in 8×8 grid. Therefore, the values in Figure 4.1 (a3) and (b3) belong to the measurements from that grid. In Figure 4.3, the power and energy plot for the grid numbered (5, 6) are shown as (a) and (b), respectively. Although this grid we chose is a grid next to grid (6, 7), which is the grid where the Trojan horse is placed, and although we have applied the 48th Input vector, which gives the best results, the effect of the Trojan horse is clearly not visible. Therefore, it is important to take the power and energy measurements from the correct grid, i.e. the grid containing the HTH in order to detect the existence of the Trojan horse. As it is unknown a priori, all grids must be tested individually for any HTH existence, which complicates the detection process.

4.5. Dependence on the Circuit Size

Finally, it is important to show the relationship between the detection performance and circuit size. For this purpose, instead of the 64-bit adder, a 16-bit adder is used. 1000 samples were formed from this 16-bit adder circuit similarly. All the results we have achieved so far are for the 64-bit adder circuit.

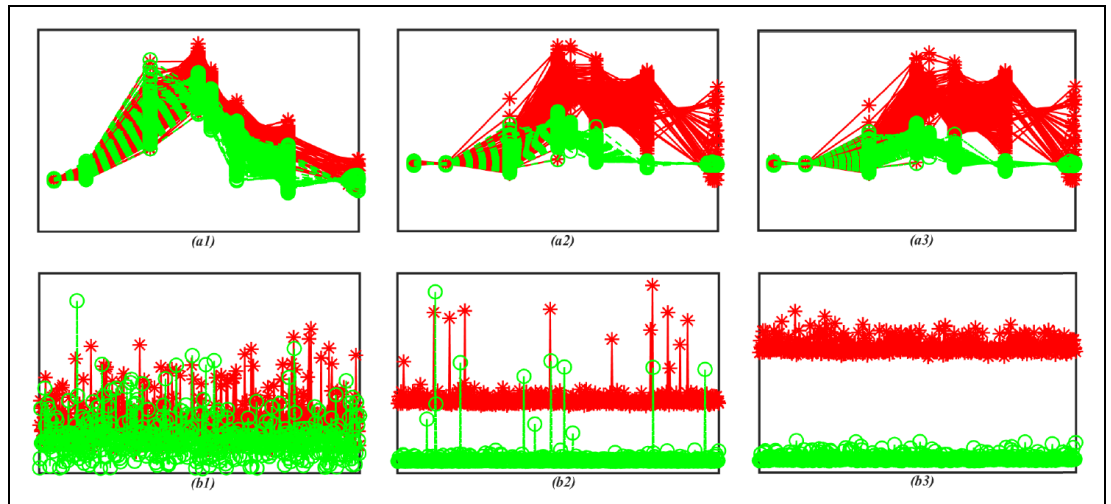


Figure 4. 4: Power vs. time and energy vs. sample no plots for 16-bit adder (same x, y axis and scale as in Figure 4.2): $P(t)$ for (a1) 2×2 grid (a2) 4×4 grid (a3) 8×8 grid and $E(tr_{82})$ for (b1) 2×2 grid (b2) 4×4 grid (b3) 8×8 grid.

The plot in Figure 4.4 shows the same results as Figure 4.1, besides belonging to a 16-bit adder. In our tests on 16 bit adder, the 82nd random input vector gives the best results. Therefore, the results in Figure 4.4 are obtained after applying this input vector. When we look at both power and energy plots, it clearly shows the effect of the Trojan horse even when the 16-bit adder circuit is divided into a 4×4 grid. The main reason for this is that fewer transistors are needed to build the 16-bit adder circuit and thus the relative effect of the HTH is revealed before further separation of power supplies.

The main flaw of side channel analysis based HTH detection is known to be the difficulty to detect relatively small Trojans. The results support that expression. Therefore, the amount of separation required for HTH detection depends on the circuit size. Considering the complex modern circuits, the number of grids can be huge, which increases the importance of the method proposed in Section 3.6 to decrease the number of additional pins to two whatever the number of grids is.

5. CONCLUSION

First of all, any side channel analysis based hardware Trojan horse detection method must report its correctness by modelling the variations accurately. This is the starting point of this thesis and an accurate variation model considering both the inter die and intra die variations with spatial correlations are used in all experiments of this thesis.

Secondly and again for precision and accuracy all computations and plots in the is correspond to highly precise and accurate transistor level Spice simulations. These make the results in this thesis closer to the real life tests.

From Figure 3.2, it is clear that, under realistic variations, the effect of small Trojans is not evident when we supply the circuit from only one source. At the same time, it is shown in Figure 4.1 that the effect of small Trojans is not clearly seen if our circuit is not divided enough number of grids. This figure also shows the detection performance can be higher if total consumed energy is used for detection instead of dynamic current or instantaneous power. This is because the amount of variations due to the HTH is accumulated in energy. We touched on the importance of the input vector given in Figure 4.2. If the given input vector does not activate the Trojan horse, the power consumption tests performed do not give successful results. Likewise, even if our circuit is divided into grids, as seen in Figure 4.3, the power consumption results obtained from a grid that does not contain Trojan horse cannot reveal the existence of a Trojan horse.

According to the data we have obtained as a result of this thesis, we need to divide our circuit into a sufficient number of grids, apply the correct input vector to the circuit, and measure the energy per transition for the grid in order to reveal the effect of the Trojan horse in the circuit. This shows the difficulty of Trojan horse detection under realistic variations.

The largest of the circuits we use in our tests here is the 64-bit adder. Considering the modern complex circuits used today, it may be necessary to divide these circuits into more grids and require different power sources in order to speak of the existence of the Trojan horse. When Figure 4.1 is examined, it is observed that the existence of the Trojan horse emerged when we divide the 64-bit adder circuit with only 2304 transistors into 64 different grids. Accordingly, if we make a rough

calculation, we will need to divide this circuit into 4096 grids in order to talk about the existence of a Trojan horse in a circuit with approximately 150,000 transistors.

But the main problem here is that, as we mentioned earlier, a physical pin must be attached to the grid in order to supply individual power to each grid. Therefore, 4095 additional physical pins are needed for a circuit divided into 4096 grids. This is almost impossible. Therefore, in this thesis, a pin number reduction method is proposed so that the number of additional pins can be decreased to 2 whatever the number of grids is. Using the proposed circuit in Figure 3.3, it is possible to reduce the number of pins exponentially and with an additional shift register it can even be reduced to 2 as in Figure 3.4.

REFERENCES

- [1] Tehranipoor M., Koushanfar F., (2010), "A survey of hardware Trojan taxonomy and detection", IEEE Design Test Comput., 27 (1), 10–25.
- [2] Potkonjak M., (2009), "Hardware Trojan Horse Detection Using Gate-Level Characterization", Proc. 46th Design Automation Conf. (DAC 09), 688-693, San Francisco California, 26-31 July.
- [3] Bhunia S., Abramovici M., Agrawal D., Bradley P., Hsiao M. S., Plusquellic J., (2013), "Protection against hardware trojan attacks: Towards a comprehensive solution", IEEE Design & Test, 30 (3), 6-17.
- [4] Web 1, (2010), <http://www.itrs2.net/>, (Access Date: 10/07/2010).
- [5] Web 2, (2012), <https://doi.org/10.1155/2012/505983>, (Access Date: 22/04/2012).
- [6] Martins J., Reis R., Monteiro J., (2000), "Capacitance and Power Modeling at Logic-Level", Power, 1(2), 2.
- [7] Web 3, (2021), <https://www.vlsisystemdesign.com/on-chip-variation-ocv-part-1/> (Access Date: 03/01/2021).
- [8] Esirci F. N., Bayrakci A. A., (2017), "Hardware trojan detection based on correlated path delays in defiance of variations with spatial correlations", Automation Test in Europe Conference Exhibition (DATE), 163–168, 27-31 March.
- [9] Jha S., Jha S. K., (2008), "Randomization Based Probabilistic Approach to Detect Trojan Circuits", 11th IEEE High Assurance Systems Engineering Symposium, 117-124, Nanjing China, 3-5 December.
- [10] Bloom G., Narahari B., Simha R., (2009), "OS support for detecting Trojan circuit attacks", IEEE International Workshop on Hardware-Oriented Security and Trust, 100-103 San Francisco, CA, USA, 27 July.
- [11] Bhunia S., Hsiao M. S., Banga M., Narasimhan S., (2014), "Hardware Trojan attacks: Threat analysis and countermeasures", Proc. IEEE, 102 (8), 1229-1247.
- [12] Wang X., Tehranipoor M., Plusquellic J., (2008), "Detecting malicious inclusions in secure hardware: Challenges and solutions", IEEE International Workshop on Hardware-Oriented Security and Trust, 15-19, Anaheim, CA, USA, 9 June.
- [13] Narasimhan S., (2013), "Hardware Trojan Detection by Multiple-Parameter Side-Channel Analysis", in IEEE Transactions on Computers, 62 (11), 2183-2195.

- [14] Chakraborty R.S., Wolff F., Paul S., Papachristou C., Bhunia S., (2009), "MERO: A statistical approach for hardware Trojan detection", in Proc. Workshop Cryptographic Hardware and Embedded Systems, 396–410, Lausanne, Switzerland, 6–9 September.
- [15] McIntyre D., Wolff F., Papachristou C., Bhunia S., (2009), "Dynamic evaluation of hardware trust", IEEE International Workshop on Hardware-Oriented Security and Trust, 108–111, San Francisco, CA, USA, 27 July.
- [16] Aarestad J., Acharyya D., Rad R., Plusquellic J., (2010), "Detecting Trojans through leakage current analysis using multiple supply pad IDDQs", IEEE Trans. Inf. Forensics Security, 5 (4), 893–904
- [17] Du D., Narasimhan S., Chakraborty R. S., Bhunia S., (2010), "Self-referencing: A scalable side-channel approach for hardware Trojan detection", in Proc. Workshop Cryptographic Hardware and Embedded Systems, 173–187, Santa Barbara, USA, 17-20 August.
- [18] Agrawal D., Baktir S., Karakoyunlu D., Rohatgi P., Sunar B., (2007), "Trojan Detection using IC Fingerprinting", IEEE Symposium on Security and Privacy (SP '07), 296-310, Berkeley, CA, USA, 20-23 May.
- [19] Cong J., Gupta P., Lee J., (2010), "Evaluating Statistical Power Optimization", in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 29 (11), 1750-1762
- [20] Agarwal A., Blaauw D., Zolotov V., (2003), "Statistical timing analysis for intra-die process variations with spatial correlations", ICCAD-2003. International Conference on Computer Aided Design (IEEE Cat. No.03CH37486), 900-907, San Jose, CA, USA, 9-13 November.
- [21] Web 4, (2021), <http://www.nangate.com/> (Access Date: 24/02/2021).
- [22] Web 5, (2021), <http://ngspice.sourceforge.net/> (Access Date: 24/02/2021).
- [23] Wang X., Salmani H., Tehranipoor M., Plusquellic J., (2008), "Hardware Trojan Detection and Isolation Using Current Integration and Localized Current Analysis", IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems, 87-95, Cambridge, MA, USA, 1-3 October.
- [24] Sylvester D., Agarwal K., Shah S., (2008), "Variability in nanometer cmos: Impact, analysis, and minimization", Integration, 41(3), 319–339

BIOGRAPHY

I was born in 1986 in Kardzhali, Bulgaria. I completed my undergraduate degree at Gebze Technical University Department of Computer Science Engineering in 2015. In 2014, I started working as an embedded system engineer at Ortem Electronic incorporated company. In December 2016, I left from Ortem Electronic and started to work in the same area in Accenture Industrial Software Technologies limited company. I am still working as a Test Engineer in the same company. I graduated from the electronics department of the industrial vocational high school in 2003. After graduation, I worked as an electronics technician at Ortem Electronic for many years. Later, in 2009, I won Gebze Technical University department of Computer Science Engineering and added my engineering knowledge to my technician knowledge and improved myself in the field of embedded systems both technically and in engineering.