VLSI ARCHITECTURE FOR HIGH LEVEL TRANSFORMATION ON HARDWARE SECURITY WITH AUTHENTICATION AND OBFUSCATION

Thesis Submitted for the Award of the Degree of

DOCTOR OF PHILOSOPHY
in
Electronics and communication engineering

By
Jyotirmoy Pathak
41500072

Supervised by
Dr. Suman Lata Tripathi (21067)

School of Electronics and Electrical Engineering (Professor)

Lovely Professional University

Transforming Education Transforming India

LOVELY PROFESSIONAL UNIVERSITY
PUNJAB
2022

DECLARATION

I, hereby declared that the presented work in the thesis entitled “VLSI Architecture for high Level Transformation on Hardware Security with Authentication and Obfuscation” in fulfilment of degree of **Doctor of Philosophy (Ph. D.)** is outcome of research work carried out by me under the supervision of Dr. Suman Lata Tripathi, working as Professor, in the School of Electronics and Electrical Engineering of Lovely Professional University, Punjab, India. In keeping with general practice of reporting scientific observations, due acknowledgements have been made whenever work described here has been based on findings of other investigator. This work has not been submitted in part or full to any other University or Institute for the award of any degree.

Jyotirmoy Pathak
41500072
School of Electronics and Electrical Engineering
Lovely Professional University
Punjab, India

CERTIFICATE

This is to certify that the work reported in the Ph. D. thesis entitled “VLSI Architecture for high Level Transformation on Hardware Security with Authentication and Obfuscation” submitted in fulfillment of the requirement for the reward of degree of **Doctor of Philosophy (Ph.D.)** in the Electronics and Communication Engineering, is a research work carried out by Jyotirmoy Pathak, 41500072, is bonafide record of his/her original work carried out under my supervision and that no part of thesis has been submitted for any other degree, diploma or equivalent course.

(Signature of Supervisor)

Dr. Suman Lata Tripathi
Professor
School of Electronics and Electrical Engineering
Lovely Professional University

Abstract

In addition to miniaturizing, today's technologies are also getting more and more linked to one another. The aggregate name for these connected, smart devices is "Internet of Things" (IoT). High-speed wireless sensor nodes, consumer electronics, and life-saving systems like implantable and medical devices are just some of the IoT applications already in use. Low-cost, high-efficiency, and high-security algorithms must be implemented on the most resource-efficient platforms for the devices used in these applications. The ever-changing nature of the application landscape necessitates fresh architectural mappings and design methodologies that go beyond conventional optimization. This thesis focuses on developing hardware architectures that can deliver robust end-to-end security and low-cost, low-energy solutions through the use of Application-Specific Integrated Circuit (ASIC) designs and Field Programmable Gate Array (FPGA) implementations. The proposed layouts are meant to provide safe semiconductor circuit production.

There are limitations on power, space, and safety that must be met in order to perform complex compute functions. Leakage current also exposes sensitive information through unavoidable side channels. The side-channel analysis suggests that attackers can uncover secret information by monitoring leakage, which remains linked to processed data. Data has a role in determining how much dynamic power CMOS devices have. Variations in high bits and bit position shifts cause power traces to sway.

For secure production, hardware obfuscation layouts are proposed. Hardware security has become more important over the past decade in response to reports of counterfeit gadgets detected in life-or-death military systems. In the next phase of the Internet of Things, when more devices are built and connected to a common network, the demand for security will increase. The goal of hardware obfuscation is to make circuits more secure by hiding their intended purpose and protecting them with secret keys before sending them off to be manufactured. Those without the correct secret key will be unable to decipher or use the circuit thanks to this technology. After acquiring the chips from questionable manufacturing facilities, they are encrypted and sent to end customers for verification of reliability. The development of novel structures and obfuscation techniques is the major focus of my research.

The suggested design uses both traditional CMOS components and a newer type of CMOS logic known as a "hybrid" (Transmission gate (TG) and pass transistor (PTL) logic). While there is no way to completely stop information from leaking out, tools like normalized energy deviation (NED) and normalized standard deviation (NSD) can help keep leaks to a minimum. The use of (TG+PTL) logic in design implementation has resulted in a 48.64% reduction in NED and a 21.8% reduction in NSD compared to static CMOS. By seeing the relationship between the amount of power used by the cryptographic block and the intermediate result, an adversary might speculate on the data that will be processed in the subsequent step. Based on statistical analysis, the most important side-channel assaults are differential power analysis and correlation power analysis. Examining input data and power usage trends using the Pearson correlation coefficient (ρ). If "ρ" is large and positive, then the power pattern and the processed data are highly correlated.

A cell that is impenetrable to assault would forget its prior connections to the data it processes. The resilience of the CMOS cell is investigated as part of the power attack countermeasure in order to provide protection against power analysis assaults. Hiding and masking are two common types of attack-resistant qualities, and they work together to shield the device from a power attack. During the course of our investigation, we made use of a boolean masking approach to produce the intermediate data in a random fashion. An intermediate value is masked in the architecture that has been proposed by making use of a random mask bit and extra cells. The output is then unmasked by making use of either the same random bit or a new random bit that has been derived from the first random bit. Cell-level masking can be used to make the cell's design more power-efficient and less sensitive to the data being fed into it. Here we see mask XOR and mask AND cells, whose true power consumption is demonstrated to be the same regardless of which internal node serves as the output. Since power is dependent on each internal node, and attackers cannot surmise information by seeing only the output node, the truth table approach ensures that the hammering weight of each internal node is dispersed evenly. Static power is increased by 84.4%, dynamic power by 77.8%, and latency is reduced by 164.8&percnt; when compared to a design without a mask cell. When compared to previously published work, the present design's use of Boolean masking at the cell level reduces the necessary gate count to 2173. There is an implication of independence between power consumption pattern and processed data when using the hammering weight power model, which

decreases the correlation coefficient between real and estimated power by 32.17 percent, and when using the hammering distance power model, which reduces the correlation coefficient by 26.65 percent. In Mask, the random number is merged with the input from the cells; however, the software used to produce the random number is not truly random and will cycle back after a fixed amount of iterations. One of the forthcoming hardware security modules is the response of a physically unclonable function to a challenge input and device/specific feature. For the purposes of constructing a physically unclonable function (PUF), a silicon-based device in which each of the circuits generates a distinct response is desirable. The delay and hysteresis bandwidth of the circuit determine the response of the Schmitt trigger PUF (STPUF). The given STPUF validates the parameter, with a uniqueness of 49.2%, and a reliability of 99.7% under severe operating conditions.

Acknowledgement

I am really obliged and appreciative to the several persons that assisted me with my research and thesis preparation.

I express particular appreciation and credit to the Almighty God for granting me the intelligence and health to finish this quest.

I would like to thank my adviser, **Dr. Suman Lata Tripathi**, professor in the School of Electronics and Electrical Engineering, for her guidance, support, and counsel throughout my Ph.D. programme. Her patience in providing me with comments over the years and her support for my work have helped me become a better researcher and complete my thesis effectively.

It is a joy to be obliged to the numerous individuals who directly or indirectly contributed to the production of this work and who inspired my thinking, conduct, and actions over the course of this research. I am quite grateful to my colleague and friend for their insightful advice.

I would like to express my appreciation to the School of Electronics and Electrical Engineering at Lovely Professional University for providing me with a conducive atmosphere for doing research.

Nobody has been more significant to me in my pursuit of this topic than my family members. I owe my parents, my siblings, and my in-laws profound gratitude for their love, drive, support, and wisdom when it was most needed. Importantly, I owe a great deal to my ever-supportive and caring wife, **Dwaraka Niranjana Pathak**, who has supplied me with continuous emotional and mental motivation. Always at a loss for words, I found it hard to articulate her support in words.

Jyotirmoy Pathak

Table of Content

List of Table

List of Figures

LIST OF ABBREVIATION

Abbreviations	Description
CMOS	Complementary Metal Oxide Field Effect Transistor
CPA	Correlation Power Analysis
DPA	Differential Power Analysis
EDP	Energy Delay Product
HDPM	Hamming Distance Power Model
HSM	Hardware Security Module
HWPM	Hamming Weight Power model
SCA	Side-Channel Attack
LUT	Lookup Table
MuxPUF	Multiplexer based Physically Unclonable Function
PDP	Power Delay Product
PTL	Pass Transistor Logic
PUF	Physical Unclonable Function
RNG	Random Number Generation
RO-PUF	Ring Oscillator Physically Unclonable Function
SPA	Simple Power Analysis
ST	Schmitt Trigger
ST_PUF	Schmitt trigger PUF
SBOX	Substitution Box
SVM	Support Vector Machine
TG	Transmission Gate

Chapter 1

Introduction

1.1 Overview

The Internet-of-Things (IoT), a global network that will connect billions of devices. The trend of computing moving away from the cloud and toward the network edge is being driven by the proliferation of interconnected devices. Edge-centric computing is the term for this approach. More and more businesses are moving their computing operations to the "edge" because of concerns about security and privacy, better data accessibility, and improved computing resources [1, 2]. However, there will be a number of issues that need to be addressed as a result.

- Manufacturing small integrated circuits and embedded chips is expected to rise as the number of devices being produced and connected grows.

- In-house foundries will see a decline in use as more companies opt to outsource their fabrication.

- The IoT framework thrives on both device-to-device and cloud-to-device communication in its network. These communications necessitate high levels of security and privacy.

- Smart cards and chips, as well as laptops and cell phones, can all be used at the edge of the network. A lack of high-quality random number generators, which is critical for a number of cryptographic algorithms, could result in low device resources in many applications.

- Cryptographic algorithm tasks like secret key distribution and management can be difficult due to the large number of devices connected to the network.

- Machine learning algorithms, such as Support Vector Machines, Neural Networks, etc., will become increasingly necessary to run on the device itself. Training algorithms can also be run on the hardware in many cases. This necessitates a lot of processing power and memory.

- IoT devices must consume as little energy as possible because they are typically powered by small portable batteries or wireless power transfer.

Because of the current craze, hardware security is becoming an increasingly important area of study. While software threats can be traced, physical tampering of hardware can go undetected, making hardware threads difficult to detect. It is extremely dangerous to distribute this hardware in an unsupervised and uncontrolled manner, as the user may be fooled by the tampered device. IC chips, which are becoming increasingly common in consumer electronics, are a result of the current smart era's emphasis on making every device as user-friendly and intelligent as possible. In the military, precise and efficient results are achieved through the use of these Hardwares. Scientific and space research organisations also use them, and the security risks they pose are frightening. IP theft, piracy, Trojan insertion, and information leaks are just a few of the many attack models. The device's security may be jeopardised as a result of this. Trojans, which are difficult to detect, and illegal recycling of the design are two additional methods of attack that are difficult to learn about. Since the hardware implementation provides more information than specific algorithms, these approaches are extremely risky.

Demand for semiconductor devices has skyrocketed as a result of the invention of semiconductor material. For the most part, other consumer electronics have utilised semiconductor integrated circuits (ICs). As a result, there was a demand for a more advanced device with additional features, and scaling technology was used to meet that demand and increase functionality. The threshold value and the supply voltage are both reduced in this method. Moore's Law predicts that this scaling of the device will occur [3]. As technology shrinks in size, new issues in the circuit arise, such as the difficulty of fabricating 10nm technology and problems with IC testing. In response to the fact that IC production had become a time-consuming and difficult process, several foundries emerged to address these issues. Some companies offer software tools to assist in the circuit's verification and design. Soft intellectual property (IP) provided by designers can also significantly reduce design time. The next step is to have the ICs fabricated in an outsourced fabrication facility, which aids in both precision and speed. This has also prompted questions about the ICs' safety. No one can guarantee 100 percent safety; the best we can do is design a chip that makes reverse engineering much more expensive and time consuming. Figure 1.1 depicts the various stages of IC production and the security issues that arise at each stage.

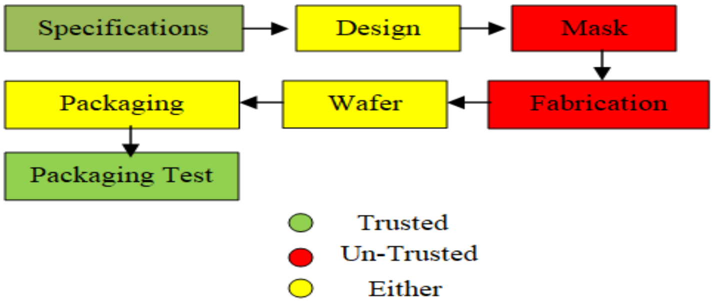


Figure 1.1. Level of IC design and security at each stage

Another security concern arises from the use of outside foundries in the fabrication process. Allegations of hardware theft have skyrocketed in recent years [4]. As a result, it is even more critical than before to focus on hardware security. The designer's ability to maintain creative control has been severely weakened as a result of the outsourcing industry. Since this design was stolen, some security measures have been put into place. The IC's threat can be categorised as:

- Hardware Trojan Insertion is used to steal sensitive data from a target device.
- Reverse engineering is a method for discovering the design's structure and functionality.
- Stolen design and manufacturing of the same IC under another name results in significant revenue loss; this practise is referred to as "IC theft".
- Illegal use of intellectual property to produce the design is known as IP threat.

Many techniques are procured at various stages of IC manufacturing in order to ensure the design's security. Authentication-based and obfuscation-based approaches are the two broad categories of hardware security technique.

The standard set of characteristics used to characterise very large scale integration (VLSI) circuits now includes security. The safety of the system must be built on a bedrock of trust. The response (output) of the hardware security module is a nonlinear function of the challenge input [5, 6], making it impossible for unauthorised users to foresee. A piece of information is given a physically specified answer by the Physical Unclonable Function (PUF). Semiconductor devices that exploit PUF take use of a physical device or circuit variance that occurs spontaneously during manufacturing [7]. According to NXP Semiconductors, a PUF system may be employed for electronic qualities that are inherently unstable but whose stable state varies. The reaction of the electrical gadget depends on both the input and its own properties. If a third party has knowledge of the feedback and the equipment, unauthorised access is impossible. PUF should offer (a)a wide range of responses to the same challenge, (b)a consistent distribution of responses, and (c)a steady response even in a hostile operating environment. PUF's two most important uses are in secure authentication and key creation for use in cryptography. [8, 9]. Both a ring oscillator PUF and a delay-based arbiter-PUF are available as PUF parts.

PUFs, or physically unclonable functions, are innovative, promising circuits that are primitive in design but hold great promise because they can be used for authentication, as well as for generating and storing the secret key, without the need for the previously mentioned expensive, power-guzzling large circuitry. PUF generates the secret key, rather than keeping the secret key in nonvolatile memory, so that we can accomplish this goal.. It is impossible to control all aspects of integrated circuits during manufacturing. Due to manufacturing variations, PUF realises these controllable parameters and converts them into a readable form. It is possible to use PUF's uncontrollable parameters as a secret key. Depending on our application, this secret key can now be used for authentication or cryptographic methods.

A few of PUF's advantages over other standard protocols are listed below.

- Because SRAM/EPROM memory sources are expensive and difficult to manufacture, PUF hardware uses simple digital circuits that are easier to manufacture, require less space and consume fewer kilowatts of power. To replace PUFs, we can use the secure hash algorithm (SHA) and replace encryption algorithms that use public and private keys with cryptographic hardware techniques.

- We generate our secret key in PUF using the physical parameters of an integrated circuit. When the device is powered on, any information about the secret key must be extracted. As a result, the chip can only be attacked physically when it is powered on.

- Attacks on PUF are more difficult because the physical parameters of PUF must not be altered to extract the secret key. As a result, invasive attacks against PUF are more difficult to carry out. Due to the lack of invasive attacks on PUF, we don't need expensive and power-consuming anti-tamper mechanisms.

- Non-volatile memory manufacturing is a costly endeavour. EPROMs and RAMs, which require external power, always require additional mask layers. These are not necessary in the case of PUF.

Watermarking and IP signatures were the most widely used authentication methods in the beginning, but they were eventually supplanted by more sound and effective security methods that protect against both copyright violations and unauthorised copying. This strategy helped to reduce the illegal distribution of ICs, protect intellectual property rights, and reduce the use of counterfeit IC chips by consumers. To protect their products from unauthorised reselling, manufacturers use watermarking and signature techniques. However, this does not prevent counterfeiters from learning about the design's internal structure and fabricating their own chip. The basis for a watermark or digital signature is that the IC displays a logo that identifies the original designer or manufacturer of the product. Because it does not provide security at all levels, this technique is ineffective in preventing overbuilding and IP piracy by protecting against illegal use of the IC. It is still possible for the hacker to study the schematics and gain an understanding of the IC's operation and structure, and then produce his own version of the IC from scratch. In the IC industry, reverse engineering has become a major threat to theft and espionage. An IC's structure, functionality, and design can all be described using Reverse Engineering. It is very easy to tamper with or illegally use a circuit once you have a good understanding of its inner workings. The number of incidents involving the theft of computer hardware has skyrocketed. Additionally, various techniques like watermarking [10, 11] and cryptography [12, 13] can be used to prevent the illegal and unauthorised use of IC. The use of a key or fingerprint as a form of authentication is also common. Keys used to encrypt designs previously resided inside of the IC chips, but new techniques for reverse engineering have made it much easier to extract the key from the fuses, which are located on the surface of the chips. As a result, a method for creating a key that can't be duplicated was devised. PUFs and FSMs are both viable options for providing a user's key. To operate normally, the circuit needed to be entered correctly. To operate incorrectly, the circuit needed to be entered incorrectly.

Physical Unclonable Functions (PUFs) exist in several forms, with each variant possessing distinct attributes. Various types of Physical Unclonable Functions (PUFs) include the Ring Oscillator PUF, SRAM PUF, Arbiter PUF, Delay PUF, Optical PUF, MEMS PUF, and

Analogue PUF. The selection of a physically unclonable function (PUF) type is contingent upon the particular demands of the application, the desired level of security, the resources at hand, and the vulnerability to environmental and manufacturing fluctuations. Scholars persist in their investigation of novel forms of Physical Unclonable Functions (PUFs) and the enhancement of pre-existing ones in order to tackle diverse issues and boost the overall security and dependability of PUF-based systems. A Delay Physical Unclonable Function (Delay PUF) is a variant of a Physical Unclonable Function (PUF) that leverages the inherent changes in signal propagation delays present in a circuit. One of the advantages of Delay Physical Unclonable Functions (PUFs) is the technique employed involves the utilization of variances in signal propagation delays. The utilization of common digital components can facilitate the implementation process. Exhibiting resistance to certain environmental fluctuations. Prone to fluctuations in temperature. The Delay Physical Unclonable Function (PUF) has been selected for our research because to its ability to operate independently of environmental fluctuations and its utilization of timing variations in the signal.

Once the chip has been manufactured, we must ensure that the adversary has a difficult time understanding the chip's structure and function. This sparked further research into the secure technique, which resulted in the development of a number of similar methods in the years to come. Obfuscation is one of the most widely used strategies. Obfuscation is the primary focus of this paper. The process of making an IC structurally and functionally safe is referred to as obfuscation. The IC's structure and function are hidden from the adversary, making it much more difficult for them to reverse-engineer the IC itself. Due to the increased difficulty and expense of reverse engineering, the design is a success in protecting against various malicious attacks.

When obfuscation is employed, there are two primary approaches: structurally and functionally.

Structure Obfuscation is when we make it more difficult to figure out the design's function by making it more complex structurally [14]. In [15], we saw how by employing HLT techniques, we were able to alter circuit structures to look similar but perform a different function. Although they appear to be structurally identical, the two circuits are functionally distinct.

A key is used in Functional Obfuscation. The design will not produce the desired result unless the key is entered correctly. It is possible to store this key in the form of fuses within the circuit.

Fusing the key into non-volatile memory of circuits makes it insecure because the adversaries can get their hands on it and use it to attack the system. The key to the circuitry was thus developed using a technique known as PUF [16]. Using dynamic obfuscation in [17], the author devised a technique that uses the circuit going into obfuscated mode, but does not always produce the wrong output, in order to make it much more difficult to reverse engineer. To make the design more difficult and more secure, a random number generates output that is sometimes correct and other times incorrect.

1.2 Researcher's Contribution

This work presents a Physical Unclonable Function (PUF) based on a Schmitt trigger whose response depends on challenge input and two device-specific characteristics. Until now, PUF responses have never been linked to multiple hardware-specific characteristics.

To reduce the number of in-line delay elements, a shifting algorithm has been proposed. Advanced sign preparation calculations are represented by iterative information flow diagrams, in which hubs and edges speak to calculations and correspondences separately because of the redundant concept of these calculations.

It has been suggested that hardware obfuscation, which involves making modifications to integrated circuits and locking them with keys, be used to ease the burden of secure manufacturing. In my research, I present a novel approach to hardware obfuscation by modifying control signals of the architecture and extending it for use in a hierarchical manner to an integrated circuit.

Attacks on obfuscated circuits have become more effective due to the availability of low-cost resources and equipment, and current obfuscation methods are insufficiently secure and easy to break. As a response to this issue, an obfuscation method that uses modes that are unpredictable and random was developed to increase the design's security by orders of magnitude.

By analysing the power supply, the attacker can decrypt the secret data. In this study, we suggest a design for mask XOR and mask AND cells that is resistant to power attacks. Due to the reduced correlation coefficient, it is clear that the pattern of power consumption does not retain a linear relationship with the processed data when using a mask cell implementation.

1.3 Outline of Thesis

There are seven parts to the thesis. Here is a rundown of what each chapter covers.

The necessity of hardware security is presented in Chapter 1. It stresses the need of constructing a physical unclonable function and gives a brief summary of the limitations imposed by side-channel information. This chapter incorporates the author's work.

Chapter 2 details the research of a third party in accordance with the goals of the author. There are three main categories that we'll be looking at: the formation of physical unclonable functions, power and assault, and defences against power attacks.

In Chapter 3, we see an original idea for an unclonable physical function that relies on Schmitt triggers. The chapter delves further into the STPUF simulation findings and details their successes.

In Chapter 4, we see how to determine the iteration bound using the column-shifting method. In this chapter, you will learn how to use the Folding Transformation to create both meaningful and meaningless blocks for use in mode-based obfuscation.

In Chapter 5, we see an attack against SBOX based on power analysis, specifically a combination of pounding weight and distance power models. The connection between actual strength and anticipated strength is investigated here.

The chapter 6 content is dedicated to the attack-resistant feature of the logic cell design. In this section, we looked at masking, a technique used to hide sensitive data from circuits.

Chapter 7 concludes the thesis by discussing the most important findings of the author's current research, the contribution of the thesis, and the scope for further study in this area.

Chapter 2
Literature Review

Concerns about the security, authenticity, and reliability of electronic components (ICs) have risen as a result of their widespread use in smart phones and other devices. Security measures have been developed to protect the design not just against algorithmic attacks but also from physical ones. There have been a number of attempts to design an authentication system that secures the owner's entitlement to the IC, but does not shield it from physical attack. Thereby, various methods were investigated and the secret key-locking method was developed. Incorrectly inputted keys do not operate the IC and it displays unknown value. Previous versions of reverse engineering techniques utilised RAMs inside IC chips to store the key required to lock designs, but newer techniques like side channel attacks and power estimates made it possible to extract the key from fuses with relative ease. It made it simple to make a copy of the key and use it to unlock the design by making power differences in the design. To ensure the IC's security, a more trustworthy method was required. PUF and FSM techniques advanced as a result of this. Because of its inherent inability to be replicated, these tactics quickly became popular. It demonstrates a wide range of inter-chip and intra-chip features.

Since 1970, CMOS has been the dominating technology in the VLSI industry. VLSI technology has made an important trade-off in the design of integrated circuits (I.C.s). yield or die area was of paramount importance in 1970. In 1980, circuit speed became the primary constraint due to advancements in technology. The 1990s scaling era was characterised by an emphasis on improving both speed and power. In 2000, it was discovered that adding more components to a wafer increases cacophony. and that noise may be used to improve both power and speed. It's no secret that security was a major focus in product development in 2010. To make, leakage information independent, of the data to be processed, additional circuitry has been added, resulting in an increase in die size and power consumption. A directed critical path, on the other hand, can reduce the time.

2.1 Hardware Security Module (HSM)

By incorporating the unpredictable characteristics of semiconductor devices, hardware security aims to raise the algorithm's security bar.

2.2 Physically Unclonable Function (PUF) as Hardware Security Module (HSM)

It's an inventive circuit that extracts a unique feature from integrated circuits. PUF circuits based on the silicon IC-based were introduced in 2002 by Gassend, Blaise, and others [18]. Even when using the same material and tools, no two silicon devices will have the exact same properties. 2007 [19] saw the introduction of PUF's use in authenticating and generating random numbers by G. Suh and S. Devdas. For each challenge, the unique PUF response should be impossible to guess by users. Horie et al. [20] in 2013 confirmed the PUF circuit's uniqueness, homogeneity, and reliability.

2.3 Obfuscating DSP circuits via High Level Transformation

It's become apparent that software isn't the only item that requires safeguarding with the unexpected explosion of smart devices in our homes, workplaces, and military forces. Moore's Law predicted in [3] that the number of transistors on a semiconductor would increase exponentially due to the pervasive use of integrated circuits (ICs) in electronic devices and the optimal utilisation of those devices' functionality. When the density of chips increases, the size of chips shrinks. Because of the difficulty and length of time involved in designing and fabricating IC devices with a shrinkage in the 6nm-7nm region, this technology has become obsolete. A number of outsourcing organisations have emerged to handle fabrication, masking, and IP production tasks as a result. The rise in outsourcing of chips has resulted in a loss of control for the designer in the design process. There are numerous ways to compromise the IC chip's security. The number of incidents involving the theft of hardware has skyrocketed [21]. Protection from Hardware Theft and Reverse Engineering threats is a must in light of these requests. A technique known as Obfuscation is discussed in this work. Obfuscation is a technique for making the design difficult to reverse engineer in both its functional and structural aspects. The more secure a device is, the more time and work it takes to reverse engineer it. This section discusses structural obfuscation via HLT, such as folding and pipelining.

Since electronic gadgets are becoming more and more interconnected, it's become apparent that software isn't the only thing that needs safeguarding. Devices need hardware security in order to function properly. Military and other sensitive areas are concerned that electronic devices could leak sensitive information. Authentication or obfuscation-based approaches can be used to accomplish this. This study takes an obfuscation technique, in which a design's structure and function are altered in such a way that understanding it and deciphering it are both tough tasks. In contrast to previous work on obfuscation DSP filters, a combinational DSP multiplier is employed in this paper. Folding Technique has been used to perform Structural Obfuscation on the Wallace Tree and Baugh Walley multipliers. Folding and retiming techniques have also been shown to significantly lower the power consumption of the original design. By altering the timing of switch connectivity, Structural Obfuscation by Folding not only made the design difficult to understand, but it also provided protection against the design's functional identification as well. The DSP circuits, with switches for timing control are difficult to comprehend at a glance, and the same structure serves multiple purposes. In order to accomplish this goal, the switches in the circuit might be programmed to act in a different manner. The behaviour of the circuit is dependent on the configuration of the switches. The output is accurate if the switches are synced with one another in the suitable manner; otherwise, it is inaccurate.

2.4 Side Channel Information

Instead of exploiting a flaw in the security algorithm, a circuit secret can be obtained by a side-channel attack. Mathematically secure security algorithms like DES and AES rely on the user providing the primary input and the secret key being used to calculate the primary output. A VLSI implementation of such an algorithm has hardware restrictions, as explained in [22, 23]. Leakage is a problem for CMOS devices, and it becomes more noticeable with scaled MOS transistors. The cryptographic algorithm's principal output is computed by a CMOS implementation, in addition to the secondary information that was leaked. By using a technique called side-channel analysis, the attacker can obtain control of the device and examine its internals. Although a power assault is the most efficient type of side-channel attack, there are others, like heat, sound, time, faults, etc., that are all viable options. Attacks that use passive side-channels take use of information that is released from the outside of a device through its interface. These attacks are discussed in [24, 25]. An active side-channel attacker actively manipulates, a device by introducing probes and defect data in order to get the sensitive information discussed in [26], which describes how this type of attack is carried out.

2.5 Literature Review: Categorization

2.5.1 PUF Literature Review

In [27], Maiti, A., Kim, I., and Schaumont, P. explain that a unique response to each challenge is a necessary condition for device authentication. A reliable PUF must generate a unique response from a pool of possible responses to the same input bit challenge, as well as a large numbers, of random numbers from which the secret key can be extracted. To improve the CRP set, the author of this study proposes a new method that relies on a recursive operator (R.O). A PUF's entropy can be adjusted during construction by adjusting various circuit characteristics. When a large key and a large numbers, of challenge-response pairs are employed in a circuit, both the hardware area and cost are increased. The author uses ROPUF to produce CRP pairs of the same size while using fewer resources. The area required for this CRP generator has been cut in half, and it is now able to generate additional functions via statistical hypothesis testing on an FPGA platform. Uniqueness, reliability, and impartiality are the metrics by which PUF quality is evaluated. High output uniqueness and dependability may be attained with the presented PUF, which has an inter-chip H.D. of an average of 50.07 percent. Even in a protected setting, a PUF circuit should be able to produce unique replies. High-temperature, threshold-voltage-variable, response-bit creation using memory. The limitations of supply voltage and the means by which it can be distributed and diversified were studied.

Different PUF structures, are constructed and their performance, is compared in paper [28] by Susana Eiroa et al. (2010) in terms of, inter and intra hamming distance. In PUF, a collection of challenges is transferred to a set of responses, both of which are bit/byte in size and are driven by physical characteristics that are difficult to predict and recreate. Two of PUF's most important uses are in secure authentication and random number generation in hardware [29]. Discussed and analysed is a silicon I.C.-based exact random generating technique for its randomness, uniqueness, and dependability in static analysis. Investigated were the challenges of implementation and how they were affected by their surroundings. Since there are many variables to tweak before getting a usable result, PUF selection can be tricky. This research covers many PUF architectures, and discusses how reconfigurable PUF can be used to dynamically alter the inherited attributes. A controlled PUF system will typically allow access to a generic processing unit via a dedicated programme. Cell leakage current is used to calculate the PUF, and the PUF is then digitalized and encoded to produce the cell's unique identifier.

Bitstream encryption is used by many FPGA manufacturers as a means of protecting intellectual property, as discussed in paper [30] by Kumar et al. (2008). PUF is an alternative physical system that extracts response from the integrated circuit's inherent physical features. PUF, as explained below, is a very secure approach to creating a temporary secret key for use in a cryptographic operation that cannot be duplicated. In this study, the author uses the unpredictability of the D-latch to achieve stability following metastability caused by the violation of the setup and hold times. A butterfly PUF is implemented with common I.P. found in any FPGA (cross-coupled D-latch structure, CLB, BRM). The effects of temperature changes, from -20 degrees Celsius to 80 degrees Celsius, on the inter- and intra-hamming distances are investigated. Measured H.D. and FPGA-based H.D. discrepancies validate PUF. In order to generate the response, Butterfly PUF makes use of the integrated circuit's inherent physical qualities, specifically the matrix of available internal resources on the FPGA. An encouraging component, Butterfly PUF creates a steady response and FPGA-friendly environment to run in. Katzenbeisser et al. (2012) of the paper [31] implement a delay arbiter and ring oscillator (RO-PUF) using CMOS 65nm technology.

The temperature, supply voltage, and noise properties of PUF have been investigated. When compared to SRAM PUF, ROPUF has complete independence in all desirable qualities [32]. Data from testing performed on 96 TSMC 65nm CMOS ASICs. The following PUF characteristics are taken into consideration during an evaluation: (a) robustness, as measured by bit error rate; (b) the number of bits in response1 differs from response2. Unpredictability challengers (b) are incapable of computing PUF's response to a known input challenge efficiently. The entropy of flip-flops and latches is temperature-dependent, whereas the Arbiter PUF exhibits minimal entropy. Authentication and key generation are two of the most important uses of PUF, hence their suitability and selection are crucial.

Majzoobi et al. (2012) offered a safe way to authenticate and reply utilising arbiter and strong PUF in article [33]. With the XORing together of the path responses from a linear arbiter and two separate parallel arbiters, we may implement PUF with improved static characteristics. Thin PUF correctly guessed a random number and discovered that a 64-bit arbiter PUF model achieved 95% accuracy. The presented slim PUF deviates from the traditional model in that just a fraction of the responses received for authentication are revealed. Lightweight error correction logic, a fuzzy extractor, and resistance to machine learning attacks all contribute to the response pattern's strength as a method of strong authentication. Short PUFs with a critical length of 128, 256, or 512 reach the hamming threshold of 33,76 or 154 bits, respectively.

According to Handschuh et al. (2010), the PUF is a feature of a device in paper that may be used to extract the key. It is also called an electronic fingerprint [34]. Physically Unclonable Function (PUF) refers to the physical characteristic of a device that is difficult to replicate due to its distinct micro and nanoscale features, which are the result of manufacturing variation. The cloning of PUF is a lengthy process that necessitates the use of sophisticated mathematical models and computational algorithms in order to predict the correct answer to a problem in a timely manner. Here, the author implemented a biometric based on differences in threshold voltage between SRAM transistors. A fast transistor and a slow transistor each make a choice based on logic to determine if a situation is 1 or 0. The author of the paper proposed a secure key storage method for SRAM and ROM, where the secret key is stored. With each transistor's length in mind, a separate threshold voltage can be attained, and a different initialization value can be provided by each cell. Operational consultation temperature, voltage variation, and radiation evaluation of PUF quality. The various PUFs that have been reported are as follows: SRAM PUF 950/1000 bits The delay PUF is 130 bits per 1000 bits, while the butterfly PUF is 600 bits per 1000 bits.

True random number generation is described in paper [35] by Barak et al. (2014). It's important that the random number be uncalculable. Although PRNG-based random numbers are typically thought to be non-deterministic, it is possible to predict the next number by looking at its past behaviour. Multiple extremely important applications, such as lotteries, Monte-Carlo simulations, cryptographic key generation, etc., made use of random numbers. The software process used to produce random numbers is not truly random; it cycles back over the same numbers after a fixed number of iterations, and it can be computed if the original seed or the pattern of numbers generated before it is known. Mixed hardware properties are used in a real random number generator. This parameter is a nonlinear function of the derived random number. The author explains how to generate random numbers using noise as the source of entropy in this study. A generating scheme's entropy comes from a random operation, or R.O., malfunction. There is still work to be done, but the author has laid out the challenges and opportunities in random number generation. To create digital data, the noise mechanism amplifies and digitises a low-voltage signal. Given that the I.C. designer can bring down the noise to any acceptable level, designing an amplifier is a tedious process. A random number generator (RNG) based on the concept of chaos combines a collection of free-running oscillators to generate numerous frequencies at a randomly selected rate, and then applies that rate to a digital block to produce a random bit.

Bhargava et al (2010) [36] Personal Uninterruptible Power Supplies that use a sensing amplifier to withstand attacks on the measured voltage. In static memory, the amplifier serves as the reading path. This circuit can detect very low voltage changes. The author of this work has produced a sizable number of SRAM cells, each of which has a threshold voltage that is distinct from the others. The input delay is determined by the unique voltage that is present in each cell. Allows, for every column sense amplifier It is the offset voltage and the mapping strategy that determine the resulting response. The minimal voltage required to resolve a 0 or 1 bit is only a few microvolts, and the sense amplifier can do this reliably. The error is fixed on the answer in a post-processing unit so that the bit is consistent and trustworthy. The research presents a low-cost error-correcting code (ECC) block method for protecting power-under-full-load (PUF) amplifiers against the effects of environmental variation. Through a scrambler, the SA PUF design obtains the required address from the S.A. array based on the inputted challenge bit. The median of the simulation results based on CMOS 65nm technology Area reduced by a factor of 2.5 and the hamming distance down to 16.000937.

NXP Semiconductor (2013) [37] published a paper titled "Processor-Based Consistently high PUF over time confirmed by studying response consistency across many years. Threshold voltage, effective gate length, and other factors are all influenced by the intrinsic randomness of the manufacturing process in silicon I.C. The final response bit is decided by combining the results of two ALU cores, with a xor gate inserted in the middle of the circuit to prevent any information from escaping. The effect of changing challenge and PUF behaviours on Hamming distance is investigated. A minuscule circuit embedded within the processor extracts response bits with a distinct temporal signature on the fly. Using an ageing parameter to evaluate the decay of logic level over time, the given PUF is able to produce an average Hamming distance of 16.1% for a 32-bit response. After that, this difference is sent into an amplifier and a digital converter block as their respective input. Following the deployment of CMOS 45nm technology, an increase of 98.1% in the typical intra-chip hamming distance is seen.

Hossain (2015) in article [38] presents a promising security module based on the ROPUF random function of the I.C.'s special functions that are distinct for each die. Producing a new variety of PUD through I.C. manufacturing variations. If you want a reliable component in your PUF circuit, RO is your best bet because it consistently delivers the desired results. All of the inverter stages in the R.O. were constructed using CMOS and FTL technology, and the author recommended using 13 sets of RO PUF to generate independently-operating frequencies. An analysis of the differences between a standard inverter and an FTL inverter is provided; the FTL inverter is more inverse since it uses a larger number of NMOS in a parallel configuration, each of which is operated by its own clock. Using an FTL inverter allows for a high frequency of operation. To some extent, each of the frequencies was chosen at random. The results suggest that we are 45.24 percent unique (the target is 50 percent), 91.14% reliable (100 percent), and 41.45 percent uniform. Low threshold voltage PUF quality reports of 46.85% uniqueness, 95.54% reliability, and 40.799% uniformity, and high threshold voltage PUF quality reports of 41.71% uniqueness, 90.12% reliability, and 42.34% uniformity.

Maiti et al. (2009) publish a study [39] where PUF is used as a (TRNG) True Random Number Generator. Both PUF and TRNG have proven to be useful for developing a secrete system. The TRNG is used to produce random padding bits and the initialization vector, while the PUF is responsible for extracting a random chip signature to use as a volatile secret key. In this work, the author investigates R.O.-based TRNG, bringing novel contributions in terms of reduced area requirement, scalability, portability between platforms, and bridge between circuit and system. The TRNG bitstream quality has been validated by both the NIST and DIEHARD test suites. Xilinx's Spartan 3S500E series FPGA was used to create a rising-edge 32-bit and 64-bit RO PUF. The originality and reliability of the RO PUF answer are, respectively, 42.8% and 80%. However, when the number of R.O. stages increases, these values begin to decline. The circuit is controlled by an XOR gate, which generates 32 bits of randomness at 3.2 MBPS and a unique device signature.

According to the findings that Kong et al. (2013) report in their study [40], device authentication requires very little power and smaller circuits than traditional methods, which makes PUF's modest dimensions suitable. Stability of silicon PUF may be improved by lowering the temperature coefficient of threshold voltage and lowering the nmos and pmos threshold voltage fluctuation. The use of inverters with a variety of widths and architectural layouts contributes to an improvement in the dependability of PUF. PUF, depending on the technology, was evaluated using a NIST test suit for CMOS responsiveness at both 90nm and 45nm. PUFs that measure leakage current make use of a leakage sensor whose output remains the same from measurement to measurement. Despite this, PUF responses can fluctuate to some degree owing to manufacturing tolerances and other variables. When compared to RO PUF, leakage PUF achieves a score of 60% reduction in power consumption while also achieving an 84% reduction in area and a 35% improvement in temperature stability utilising Monte Carlo simulation experiments with 2000 chip instances.

Using transitory effects, Maiti et al. (2013) [41] suggest a ROPUF. The planning, testing, and fine-tuning of the ROPUF impulse response. The ring oscillator's brief effect at startup increases randomness and produces a more consistent response than ROPUF. The TERO PUF response generating system was evaluated on 30 chips fabricated using a CMOS 350nm manufacturing technology at room temperature with standard corner and voltage settings. The design goals of TERO PUF are to maximise efficiency and minimise footprint without sacrificing output quality. To further ensure stability, a second balancing block is used to draw out maximum entropy. By isolating TERO's oscillation, TERO PUF generates multiple response bits from a single circuit. There is 49% originality, 5% consistency, and 99% chance in the production of responses.

PUF design with remanence decay is reported in paper [42] by Shaza et al. (2015). Remanence is the property used in conjunction with non-volatile memory to retain the binary value for an extended period of time. In this paper, we propose a CMOS 65nm PUF array with 8KB of SRAM as a lightweight security primary. In order to increase cloning resistance in SRAM PUF, remanence decay is utilised as a countermeasure against side-channel attacks. The goal is to achieve a time remanence of 43.455% and a voltage remanence of 44.82%.

According to study [43] by Herder et al. (2014), PUF can be broken down into two categories: those that are weak and those that are strong. The author has developed a PUF based on a ring oscillator, SRAM, and arbiter using 180nm CMOS technology. According to the application, PUF is classified as a string and feeble PUF. The strong PUF can be used for secure authentication, while the weak PUF can generate cryptographic keys. While a powerful PUF requires a large number of CRP, a weak PUF can get by with a smaller number by taking advantage of the manufacturing variation. In addition, results obtained by PUFs were previously steady and dependable thanks to error correcting algorithms. Various PUF topologies, including optical PUF, Arbiter PUF, and RO PUF, have been tried and tested by the author. The 6% inter- and 5% intra-variation in PUF responsiveness obtained across a -25 to 85C temperature range is impressive. Since it is difficult to test to clone PUF and predict its past CRPS, it is also difficult to teach it to be vulnerable to attacks.

Meguerdichian (2011) introduced in article [44] a new design of Dynamic PUF (DPUFs), whose properties fluctuate randomly between applications. Since device ageing allows for fast reversibility and low-energy customisation, it was a natural fit in this scenario. For this purpose, the author suggests using a technique from numerical algebra to perform reverse engineering. As predicted by the ageing model for CAD tools based on negative bias temperature instability (NBTI) and hot carrier injection (HCI), particular hardware properties for the PUF non-linear function degrade over time. When a module pair ages, there is a difference in their state, which can be decoded as a logic one or zero by a variation arbiter. A buffer chain must be included to the design in order to provide protection from attacks. High unpredictability and rapid stabilisation are hallmarks of dynamic PUF.

In their article [45], Zouha et al. (2012) present a method for assessing a silicon PUF based on delay elements during the design phase without the circuit. The method is predicated on a Monte Carlo (MC) simulation of the netlist that accounts for process variation and the surrounding environment. When endeavouring to evaluate PUF, numerous parameters are utilised. Maximum entropy randomness. PUF constructed with HDL and simulation results based on Cyclone II FPGA to guarantee uniqueness and dependability (response bit unaffected by environmental and operational conditions). This method verifies the simulation results by comparing them to silicon measurements, which indicate that the arbiter PUF is 97% unique and 98% corrupt. Loop PUF is able to accomplish a 95% rate of uniqueness and a 98% rate of stability by utilising a delay chain, an odd number of cascaded structures such as a ring oscillator, and a frequency measurement device.

Pegu et al. (2015) of article [46] make use of an uncontrolled intrinsic random characteristic of the integrated circuit during manufacturing variation in order to develop a non-linear and unique response for each challenge input based on a multiplexer PUF. This is accomplished by using the inherent randomness of the integrated circuit. In this study, an analysis of the different MUX-based PUFs is provided, and it is based on layout-based simulation in CMOS 50nm.rul utilising tools like as Microwind and DSC2.7. The analysis has been tested extensively. Different MUX-based arbiter PUF topologies are compared. Mux PUF's feedforward topology boosts the mux's nonlinearity. With MUX PUF, you get a randomness of 86%, a combination of rarity (79%) and reliability (88%). There is 11% variation between chips under one scenario and 9% variation under another. Nonlinearity is enhanced and mistakes due to the environment are reduced in a feedforward architecture. Mux-Demux PUF is a reconfigurable PUF that allows you to choose the propagation direction.

In paper [47] by Ozturk et al. (2008), the pseudo-random function is implemented with a delay-based pseudo-uniform generator (PUF) based on intrinsic process variation and random switching via input challenge. One- and zero-based logic tristates also feature a third stage, which acts as a high impedance (Z). Output tri-states IF gate is disabled. Since the delay of BUF0 and BUF1 stage is the different reaction time of tristate buffer, a delay unit is implemented with BUF0 and BUF1 stage while allowing the same input to pass. Tristate buffer acts as a barrier between two states, much like an arbiter PUF does except the inverter is removed. BUF0 has more delay than BUF1 because of an extra invert gate. Input lag can be calculated using a mathematical model. The simulated results based on the TSMC 130nm technology library reveal that at 100MHz, mux PUF require 193.67W and tristate PUG require 152.93W, with a reduction in power consumption of 49.

On-line evaluations of the NIST test, the Diehard test, and the Dapra Shield test, among others, are presented in article [48] by Hussain et al. (2014). In terms of hardware predictability and stability, BIST-PUF offers real-time measurement of the desired PUF features. The response generator for BIST PUF is incorporated into a test suite. Quantitative reporting and evaluation of PUF stability and predictability via a web-based assessment tool.

Sensor-based analysis, parametric probing, and multiple probing are the three methodologies used in BIST PUF evaluations. High entropy TRNGs use BIST PUF to eliminate challenge generation bias. The 18.92mW of power that is needed to run the BIST PUF on a Xilinx Vertex-5 based FPGA is eliminated by the stability tool.

A new research by Yao et al. (2013) [49] demonstrates how to use pairwise skews between clock network sinks to recover a secret using PUF techniques. These methods improve the reliability of the clock network in order to isolate individual chip signatures caused by manufacturing flaws. The CMOS clock network was built at the 45nm node. To make the chip more secure, an on-chip clock network is constructed; a delay that may be adjusted increases the entropy of the clock PUF. Multiplexors execute pairwise comparison using an algorithm, and the addition of a return oath helps keep traffic on the network's roads to a minimum. When running a Monte Carlo analysis with 1000 keys, SPICE uses the Inter and Intra hamming distances as a verification metric. Inter-Hamming distances in the PUF configuration are as desired (50%). The ability to reliably reproduce a result depends on the response being invariant throughout a voltage range of 10% and a temperature range of -20°C to 120°C. PUF clocks are twice as reliable than ROPUF ones.

2.5.2 Literature Review of Obfuscation Technique

With so many ICs on the market, it's getting harder to keep their secrets safe while keeping their functioning intact (2012) [50]. Overbuilding and hardware Trojan infection are both options for attackers. Therefore, measures are taken to secure the ICs, such as the use of logical obfuscation, PUFs, and IC camouflage. This paper compares and contrasts the efficacy of several methods of logical obfuscation from the perspective of security. By inserting special gates called key gates into the circuits, this method hides the ICs' true performance. Only the right key will unlock the IC's capabilities; using the wrong one will result in the IC's false functionality being shown. These writers also include attacks on logic obfuscation circuits in their article. Having the secret key allows the attacker to deduce the properties of the logic obfuscation. Once the attacker has the secret key, they will be able to decipher the obfuscated gate netlist and replicate the ICs. This paper proposes an approach wherein the adversary applies several input patterns and, by observing the resulting output patterns, learns the secret key. This attack requires both a working IC and the netlist of the obfuscated version. Therefore, logic obfuscation is used to safeguard the key, where key sensitization is accomplished by introducing a gate within the ICs. The key gates are placed in the network in such a way that key value propagation is only allowed under certain forced input conditions. For this reason, brute force is required for an attacker to determine the correct key values. The study detailed a wide variety of assault methods for a variety of setups. Cascade connections between configuration key gates make up the run of key gates. Therefore, the keyspace is diminished and the attack probability is raised thanks to these run-key gates. Once the attacker has reduced the number of key bits, they need to replay the run key gate using single hatred. An attacker can deduce the entire key sequence by just knowing the value of a single key gate. When there is no way for one key gate to communicate with another, the result is a setup known as an isolated key gate. This arrangement is favoured by attackers because it allows them to discover the keys' individual propagation patterns and so obtain the proper ones. Dominating key gates refer to a setup in which two key gates are present but only one can be reached from the output. As a result, attackers favour designs where they only need to know the key bit of one gate in order to extract the entire right pattern, a tactic known as the "golden key pattern." Therefore, authors have proposed Strong Logical Obfuscation, the insertion of keygates in the ICs with complicated interferences between the key gates, as a means of shielding the ICs from these various attack strategies. An interference graph is constructed after the authors insert the necessary gates. Key gates are represented by nodes in the graph, with single nodes standing in for cascaded key gates and isolated nodes for standalone key gates. The ISCAS'85 benchmark and the Atalanta testing tool were utilised to evaluate the suggested method by determining the input pattern for the propagating key gates. The circuits' overheads in terms of size, latency, and power are obtained using a Cadence RTL compiler. First, a key graph is built with only 10% of the possible key gates added. From there, the keys can emerge as either dominating, isolated, or convergent nodes. Even if the number of substituted gates is only 5% of the total number of gates in the original circuit, the results reveal that the area overhead is significant. When using a weighted insertion, you should expect an overhead of up to 25%, while using an unweighted insertion will only cost an average of 21%. If changeable gates are introduced, the attacker can discover the proper key in a matter of seconds, as was the result. so that mutable gates can be added to the circuit and it will become modifiable. In addition, IC design aids designers and testers in visualising the design through the manipulation of inputs and monitoring of outputs. Logical obfuscation can be compromised by an attacker, but it can be strengthened by the defence through circuit testing.

As the number of ICs in use grows, it becomes harder to prevent their functionality and IPs from being copied, overbuilt, or reverse engineered. These counterfeit ICs have a negative effect on systems and operations, in addition to harming research and the reputation of associated brands [51] (2012). As a result, several other methods have been proposed to prevent the ICs from being stolen, including as reverse engineering, logic obfuscation, and IC camouflaging. The authors suggested obfuscation methods in which a modified netlist would be used in place of the original. When the right key is used, the performance of an encrypted netlist is identical to that of an unencrypted one. Therefore, the authors proposed an obfuscation structure consisting of a Multiplexer and an inverter, with the select input of the Multiplexer serving as the key. Such a structure is constructed to allow the IC designer to safely unlock each IC. Attackers employ a wide variety of RE tools and techniques to decipher the encrypted source code and recreate the original design of the stolen ICs. In order to change the netlist and run simulations, attackers must first acquire the obfuscated netlist and then conduct extraction-based IC Reverse Engineering. And you can legally buy the ICs with the unlocks in stores. Therefore, the suggested obfuscation cell (OC) can be integrated into inverter ICs by simply swapping out the wires. Because the keys in OCs are so dynamic, attackers can only guess at the OCs' performance even if they know their structure. The obfuscated circuits rely heavily on the XOR and XNOR gates.

For XOR and XNOR, the critical bits would be 0 and 1, respectively. Given this, it is clear that an attacker may perform extraction-based reverse engineering to determine the right obfuscated gate netlist by inserting key bits. This can be avoided by switching to inverters and XNOR gates instead of XOR gates for security, but this requires additional resources in the form of power and space. However, the suggested construction has low overhead and doesn't necessitate any such alterations. Even if an adversary uses extraction-based Reverse Engineering to decipher the gate-level netlist, the OC structure will keep sensitive information safe. To obtain the important information they have to employ brute forces.

To safeguard the IC from attackers, the writers configured the designed OCs to communicate with the PUFs; nevertheless, only the designer can utilise the IC's licence to turn it on. As the IC is being powered on the licence will be XOR with the PUF answer so that the key bits which are correct for OCs can be created, after that all these key bits configurations are saved in the flip-flops to unlock the IC. If the attacker tries to recover the obfuscated gate-level netlist by performing Reverse Engineering, not be able to derive the original key bit configuration of the proposed OCs. ISCAS benchmark circuits were utilised for this proposed architecture, and synthesis was performed with the 45nm Nangate open cell library in conjunction with synopsis dc. Up to 5% of cells were disguised, and OCs were used to replace 50% of the inverters in the ISCAS reference design. So, not only does the suggested obfuscated technology offer reduced power and area overheads, but it can also withstand Reverse Engineering methods based on extraction. Experimental findings indicated this technique sustains average area overhead and power overhead up to 0.63% and 2.6% respectively.

Security risks like IP piracy, overbuilding, SAT attacks, reverse engineering, modelling attack, and many others have grown in tandem with the expansion of integrated circuits. In order to make the IC completely impervious to any kind of attack, numerous different methods, such as logic locking, have been put into practise in recent years. Many different methods of logic locking were discussed, and their relative merits were weighed and compared in this work.

Since the cost of producing integrated circuits (ICs) continues to rise, many companies are switching to fabless fabrications, which makes them more vulnerable to outside attacks. Consequently, numerous strategies have been developed to protect against security risks such as trojan insertion, overbuilding, reverse engineering, intellectual property theft, etc., including PUF s, watermarking, IC camouflaging, and logic locking. The flexibility and effectiveness of logic locking are attracting the attention of the research community. Logic locking can prevent attackers from gaining access to ICs at any point in the supply chain. To prevent unauthorised use, logic gates are implanted into the chip using logic locking techniques. The locked circuit retains the original inputs and is powered by the tamper-proof memory alone, but nobody else can access the chip's original functionality. XOR gates and LUTs (lookup tables) are two examples of these supplementary logics. At first, methods were employed to determine where in a circuit to insert locks, such as in interference-based or fault-analysis-based logic locking. However, as time went on, attacks were launched that could retrieve the keys, leaving the logic locking methodology vulnerable. It is discovered that the biggest problem for logic locking methods is attacks that rely on Boolean Satisfiability. Logic locking that can withstand attacks from SAT stacks is now the focus of much study, and solutions like SARLock and AntiSAT are posing a serious challenge to SAT attackers. This paper provides a taxonomy of the Logic Locking method. This method can be broken down into two subcategories: Traditional Logic Locking and SAT attack resistant Logic Locking. Strong logic locking and logic locking based on fault analysis are examples of conventional logic locking techniques in which algorithms are constructed to determine where to introduce the key gates. SAT attacks, however, pose a threat to all SAT solver-based SAT assaults, hence a new technique called SAT attack robust logic locking has emerged to counter them. The authors also classified the many kinds of assaults that can be used to circumvent logic locking. Attacks based on algorithms probe the vulnerabilities of the target algorithm and use that information to deduce the secret key needed to bypass the locking logic. An attacker performing an approximation attack will typically extract a netlist that is very similar to the original netlist but not an exact match. This approximate netlist may initially generate incorrect results, but when more data is fed into it, it will eventually settle on the proper solution. In other words, the attacker can use this method to gain access to the circuits' functionality. When attempting a structural attack, the hacker first dismantles the protective logic block, if present, and then focuses on isolating the correct netlist's functionality. To compromise a system with the correct key, side-channel attacks alter its physical properties, such as its timing or power supply. To learn about the safety characteristics of the logic locking method, side-channel attacks based on test data are used. Sensitization attacks, in which an attacker analyses the entire netlist, inserts algorithms to sensitise the individual key bits, and then examines the replies they receive, are just one type of algorithmic assault. The SAT assault, using the DIPS pattern. To minimise the attacker's calculation time even more, these DIPs are used as input to the key gates, which then generate outputs and eliminate potentially valid but wrong keys. The attacker in an in-circuit partitioning attack divides the circuit into a series of logical cones, each of which receives a separate round of brute-force attacks. The AppSAT and the DoubleDIP are two types of approximate attacks. Similarly, the three sections of this study that deal with conventional logic locking methods are called Random Logic Locking. Utilisation of robust logic locking and logic locking based on defect analysis. Random logic locking, which involves inserting XOR gates into a specified area and distributing them evenly without interfering with one another, is vulnerable to a broad range of security risks. When the incorrect key is used with Fault analysis-based logic locking, an excessive amount of output-level data corruption occurs. The amount of corruption in the output can be quantified by calculating the Hamming distance between the expected and actual results. Key gates in Strong logic locking are placed in the circuit in a way that causes the most interference, making the method impervious to attacks. This makes the key bits less vulnerable to being compromised. In this study, we examine the effects of attacks on several logic locking methods and make comparisons between them. It was discovered that SAT assaults can circumvent conventional logic locking, and that only the Stripped Functionality logic locking technique can withstand such attacks. As a final point, this paper demonstrates how Logic Locking approaches have progressed through time. At first, researchers concentrated on finding a safe spot to insert the gates, but when these methods have been shown to be sensitive to SAT assaults, they have moved their attention to improving the Logic locking mechanism so that it is immune to all SAT attacks.

The expansion of integrated circuits creates security risks such as IP piracy, overbuilding, SAT assaults, reverse engineering, and modelling attack, among others. In recent years, several approaches such as logic locking have been employed to make integrated circuits resistant to all attacks. In paper [52] Yasin et.al (2017), the authors compare and contrast the numerous logic locking strategies. As the cost of producing integrated circuits (ICs) rises, an increasing number of manufacturers are turning to fabless production, which increases the risk of third-party assaults. There are many significant security threats, such as trojan insertion, overbuilding, reverse engineering, and intellectual property theft, among others. To combat these threats, many approaches, such as PUF s, watermarking, IC camouflage, and logic locking, have been developed. Researchers are interested in logic locking due to its ability to safeguard integrated circuits (ICs) from supply chain-based attacks and its adaptability. In logic locking approaches, logic gates are introduced into the chip to lock the functionality, and in addition to the original inputs, the locked circuit additionally possesses the key inputs, which are driven exclusively by the tamper-proof memory. These additional logics may consist of XOR gates or LUTs, sometimes known as lookup tables. In the early stages of logic locking, algorithms were used to select the area for the insertion of gates, i.e. interference-based logic locking or fault analysis-based logic locking. However, as time has progressed, attacks that can recover the keys have made the; logic locking technique susceptible. It is discovered that Boolean Satisfiability-dependent attacks provide the greatest threat to logic locking approaches. Now, research efforts are concentrated on developing logic locking that is resistant to SAT stacks, and solutions such as SARLock and AntiSAT are becoming a danger to SAT attackers. In this article, the writers classify the Logic Locking method. SAT attack resistant Logic Locking and Traditional Logic Locking are the two subcategories of this approach. In conventional logic locking techniques, algorithms are constructed to select the position for inserting the key gates. These techniques include strong logic locking and logic locking based on fault analysis. But these standard logic lockings are susceptible to SAT attacks, hence a new technique called SAT attack-resistant logic locking is employed. Techniques such as Anti-SAT and SARLock pose a danger to all SAT solver-based SAT assaults. The authors have categorised the attacks that can be used to circumvent logic locking. Based on the vulnerability of the applied algorithm, algorithm-based attacks are able to derive the secret key to unlock the logic locking. In approximation attacks, the attacker extracts a netlist that is similar to the original netlist but not identical. This approximation netlist can only provide incorrect outputs for new inputs; thereafter, it will generate the right output. Thus, by means of this technique, an attacker can gain access to the circuits' functionality. In structural attacks, the attacker circumvents or removes the protective logic block before isolating the original or correct netlist's function. In side-channel attacks, physical channels like as time and power are altered such that the correct key can be easily leaked. To comprehend the security properties of the logic locking method, side-channel attacks based on test data are employed. Algorithmic assaults include a variety of methods, such as the sensitization attack, in which the attacker analyses the entire netlist and inserts algorithms to sensitise the individual key bits, before analysing the output answers. SAT attacks that employ Distinguish Input Pattern (DIP) input patterns. These DIPs are considered inputs of the key gates in order to produce outputs and eliminate repeated wrong keys, hence reducing the computational time of the attacker. In an In-Circuit Partitioning Attack, the attacker divides the entire circuit into logical cones and then applies brute force on each logical cone. Approximate assaults are further divided into two subcategories, namely AppSAT and DoubleDIP. Random Logic Locking approaches are also divided into three parts in this paper with regular logic locking techniques. Strong logic locking and Fault Analysis-based logic locking. Random logic locking inserts XOR gates in a certain area and distributes them uniformly, without interfering with one another. As a result, this technique is susceptible to a variety of security threats. In Fault analysis-based logic locking, when the erroneous key is applied, the output bits are corrupted excessively. And this output corruption can be measured by the difference in Hamming distance between the correct and wrong outputs. In Strong logic locking, the key gates are introduced into the circuit with the greatest possible interference to prevent the sensitization of the key bits, making this technique resistant to attacks. In this work, a comparison is made between the effects of assaults on the various logic locking approaches. Only the Stripped Functionality logic locking technique provides resistance to these SAT assaults, indicating that the SAT attacks are capable of defeating conventionally-based logic locking. As a conclusion, this study demonstrates the evolution of Logic Locking approaches. As these strategies become susceptible to SAT assaults, the researcher's attention eventually shifts to altering the Logic locking technique to make it more resistant to all types of SAT attacks.

In [53] Kamali et.al. (2018) presented a Look Up Table-based netlist obfuscation strategy to safeguard the intellectual property of FPGA bitstream and ASIC netlist from SAT attacks. LUT-based locks are more resistant to SAT assaults since they exponentially increase the SAT solver's execution time in proportion to the number of obfuscated gates. As hardware security has been a key concern in recent years, so is the protection of ASIC from offshore manufacture. Due to the high cost of construction, operation, and maintenance, many businesses are opting for offshore fabrication. However, these offshore fabrications are unreliable due to many security risks, such as reverse engineering, overbuilding, Trojan insertion, etc. At FPGA, it is difficult to safeguard bitstreams in their earliest and distant phases from third parties. Through Reverse Engineering the netlist or the addition of hardware trojans, an attacker is able to send FPGA bitstream to unauthorised third parties. The author demonstrates in this work a LUT lock-based obfuscation netlist that makes it impossible for an attacker to execute SAT assaults to the FPGA bitstream and ASIC. In FPGA solutions, cutting-edge FPGA offers a wide variety of applications that result in a large number of unused LUTs once mapping is performed. So that these unused unmapped LUTs can be used for netlist obfuscation, large-sized LUTs are mapped with two input logic gates, and other inputs are taken from the output of the internally built Physical Unclonable Functions (PUFs). By altering the input pattern dependent on the key inputs, it is possible to increase the likelihood of obfuscated circuits. When obfuscation is performed on ASICs using LUTs, there is an increase in size and delay overhead. In LUT, the area and delay overhead of the memory devices increase exponentially with the input size. Consequently, only a small number of LUTs are used to substitute gates from the netlist. Due to the employment of STT and MTJ-based LUTs, the area overhead has been diminished. Therefore, the LUT-Lock technique introduced in this research protects both the FPGA bitstream and the ASIC from SAT attacks. therefore, the essential characteristics of this are Focusing on the fanin of the fewest number of primary outputs (FIC), focusing on the higher skew gates in FIC (HSC), focusing on the minimum fan-out (MFO-HSC), focusing on the gates with the least influence on primary outputs (MO-HSC), and avoiding back-to-back insertion of LUTs (NB2-MO-HSC). Authors used an Intel Core i5 processor with 8 GB of RAM for experimental setup and benchmarking. They compared the SAT solver's execution time to determine the effectiveness of the algorithm's essential components. In addition, they compared the new LUT-Lock-based obfuscation technique to the standard STT-LUT. As a result, the primary objective of obfuscation was to affect a small number of output pins; however, this increases the difficulty of obfuscation, and gates with minimal controllability are optimal for obfuscation. And the LUT – Lock based on NB2-MO-HSC is more resistant to SAT assaults than others since it lengthens the execution time for SAT attacks.

In [54] S.Patnaik et. al. (2018), As technology advances, it becomes increasingly difficult to secure the intellectual property (IP) of electronic gadgets. Therefore, the approaches employed most frequently to protect these devices include layout camouflage, Logic locking, and encryption. Therefore, the majority of these strategies are used to CMOS integration in order to reduce design overheads by adding additional circuits or making design-level adjustments. Therefore, the authors of this work propose that the Giant Spin Hall Effect (GSHE) serve as a switch to simultaneously activate both strategies, namely logic locking and IC camouflaging. This GSHE is capable of clocking 16 Boolean functions with two potential inputs. As IC camouflaging is a technique in which the layout aspect of the IC is modified so that it is impossible for an attacker to determine the under process and intellectual property of the ICs. Other approaches, like as threshold-dependent camouflage and connectivity obfuscation, are being explored for CMOS integration. In logic locking, key gates are put into ICs, and the performance of the ICs can only be decoded with the correct key. Based on Boolean Satisfiability and Discriminate Input Patterns, attacks that modify camouflage are being launched (DIPs). Other physical assaults, including as fault injection and side-channel attacks, may also be employed to obtain sensitive data from a protected chip. To enhance hardware security, low-power dissipation technologies such as nanowire transistors, spin and carbon-based devices are utilised. Polymorphic gates that can implement different Boolean functions and support logic locking and IC camouflaging are the most prominent aspect of emerging devices. Due to their uniform layout, it is difficult to determine their performance using optical image-based Reverse Engineering, and their functionality is dependent on the key inputs, which are also their controlling pins. In this article, the GSHE switch consists of a heavy metal-based spin hall layer, read and write nanomagnets that exhibit dipolar coupling, and two fixed antiparallel ferromagnetic layers on the read nanomagnets. When current is delivered to the bottom layer, transverse spin buildup is produced. The performance of the GSHE switch is determined by nanomagnets. In accordance with the Landau-Lifshitz-Gilbert-Slonczewski equation, the GSHE dissipates up to 0.2125 uW of power. As a result, these GSHE switches are more resistant to electron microscopy and optical imaging-based Reverse Engineering and are smaller than CMOS devices. By adding polymorphic gates to GSHE switches, runtime polymorphism can be accomplished. As a result, the performance of the chip will no longer be static; it will become dynamic, and SAT attackers will be unable to reverse engineer the dynamic functionality and features of the semiconductor. As CMOS devices release photons during operation, they are susceptible to side-channel assaults, but GSHE emits no photons and is therefore also resistant to readout-based attacks. Consequently, these GSHE-based polymorphic devices offer logic locking and camouflage techniques and are resistant to a variety of attacks, such as side-channel attacks and analytical attack.

Obfuscation becomes a crucial approach for protecting devices from attacks like as overbuilding, SAT attacks, and reverse engineering. The [55] Sengupta, A, et.al. (2018) applies an obfuscation procedure to Digital Signal Processing (DSP) cores with encryption and logic locking such that it will incur less overhead, i.e. design overhead, and as compared to state-of-the-art approaches, the design cost can also be lowered. The authors of this study suggest a PSO-DSE (particle swarm optimization is driven space exploration system) to lower the cost of designing DSP core designs. Here, logic locking is implemented utilising logic blocks to strengthen the locking of bits of dad output of functional resources. As a result of the rising expense of maintaining, managing, and operating circuits as a result of technological advancement, fabless manufacturing is on the rise. These fabless design firms are also associated with third companies that pose a threat to the security of the circuits. These threats include IP detection, overbuilding, and trojan insertion. Functional Obfuscation is the most recent strategy introduced to thwart attacks, albeit there are many others. The purpose of this functional obfuscation, also known as Functional Locking, is to incorporate locking components inside the circuits that require a valid key before the right output can be extracted. This logic locking is comprised of MUXs, XOR/XNOR gates, and OR gates. The inability of this technique to integrate multi-pair-wise securities is its restriction. And unlike the proposed method, this strategy does not contain the mechanism of the obfuscated design, i.e. to generate optimal functionality for the obfuscated design, nor does it target the digital signal processing cores. In this research, they offer an IP functional Locking block (ILB) that is used to obfuscate digital signal processing cores and make them resistant to key sensitization assaults through multi-pair-wise security.

Numerous security analyses have been performed on the suggested functional block technique to determine its resistance to assaults. Strong ILBs characteristics include correct key spacing, prevention of key gate seclusion, and multi-pair-wise security. If an attacker cannot sensitise a key bit without adjusting the other key bits, then the key bits are pair-wise secure. In multi-pair security, no key bit can be modified to obtain the output without adjusting other key bits, hence making this security mechanism resistant to attackers. The isolated key gates, those not connected to the other gates, are vulnerable to the numerous attacks. However, the suggested ILBs are interconnected and interdependent on the key inputs, preventing the creation of an isolation gate in the circuit. This security technique is known as key gate seclusion prevention. Therefore, adjacently connected gates can sometimes be substituted by a single gate, making the circuit susceptible to attacks. However, the proposed ILBs are resistant to such attacks due to the intricate interleaving of gates, which prevents key runs. Muting is a technique by which attackers can influence the inputs between two gates; however, with the proposed ILBs, the security of each key input is bolstered by the other seven key inputs, and there are no controlled inputs, making it more resistant to attacks. In this research, many attack scenarios against resilience are also presented. As demonstrated, circuits including both mutable and isolated key gates are vulnerable to attack, but the proposed ILBs do not contain either type of gate and are hence resistant to key sensitization assaults due to their multi-pair-wise security characteristic. The primary objective of the attacker is to make a profit by reselling the Intellectual Properties; therefore, the attacker must unlock the locked IP in order to determine its functionality. In order to do so, the attacker inserts a hardware trojan in a secure location so that the performance of the IC and IP can be decoded. However, in the case of the Proposed ILBs, this approach becomes extremely difficult due to its resistance to both the Trojan Insertion attack and IP piracy. SAT assaults are not viable since the DSP core consists of numerous multipliers and adders, resulting in a massive CNF. Therefore, this proposed technique is also resistant to the SAT solver employed in SAT attacks. For this execution, the author used a 4 GB Intel Core i5 3210M CPU to execute Java, which is needed to implement this method. PSODSE is integrated with the proposed method in order to achieve the cheap cost. Due to the addition of ILBs, this strategy is hampered by a rise in overhead costs and a lengthening of critical path delays.

Logic obfuscation plays a significant part in the hardware security industry in order to protect Intellectual Properties(IPs) and the performance of Integrated Circuits (ICs). Recently, SAT attacks have been susceptible to a variety of obfuscation methods. To strengthen the resistance of the circuits against SAT attacks, Look Up Tables (LUTs) are used as a new approach. In this study [56] Kolhe, G et.al. (2019), employed Custom LUT based on logical encryption. LUT+MUX and LUT+LUT-based obfuscation are the two customised LUTs used to secure the design by MUX routing obfuscation and to reinforce the circuit with additional logic and routing concealment respectively. Many IC businesses are going Fabless to reduce production, maintenance, and operating costs, which can increase the danger of third-party attacks on the designs. To prevent these security threats, numerous solutions such as PUFs, IC camouflage, split manufacturing, logic locking, etc. are employed. As it requires the proper key to determine the performance of the original circuits, logic locking obfuscation aids in preventing a huge variety of attacks. So a variety of logic locking approaches are being developed to combat SAT assaults. And the author of this research proposes custom-based LUTs, namely LUT+LUT and LUT+MUX obfuscation. Because these Look-Up Tables are crucial programmable logics that offer strong resilience to SAT attacks, but require additional power and space. The important points the authors utilised for custom-based LUTs are that they cut overheads without compromising security. The STT is utilised for concealment. By replacing the number of gates with LUTs, the content is kept in tamper-proof memory, and only the IP holders are aware of the IC's original performance. Therefore, it becomes challenging for attackers to perform Reverse Engineering on the IC. As the size of the LUT is raised, the complexity of the obfuscation circuit's performance likewise increases. Here STT-LUT is another type of custom LUT proposed by the authors. LUT+MUX-based obfuscation is utilised for CMOS devices that make less use of non-volatile parts, whereas LUT+LUT-based obfuscation is used exclusively for developing technology. During LUT+MUX obfuscation, MUX are put into the circuit alongside LUTs in order to increase the security of the circuits; in this article, the size of MUX is limited to 2 in order to keep PPA overheads to a minimal. As LUTs are totally changeable, MUXs increase the likelihood that LUTs will be altered. In LUT+LUT obfuscation, LUTs are coupled in a cascade configuration to increase the circuit's complexity. Here, the LUTs are incorporated into the circuit by substituting the input gates with small LUTs, hence increasing the reconfigurability of the LUTs. The authors utilised a cluster computing environment with 1060 cores, 3TB of memory, and 50 nodes for the experimental setup. To execute the SAT solver, they allocated 128 GB. Consequently, it was determined that both obfuscation techniques are resistant to SAT assaults. If the attacker wishes to determine the right functioning, he or she cannot remove the LUT and MUX from the circuit. Eliminating them will also eliminate the functionality of the entire IC. Experiment findings demonstrated that the bespoke LUT is extremely resistant to SAT attacks and protects circuits. LUT+MUX-based obfuscation is more secure than LUT+LUT since it employs fewer non-volatile elements. In contrast, LUT+LUT-based obfuscation helps reduce power and space overheads. Consequently, there is a reduction of 3 times in power overhead and 8 times in area overhead. Consequently, LUT-based obfuscation is effective against SAT attacks.

Providing hardware security for the performance of ICs and their intellectual property is getting increasingly complex as technology and the internet continue to develop at a rapid rate. Numerous approaches, such as PUFs, Logical Obfuscation, Split Manufacturing, and IC Camouflaging, have been developed. Toggle Count based Logical Obfuscation of the IC [57] A.Kumar et.al. (2019), where obfuscated cells are introduced in the circuit based on the controllability and toggle count of the node. In this manner, when the incorrect key is entered, the output is garbled. Similar to logic obfuscation, key gates are added to ICs to conceal their functioning. Further, logic obfuscation can be combinational or sequential. In sequential circuits, key gates are inserted into a state transition graph, but the functioning of heygates is concealed in combinational circuits. The proposed approach is intended to conceal the design of obfuscated cells (OCs). Two inverters coupled to two NAND gates and then connected to an OR gate were utilised in the design of the OC, and its performance is comparable to that of the XOR/XNOR used with the decoder. To implant these OCs, a switch count process is activated. Nodes with a high toggle count are more sensitive and can be influenced by any change in the nodes that follow them. Thus, obfuscated circuits are introduced exclusively in these nodes. The controllability of additional factors also plays an essential part in the installation of OCs. Low-controllability nodes are preferred for the insertion of obscured cells. The ISCAS'85 benchmark is included in the experimental setup for the suggested methodology, and Python is used to calculate the toggle count and controllability. Python code is created by utilising the Hamming distance of the obfuscated netlist. With 90nm technology, the Synopsys dc design compiler is utilised to calculate area and power. Calculating Hamming distance by inserting random input patterns into the obfuscated netlist. As a consequence, the overhead switching count-based obfuscation differs from the randomly-based insertion method. As attackers employ brute-force tactics to determine the right key for disguised ICs. However, the complexity of the circuit based on the toggle count and controllability method makes the design resistant to brute force attacks by exponentially increasing the attack time. As in this work, the examination of the suggested technology is performed on each of the benchmark's five distinct circuits, and observations are made by comparing them. Consequently, comparison results demonstrate that toggle-based obfuscation yields a greater hamming distance than the standard random-based insertion strategy. The key gates inserted result in a 50% hammering distance. Incorporating a greater number of key gates, however, can increase the amount of overhead space; this is the most significant drawback of the proposed strategy.

As logic lock is a common strategy against intellectual property theft and counterfeiting, it is susceptible to numerous attacks, such as Boolean-based satisfiability attacks. Modern techniques for logic locking therefore struggle against SAT assaults. Logic locking aids in disguising functionality from attackers by inserting key gates into ICs. Therefore, untrustworthy foundries are unable to extract the chip's performance, as the chip will only operate successfully when the correct key is applied. Therefore, an attacker can be anyone in the supply chain, including third parties or post-designing houses, and have access to the following: attackers obtain the functioning IC from the market and use it as a black box to obtain the input-output pairs, also known as an oracle. After using reverse engineering to the GDS file, the un-trusted founder can also obtain a gate-level netlist that has been locked. SAT attacks are the most potent attacks capable of defending any existing scheme type. These SAT assaults employ SAT solvers to seek Distinguished Input Patterns (DIPs), which aid in determining the correct input patterns. ICs can also be encrypted utilising weak key gate positions that leave them vulnerable to numerous assaults. This research [58] J.Rajendran, el.al. (2012), demonstrates evolution strategies such as logic locking techniques. Key gate location-based strategies either enhance the interdependence of keygates or raise the output's susceptibility to corruption for every incorrect input. SAT attack blocking logic locking approach exponentially increases the number of DIPs required to obtain the correct key. This technique's primary shortcoming is that less output corruption occurs when the incorrect key is applied. However, some of the strategies are adequate against attacks such as double DIPs, AppSAT, bypass, and many others. By resisting the functional scan-out reactions, some approaches are resistant to the attacks. In this method, scan cells that contain the values of the previous iteration's test answer are replaced with secure cells so that no key bits can be captured in the secure cells. By combining MUX and flip-flops, a new method of logic locking has been proposed here. Here, the inputs of MUX are connected to the flip-flops' two outputs. Here, the MUX acts as an inserted key gate, with the select lines of the MUX serving as the keys. To obtain the correct key, one must therefore apply the input patterns to the MUX selection line. Due to the insufficient number of key gates at the outputs, this logic-based encryption method is occasionally vulnerable to attack. In order to prevent this, these key gate positions are constrained by several outputs. In this work, SAT assaults are also the subject of a security examination. Consequently, the use of SAT assaults involves a number of distinct phases. In combinational circuits, these flip flops function as pseudo outputs and inputs, whereas gaining access to the scan chain enables an attacker to quickly access the outputs of the flip flops and determine the information stored within them. During scan operation, flip flops are a part of the scan chain and also contribute to the obfuscation of the scan data. This scan data undergoes many inversions during scan shifting, but only after depending on the secret keys. The scan shift process embeds information about the keys into the scan in-scan out modified vectors, which can be attacked by very advanced ScanSAT-based SAT attacks, resulting in the deobfuscation of the obfus So writers offered key-based procedures that provide access, i.e. scan access, only to the trusted designer or engineer who has access to the valid key. To protect the circuit, one of the MUX's inputs acts as an obfuscated key, while other keys function as pseudo-random number generators. The chip test engineer has now applied a valid test key that is identical to the obfuscated key. This key was utilised to retrieve the original responses from the obfuscated scan input and output responses. In this work, experiments are conducted using ISCAS'89 benchmarks, and algorithms are constructed using C code. The Intel Core i5-3470 processor and 4GB of RAM were used to accomplish the encryption time of the flip flop technique. Here, the area and power overheads of common logic locking approaches have been compared. As a result, it has been shown that the area overhead of the encryption flip-flop technique is greater than that of previous techniques, although none of them can prevent SAT attacks. The result of this work is that they proposed the encryption flip-flop approach in which the MUX inserted in flip-flops encrypts the outputs. The output values of this approach are corrupted when the incorrect input keys are used. And this strategy is useful for preventing modern-day attacks.

Due to the expensive cost of building, manufacturing, maintaining, and running ICs, many businesses are going fabless; as a result, third parties are exposing the performance of the

ICs as well as the product, which will increase security concerns. Therefore, the risks of overbuilding, reverse engineering, and SAT attacks always exist. Many hardware security approaches, including as Physical Unclonable Functions (PUFs), logical obfuscation, IC camouflage, and watermarking, are employed to counteract such attacks. In [59] M.M.Shihab et.al (2019), the authors present an alternative design obfuscation in which the functions are not placed within essential protected functions but rather are missing from the produced silicon and can be reprogrammed post-manufacturing. This also results in lower performance, power, and area overheads than LUT-based FPGAs, hence increasing the difficulty for SAT attackers. The primary contributions of this paper are the Trap-based design obfuscation method and the proposed fabric of transistor-level programmable for logic obfuscation. Numerous security analyses, such as brute force attacks, have been performed by the SAT attacker on this TRAP obfuscated design. As a result, this TRAP obfuscated design is resistant to SAT attacks. Fab Scalar microprocessor cores analyse the effectiveness of the TRAP obfuscation design. Additionally, security evaluations are performed on the proposed technique to evaluate its resistance to assaults. The objective of brute force attacks is to decrypt the design by employing all possible key combinations. The number of combinations used in a brute-force attack reveals its efficacy. Due to the omission of these obscured designs from the layouts, TRAP can be immune to conventional attacks. Therefore, SAT attacks necessitate the use of the CNF formula to attack all Boolean functions that target fabric execution. TRAP-based logic obfuscation achieves resistance against SAT assaults, albeit at the expense of power, performance, and area overhead. Here, it is demonstrated that the area overhead for Virtex 5 and TRAP-1L is normalized to the full – ASIC execution area of 65nm, and that the area overhead is favorable compared to FPGAs. Additionally, the latency overhead is being standardized to that of the full-ASIC Fab scalar execution. Moreover, the latency overhead is less than that of FPGAs. Power overhead for entire ASIC Fab scalar operations is always lower. Unlike other techniques, TRAP-based obfuscation does not require a key to de-embed the obfuscated performance of huge functions. Also, it is seen that the differences between FPGA and TRAP bit stream are minimal. The secret key can be kept both internally in tamper-resistant memory and outside using externally applied power modes. As a result, it has been determined that the suggested transistor-level programming-based obfuscated design can prevent un-authorized access to the design, but incurs overheads in terms of space, performance, and power. And by integrating the TRAP-based obfuscated design with ASICs, the design can become complicated and impenetrable to SAT attacks.

With the rise of risks such as reverse engineering, overbuilding, IP piracy, and modelling attacks, hardware security has become crucial. Physical Unclonable Functions (PUFs) play a major role in the security of integrated circuits (ICs). This technology helps resist many attacks and cloning attempts of the design. This PUF serves as a fingerprint for specific things. There are numerous types of PUFs in use, including SRAM PUFs, Arbiter PUFs, and ring oscillator PUFs. In [60] S.Batabyal, et.al.(2019), the authors present a ring oscillator-based architecture with enhanced CRP, or challenge-response pair. This method is more reliable because it is dependent on relative values and not absolute values. RO-frequency PUF's oscillations are dependent on the interconnects as well as the gate delays and are composed of cascaded inverters. The RO-PUF falls under the weak PUFs and possesses the challenging response bit features. This challenge-response bit helps generate the key bits. Consequently, as the number of challenge-response pairs increases, it becomes more difficult to collect or generate the Key bits, making this an active field of research as of late. Due to the delays of the gate, if an attacker attempts to decode one of these PUFs, they will receive various responses even for the identical challenges, making it resistant to attacks; consequently, the PUF is utilised for IC authentications. In addition to being dependable in situations like as supply voltage, temperature, and environmental noise, PUF circuits are also capable of maintaining their responses. The proposed RO-PUF can be implemented on both FPGA and ASIC, which is their primary benefit. For the execution of PUFs, digital circuits such as gates, decoders, and MUXs are utilised to save a great deal of space and energy. Due to the ageing of the components used in the design of PUF, the biassing points change, resulting in incorrect IC authentication and key changes. Due to environmental differences, the PUF replies also change, making Machine Learning-based attacks more susceptible to knowing the authentication of the ICs. Uniformity, defined as a uniform distribution of 0's and 1's in the response pattern, and uniqueness, defined as the capacity of the physical unclonable functions to generate random replies and with those random responses able to identify the chips, are two acceptable criteria for the PUF. The proposed modal has two phases in this work. In the first phase, the maximum frequency of the ROs is stored, and in the second phase, the selection bits of the MUXs are used as the comparison challenge bits when two ROs are engaged for comparison. In this article, the Xilinx Spartan 3E FPGA, the Spartan 6 board, and Xilinx ISE design Suite 14.5 are used to implement the suggested RO PUF. Various experimental circumstances, primarily Frequency Measurement, are utilised to analyse the PUF response. It was discovered that Spartan 6E has a maximum frequency of 114 MHz and a maximum average frequency of 104 MHz for its hardware. To determine the PUF's dependability, calculations are performed at room temperature. As a consequence, the PUF's reliability is found to be 99.2%, and the entire design uses fewer FPGA resources than typical RO PUFs. In conclusion, the proposed RO PUF had more difficult responses than standard approaches, and the uniqueness of the design increased both the response bits and the reliability. Also here, the frequency has been measured to determine the effect of ambient circumstances. As a result, the greatest frequency of the Spartan 6E, 114 MHz, can be expressed as 27 bits, causing an increase in the number of devices requiring authentication. Due to the execution of this process, expenses will also be reduced.

As the active participation of external organizations in the manufacturing of numerous products has raised numerous hardware security issues such as overbuilding, SAT attacks, etc., the goal is to protect the products from un-authorized access to their functionality using logic encryption in which key gates are inserted into the circuit in such a way that the performance of the circuit can only be detected by applying the correct key. Therefore, the objective of the designers is to ensure that neither the original circuit nor the proper key pattern can be revealed to various attackers. In [61] A.Rezaei et.al. (2020), propose bilateral logic encryption, whereby great security can be obtained with a minimal circuit, and the robustness of the circuit is evaluated using a variety of relevant and precise attacks. As SAT assaults are capable of thwarting all conventional methods of Logic Encryption, this attacker employs two copies of the encrypted circuit with different key values, but the same input, under various circumstances. Moreover, these input patterns are referred to as Differentiating Input Patterns (DIPs). The primary objective of these SAT assaults is to remove as many incorrect keys as possible. In the post-SAT era, logic encryption consists of two components: logical circuit obfuscation and logic locking. The primary objective of Logic Locking is to conceal the right keys so that they cannot be easily extracted by encryption. And Logical Obfuscation is implemented so that structural analysis cannot be used to crack the true structure of the circuit. Therefore, the proposed strategy is to employ the locking mechanism with minimal logic obscurity. In addition, this proposed method proposes logic locking techniques and better versions of SAT attacks. And very little logical obfuscation is utilised in order to maintain low overhead costs. Standard and reduced functionality logic locking have also been implemented, although introducing differing logic explicitly might potentially lead to structural vulnerability. To encrypt the circuit, the logics have been XORed with the original circuits, exposing the circuit to piracy; hence, minimal obfuscation is performed to prevent attacks on the logic-locked circuit. Therefore, the suggested Bilateral Logic Obfuscation comprises four major components: extraction, logic locking, obfuscation, and concatenations. The sensitive component of the circuit is extracted as sub-circuits with a low corruptibility ratio during extraction. And if the principal output of the original circuit is located between the subcircuit, the ratio of corruption will become 1. When the incorrect input is supplied to a subcircuit, that input will act as a don't care condition, and when the proper key is pressed, the inputs of the original circuits will act as a fan in signal. To lock, the stripped or normal functionality locking is applied to the subcircuit with a high Error Number and Logic Complexity. obscuring the locked circuit Signal Routing is activated, the key size is decreased, and obfuscation is applied to the gates that are connected to the inserted lock's sub circuits. Consequently, logic locking components and subcircuits will become structurally intertwined. Currently, the subcircuit is joined to the original circuit. The developers of this proposed method utilized the ISCAS'85 benchmark. Additional gates are forcibly added to the fan out in order to reach the sensitive component's outputs. The outcome of the SAT attack on the benchmark circuit revealed that only small-sized circuits can be decoded. Additionally, AppSAT assaults are executed, resulting in a minimum of five AppSAT attacks. On the other side, each benchmark was secured with the same key size, and re-synthesis was performed using the ABC synthesis tool. Consequently, an entire logic-encrypted design is safe from SAT attacks. With the lateral Logic Encryption approach, it has been discovered that circuits can be safeguarded with minimal circuit alteration.

As reverse engineering becomes commonplace for determining the functionality of circuits, numerous approaches such as IC camouflage and logical obfuscation are employed to safeguard ICs against piracy, trojan insertion, and overbuilding. In [62] B.Park et.al. (2020), the author proposes an enhanced threshold voltage (E-TVD) logic family that helps conceal circuit performance. In comparison to typical TVD logic families, the number of cascaded transistors between the supply and ground points is reduced in the ETVD logic family. To secure the performance of ICs from all attacks, such as reverse engineering and SAT attacks, various logic families with threshold voltages are now employed. The design or structure of these devices is the same for a variety of functionalities, making it difficult for an attacker to determine which functionality is being attacked. These devices require an external voltage to regulate the switching mechanism, resulting in PVT changes that render the ICs inoperable. Incorporating TVD logic families with differential circuits that directly rely on threshold voltages will result in less PVT variations. The problem with TVD logic families is that the number of transistors depends directly on the input. As the number of inputs increases, so does the number of transistors. Therefore, the authors of this study presented the Enhanced TVD logic family to overcome all limitations of traditional TVD in terms of area, power, and delay consumptions. In the typical 2-input TVD logic family, the sense amplifier input pairs are replaced by parallel pull-down transistors. And working consists of two phases: the pre-charge period and the evaluation phase, during which the clock is low and then high, respectively. However, as the number of input patterns increases, so does the number of transistors, which makes the circuit resilient to PVT changes. Instead of stacking the transistors, the suggested ETVD has a parallel architecture. When the input pattern is applied to such a setup, just one transistor on both differential sides is activated. When the threshold voltages of these transistors are set for particular input combinations, such as 00 and 11, input combinations such as 10, 01 malfunction. In order to tackle this issue, transmission gates that function as XOR logic and give an alternative current path through the differential sides to detect the 01 and 10 input patterns are being developed. The parallel structure of the ETVD logic families simplifies the entire circuitry. In addition, the number of TVD transistors within the circuitry is diminished. In order to compare this suggested ETVD logic family with traditional TVD, a 65 nm CMOS experimental setup was utilised. For simulation, they utilised a supply voltage of 1.2 volts and a temperature of 270 degrees Celsius at 100 megahertz. Due to the reduction in the number of TVD transistors, the ETVD has a 13% smaller footprint in the 2-input logic family, a 34% smaller footprint in the 3-input logic family, and an 80% smaller footprint in the 4-bit input logic family. Maximum delay deviation is 4.9%, while maximum current consumption is 4%. In comparison to conventional TBVD logic families, the ETVD logic family exhibits a substantial current and delay divergence. Furthermore, both E-TVD and TVD logic gates are resistant against reverse engineering and side-channel attacks. Consequently, the authors determined that the ETVD family of multi-input logic gates is more significant. And the suggested ETVD is exceptionally resistant to reverse engineering.

As reverse engineering will rise with the proliferation of ICs, several strategies are employed to protect the Intellectual Properties (IPs) and functioning. The camouflage of ICs also plays a crucial role in defending them from attack. The [63] V.Patil, et.al. (2020), proposes multi-threshold FinFET transistors to facilitate the camouflage of logic cells. Consequently, the camouflaged structures increase the complexity of the ICs' construction, making it difficult for attackers to determine the ICs' performance. Also, due to the high cost of maintaining, controlling, and running, the majority of enterprises will be fabless, which increases the risk of third-party functionality and IC piracy. Numerous potential techniques include tamper-proofing and watermarking. With camouflaged cells, logical obfuscation techniques are also employed, resulting in an increase in complexity. However, these strategies are all susceptible to reverse engineering attacks. In order to counteract this susceptibility, camouflaged structures employ intrinsic properties such as threshold voltage transistors. These properties of the devices are impervious to image processing techniques and reverse engineering, and they give a high level of resistance against attacks. In this study, the authors suggest an n-input gate design that employs multi-threshold voltage transistors and is manufactured using the 16-nanometer FinFET process. Using a 2,3,4 input gate design, they demonstrated that it is possible to create huge functions using the same structure but transistors with various threshold voltages. In this article, various surveys were conducted to evaluate the trustworthiness and hardware security, as well as the logic obfuscation. Logical locking can only be decoded at the system level if the correct key is applied; otherwise, it cannot be. In circuitlevel, the purpose of obfuscation is to conceal the performance of the circuit by introducing camouflaged gates; however, this technique can also result in an increase in power consumption and area overheads. Therefore, a modification is required to make these tactics impervious to attack. Therefore, the primary objective of this study is to build an n-input gate capable of executing several function realisations. This author also explored a 16nm PDK with 0.8V nominal voltage to enable the four threshold voltage option. Here, they obtained stable logic functions for gates with 2, 3, and 4 inputs. And a simple inverter was utilised for the synthesis. The authors also analysed SAT solvers over the design's obfuscation. Therefore, they assumed that the IC manufacturer is a trustworthy vendor, and that the attackers are third parties who wish to perform reverse engineering on the IC in order to determine its functionality and Intellectual Property, as well as extract the physical information using an image processing extraction technique. In addition to identifying the disguised gates and their size, the attacker was also able to identify the inverters. The sole limitation is that the attacker is unaware of the logic function provided in the cell library. Therefore, an attacker will now use SAT solver approach to decipher the functionality. The attacker modelled n-input camouflaged cells using a SAT-based DEobfuscation approach guided by Oracle. However, because the disguised structure has a wide variety of functionalities, the attacker must consider all conceivable possibilities when programming the vector bit. The authors employed the ISCAS-85 benchmark circuit and a 16nm standard cell library for the synthesis process. In order to acquire de-obfuscation, it was believed that up to 25% of the gates have been obfuscated to the attackers and can be deciphered using SAT assaults alone. In addition, there are some logic functions whose outputs can be determined exclusively by the input subsets. Consequently, it raises the difficulty for SAT solvers. Consequently, they studied all of these in three circuits: c499, c432, and 2670. They explored obfuscating the entire design by incorporating disguised dummy input gates. Therefore, 5% more dummy gates were introduced to the circuits than were present in the original circuits. Then, with dummy input gates inserted, they produced up to ten netlists, followed by a 24-hour SAT attack. As a result, the addition of dummy connections increased the effort required for SAT solvers. Thus, the conclusion is that camouflaged gates suffer space and time costs, and also increase the effort required by reverse engineers and SAT solvers.

The authors of [64] C.Cheng and K. Parhi (2006), implemented DCT with a new approach based on the reformulation of N-DCT into two cyclic convolutions. Additionally, the systolic design for these cyclic convolutions was proposed here. According to their proposed methodology, it demonstrates that 2N registers can protect up to (N-1)/2 multipliers. According to their method, this cyclic algorithm retains all the advantages of VLSI algorithms that are suitable for it. To develop their proposed method, they substituted 13 for N. The execution of this thirteenth DCT requires six clock cycles, six multipliers, twenty-six registers, and eighteen adders, correspondingly. Therefore, the proposed design conserves up to six multipliers, four adders, and twenty-six adders. The authors also compared the multipliers, adders, and registers of the proposed system with those of existing designs. In addition, a comparison chart is developed to illustrate how their proposed methodology is superior to the alternatives in terms of multiplication, addition, and registers. The author drew the conclusion that this topology has less hardware complications and is also inexpensive. This structure also has a high throughput rate and low input-to-output costs.

As is well known, cyclic convolution is currently having a significant impact on signal processing. Moreover, this cyclic convolution is predominantly utilised in systolic systems. However, the complexity of the design rises as its length increases. In addition to increased hardware costs and a slower executed design, the price of hardware also rises. Consequently, the authors of this study presented a novel cyclic convolution method that is not only hardware efficient, but also gives a rapid speed to the algorithm comprising of the convolution of huge or lengthy designs. In [65] Cheng, C., & Parhi, K. K (2005), propose technique on the DCT prime length architecture. Here, the procedure of cyclic convolution is performed on two layers. They factorised the matrix into numerous pieces using the matrix factorization procedure. And this factorization is performed when both the cyclic convolution's size and factorization level are large. As the size of the cyclic convolution becomes excessively huge, the second level involves measuring the short cyclic convolution and constructing processing elements based on their results. In addition, a comparison chart is developed to illustrate how their proposed methodology is superior to the alternatives in terms of multiplication, addition, and registers. Therefore, if the length of the cyclic convolution is modest, simply a simple structure is required for quick convolution, as determined by the authors. However, if the length of the cyclic convolution is long, processing elements with the same structure for convolutions of shorter length are designed so that the system can be made more efficient. And when the proposed mode is compared to existing systolic architectures, it is discovered that the proposed design is quicker, more hardware efficient, has a lower input-output cost, and is superior for execution in VLSI domains.

In [66] R.L Chung et.al. (2021), the authors offer a high-quality, cost-effective, and hardware-efficient 2-dimensional DCT signal analyzer. They proposed this 2D-DCT mode for video and picture encoding. Instead of the unfolded CORDIC architecture, they suggest a Loeffler DCT that relies on the recursive CORDIC architecture method to alleviate the memory issue and improve image quality. And to enhance operating frequency while maximising throughput efficiency, they devised a pipelined architecture. In this research, the image compression technique utilised 2D-DCT, a quantization module with a quantization table, and an entropy encoder with a Huffman Entropy encoding table. The resulting image has a loss deviation of only 0.01db. For picture compression, the quantization table of the luminance components was utilised. Over the 0.18 um CMOS technology, the intended modal is being implemented. The computed power consumption and operating frequency are 4.17mW and 100MHz, respectively. Therefore, the advantages of the presented method are high quality, low cost, and the provision of excellent computational resources for the execution of extremely large scale integration. Using Loeffler DCT to lower memory needs depends on the recursive CORDIC algorithm. In addition to its use in IoT applications, this proposed mode and method can play a significant role in the WSN area due to its high image quality and minimal complexity.

The authors of [67] M Perera et.al. (2019), propose the DCT II and DCT VI algorithms in order to introduce the SFG of these two methods. These proposed algorithms produce identical results to the usual DCT, but at a faster rate. This research also examines the computational parallel design of the Discrete Cosine Transform II. On the 40 nm virtex6 FPGA, prototyping, designing, and simulations are performed. Standard 0.18um CMOS cells are utilised for customization. Here, SFG is employed to demonstrate the factorization relationships between the DCT IV and DCT II algorithms. DCT II and DCT IV algorithms have a higher operational frequency than 1D- DCT. As an instance of their proposed methodology, the authors used a 16-point signal flow graph with n=16 points. For the power consumption and area realisation, both ASIC and FPGA implementations are performed.

In [68] N. Banerjee et.al. (2007), N, the authors demonstrate the various implementations and techniques of 1-D 8-point DCT on FPGA. The primary objective of this work is to reduce architectural complexities and delays while increasing the architecture's speed relative to standard 8-bit discrete cosine transform systems. This discrete cosine transform is useful for encoding images and videos, hence it must be of good quality and affordable. In this research, the arithmetic technique has been developed on an 8-point discrete cosine transform in order to make it cost-effective and quick so that the FPGA can execute quickly. Using the row-column decomposition technique, a one-dimensional discrete cosine transform can be implemented in two dimensions. The authors also reviewed the algorithms and computations of the 1D DCT in this study. In this research, numerous algorithms, such as the Lee Algorithm and the Loeffler Algorithm, were utilised. In the Lee Algorithm, matrix representation occurs. This algorithm divides the butterfly into odd and even sections. By multiplying matrices, we can obtain the odd and even components of a discrete N/2-dimensional cosine transform with a single dimension. In Loeffler Algorithm, the steps or components are executed in serial order. In the first step of application, this approach is divided into even and odd portions. Even portion is nothing more than a discrete cosine transform of four locations. In addition, the peculiar portion is transferred to the second stage. In addition, the author employed only three multipliers and adders in the second phase of calculations. The Liu and Chiu Algorithm is an additional algorithm. In this approach, the discrete cosine transform values are changed each time a new sample is added to the system. This algorithm employs the discrete sine transform (DST) to construct the discrete cosine transform (DCT). Therefore, both DCT and DST values are available at the output. As the authors have utilised a distributed arithmetic technique, it would be advantageous to substitute multipliers with adders, as multipliers take significantly more time than adders. Therefore, it accelerates the DCT architecture. The proposed methodology is entirely executed on Xilinx SpartanII FPGA. This version makes use of lookup tables. Therefore, with lookup tables, the design is utilised effectively.

In [69] D. Chiper et.al. (2021), a low-cost, hardware-oriented, high-quality 2D-DCT signal analyzer. This signal analyser is used for image and video encoding. The authors of this study employed the Loeffler dct-based CORDIC approach to improve image quality and reduce memory requirements. For this proposed architecture, a pipelined architecture was developed to boost both frequency and output. They can lower computational complexity by 0.01db as a result. A comparison has been made between 2D-DCT and standard CORDIC procedures. Consequently, they discovered that the 2D-DCT signal analyzer has a strong peak-to-average PSNR ratio and is cost-effective for usage in VLSI. The authors have utilised the 0.18um CMOS technology from UMC. The maximum operating frequency is 100MHz, and the maximum power consumption is 4.17mW. With these features, the suggested architecture is hence ideal for IoT and WSN networks.

As adaptive filters play an essential role in signal processing systems, its implementation is based on a set of programmed instructions. The programmes are then executed on devices such as DSP chips, microprocessors, VLSI, and FPGA. Thus, they contribute to the acceleration of the system. Folding is a technology renowned for its minimal hardware requirements. Combining a folding design with a systolic architecture leads in an improvement in speed and a decrease in area. Therefore, [70] G. Swaminathan et. al. (2020), propose merging the folding architecture with the systolic architecture and designing it with several adaptive filters such as AP, RLS, and Kalman Filter. MATLAB2015 and Xilinx tools were utilised for this objective. In addition, they've implemented the Vertix-5 FPGA. As a result, they've determined that the folding has enhanced SNR=6.77% and RLS=4.67%. In the instance of AP, the proposed design simply demonstrates an enhanced SNR in comparison to others. In terms of size and delay, the proposed architecture demonstrates a decrease of 18.35%.

2.5.3 Review of the Literature on Power Attack Analysis and Countermeasure

Using the SASEBO-GII board, Mestiri et al. (2013) investigate the amount of power traces needed to estimate the AES SBOX key in their article [71]. This board has two FPGAs, one for control and one for cryptography, and both are HDL-programmable. In order to calculate output, CRO displayed the power trace. For this application of CPA, we use the hamming weight power model or the hamming distance power model in conjunction with the switching distance to establish a correlation between the actual and predicted power. In this article, the author examines the power-trace complexity required to derive each bit of the correct key. The 10th key makes use of a correlation coefficient calculated from 4000 traces to distinguish between the real and estimated model. It takes less work for the switching distance model to accurately forecast the key than the Hamming distance model. There is an H.D. of 5.7 and a standard deviation (SD) of 5.6 for the first key, an HD of 6.1 and an SD of 6.5 for the fourth key, an SD of 8.5 and an HD of 11 for the eighth key, and an SD of 6.5 for the twelfth key. HD 9.3 and a 16-key HD version.

A linear relationship between power consumption and the amount of switching bits in the crypto module's input is demonstrated in a 2017 article by Takeshi Fujino et al. [72] Multiple methods for protecting against side-channel assaults have been discussed. Under a power attack, a countermeasure cell supplies either random or consistent power to the inputs and does not divulge any concealed circuit information. An increase in circuit area and power dissipation can be seen in an attack-resistant cell. Excellent resistance to side-channel attack is demonstrated by both the Threshold implementation and the HMDR-ROM, albeit at the cost of a five-fold increase in area and a fifteen-fold increase in power in the case of T.I., and increases of 6% and 20%, respectively, for HMDR-ROM. This implementation of SBOX in MDR ROM demonstrates a stable capability to read a replaced byte from storage.

Power leakage countermeasures are described in depth in a research [73] by Massimo Alioto et al. (2014). There will always be some amount of power loss in a CMOS circuit. Examining how differential power attacks (DPAs) may benefit from and be protected against leaking power. The author of this work presents an approach of recovering the crypto module's key by determining the leakage power, which is always smaller than the total power of the input but can change from input to input. The first component of the leakage current is determined by the hammering weight of the text typed and the round key applied to the add round key. To determine the second variable, press the key followed by the round input. Two things are constant: (1) the leakage current is always present, and (2) the quantity of high bits rises linearly with the leakage current. The probability of a successful kay prediction is reflected by the correlation coefficient. Innate cells like WDD, SABL, and MDPL use a CMOS architecture that allows for a smaller relationship value while producing a bigger area.

Secret decipherment can be aided by correlation power analysis, as demonstrated in Junrong Liu et al(2010) .'s study [74]. If you want to use CPA with the steam cypher MICKEY v2, the two most common models are H.W. and H.D. Power consumption is correlated with both the number of high bits at the output and the number of transitions in the H.W. and H.D. models. The estimate is modelled after the power used to charge the load capacitor during an output change of 0-1 and to discharge it during production of 1-0. If you're switching between 0 and 1, you're not using any energy. While the Hamming weight model may retrieve the key more efficiently, this is also its main weakness. Power consumption is inversely proportional to the hamming distance, and the correlation impact is -0.43 when k=0 and 0.4 when k=1. In this article, the authors explain how the gap between k=0 and k=1 can be bridged with just ten power traces.

You can find a mathematical model of DPA and CPA in paper [75] by Eiric Brier (2004). The crypto module's side-channel analysis can be tested with standard power attack analyses like the common power attack (CPA) and the differential power attack (DPA). DPA uses the average power from the traces of LSB 0 and LSB 1 as inputs. Additionally, the CPA makes use of the correlation coefficient to evaluate the similarity between predicted and real power. This article reveals the flaw in the conventional differential power attack method and offers a theoretical framework for the more modern correlation power assault. The CPA model establishes a correlation between power usage and the number of high bits transmitted to the SBOX input. The CPA attack was portrayed as a greater value of the correlation coefficient, while the DPA attack was limited by a ghost peak. Eight SBOX were attacked using a CPA with only 40 power traces in the first round. For the first SBOX, we should expect a correlation of at least 65% and up to 92%. DPA calls for larger sample power traces than CPA does. An attack's analysis can still be exploited with only a partial correlation. Reverse engineering (leakage rate) on existing data should be required by the CPA.

A method for recovering a key from cache memory between the CPU and memory system is described by Satok et al. (2001) [76]. Cache memory is used in a novel process-sharing technique for a cryptographic algorithm. Table accesses are limited by the number of bytes in use and whether or not they are cached due to the short duration of each time slice supplied by b victim. It is necessary to load a new byte after a cache miss and clear the cache. Using this data, an attack known as a side-channel was successful. By using only the most up-to-date information stored in cache memory, the author of [76] demonstrates a side-channel attack (SCA). The requirement for massive quantities of cache memory is shown by the fact that an analytical model for leaking may be constructed if the victim has up to 60 cache accesses open. A 128-bit AES key may be decrypted or encrypted in only around 7 repetitions using a caching attack.

Cecile Canovas and Jessy Clediere (2005) divide power attacks into three categories (DPA, DEMA, and CPA) in their study [77]. While leakage power data from the SBOX is used by DPA and CPA, DEMA relies on electromagnetic radiation produced by a computing module. Gathering and processing the E.M. radiation at each SBOX input is necessary for a capture probe. In contrast to DPA and DEMA, where the incorrect key appears as a ghost peak, CPA identifies both the correct and incorrect keys through subtle differences in the correlation coefficient. A CPA attack using the hamming distance model may be used to compromise both multi-bit and single-bit SBOX output. An imaginary model in which the hammering weight and hammering distance are scaled by a factor of 32. In place of the ghost peak issue, correlation value provides improved categorization based on the correlation coefficient. The approach's health improves as the degree of imbalance rises. These tests help ensure that the SBOX architecture is correct and up-to-date before the real test.

In their 2016 study [78], Davide Bellizia et al. made advantage of the relationship between a submicron CMOS chip's static current and its internal processing capacity. Power assaults are able to derive the secret key using a univariant attack by statistically assessing the correlation between the actual leakage current and the leaking model. For example, the WDDL, MDPL, and

SABL demos show how to conceal information in the power trace or how to randomly generate the power so that the power paternal is always the same. Time-closed logic (TEL), when implemented in dynamic CMOS circuitry, is a novel type of power-resistant logic. The data route and CMOS technology determine the observed security level, not the clock signal. The evaluation criterion of 10ps is shorter than the time it is expected to take for the transient to settle. This means the attacker won't have time to gather the power trace.

In their study [79], Y.J. and Noh, M.J. (2005) provide a comprehensive overview of differential power assaults and related defences, such as masking. Power attack analysis relies on leakage power since the value of power is proportional to the information fed into it. The masked cell's power curve does not have a linear relationship to the input. Using a standard Boolean logic gate, we can implement masking where the principal data has no effect on the amount of power used, instead of having that power split throughout multiple stages of the circuit. The input data is hidden by a series of random bits, and the decoded output data is once again hidden by a series of random bits. Combinations of logic gates with no effect on the final result; the author of this article has used mask gates to verify the security of the AES, SEED, and SHA functions. The number of gates and the time along the crucial path are both inflated due to masking. The number of gates in an AES system using masked SBOX rises to 25.9K from 17.4K.

In their paper, [80], Owen Lo et al. (2017) demonstrate a DPA and CPA attack on SBOX output. The DPA attack demonstrates that the mean values of the power traces for LSB1 and LSB0 are distinct. With the right key, a peak appears in the differential curve, while the erroneous key shows a "ghost" peak. The CPA attack makes use of a simple Pearson correlation value, a Power trace storage system built with an Arduino university, a key-sight oscilloscope, a probe, and an information analysis machine. However, DPA has limitations due to the public availability of cryptographic library source code that can record the power consumption of real-world devices in use. Obtaining a known result from AddRoundKey for a given plain text is possible if one knows the exact attack point for DPA and CPA. For a right guess, the original key to the secret can be yours.

In article [81] from 2015, authors Saravanan P and Kalpana P. explain the role of the XOR gate in cryptography. Power attacks are a problem for XOR gates implemented in complementary metal oxide semiconductors (CMOS). There is no information leakage and a power trace is displayed, thus we know that the XOR gate is secure. by Here, the author proposes a novel design for the XOR gate in adiabatic logic that includes a charge-sharing alternative. Two energy-based metrics, NED and NSD, exhibit resistance to power attacks. Difference between minimum and maximum energy expressed as a percentage; input plain text energy deviation expressed as a percentage. There are already XOR designs that show reduced NED and NSD, such as CPL, DCSVL, Adiabatic, SABL, SyAL, and CSSAL, albeit at the cost of higher gate count and power consumption. The proposed XOR gate can conserve energy by 79.6% compared to SyAL and by 82.5% compared to CSSAL, while decreasing NED and NSD by 20.8% and 26.1%, respectively. The SyAL XOR gate occupies 409% as much room as the CSSAL XOR gate, whereas the latter requires 463% more room.

Paper [82] by Lu Zhang, Luis Vega, and Michael Taylor (2016) uses a Power model to verify the efficacy of defences against DPA and CPA assaults on the crypto module. High-bit-rate CPA assault based on hamming-distance power usage patterns. Assuming that a CMOS circuit's input data may shift between 0-0 and 1-1 without requiring any additional power, the power traces of the XOR and AND gates revealed the bits at the input. Both DPA and CPA are able to effectively retrieve the secret key with a strong correlation. An effective method for creating a trustworthy crypto module is masking. The correlation factor can be decreased by using the masked gate to implement masking in the method. If you're looking for an alternative to masking logic like WDDL, RSL, or SABL, the mask gate is widely regarded as the superior countermeasure.

In 2005, Stefan Mangard and co-workers found, in paper [83], that glitch reduces the performance of the mask gate, making it more susceptible to DPA attack. DPA countermeasures aim to lessen the degree to which theoretical power predictions and measured power levels are correlated. The goal of the countermeasure is to ensure that the CMOS circuit's power consumption is constant regardless of the input pattern. Energy input E is modelled as a function of output values, so we may examine the effects of input values of 0 and 1 on energy input and vice versa. To generate a single bit takes a larger amount of energy, and the time and power required to flip the output 0 opens the door to a DPA attack. To implement an attack with a mask gate, you'll need a 45*16 power trace, while a conventional gate just needs a 25*16 power trace. Mask gates not only increase computational complexity but also distort the differential curve of mean necessary for DPA analysis.

Maneesha Jayakumar (2018) demonstrates in article [84] that the efficient architecture of mask AES engines is safe from DPA assaults. The AES technique is masked using a Boolean questioning methodology. Each element of AES is represented by a logic gate circuit that is a boolean equation. SBOX's input and output are both disguised due to its significance in the encryption procedure. When the input to SBOX is modified by XORing it with a random bit set, the resulting output is masked and must be unmasked with yet another random bit set. For SBOX's output to be consistent with what would be received from unmasked data, there must be a relationship between the random bits used for masking at the input and the output. Unmasked SBOXs have different power lines than masked ones. FPGA implementations of unmasked and masked AES have a latency of 36.886 nS and a memory need of 342.532 Mb, respectively.

Tiri, Kris, et al. (2005) examine power-resistant logic models SABL; WDDL and TDL, all of which are centred on the random charging and discharging of the load capacitor. A method called SBOX that minimises energy consumption through the use of adiabatic charging is proposed [85]. The energy required to charge the low-to-high transition is $2(R/t_p)C_LV_{DD}^2$, which influences power traces and assessment. Charge-sharing symmetric adiabatic logic is the technology used to achieve this output power independence from input bits. For each change, SBOX offers a stable stream of electrical power. At 50MHz, the eCSSAL SBOX is capable of producing an Emin of 4.85pJ and an Emmax of 5.87pJ. At 50MHz, the NED and NSD for adiabatic charging and discharging of a DPA are 3.7 and 0.65%, respectively.

M. Masoumi (2019) publishes a work in which he employs a space-saving variant of the Boolean masking technique to create an AES architecture [86]. Welch's t-test is a common example of a randomised table that severes the connection between real-world power and a mathematical power model by using the difference in the capacity to compute outputs 0 and 1 as the dependant variable. Based on the XORed output of the add-around key step, SBOX is utilised to locate a byte and mask table that may be replaced. Each XORed output has its own distinct mask value stored in a 16x16 matrix known as the mask table. Unmasked output is obtained by XORing the mask and output from the SBOX. According to the results of the article, the affine transform results in a 300% gain in area at the expense of a 60% reduction in speed.

In their publication [87], Tena-Sanchez et al. (2014) offer a technique of cryptanalysis including a third-party power attack with a connection between power and input that is resistant to DPA assault. Some of the drawbacks of the mask logic approach are mentioned, such as SABL's sensitivity to unbalanced output, SyCML's decreased swing, TDPL's extra control signal, and DDPL's need for an additional inverter. In this work, the author adjusts the pull-down criterion in both the charging and testing phases by using either a single or double switch. An increase of 0.17 percentage points in NED value was noted during precharge for AND gates while using a single switch. There were 35.15 percent of people using two switches, 10.9 percent using one, and 27.0 percent using the double witch. During the evaluation, a single switch increased the NSD value by 32.95 percent. When comparing single- and double-switch NSD usage, we see a difference of 53.05% for the latter and 157.17% for the former. The evaluation step for the XOR/XNOR architecture uses 1.28 times the energy and 1.02 times the delay of a conventional implementation, while the double and single switch pull-down operations use 102.1 and 102.1 times the energy, respectively.

Elisabeth and Stefan Mangard (2005) [88] wrote a study reviewing the hardware restrictions associated with sidechannel attacks. Timing, power analysis, and electromagnetic attack are the three main types of SCA. The three subcategories of power attacks are special, direct, and common. A DPA attack on SBOX based on power traces and the difference between the mean curves for keys with LSB=0 and LSB=1 reveals both the bogus ghost peak and the correct key peak. Adjusting the device's power usage so that attackers can't tell how heavily the gadget relies on data is one mitigation mechanism against DPA assaults. Options for applying concealment include dual-rail precharge, current mode logic (CML), and dynamic current mode logic (DML) sensing amplifier-based logic. Insecure implementations, such as hiding, might affect evaluation times and/or efficiency. Each cell's power value in the cryptographic device was created at random and disguised. Overhead, memory requirements, and bugs stand out as particular challenges for a DPA-resistant logic design approach.

Popular ways to defend SBOX against power attack analysis include gate-level masking, as explained in paper [89] by J.Zeng at. el. (2012). Attack-resistance gate is implemented in WDDL and SABL logic. There is still a potential for them to slip inside the masked gate and steal secret information. This study proposes a mask AND gate and a guarded XOR gate that may be used to calculate the SBOX non-linear function GF28. The masked gate, modified the input, to the logic gate as a non-linear function of the normal and masked information. In a masked SBOX AES-SBOX implementation in 180nm CMOS technology with a masked AND gate, the latency rises from 7.02 ns to 9.44 ns, the static power rises from 22.973 nW to 51.151 nW, and the dynamic control rises from 4.187 mW to 16.2 mW. Using 594 gate slices, the Virtex-5 XCVLX30 may create a 32-bit masked SBOX. The masking algorithm used to restrict the S-box is 34% less efficient than the unmasking algorithm. When implemented using the CMOS 0.18m CMOS standard cell library, the power consumption and footprint are twice as large as those of an unmasked SBOX.

In their study [90] from 2018, Shuaiwei Zhang and Weidong Zhong propose a protection system for SBOXs that converts multi SBOXs and number input to 4*4 cells rather than 8*8 cells. Highlighting the SBOX framework's reusability. SBOX's reduced footprint demonstrates that it uses less energy while storing the same number of bits. Since the difference in power equals DP=p1-p0-/2, the logic level's safety is evaluated by computing the high or low output. Power and input data have 0% correlation, as indicated by the D.P. value of zero. Protecting a minimum level of D.P. is key to the DPA-resistant logic approach. Noise and masking are offered as a countermeasure to power attacks. To obtain power data, attackers aim for the weak spot in the SBOX. By increasing noise, parallelizing the SBOX, and aligning the power trace, we can reduce the percentage of times an estimate is true. The success rate ranges from (1/2)9 to (1/2)51 as the key is extracted from 8 SBOX. The first thing to know is that SBOX is more vulnerable to attacks, and the last thing to know is that it has a lower success rate.

To conduct a more damaging leakage power analysis attack, as detailed in article [91] by Siva Nishok Dhanuskodi et al. (2016), it is necessary to be aware of the internal state of the circuit. LPA are may exploit circuitry designed to withstand a side-channel during a dual-rail precharge. The leakage current may be determined now that the SBOX design is available to the public. Leakage current is determined using a statistical method based on the on and off states of individual transistors. Accurate flow predictions are possible given knowledge of the circuit's crucial internal condition. Calculate the Pearson correlation coefficient between the expected current and the actual current using the hammering weight (H.W.) power model. In other words, the Pearson correlation coefficient is quite high for the right answer. Key recovery at 300MHz using 1000 samples is successful 59% of the time due to the effect of process variation and temperature.

When it comes to space and latency, the ROM-based SBOX implementation presented in article [92] by Craig Teegarden et al. (2010) is unrivalled. Sidechannel attacks originate from power leaks, and their goal is to establish a connection between the resulting fluctuations in power consumption and the underlying input data. WDDL and TDPL are two examples of combinational logic that aim to reduce power consumption by making stronger assumptions about the system's state. When compared to a regular CMOS cell, a WDDL-implemented SBOX uses 3.1 times as much space and generates 3.7 times as much power, but its throughput is 3.8 times slower. Here, the current equalisation via switched capacitors consumes practically the same amount of power regardless of the source. It was discovered that using switched current logic to implement SBOX results in a 1.07-time area overhead, a 1.33-time increase in power consumption, and a two-time decrease in system throughput.

Paper [93] by Halak, B., Murphy et al. (2015) provides an illustration of a leakage-power-attack-resistant layout. Static power consumption prompted the development of leakage power analysis. When the weight of the hammer is high, a lot of power is being used, and when it's low, very little. The objective of the power-balanced logic method is to generate n-input data in the form of equal power value-form output. Leakage current may arise in CMOS 90nm technology in three different ways: subthreshold leakage, gate leakage, and band-to-band tunnelling. Based on the hamming weight power model, a leaky power differential attack is shown here. One technique to learn how to create a safe logic design is to play about with the input pattern for an AND, OR, and XOR gate (0101, 0110, 1001, and 1010) and observe the leakage power dependency of the output. The leakage power was observed to be greatest for the "0101" input and least for the "0010" input. Leakage of power increases when the most important bit is high. When compared to the OR gate, the XOR gate's Pearson correlation coefficient is lower. In power-balanced logic, temperature changes little affect the coefficient of variation.

Adiabatic logic, introduced by Cancio Monteiro et al. in article [94], is a power assault resistant logic (2011). An ideal power-resistant crypto module would be concealed. But masking isn't foolproof because it needs an arbitrary bit for both masking and unmasking. In this work, we explore the single-rail and dual-rail logic families for their applicability to the implementation of adiabatic logic's security. To pre-charge and evaluate, a CMOS inverter with adiabatic differential logic like SABL or 2N2N2P only needs a single clock input because of the inverter's complementary parallel construction that uses differential pull up and differential pull down. The SPICE simulation results show that the XOR implementation of adiabatic differential logic has NED 91.22% NSD 46.8%, while the 2N-2N2P implementation has NED 65.178% NSD 29.8%, and the SABL logic implementation has NED 0.79% NSD 0.329%. SABL and 2N-2N2P logic are significantly more resistant to power threats than adiabatic dynamic logic.

In article [95] by Tiri et al. claims that the power consumption, of CMOS circuits is invariant with respect to the input processed data. The power attack on CMOS dynamic and differential logic (DDL) is mitigated in this work, making the technology secure enough for cryptographic applications. The method that has been presented displays the amount of power usage irrespective of the data sequence or the logic value. AND-NAND logic, as well as flip-flop logic, have both been explored by the author. Since the output node discharges while the input value is hidden, the problem was detected by dynamic logic. There was no discernible difference between the 0-0 and 1-1 switching events; nevertheless, the 0-1 and 1-0 transitions consistently ate up energy. The input data sequence is broken by dynamic logic, and everyone uses up energy as the capacitor charges and discharges from zero to one. The upper half of a SABL logic gate functions as a dynamic logic gate, while the lower half functions as a differential pull down, sharing both features. SABL SBOX uses NED and NSD as its security measures. At the expense of 38541.1 m2, SABL SBOX is able to achieve NED 0.032 and NSD 0.006.

Popp, T., and Mangard, S. (2005) [96] demonstrate a side-channel attack-resistant cell for SBOX implementation with energy utilisation that is independent of input data. The asynchronous logic hardware implementation was shown to be safe from side-channel attacks. It takes more energy to pull up an output node to a high state than to bring it down, according to a model of data switching probabilities. A cell resistant to side channels attempted to bring this disparity down to an extremely small number. The flaws in MDPL logic are the cause of the issue in the masked cell. If a disguised cell has a transition at an off-time (glitches), it becomes more vulnerable to attack. Energy expenditure during input switching with and without a masked cell, in order to demonstrate assault resistance. In MDPL logic, the XOR gate is used to mask all signals with the same mask bit. MDPL cell-based AES implementation has a 4.54x area overhead and a 0.58x speed overhead compared to CMOS circuitry.

Side-channel analysis was first proposed in a 2003 work by Fournier et al. [97] to collect and evaluate statistically large amounts of power trance data. Analysis using electromagnetic waves produced by the conducting component. Just like a power attack, an E.M. wave will take in data and look for a statistical connection between the size of the data being exploited and the amount of hammering force applied to it. Intrusive techniques like fault injection analysis, which involve inserting an optical probe or deliberately producing an external malfunction, are used to try to determine the secret. Temperature fluctuation during the process of computing. If the countermeasure strategy discussed in this study is to be effective, it must be based on switching activity, which is a good approximation of DPA and DEMA attack. When compared to a traditional synchronous circuit, a balanced asynchronous circuit is far superior. Analysis of security at the design stage takes many factors into account to determine the likelihood of errors occurring.

Mentens et al. (2003) argue in their study [98] that security measures should be built into the VLSI design process at multiple stages. The necessary steps in the development process of a secure side-channel attack. VHDL modelling is the first step in the design process, while attack-resistant layout is the final step. Based on experimental evidence, it has been determined that 200 power traces are sufficient to compromise the security of a CMOS standard cell using a DPA attack. While DES implementations can recover hidden kay over 2000 power traces, this is not the case for DES implementations that use stack-resistant cells. Additional attacks can be made, such as a timing attack based on the ciphertext's arrival time, which can reveal the dependent operation, which in turn requires knowledge of the secret key. The first step in creating a SCA-proof design is to use the industry-standard Verilog HDL digital design language. Electromagnetic analysis is a form of power analysis attack in which electromagnetic power waves are studied rather than electrical power. A differential fault attack occurs when an adversary takes advantage of a vulnerability in a system by deliberately introducing a fault into the system.

Nonlinearity in computation is included in the SBOX building blocks of AES algorithms, as described in paper [99] by Khan et al. (2003). New XOR logic was developed by Nabihah Ahmad and S.M. Rezaul Hasan in 2013 and is now being used to implement SBOX and inv SBOX. Composite field arithmetic is implemented in GF2 within an internal SBOX block. Each brick also comes with a Boolean equation and a logic gate. At a supply voltage of 00.8v, the SBOX requires 948 transistors and has a latency of 7.32nns, while its maximum throughput is 130MB/s. When using CMOS 65nm technology, the proposed XOR gate in Galois field arithmetic necessitates symmetric coupled inverter pain and an additional not gate design. PDP for the proposed 6T XOR gate is 23.453 YJ at a voltage of 1v. The simulation results for the

SBOX demonstrate a delay of 7.322 ns, a power dissipation power of 0.659 fJ, and a throughput of 1 GBPS with a voltage supply of 0.8v.

A review of a particular kind of cryptanalysis can be found in paper [100] by Franc ois-Xavier Standaert (2009). An adversary's ability to acquire access via hardware by exploiting the information's physical specification is a metric by which their efficacy can be judged. In this case, attackers bypass the encryption technique altogether by studying the leakage information emitted by the underlying hardware. We've covered the basics of what's needed to collect the power draining away from the algorithm's working space as it calculates. The chip is physically damaged by an invasive or active attack because it is repackaged with an external cable or connection and the trace data is collected. Passive or non-intrusive assaults rely on data that is already publicly available. An extensively researched attack vector against DES and AES is the power analysis attack (DPA, SPA). Dynamic differential CMOS logic is a defence mechanism against power threats at the cellular level. Randomizing the timing of operations, encrypting bus a, masking leakage current, and injecting noise into computations are all algorithmic techniques for evading detection.

Using hardware description language, a pseudo-noise sequence generator, and SBOX's ability to apply a dynamic permutation-combination, the AES128 encryption algorithm devised by Eman Mohammed Mahmoud et al. (2013) [101] is implemented. The sequences produced by the permutation block are then utilised to organise the rows and columns of the SBOX.

The columns and rows of a key-dependent SBOX are organised according to a permuted sequence. Randomness, avalanche effect, correlation factor, and simulation time are used to evaluate the security of a designed SBOX. The simulation time difference between AES and DES is 0.0031s, the avalanche impact is between 41% and 61%, and the correlation factor is between -0.3 and +0.3.

The results of a comparison of two SBOX implementations using a lookup table and on-the-fly computation with the Xilinx XCV100e-8bg560 and XC2VP20-7fg676 FPGA are provided in paper [102] by Pammu et al. (2016). It takes 7215 logic gates to build a look-up table SBOX, but only 4752 gates to build an on-the-fly SBOX that does the same thing. SubByte is a non-linear substitution byte task that is realised as a matrix and is byte-independent. SubByte uses a multiplicative inverse and an affine transform on the supplied data. As a function of clock frequency, the number of clock cycles, and the number of bits generated, throughput may be determined. The LUT SBOX implementation requires 2.229mW of power, whereas the on-the-fly SBOX implementation requires 2.849mW of power; however, the LUT SBOX chip is 1.4 times larger than the on-the-fly SBOX chip.

Using the cryptographic technique on an 8-bit Atmel microcontroller disclosed in paper [103] by Han Yu et al. (2018), Dey et al. (2018) explain and analyse the hardware and software vulnerabilities (2018). The power trace at the SBOX output was collected in a simulation-based experimental setting so that the 1-bit DPA and CPA attack could be studied. The input transition's accompanying power consumption is recorded in a file called a "value changes dump" (VCD). The process of analysing the data reveals that the CPA assault shifts the correlation from -0.3 to 0.3 at the right data points. The difference between the actual power needed for a DPA assault and the potential power needed for a hamming model to return the secret key is the 5120 power trace, which takes a few of SBOX. Compared to DPA, which has a medium computational complexity, and enhanced DPA, which has a moderate computational complexity, CMOS has a very high one.

In paper [104] by Kazuyuki Tanimura and Nikil D.Dutt(2012), differential power analysis(DPA) relies on a correlation between power trace at the output of SBOX and hypothetical power model. Exhibits the WDDL cell logic's ability to guarantee a 100% switching factor, making it possible to accurately predict power consumption. In order to create secure cells, the authors of this research recommend using homogenous dualrail logic. HDRL exhibits increased resilience to attacks, thanks to its shrewd fusion of the Vss current wave and the suppression of the differential power curve. Considering that the SBOX itself uses 75% of the power in an AES circuit, it is evaluated using HDRL when used in a 128-bit AES circuit with 16-SBOX. The hypothetical power computing model can reduce the differential power to zero. HDRL demonstrates a 200% increase in both space and power consumption relative to SBOX implementation. In comparison to WDDL's estimated 2371.7% energy overhead, the HDRL logic implementation provides stronger security at no extra cost in terms of energy. Reduced space is unnecessary for HDRL to function. Since HDRL promotes SCA-resistant cells, it is a promising treatment.

2.6 Research Gap

The randomness of mathematically generated secret keys is questionable. If you know the first seed or a pattern of responses, you can predict what will happen next. The randomness of the secret key is mitigated through statistics. Both the NIST [12] and the Diehard [13] tests, which are widely used in the statistical community, find problems with these claims of randomness. Certain non-replicable features must be taken into account in order to get a truly random result. The expected response from the created answer is that it will be surprising. By manipulating the IC's physical properties, PUF creates a secret key that can't be duplicated. Keeping the key in nonvolatile memory is optional. Software and protocol-based security systems are vulnerable to side-channel assaults [27], which involve information leaks through power, radiation, and timing. While PUF does not store the secret key, it does produce the essential runtime. A collection of challenge-response pairs that were generated and kept in a safe place before being used to generate an authentication key for a risky challenge. To pass the authentication test, the new answer must be very similar to the one that was previously saved. Several PUF designs have been compromised, showing that they are not all clone-proof. Within twenty hours, a Berlin Institute of Technology research group used failure analysis techniques to duplicate a static random-access memory (SRAM) PUF [1]. Side-channel assaults can be used as an entrance point to decode PUF. Both existing MuxPUF and ROPUF suffer from slow response generation rates and convoluted designs. In order to create a PUF structure with improved performance, it is necessary to look at new CMOS device properties [5].

The S-Box is an essential aspect of the encryption mechanism since CMOS-based devices leave an input signature that grows in proportion to the amount of power they use. It uses up 75% of the world's electricity [14]. Side-channel assaults (SCA) can be performed because of the relationship between power usage and data processing. Extraction of secret keys using statistical analysis of side-channel information has recently been shown possible. The independence of power and data is computed using normalised energy density (NED) and normalised standard deviation (NSD). In hybrid CMOS logic, the input pattern should not affect the power consumption pattern. In order to deduce the hidden secret key from the cryptographic key, an attacker can perform a power assault while the machine is in use and analyse the power trail statically. Countermeasures for a damaging assault include concealment and masking, which include making changes to the cell architecture or algorithm. In order to counteract SBOX, existing masking countermeasures mask the input, calculate using the masked value, and then demask the result. A mask pattern must be generated using a finite state machine (FSM) controller [57]. To construct an SBOX, complicate the prediction algorithm in a side-channel assault, and secure the key, it is recommended to mask certain input bits and supply the same power consumption for all input vectors [64]. Existing defences place a premium on reducing gate numbers but increase the amount of area and energy needed.

2.7 Objectives

1. To design and validate PUF based secure system
2. To design an obfuscated architecture for signal processing application using High Level Transformation.
3. To secure the optimized architecture with obfuscation
4. To authenticate the overall system against the attack

Chapter 3

Secret Key Generation

3.1 Introduction

VLSI integrated circuits are defined by the three characteristics of area, power, and delay. With this new technique, designers have an extra layer of protection against vulnerabilities. How likely is it that a genuine user of IC is using it? Is it safe to rely on IC to do its job? This calls for a complex cryptographic method, which raises both the required storage space and the risk of a side-channel attack. In [19], PUF stands out as a major hardware module that incorporates safety measures into the IC design process. A PUF-circuit can be created using any electronic component with changeable attributes and a stable state. The output of a computational PUF circuit depends on the input to the challenge and the unique behaviour of the circuit itself, both of which contribute to the unpredictability of the PUF response (as illustrated in [30]), hence it is necessary to first define a distinguishing feature. We invest a lot of trust in others without understanding whether or not they can be trusted when we communicate with them or when we connect with a system. In order to have safe online conversations, people rely on system and software protocols. PUF is becoming more and more commonplace in digital transactions including smart cards and other forms of secure authentication, such as RFID tags. More information about the application setting is needed to better understand how secure smart cards are. As such, the smart card needs to be able to reliably authenticate users and communicate securely amongst gadgets. The practise of encrypting and decrypting data is known as cryptography and is a subfield of computer engineering. In order to function properly, the smart app needs a protected cryptographic component. Two crucial but easily exploitable security weak spots are the random number generator and the authentication process, both of which are examined in [34]. The generation of random numbers is a sophisticated mathematical function essential to the encryption protocol. It takes a lot of potentially dangerous power for hardware to perform such a sophisticated task. An unpredictable quantity that cannot be predicted or calculated. To some extent, the software's random number generator can be predicted.

Using an initial seed value, and XOR in the feedback channel, the PRNG shown in [105] may create a random integer. The PRNG sequence does have some discernible regularity. Subsequent series are dependent on the present one; given a set of series, it is possible to statistically forecast future random sequences. Authentication and encryption both benefit from using secret keys derived from a TRNG, which is why it is preferable to utilise a TRNG over a standard random number generator. It has been shown by TRNG in [35, 39] that combining hardware features during computation yields an output sequence that is based on both the beginning value and the hardware feature; however, in order to predict the output, an attacker must be familiar with the latter. For starters, John von Neumann discovered that computers can't generate really random numbers and declared "*Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin*".

Passwords are essential in the digital age, and it's not easy to come up with new ones in a safe setting. The method of key storage illustrated in [106] typically makes use of non-volatile memory, such as EEPROM or SRAM. While effective, this approach is time-consuming, taxing on resources, and potentially vulnerable to a side-channel attack when the user's current key coincides with the secret key stored in non-volatile memory. Nonvolatile memory adds complexity and a cost, as data is constantly being lost due to leakage current. An adversary can use a leak to discover secret data during a side-channel attack. There has been extensive research over the past decade into hardware-based security primitives, or those whose behaviour is determined solely by the characteristics of the underlying hardware. As a result of its inherent physical flexibility, electronic circuits based on integrated circuits are extremely popular [40, 107]. Corner analysis can be used to examine the differences between two silicon ICs, which are not identical due to process variation, which includes variability into parameters during production. The physical unclonable function (PUF) is a hidden innovation in integrated circuits that cannot be replicated by any other device. PUF is an inexpensive authentication and security-attack-resistance hardware module. PUF computes secrets at runtime, which is unexpected because it depends on physical properties rather than memory locations [108].

3.2. Classification of PUF

Hardware security modules like PUF have a response that is a complex function of the challenge input and the hardware's particular physical features. It's useful for things like cryptographic key creation and random secure authentication. Different FUF answers to the same challenge input must be distinguishable; responses from each channel must be handled separately; and a response must be invariant in the face of noise and environmental variation; these are the criteria for PUF that are outlined in [43, 113]. There are two types of PUF circuit: strong and weak. Distinguishing characteristics [114] are what set PUF apart from similar schemes; three types of PUF are recognised by [115]: (i) delay-based PUF, (ii) frequency variation PUF, and (iii) initial value of SRAM PUF. It is clear from Table 3.1 that there are two distinct types of PUF, one powerful and one weak.

Table 3.1 Comparison between Weak and Strong PUF

Weak PUF	Strong PUF
Uses fewer CR pair	Needs many CR pairings
No noise affects reaction.	Noise-resistant reaction
Intrinsic variability drives reaction.	Responses are uncorrelated.
Response processes through the error-correcting core	Error-correcting core not required
Output response must preserve	No need to preserve the response.
It is vulnerable.	Not susceptible to attack
Example are SRAM PUF	Examples are Arbiter and RO PUF

3.2.1. Arbiter PUF based on delay

In order to generate a nondeterministic delay row (route), 2:1 multiplexers and a decision device (often arbiters, D-latches, or XOR gates) are used in this PUF design. The output Y of this PUF circuit is determined by the difficulty of the input C and the time lag introduced by the AND-NOTOR cell. Due to the added need for the NOT cell, the delay of the multiplexer circuit caused by a high selection input is less than that caused by a low selection input [116]. Using a challenge input at select, as shown in Figure 3.1, the input signal is passed along the parallel row.

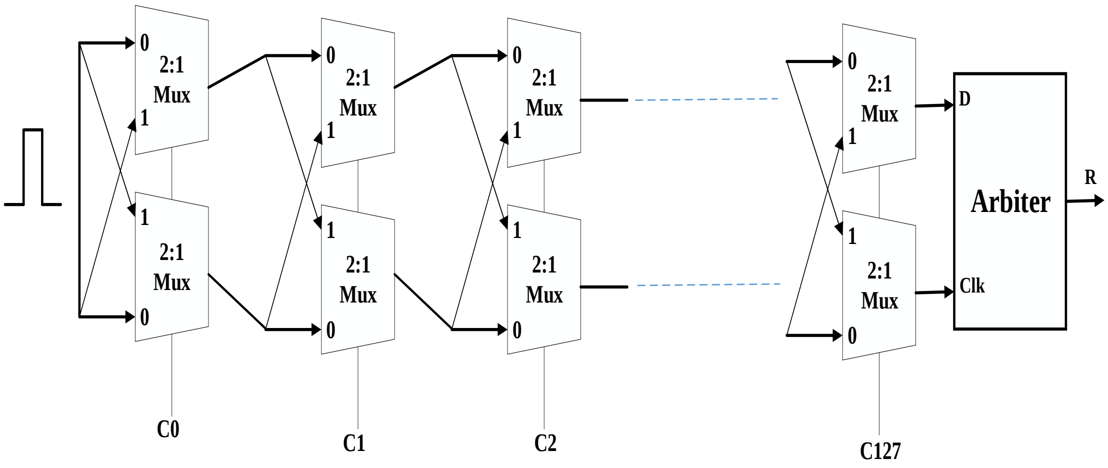


Figure 3.1 Delay based Arbiter PUF

Upper and lower paths to a destination are always competing for the same pool of input data. In [117], we observe a pair of multiplexers that respond to the identical input C[i].

To summarise, multiplexers select the higher path in the top row and the lower path in the bottom row when C[i] is small, and the opposite when C[i] is big. Each input C[i] causes a unique pair of delays inside the cell. The path is changed several times by challenge input C, giving you a unique delay path to overcome the test. The delay on the upper and lower paths are distinct due to the fact that at the arbiter's end, the delays are cumulative. The arbitrator determines which route is the most efficient, and the result is either one (if the higher path is the fastest) or zero (if the lower route is the most efficient).

As can be seen in [45], the multiplexer functions as a delay module consisting of two internal pathways with varying delays. Wait for ai and bi to reach the multiplexer on the top path. Assuming a total of n/2 stages, the delay in the upper path can be computed as Hi + $(-1)^{ci}y_{i,}$ where $H_i = (a_i + b_i)/2$ and $y_i = (a_i\text{-}b_i)/2$ are the signal's delay components (3.1)

$$D_H = \sum_{i}^{n/2} H_i + (-1)^{C_i} Y_i \qquad (3.1)$$

For each mux, di and fi represent the delay on the lower path. A signal travelling through the lower stage is delayed by Li + $(-1)^{ci}u_i$, where Li = $(d_i{+}f_i)/2$ and $u_i = (d_if_i)/2$. As a result, the sum of the lower path delay, as determined by equation (3.2)

$$D_L = \sum_{i}^{n/2} L_i + (-1)^{C_i} u_i \qquad (3.2)$$

The final destination of a signal following both the upper and lower path is an interaction decision device. The equation describing the conditions for a bit response at the output (3.3)

$$D_H < D_L \Rightarrow R=1 \qquad (3.3)$$
$$D_H > D_L \Rightarrow R=0$$

Arbiter delay PUF has the drawback of needing a complex, cascading architecture and a reliable decision device. The rate at which responses are being generated is really low. Researchers in [33, 47] found an unpredictable distribution of results when they used non-linear arbiter, DFF, and XOR gate analysis to Slender-PUF. LOOP-PUF is an alternate strategy for achieving high unpredictability via the non-linear delay stage [117]. The number of parallel routes, where many arbiters function in parallel, is dependent on the width of the response bit. Alternate delay PUF designs are explored in [118], where a current-starved inverter chain is used in conjunction with a control element to produce a final delay. In a continuous manner, Arbiter uses the arrival time of a parallel route to create a response bit. In [119], a PUF with a non-linear delay is constructed using a Feed-forward cascade and Feed-forward overlap structure.

3.2.2. Ring Oscillator PUF based on frequency

A PUF circuit that uses ring oscillators (ROs) requires a large number of the oscillator. Due to manufacturing variation (at the various corner), the oscillation frequency deviates from its true frequency (f±Δf), as illustrated in [27]. When the number of delay stages is odd, the resulting frequency is $f= 1/NT_d$. Figure 3.4 of [120] depicts ROPUF with frequency variation, a collection of N frequencies each of which has an active edge at a slightly different point in time.

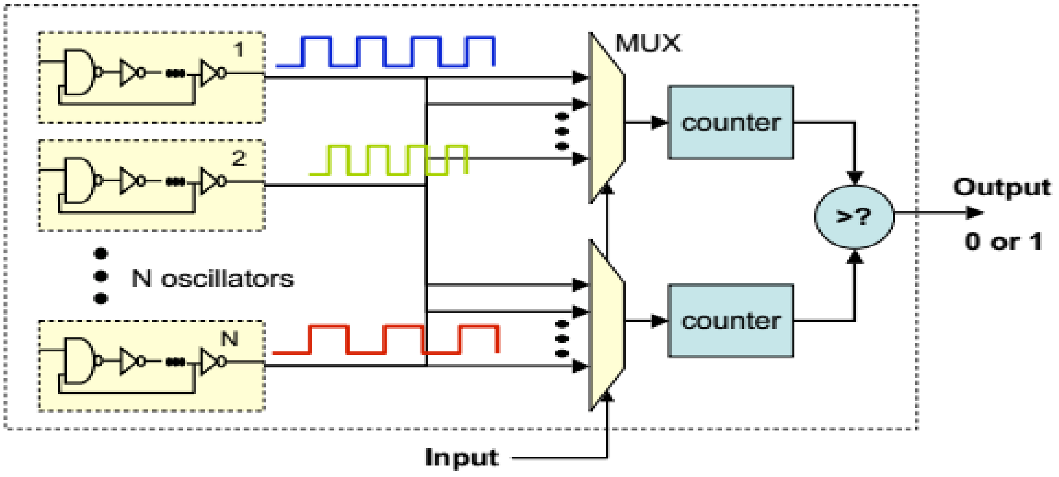


Figure 3.2 Frequency based Ring Oscillator PUF[119]

A big multiplexer with a predetermined frequency pair. Two separate counters can begin an ascending count [121, 122] in response to a challenge input at the selection line. Depending on the parameters of the test, pick two different frequencies, f_1 and f_2. The counter numbers differ by a small amount depending on the result. The counter's size should be kept somewhat large so that it has enough entropy; discrepancies in the count value should persist even after a large number of cycles have been counted. Boosted, with a high reaction if the top counter is generating results quickly, and a low one otherwise [38]. When the number of ring oscillators increases, the output power of PUF rises. Due to power consumption, multiplexer size, and frequency range counts to provide a level playing field in any comparisons made. The RO-PUF entropy is conditional on the Using a seemingly arbitrary method of frequency selection for oscillation, [29] has examined the transitory influence of ring a transient oscillator/oscillator (TERO) that counts how many peak-to-peak between two identical oscillators in terms of seconds. Two elicit a number of responses in the comparison is changed with two multi-output operations (adding, subtracting, etc.).

3.2.3 SRAM PUF based on Initial Value

Back-to-back The two CMOS inverters in an SRAM cell are controlled by an access transistor. Both Q and Qb are stable states for an SRAM cell. The Q transition is low and the Qb transition is high during a write operation if the data bit is zero, and vice versa. Cells in SRAM enter a metastable condition when they are switched on but there is no data write procedure taking place. With a perfectly matched set of transistors, the cell can stay in its metastable state indefinitely [123, 125]. It is certain, however, that a threshold voltage mismatch will occur in the transistor during real implementation [37, 124]. In the presence of noise, the circuit's feedback loop is triggered, and the state transitions to either a high or low value; the resulting cell state is unpredictable since the process variation produces output bit fluctuation. Only those SRAM cells are chosen which give steady behaviour, as shown in Exhibit 10% error for SRAMPUF in figure 3.5 [126].

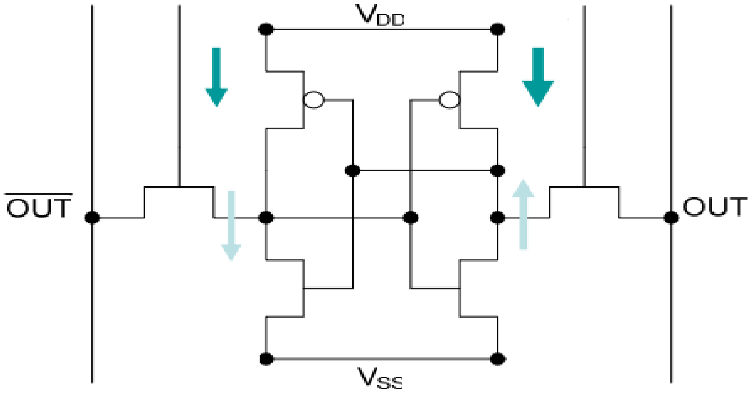


Figure 3.3 SRAM PUF

3.3 Schmitt Trigger PUF

As a common electronic circuit for shaping waves, the Schmitt trigger is a popular choice. The threshold voltages of a Schmitt trigger circuit are split in half. The higher and lower threshold voltages are determined by comparing two measurements of a varying input signal during the same time interval. Multiple threshold points are what set the Schmitt trigger apart from the comparator circuit [127]. A comparator circuit only has one threshold point to compare the analogue signal. For both increasing and decreasing input, Schmitt triggers display contrasting thresholds. In order to reduce noise creation and create a more stable signal, hysteresis can be implemented in the trigger circuit. Schmitt triggers change their output state when the input voltage rises over the upper threshold voltage (UTP) or falls below the lower threshold voltage (LTP) [128]. The gap between the voltages of the UTP and LTP states called the hysteresis width. Hysteresis breadth is unique to and remains constant for the Schmitt trigger. In the case of Schmitt triggers, the circuit's behaviour close to the transition point isn't a feature;

the switching voltage Schmitt trigger won't react to noise if the signal strength is weak; and hysteresis dampens the sensitivity to noise and disturbance. A Schmitt trigger PUF circuit response is generated using the threshold voltage and hysteresis of the trigger. In comparison to conventional comparator circuits [129], the transition time of a Schmitt trigger is much shorter, making it ideal for generating loads that benefit from quick switching, low power consumption, and a low DC supply voltage. The current work of Schmitt triggers was built using a CMOS 90nm technology at 1V supply voltage and a virtuoso schematic composer from Cadence. Figure 3.6(a) is a schematic for a typical Schmitt trigger circuit, which uses both NMOS and PMOS transistors. The circuit uses a two transistor inverter to generate Schmitt triggers. A trigger circuit's threshold voltage is adjusted by adjusting the PMOS/NMOS width ratio. Additional feedback transistors PM2 and NM2 supply hysteresis width, and the thresholds of NM1 and PM3 are raised above those of NM0 and PM4.

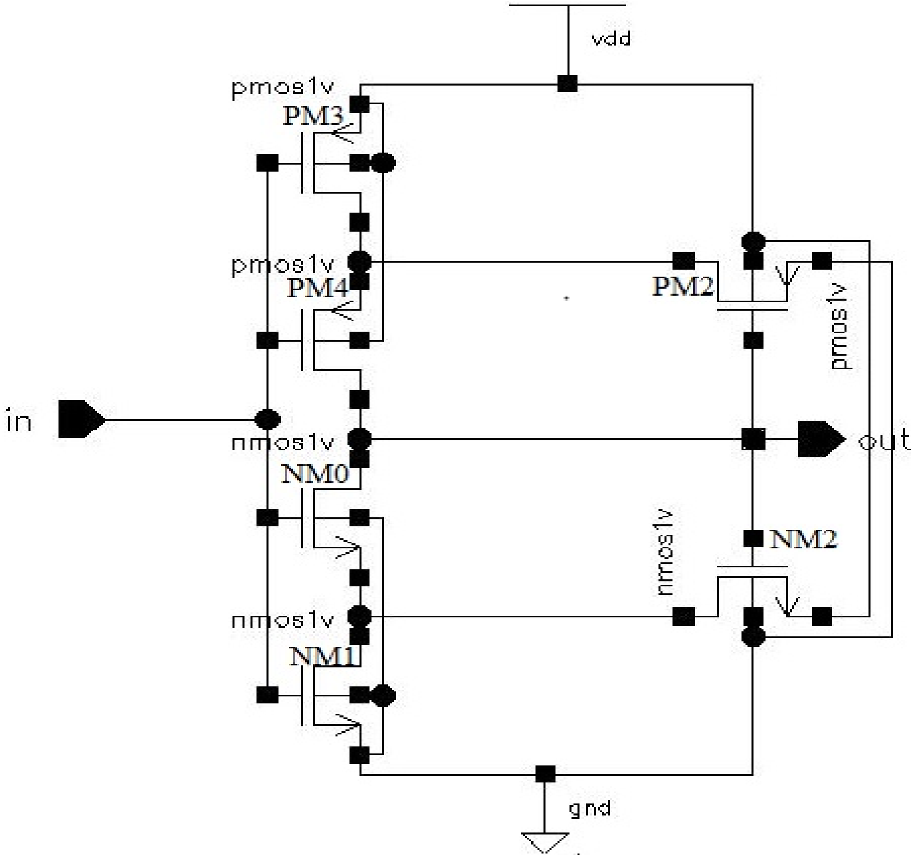


Figure 3.4(a) Schmitt Trigger PUF

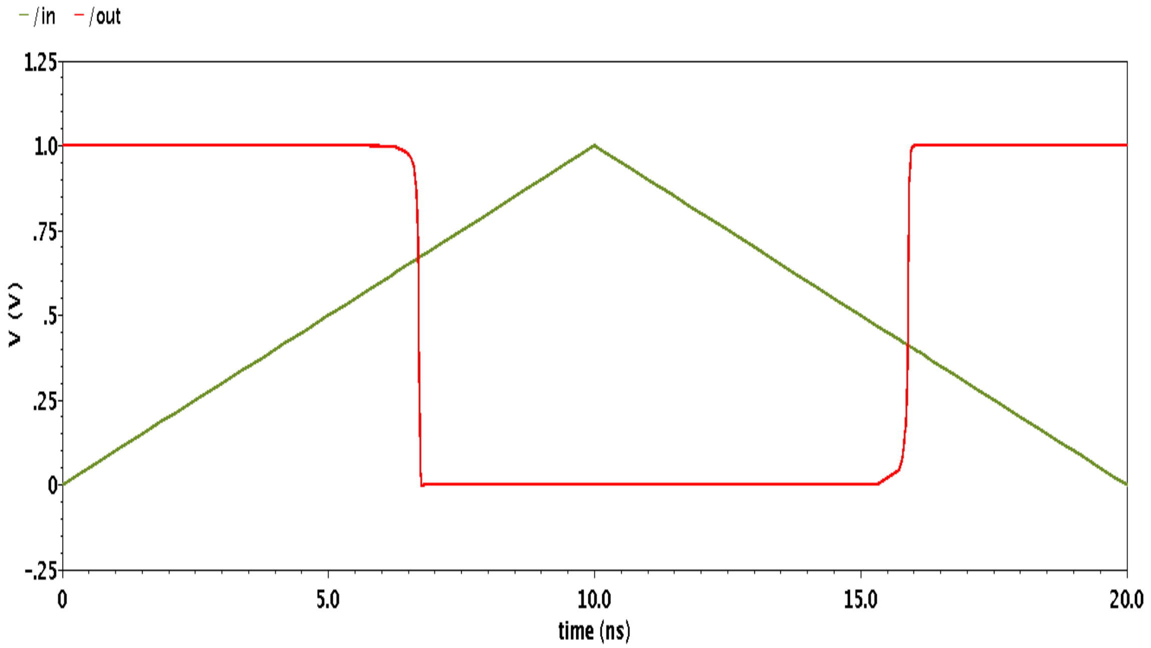


Figure 3.4(b) Transient Response of Schnitt Trigger PUF

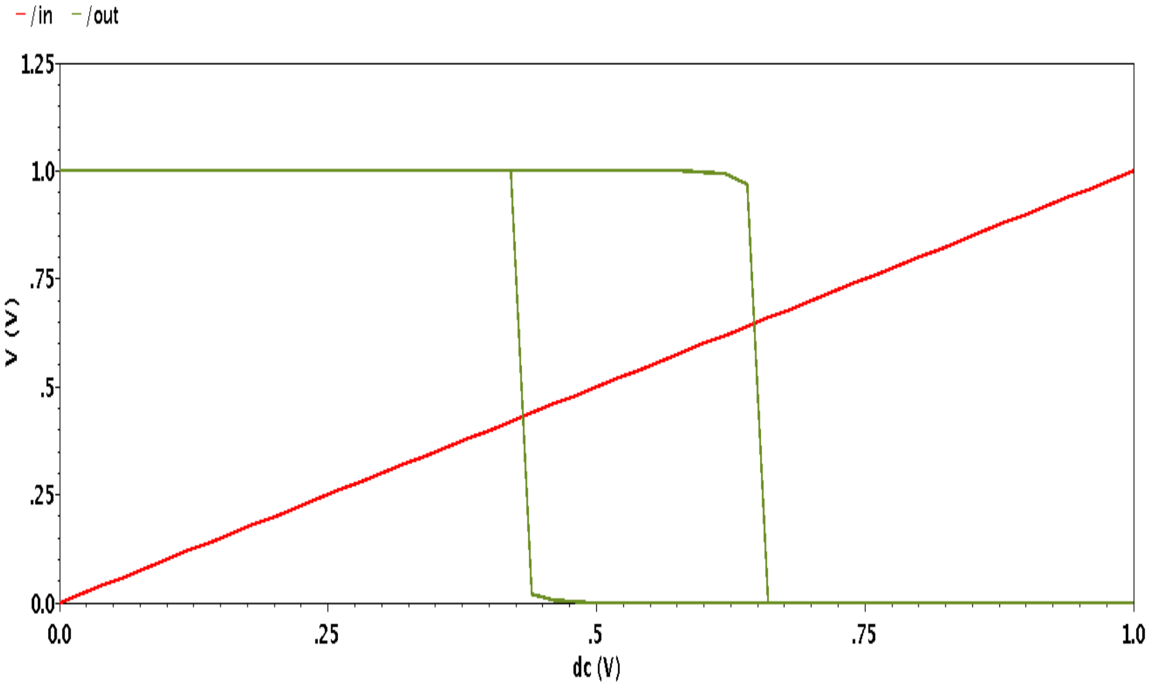


Figure 3.4(c) DC response of Schmitt Trigger PUF

By changing the transistor's W/L ratio, the hysteresis width can be modified. Transistors NM0 and NM1 are turned off and PM3, PM4 are turned on when the input voltage V_{in} is less than V_{th}; this pulls the output to a high level and turns on the feedback transistor NM2. Sets NM0 to bring down the output level and NM2 to pull it up. When Vin gets close to V_{th}, the NM1's pull-down to V_{DD}-V_{th} switching point occurs, resulting in a lower switching point. V_{LPT} Likewise, when V_{in} is greater than V_{th}, NM0 and NM1 switch on, PM3 and PM4 turn off, and PM2 is forced into a on state by feedback. PM4 is activated and PM3 is turned off when V_{in} drops to a level close to its threshold. PM4 raises output and PM2 lowers it. The output level has been held constant at a very low value (V_{th}). Pulling up the output yields upper threshold point V_{UTP} when input V_{in} approaches the threshold voltage (V_{th}) of PM3. In the simulation, V_{LTP} and V_{UTP} were able to find a happy medium at 15% of the supply voltage. When a Schmitt trigger is used in linear mode, as shown in Figure 3.4(b), the transient response of varied signals from one stable point to others is displayed. We're looking for a dropping trigger input voltage of 0.357V and a rising sample input voltage of 0.742V. When the input ramp voltage drops below 0.742V, the ST output goes low, and when the input pulse voltage drops below 0.357V, the ST output goes high. The additional voltage supplied to a low logic level at the output (V_{UTP}) or subtracted from a high logic level (V_{LTP}) is seen as hysteresis in Figure 3.4(c). High output in cutoff, low output in NM1 and NM3, and NM2 and NM3 saturation are all possible. The cutoff PM2 and PM3 in the saturatin area control the low-to-high PM4 output switches. Calculated by using equation (3.4), the Schmitt triggers' critical point (3.5).

$$V_{T+} = \frac{V_{DD} + V_{Tp} + \sqrt{\frac{\beta_n}{\beta_p}}V_{Tn}}{1+\sqrt{\frac{\beta_n}{\beta_p}}} + \frac{\sqrt{\frac{\beta_n}{\beta_p}}(V_{DD} - V_{Tn})}{(1+\sqrt{K_n})(1+\sqrt{\frac{\beta_n}{\beta_p}})} \tag{3.4}$$

$$V_{T-} = \frac{V_{DD} + V_{Tp} + \sqrt{\frac{\beta_n}{\beta_p}}V_{Tn}}{1+\sqrt{\frac{\beta_n}{\beta_p}}} - \frac{\sqrt{\frac{\beta_n}{\beta_p}}(V_{DD} + V_{Tp})}{(1+\sqrt{K_p})(1+\sqrt{\frac{\beta_n}{\beta_p}})} \tag{3.5}$$

Where VTp is the threshold voltage of a PMOS transistor and VTn is that of an NMOS transistor [19]. It is the load capacitance that determines the hysteresis width. An expression for the hysteresis voltage can be written as

$$V_H = \frac{1}{(1+\sqrt{\frac{\beta_n}{\beta_p}})}\left(\frac{\sqrt{\frac{\beta_n}{\beta_p}}(V_{DD}-V_{Tn})}{1+\sqrt{K_n}}+\frac{(V_{DD}+V_{Tp})}{1+\sqrt{K_p}}\right) \qquad (3.6)$$

In this work, we introduce a Schmitt trigger-based PUF circuit with a delay and hysteresis; the STPUF's design is similar to that of a delay-based arbiter PUF. The delay time can be varied by using a Schmitt trigger that is managed by a multiplexer in place of the traditional delay timer. Figure 3.5 [29] depicts a STPUF with two parallel routes; an incoming analogue signal must compete with itself to reach the far end first (arbiter). Multiple up and down transitions in the input signal are triggered along the journey. With the help of a multiplexer, the race signal can be sent down one of two paths depending on the difficulty of the input challenge. At each stage of a Schmitt trigger, the input signal is delayed by performing numerous comparisons with the threshold voltage. Signal delay at the Schmitt trigger stage and hysteresis breadth determine the data received at the arbiter end. If the higher path is coming first, the arbiter at the end decides to set the response bit to high, otherwise it sets the bit to low to indicate a slower way. During hysteresis, a ST is unpredictable, but its output is steady past the switching point [130]. The generated response depends on the difficulty of the input, the latency, and the width of the hysteresis. Response bits are shown to be a function of numerous hardware attributes for the first time in the literature.

$$R = f(C, Delay, Hystersis) \qquad (3.7)$$

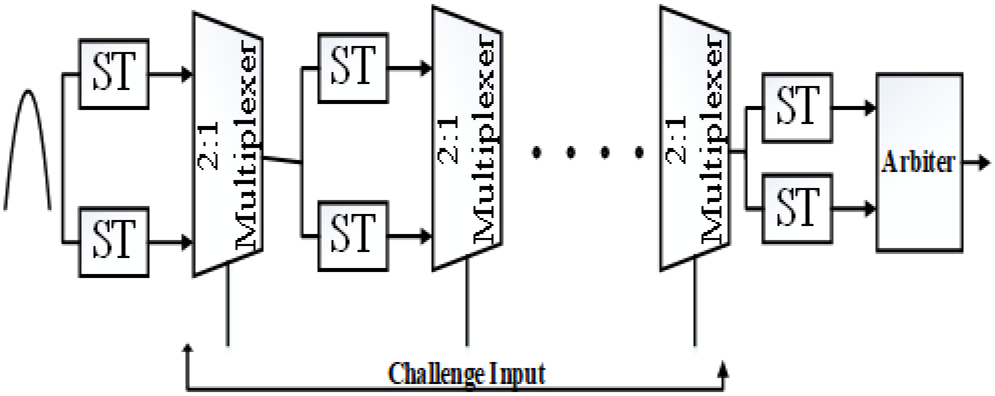


Figure 3.5 Module of Schmitt Trigger PUF

3.3.1 Different Topologies of Schmitt Trigger

The entropy of the path has been increased by using eight distinct ST architectures; at each ST stage, the incoming signals are modulated at varying thresholds and subject to atypical delays. Schmitt trigger takes a sample of input signal at its upper and lower trigger points. various STs offer various trigger points for the same input signal due to variations in delay and hysteresis width. STPUF is a joint effort of eight different ST architectural universities. Figure 3.6(a) depicts the current NOR based design [131], which uses parallel connections of NMOS from the Schmitt trigger (NORST) series to create a NOR circuit. Figure 3.6(b) depicts a Schmitt trigger with a dual threshold-based design, allowing for a high threshold for UTP and a low threshold for LTP. Figure 3.6(c) depicts the current Hex architecture MC14584B of Schmitt trigger (HexST), which is advantageous due to its low power consumption and good noise immunity. As shown in Figure 3.6(d) of [135]'s self-adjusting hysteresis (SAHST) design, the threshold voltage is determined by configuring the feedback path into ST. Figure 3.6(e) shows a standard CMOS Schmitt trigger (STDST) circuit with a programmable hysteresis and an on/off threshold from [132]. There are two conceivable configurations of feedback devices in the idealised scenario, one using UTP trigger edges and the other using LTP trigger edges. Figure 3.6(f) displays a LadderST [133] compensated ST design, which exhibits a monotone hysteresis response. Figure 3.6(g) [134] shows a 3-stage variation of the SOIST seen in Figure 3.6(e) [135], which is an SOI-compatible ST circuit.

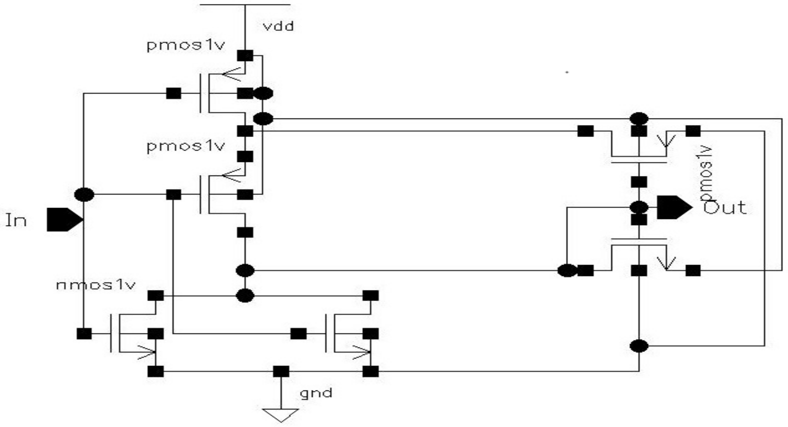


Figure 3.6(a) NOR based Schmitt Trigger

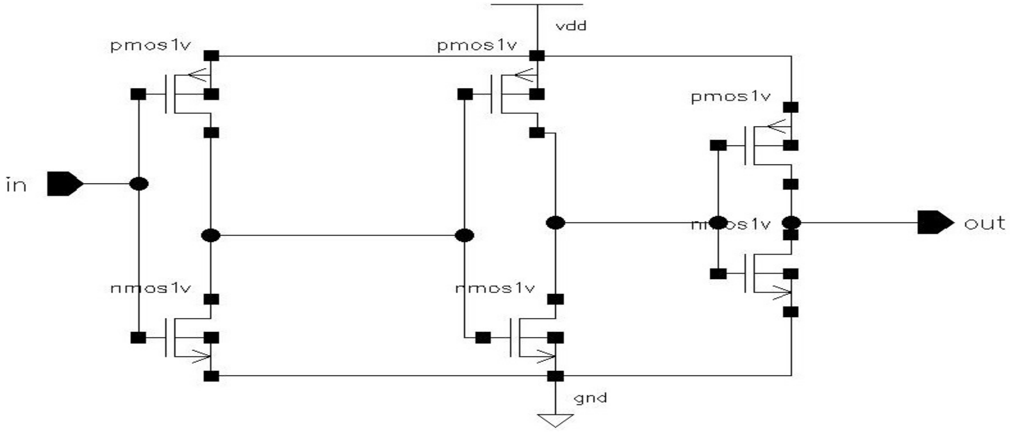


Figure 3.6(b) Dual Threshold based Schmitt Trigger

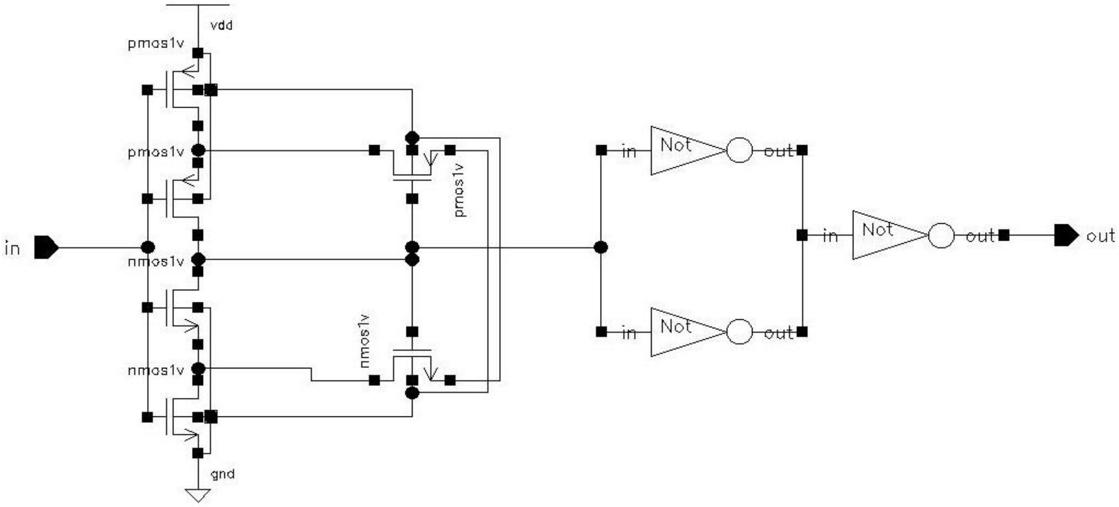


Figure 3.6(c) Hexadecagonal Schmitt Trigger

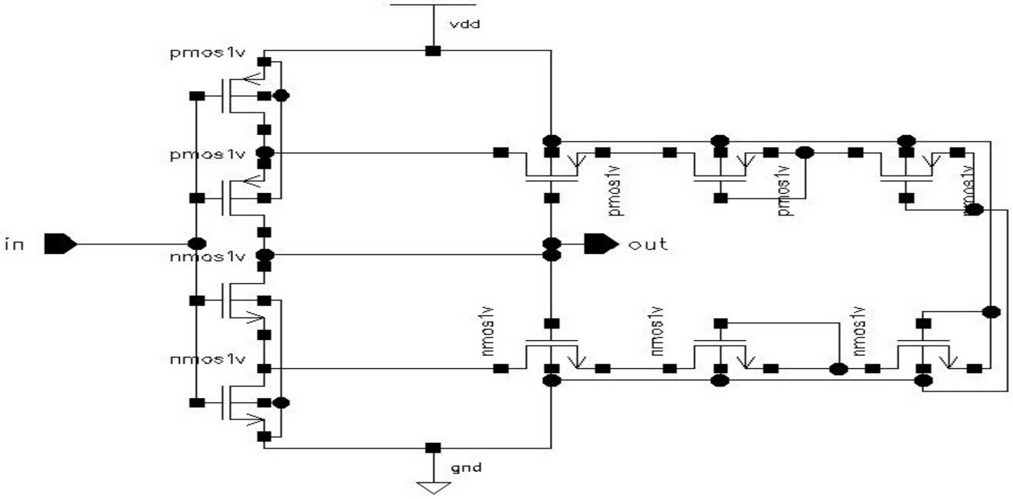


Figure 3.6(d) Self Adjusting Hysteresis Schmitt Trigger

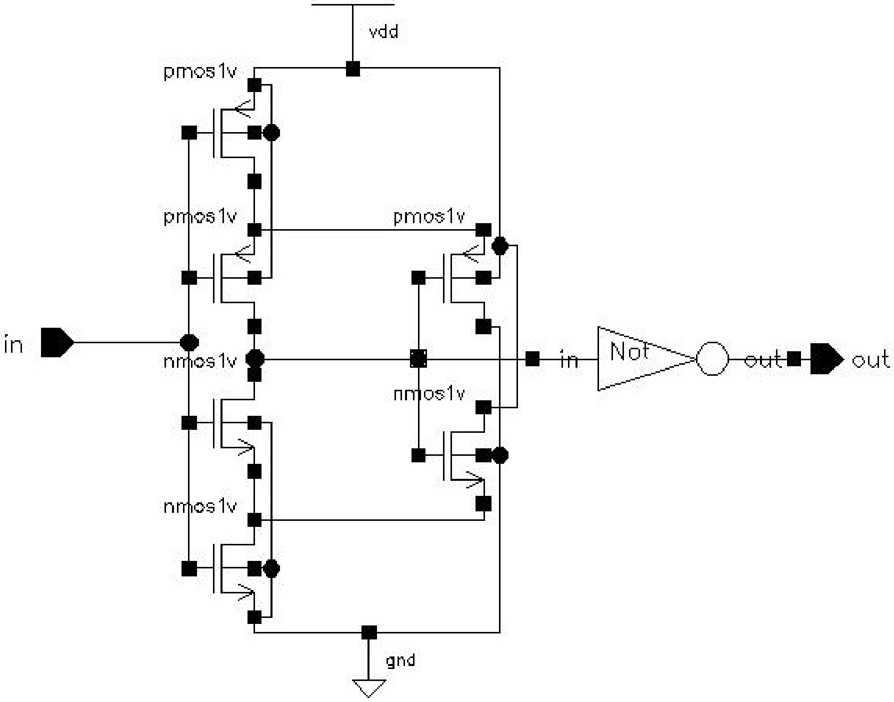


Figure 3.6(e) Standard CMOS Schmitt Trigger

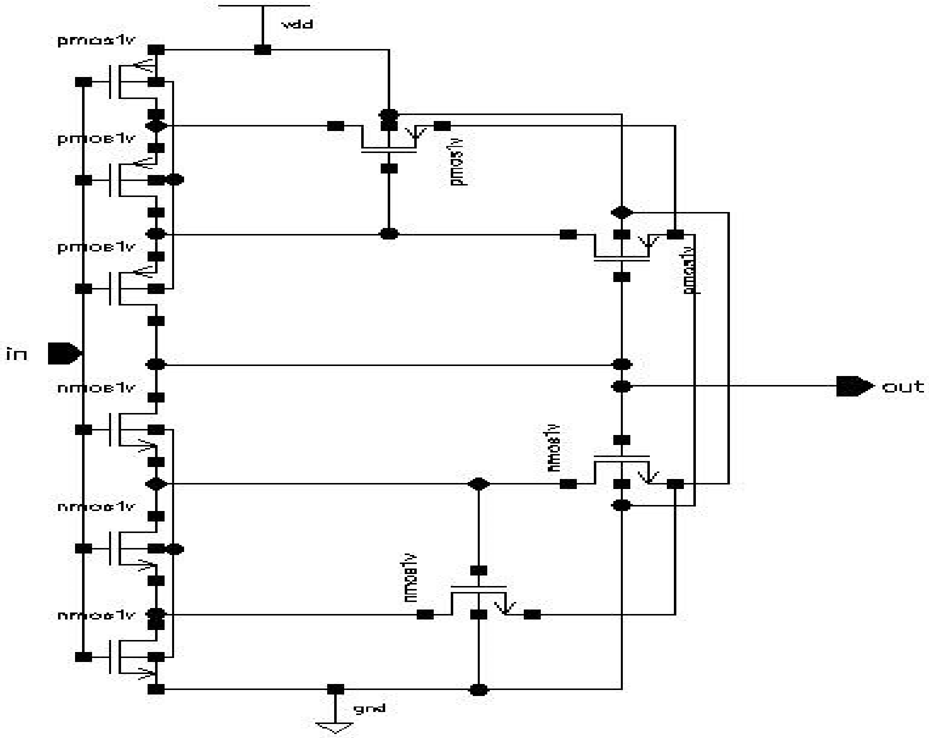


Figure 3.6(f) Ladder Schmitt Trigger

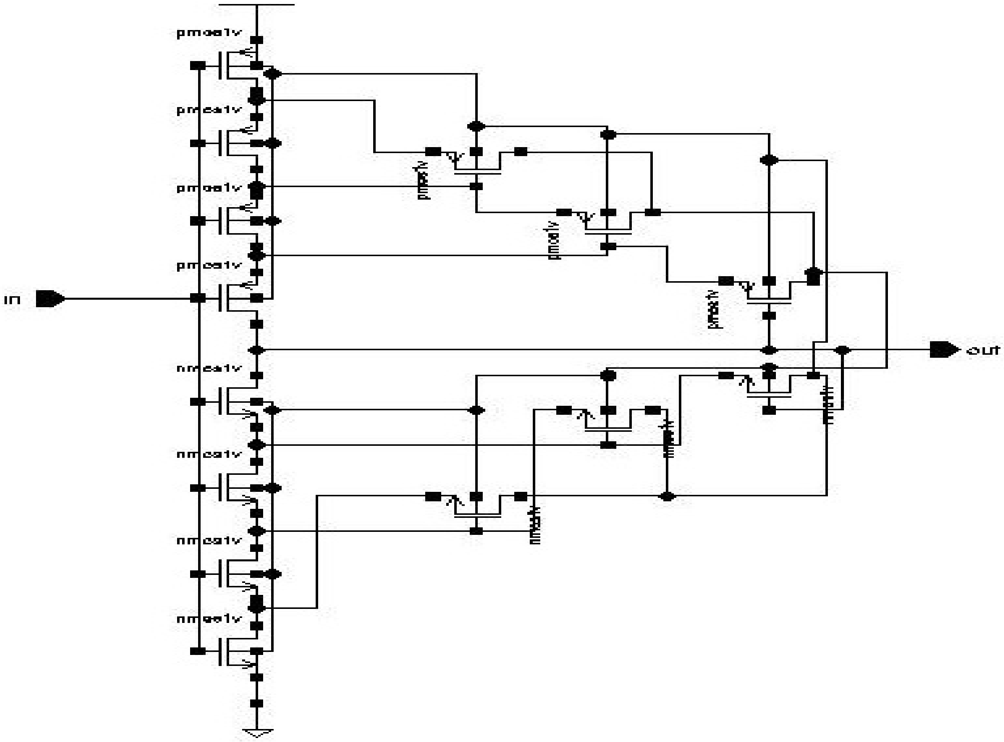


Figure 3.6(g) Silicon on Insulator Schmitt Trigger

Table 3.2 Comparision of Schmitt Trigger

Architecture	HTP(v)	Power (W)	LTP(v)	Hysteresis (v)	Delay (s)
ST	0.742	1.869u	0.357	0.385	53.43p
DTST	0.6	9.976u	0.42	0.234	531p
NORST	0.311	1u	0.286	0.025	49.19p
HexST	0.592	3.47u	0.358	0.234	96.74p
SAHST	0.466	1.422u	0.415	0.051	45.97p
SOIST	0.6	2.31u	0.28	0.32	101.2p
STDST	0.6	2.407u	0.368	0.232	69.96p
LadderST	0.597	6.919u	0.294	0.303	139p

Above table examines many different Schmitt trigger designs with respect to trigger point, hysteresis breadth, power, and latency. Adding PMOS in parallel with NMOS in series helps reduce the Schmitt trigger resistance's switching time (delay) and hysteresis width during pullup and pulldown. Modulation of the circuit's ferocity by the use of PMOS and NMOS in series and parallel. Given that adding NMOS devices in series raises the resistance and adding PMOS devices in parallel lowers it, the two should be connected in parallel whenever possible. Schmitt triggers based on NOR gates are imphasis resistant, and NORST provides more delay. In the instance of NORST, the LTP is at its lowest, whereas in the case of SAHST, it is at its highest.

Similarly, DTST/SOIST/STDST provide high HTP with declining input, whereas NORST provides a lower value of HTP. The hexadecagonal ST (HEXST) design outperforms the standard ST and STDST in terms of entire hysteresis width and predicted latency. The threshold voltage can be more precisely adjusted using a Silicon-on-insulator Schmitt-trigger (SOIST) or one with a two-stage feedback resistance. Huge latency and hysteresis width are features of SOIST. Two extra feedforwards are present in the CMOS SOIST circuit at the V+ and V- trigger edges, respectively. SAHST features a tunable threshold voltage and a higher amount of resistance because to the use of more MOS in series along the feedback circuit (3 PMOS and 3 NMOS). The upper and lower threshold voltages are set by the in-built switching voltage of the inverting stage in a dual threshold ST circuit, which has three inverting stages. The ST hysteresis width is between 0.025 and 0.385V, while the LadderST hysteresis width is between 0.385V and 0.625V. While LadderST produces a much longer period from the same input variability, NORST samples it into a small period rectangular pulse. Information is sampled into rectangles of the same width in all four sampling techniques (ST, DTST, HEXST, STDST). The bare minimum propagation delay for the ST circuit is 53.43ps. Trigger voltage is set by the DTST circuit since the inverting stage's built-in switching voltage allows for 531ps of top speed. Increasing the number of available components in an over-circuit will cause an increase in static power consumption, while an increase in switching activity will cause an increase in dynamic power consumption. DTST has the largest power consumption at 9.976W, while NORST has the lowest at 1W.

3.3.2 Arbiter

The arbitrator, represented by the decision device in Figure 3.9(a), sits at the very end of the PUF circuit and makes a "1" decision if the higher path is faster than the lower one, and a "0" choice otherwise. Figure 3.9 contrasts the upper edge with the trajectory of PUF [115], whose particles arrive first (b).

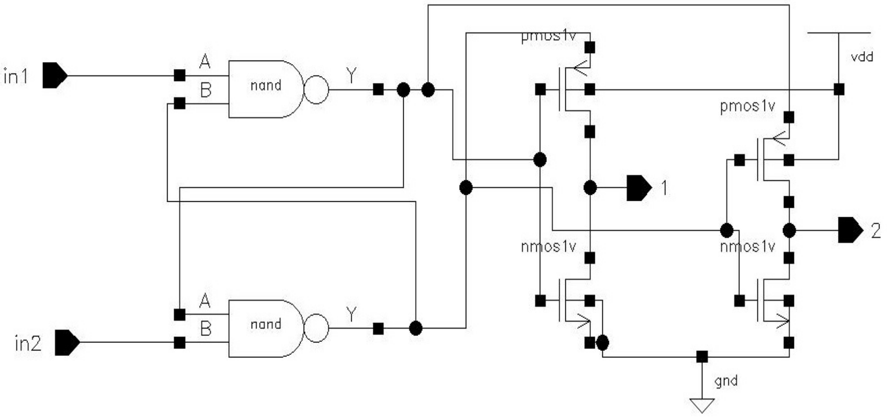


Figure 3.7(a) Internal architecture of Arbiter

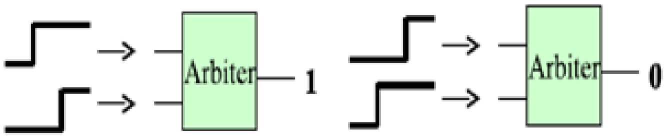


Figure 3.7(b) Response of an Arbiter

3.3.3 Schmitt Trigger PUF

Here, the response is dependent on not one but two CMOS parameters—delay and trigger voltage—whereas in prior PUF designs, the response was dependent on the input challenge and a single CMOS parameter. One such STPUF circuit is shown in Figure 3.10; it has 8 rows and 8 columns. For each column, a multiplexer-controlled Schmitt trigger group raced against one another. The delay, hysteresis breadth, and kind of ST architecture are all laid down in Table 3.2. A separate parallel line, made up of a distinct trigger circuit and then the multiplexer, connects each row of STPUF. Take PUF0 as an example; it has eight STDST and NORST routes. One Schmitt trigger has a maximum threshold voltage, whereas another has a minimum threshold voltage while still preserving the same hysteresis width as the first. A two-to-one multiplexer is built into each branch so that the choose line's challenge input can be routed to the appropriate branch. If the challenge bit on the input is set to 0, the multiplexer will switch the top and bottom paths, otherwise the signal will go through both paths. In PUF0, the input challenge C= 8'b10101010 causes the multiplexer to choose the path generated by STD, NOR, STD, LADDER, STD, STD, HEX, SOI. In each PUF, the trigger is programmed with a unique amount of hysteresis. Choose SOI, NOR, STD, LADDER, STD, ST, HEX, ST for PUF1's input, ST, LADDER, LADDER, SAH, NOR, SOI, SOI, SAH for PUF2, and LADDER, STD, LADDER, ST, ST, HEX, LADDER, SAH for PUF3's input. In a race to the finish, two parallel routes both lead to the arbitrator, with the intermediate signal cycling through the top and bottom rows according to the difficulty of the challenge. The input of the arbiter experiences a delay difference of D as a result of random process variation and a controlled trigger point. Additionally, a delay is introduced by the inclusion of an AND-OR or AND-NOT-OR gate in a common standard multiplexer when the output of the trigger is routed via the multiplexer. This is true even if the input signal from each trigger is identical. The input reaches the arbiter at a later time after going through eight steps of trigger-mux. Each trigger sends a unique signal that arrives at the arbiter at different times due to the delay associated with each. If the top signal arrives first and the bottom signal arrives quickly, the arbiter will return a high value; if the bottom signal arrives first, the arbiter will return a low value. Different voltage levels are produced in each column as a result of sampling and inverting the input signal many times. The cadence spectre calculator uses an A2D function to arrive at a numerical result.

Table 3.3 Placement of Schmitt trigger circuit into Row0 of PUF0

PUF0	STD	LADDER	STD	NOR	STD	SAH	HEX	ST	Arbiter
	NOR	NOR	HEX	LADDER	LADDER	STD	LADDER	SOI	

Table 3.4 Placement of Schmitt trigger circuit into Row0 of PUF1

PUF1	SOI	STD	STD	HEX	STD	STD	HEX	SAH	Arbiter
	SAH	NOR	HEX	LADDER	SAH	ST	SOI	ST	

Table 3.5 Placement of Schmitt trigger circuit into Row0 of PUF2

PUF2	ST	NOR	LADDER	STD	NOR	HEX	SOI	LADDER	Arbiter
	STD	LADDER	STD	SAH	STD	SOI	ST	SAH	

Table 3.6 Placement of Schmitt trigger circuit into Row0 of PUF3

PUF3	LADDER	SAH	LADDER	NOR	ST	STD	LADDER	SOI	Arbiter
	SAH	STD	NOR	ST	STD	HEX	SOI	SAH	

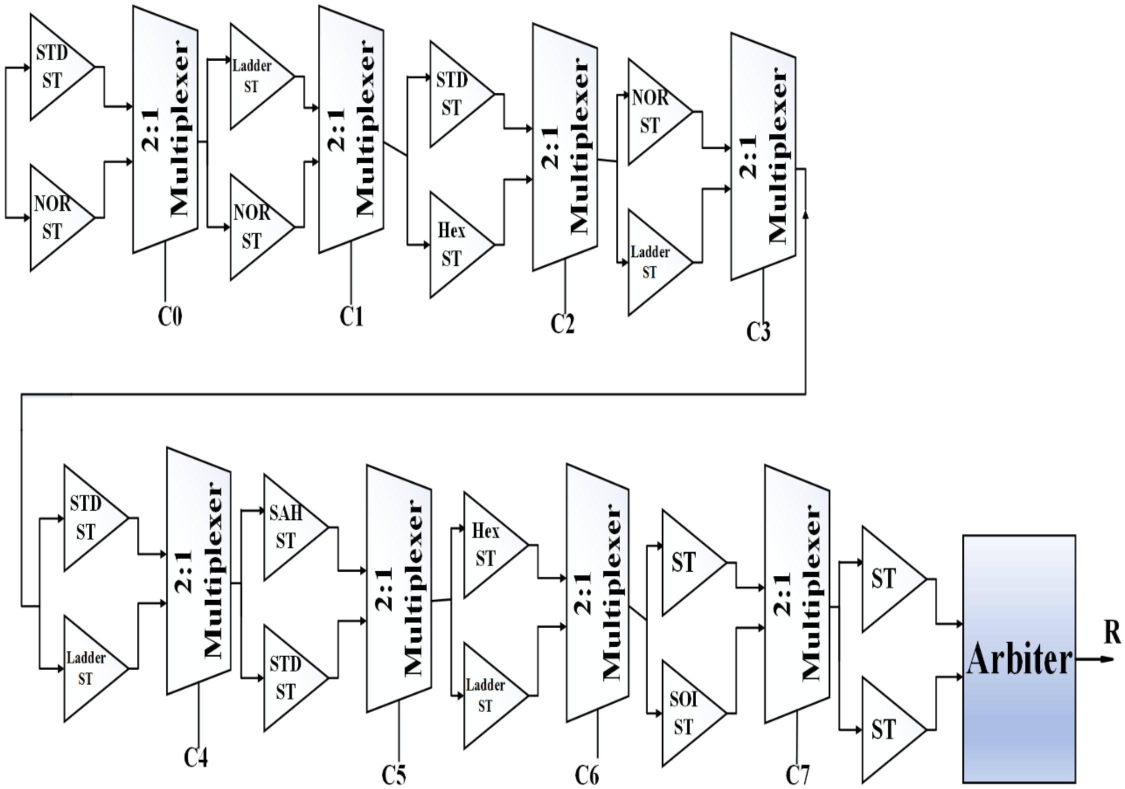


Figure 3.8 PUF0 first row schematic

By applying a ramp wave at the input and sampling each row 16 times, we may generate eight unique levels at arbiter response R0-R7. As can be seen in Figure 3.8, the arbiter's output is written to a D flip flop (DFF) at a rate of 1/8th the frequency of the input ramp wave. For each challenge, half of ST is triggered while the other half is left in an idle state, and the response at the output depends on the challenge, the trigger voltage, and the delay. All of the replies become a sortable database of problems. Due to the nonlinearity in the response pair, PUF design cannot be modelled using machine learning. Random replies are produced at varying intervals after being triggered by Schmitt's sample ramp waves. The generated answer depends on not just the input challenge, but also the delay and sampling nature of Schmitt triggers.

The Schmitt triggers circuit is used to propose a new design for the challenge-response generation technique in this work. A 32-bit solution is generated from an 8-bit input. Four 8-bit PUFs (PUFs 0 through PUF 3) collaborated when a ramp signal was entered. Delay flip-flops with parallel outputs rely on the host computer's clock. The triangle input period of 2000 ps was sampled ten times at 200 ps per sample by the clock signal. A variable input with a ramp signal width of only 200 ps is all that's needed to set off the trigger and maintain a steady output level. Each PUFMUX stage takes ten samples of the input per period at a clock frequency of 200 ps due to the fact that the HTP and LTP of a Schmitt trigger selective ST trigger input are in separate voltage ranges.

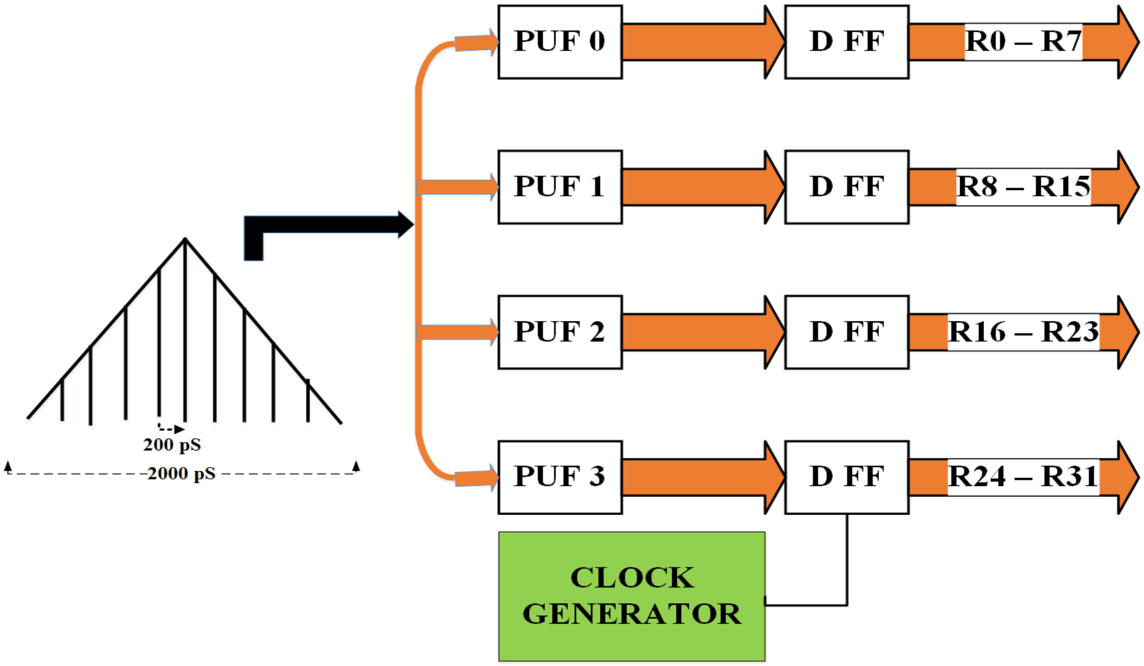


Figure 3.9 Simulation setup for PUF

Because each ST has its own delay-inducing trigger during this minor transition, the signal will arrive at the arbiter at different times. After 200 ps, DFF takes effect, and the arbiter's responses are "latched" to DFF's 8 bits. Figure 3.10 depicts the transient response of STPUF, obtained using Cadence spectre for the test input C=8'b10101010. There is a 5CH, 4CH, B3H, and 72H response from PUF0, PUF1, and PUF2 during the first clock period. We sampled the rising and falling input five times throughout one input one cycle in each PUF to get ten responses. This method greatly improves throughput since it allows for the reception of 40 answers inside a single clock cycle.

3.4 Result and Discussions

PUF circuits stipulate that the response bits from different PUFs must be distinguishable for the same challenge input, that the response bits must be consistent with operating circumstances, and that the bit distribution must be uniform. The PUF circuit's efficiency was measured with both external and internal variations in the response bit. The author proposes a number of parameters, [122] measuring attributes in terms of randomness, correctness, stability, diffusionism uniqueness, and [39] measuring qualities as uniformity, uniqueness, bit aliasing, and dependability. Uniformity, originality, and dependability are the three conventional measures of quality (described in [136]).

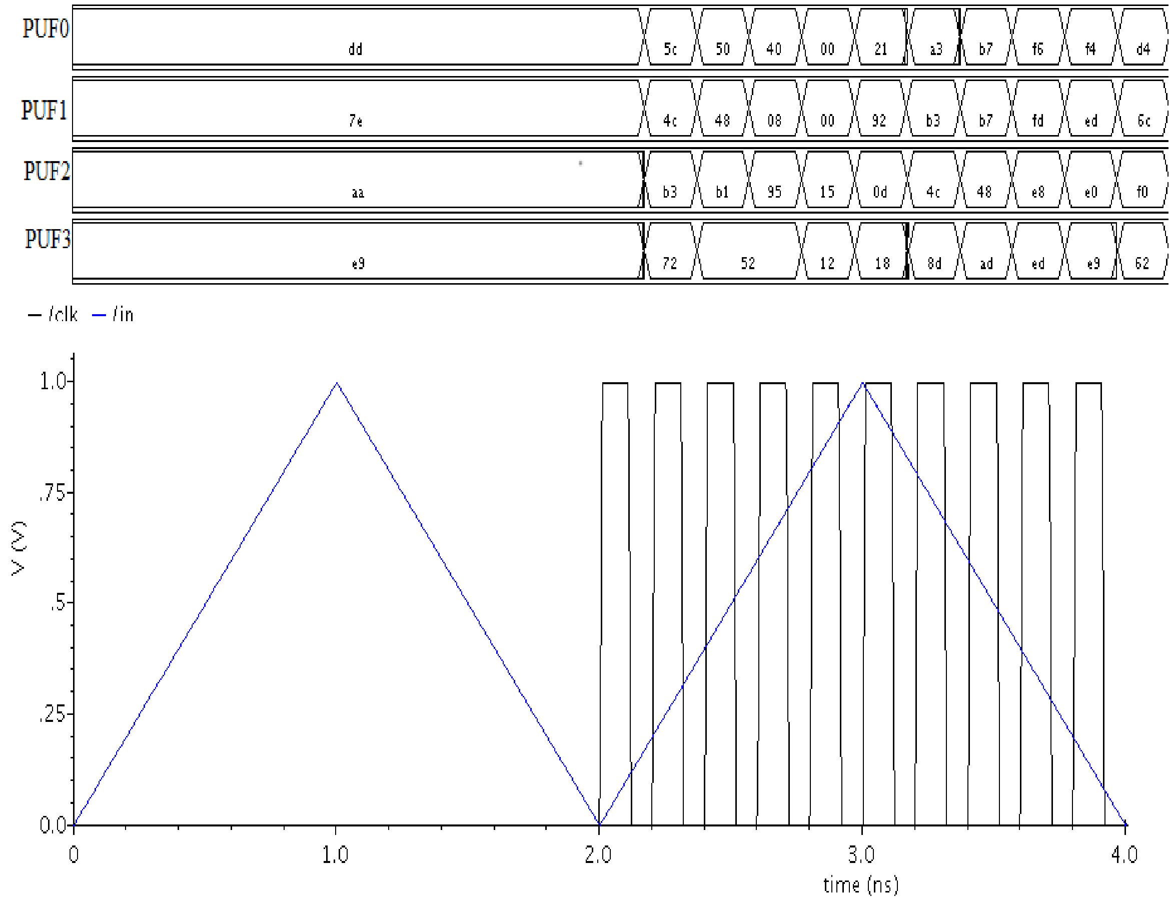


Figure 3.10 Transient Analysis of PUF

3.4.1 Inter PUF Variation

Differences between PUF instances or chips are called inter-PUF variance. Manufacturing process variances, temperature fluctuations, and other variables might affect PUF reactions. PUF design should encourage inter-PUF variance to make each PUF instance distinctive. PUF assumes that input to subsequent PUF responses for the same challenge must be distinct. The amount of bits that separate two PUF's continuous responses to the same input challenge is an estimate of the inter-PUF variation. The number of bits that are different when the identical input is applied to two separate PUFs is measured using the hamming distance [36]. When calculating the inter PUF variance for n bits per response from k devices, the typical measurement is calculated using the following equation (3.7)

$$InterHD = \frac{2}{k(k-1)} \sum_{i=k}^{k-1} \sum_{j=i+1}^{k} \frac{HD(R_i, R_j)}{n} \times 100\% \qquad (3.7)$$

To calculate the HD between two PUF responses, R_i and R_j, for the same challenge input, we use the Hamming distance (HD) between any two responses.

3.4.2 Intra PUF Variation

Intra-PUF variance is the difference in PUF responses over numerous measurements or incidents. Environmental variables like temperature and voltage variations and ageing affect this variation most. particular intra-PUF variance is expected and accepted, but excessive variation might be undesirable in particular applications. If you make a one-bit change to the input, the intra variation is the distance between the bits in the responses you get from different PUFs. A one-bit flip at the input terminal should result in a change of at least half the bit, ideally more. Variation within a PUF measured by the hamming distance of the reaction to an external stimulus. An estimate of the robustness of the response bit is its intra PUF variation. In order to calculate the intra hamming distance for a given number of bits per response from a certain number of devices, we use the following equation: (3.8)

$$Intra\,HD = \frac{1}{K} \sum_{i=1}^{k} \frac{HD(R_{i1}, R_{i2})}{n} \times 100\% \qquad (3.8)$$

The hamming distance between the first and second response bit samples from the same PUF is denoted by HD (R_{i1}, R_{i2}).

3.4.3 Uniformity

Equal bit-rate responses are evaluated. The response from a PUF should be completely unpredictable (on that particular chip), with no bias in the response bits. The likelihood with which PUF generates the responses "1" and "0" is what we call its uniformity. The hamming weight, as shown in equation (3.9), measures consistency by tallying the number of high bits in the PUF's response. 50% is the sweet spot for this.

$$Uniformity = \frac{1}{n}\sum_{j=1}^{n} R_{i,j} \times 100\% \qquad (3.9)$$

Where $R_{i,j}$ is the ith chip's jth bit of an n-bit response.

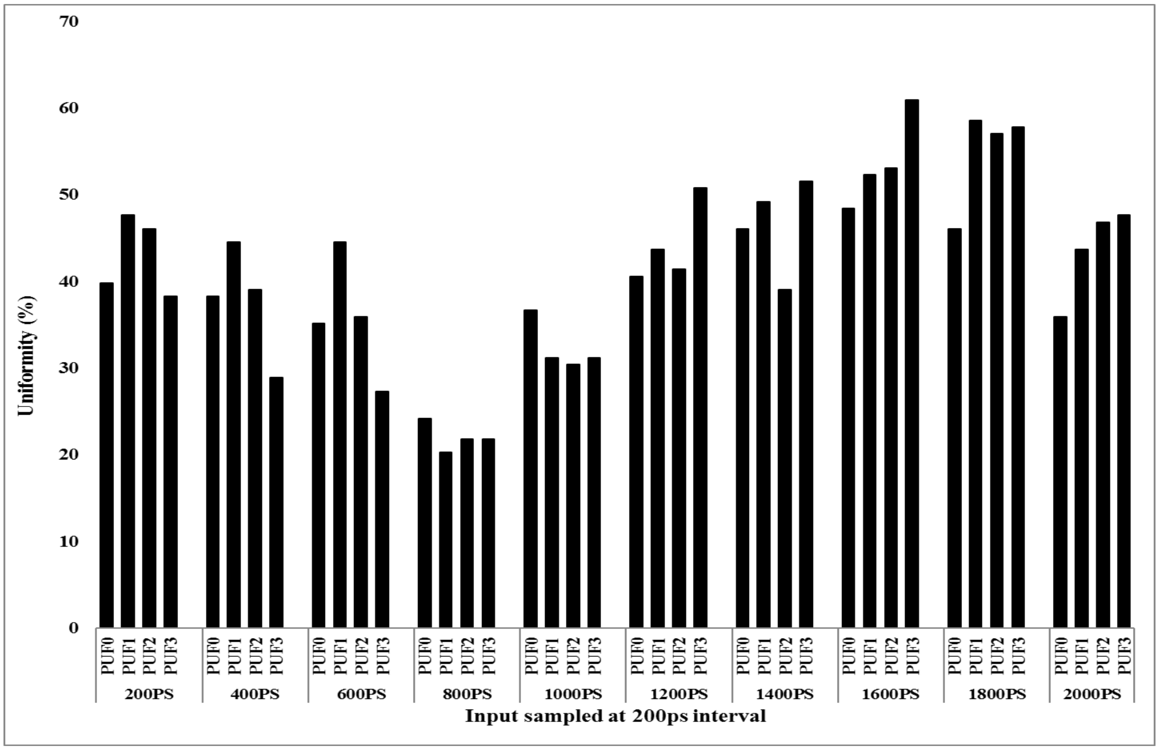


Figure 3.11 Uniformaty of STPUF

Bits in responses are sampled at a rate of 200 ps to calculate STPUF uniformity; uniformity decreases at the rising and falling edges of the input ramp signal. Figure 3.11 shows that the homogeneity is nearly at the 50% mark for the seventh clock cycle. For 4 PUF, the average uniformity value is between 20 and 60.93 percent.

3.4.4 Bit Aliasing

The aliasing of bits is a proxy for the bias of bits in a response that can be attributed to static hazards. Some of the response bits are hard-coded as '0' if the bias is to ground or '1' if it is to the power supply, as shown in [29]. Repeated PUF responses to the same challenge from different PUF provide the attacker with valuable, non-public information that can be used to 0forecast the PUF's next move. If bit aliasing happens, various chips will be responsible for producing the response. This means a potential attacker can accurately anticipate the defence. Bit aliasing for the ith bit of a PUF over K individual chips for a challenge determined as shown in Figure 3.12. (3.10).

$$Bit\ Alia\sin g = \frac{1}{k}\sum_{j=1}^{k} R_{i,j} \times 100\% \qquad (3.10)$$

In which $R_{i,j}$ represents the i-th sample's response.

It was found that the 2nd bit of the PUF response in the 4th sample is skewed, although the ideal bit aliasing metric is 50%. A ramp wave with a rising input is more likely to cause a stuck-at-0 fault, while a ramp wave with a falling input will more likely to cause a stuck-at-1 fault.

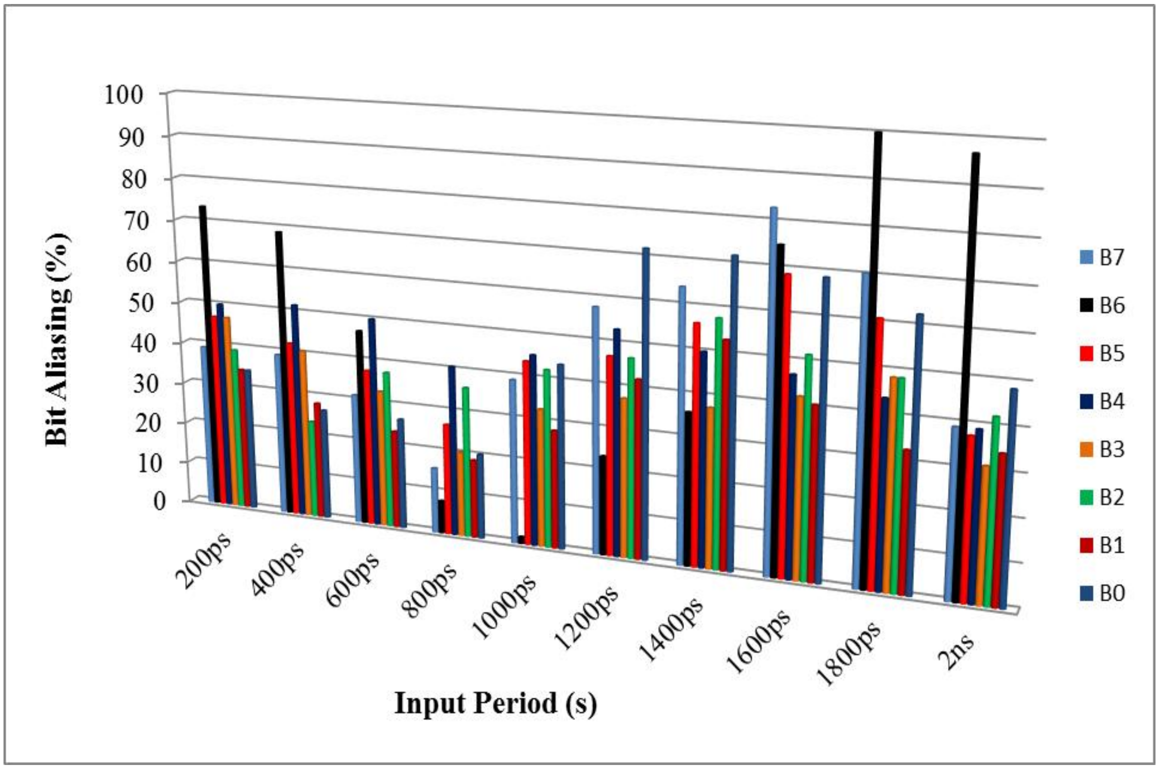


Figure 3.12 Bit Aliasing of ST-PUF

3.4.5 Uniqueness

The ability of PUF to generate novel solutions to a problem is a criterion for comparing PUF. Each PUF must have its own unique response to the same input. Inter PUF variation (inter hamming distance) [118, 137] measures how differently two PUF give the same unique response and may be compared using this formula: (3.11). The ideal uniqueness of a PUF is half, indicating a bias toward '0,' while a larger biasness indicates a bias toward '1,' with 50% of the answer bit being between two PUF.

Uniqueness = 1 – Average inter-PUF HD

$$InterHD = \frac{2}{k(k-1)}\sum_{i=k}^{k-1}\sum_{j=k+1}^{k}\frac{HD(Ri,Rj)}{n}\times 100\% \qquad (3.11)$$

If two chips i and j get the same input challenge and chip number k, then their n-bit answers, R_i and R_j, will be the same.

Figure 3.13 shows that during the fourth clock period, uniqueness has dropped to 34%, and that it approaches an ideal value of 50% during the falling input of the ramp input. Uniqueness often falls between 34% and 55% on average.

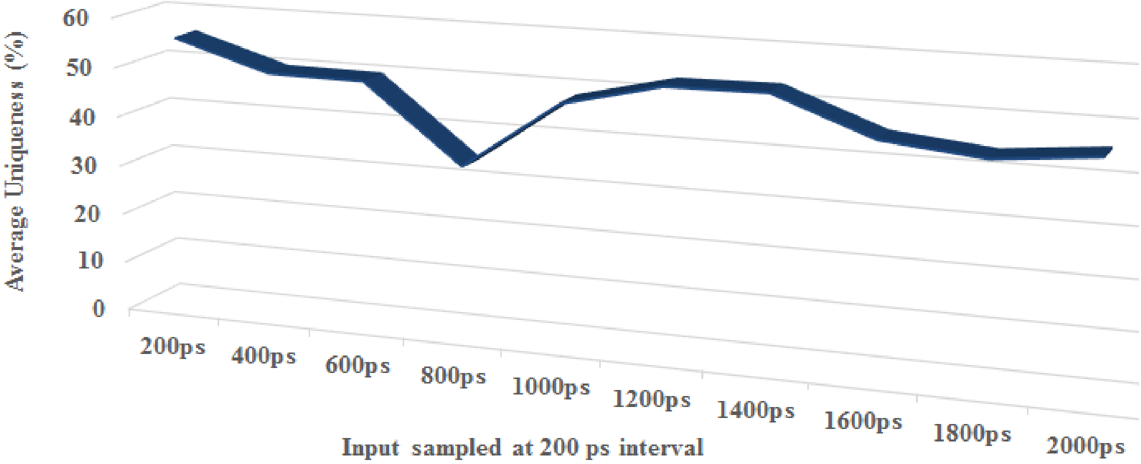


Figure 3.13 Uniqueness of ST PUF

3.4.6 Reliability

The behaviour of a PUF under extremely varied and hard operating conditions while responding to an input that requires a steady response every time, such as a ramp. For PUF to be reliable, it must produce the same output regardless of changes in environmental conditions like temperature and power input voltage [138]. Intra PUF variation, calculated by analysing the hamming distance produced in a variety of settings, is a common metric used to assess the quality of a product's reliability. The equation can be used to determine the k response based on data obtained at varying voltages and temperatures (3.12). The optimal intra-chip variance for a 100% reliable response is zero.

$$\text{Reliability} = 1 - \text{Average intra chip HD}$$

$$\text{Average Inter chip HD} = \frac{1}{K}\sum_{j=1}^{K}\frac{HD\,(Ri,Rij\,)}{n} \times 100\% \tag{3.12}$$

This study investigated the reactions from 4 PUF in figure 3.14 across a temperature range of 20°C to 80°C and a supply voltage range of 1.2 V to 1.5 V in the laboratory. According to Fig. 3.16, PUF1 and PUF3 have a 99.54 percent and 99.86 percent dependability for the first and fourth clock periods, respectively. Figure 3.15 displays the simulation result showing that increasing the voltage improves reliability. Maximum value of reliability 99.875 is attained when comparing rising and decreasing input (1000 ps - 2000 ps). The results of comparing the proposed STPUF to the already existing MUXPUF and ROPUF are shown in Table 3.7. Assuming a linear input, STPUF can produce ten replies in a single period, with the parameter displayed as a range, and the best result being the segment number measure, where a segment is a delay in time of 200 ps.

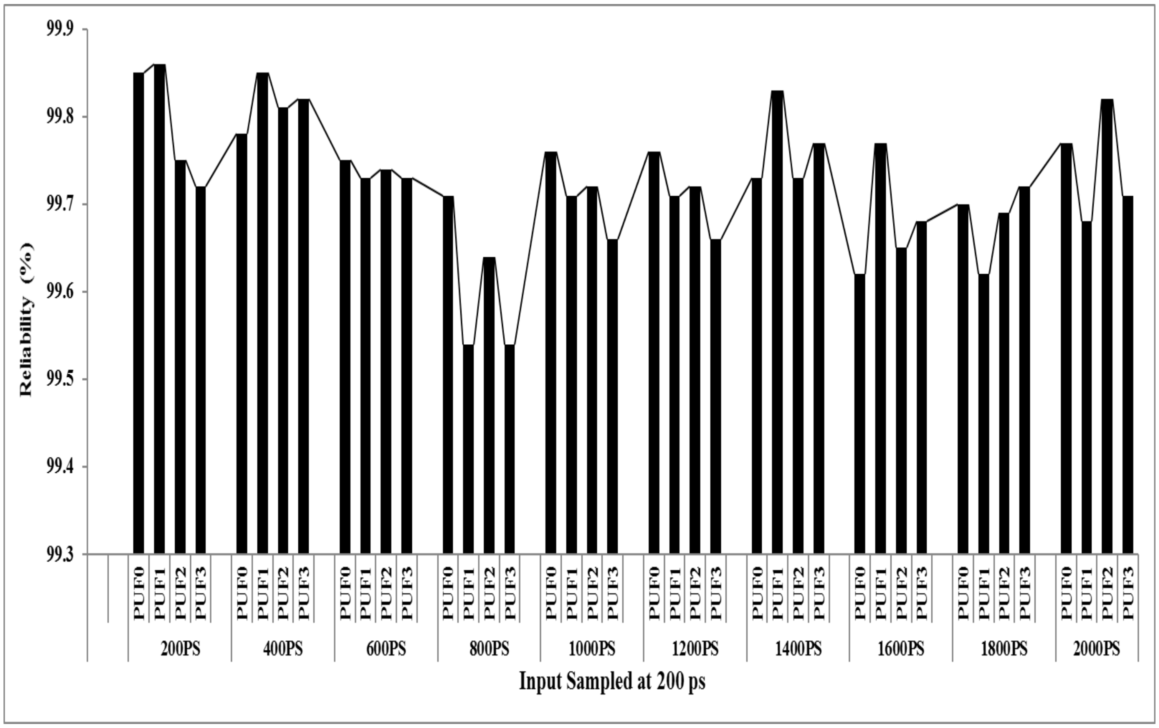


Figure 3.14 Reliability of ST-PUF with temperature variation

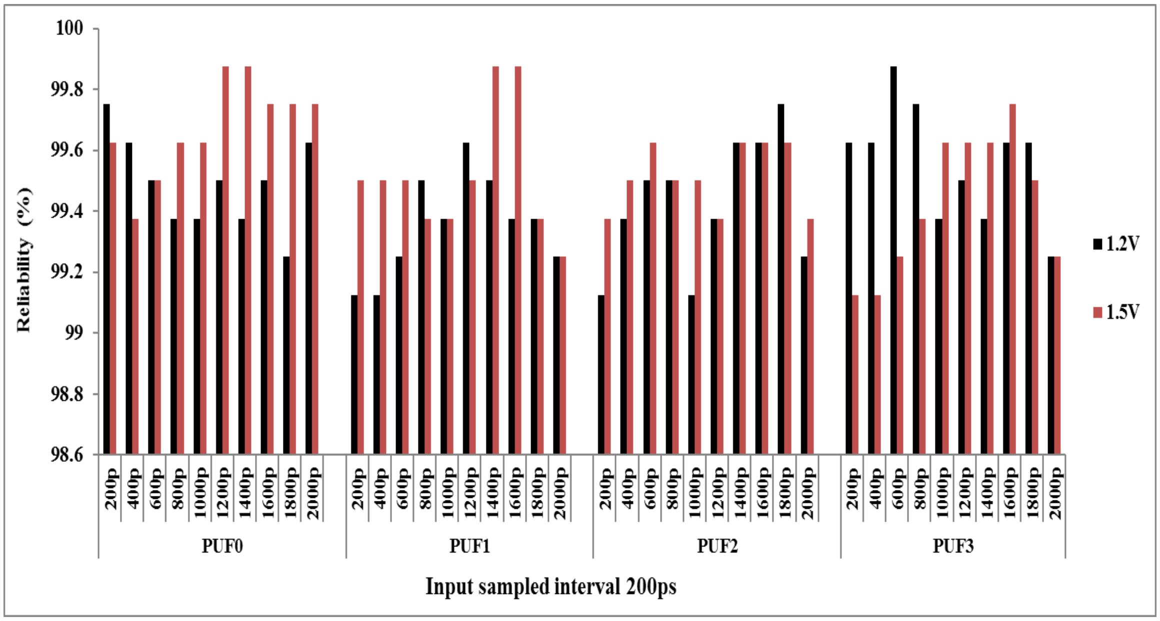


Figure 3.15 Reliability of ST-PUF with supply voltage variation

There is maximum uniformity (49.8%) in the seventh segment, and maximum distinctiveness (49.2%) in the second. At the optimal temperature and combination, 99.875% reliability is seen, while at the optimal temperature and combination, 99.875% reliability is observed for supply voltage variations. We see that the linearly falling input yields the uniform response, but the linearly increasing input can yield more varied outcomes.

Table 3.7 Comparison of PUF's Parameter

Parameter	Ideal Value [107]	STPUF		[105] Mux PUF	[19] Mux PUF	[39] RO PUF	[40] RO PUF
		Range	Best Observation For ramp input				
Uniformity (%)	50	34 – 55.5	7th Segment 49.8	30.26	_ _	50.56	_ _
Uniqueness (%)	50	20.3 – 60.93	2nd Segment 49.2	67.44	46.15	47.24	38.96
Reliability (%)	100	99.25 – 99.6875* 99.125 – 99.875**	1st, 7th Segment 99.6 2nd,10th Segment 99.8	98.01	48	99.14	98

3.5 Proposed PUF Evaluation

Based on the physical characteristics of the challenge, such as delay and hysteresis, proposed STPUF creates a customised answer. It's worth checking out for use in authentication and key management.

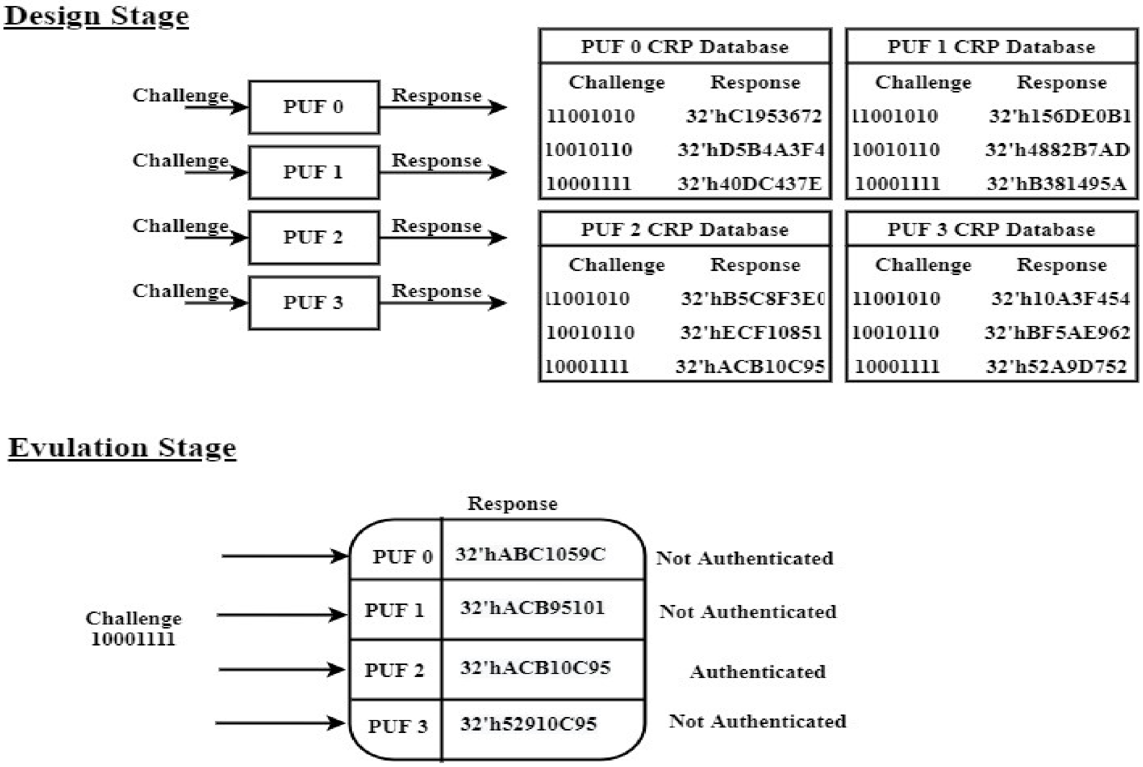
3.5.1 Authentication

We create four distinct PUF architectures, each of which generates a unique challenge and corresponding response pair that we then save in our database. Presented in Figure 3.18 is a collection of three different challenges and responses. Each PUF has a unique reaction to the specified challenge. The authentication system on a device is safe with PUF's protection. If the user wants to verify the device's authenticity during testing, they can issue a challenge like C=

8'b1000111 and receive a response like 32'hACB10C95. Responses are compared with those in the CRP database once they have been acquired. Since the response matched the PUF2 CRP, only PUF2 users with authentic access to the device are shown. If a counterfeiter stores every potential CRP, they will be unable to evaluate the response because it is computed as a function of a physical property of the Schmitt trigger, which they do not know.

3.5.2 Secret Key Generation

While a PUF won't save the actual key, it will save a trace of the key in the form of "helper data." Enrollment and generation phases of the error correction procedure are complete. Errors in the ECC code are corrected by the surrounding environment. Helper information, often a parity bit, is generated by XORing the responses from PUF(R) and ECC encoding during the enrolment step. In the Hamming code method, the position of the 20th, 21st, 22nd, and so on, weight bits is augmented by a parity bit. By inserting four parity bits into the 12-bit hamming code C= 12'b000001100110, the input to the ECC encoder, S=8'b00001101, is improved. In this case, the XOR operation between ECC syndrome and a 12-bit PUF answer (12'b010100101001) yields the assistive information H=12'b010101001111. Data used as an aid are kept in non-volatile memory, where they can be accessed by anyone and can be a security risk. Due to the obfuscation of PUF's answer, an attacker with access to this data cannot determine the key. To do error rectification, we need the supplementary information.



PUF 0 CRP Database	
Challenge	Response
11001010	32'hC1953672
10010110	32'hD5B4A3F4
10001111	32'h40DC437E

PUF 1 CRP Database	
Challenge	Response
11001010	32'h156DE0B1
10010110	32'h4882B7AD
10001111	32'hB381495A

PUF 2 CRP Database	
Challenge	Response
11001010	32'hB5C8F3E0
10010110	32'hECF10851
10001111	32'hACB10C95

PUF 3 CRP Database	
Challenge	Response
11001010	32'h10A3F454
10010110	32'hBF5AE962
10001111	32'h52A9D752

	Response	
PUF 0	32'hABC1059C	Not Authenticated
PUF 1	32'hACB95101	Not Authenticated
PUF 2	32'hACB10C95	Authenticated
PUF 3	32'h52910C95	Not Authenticated

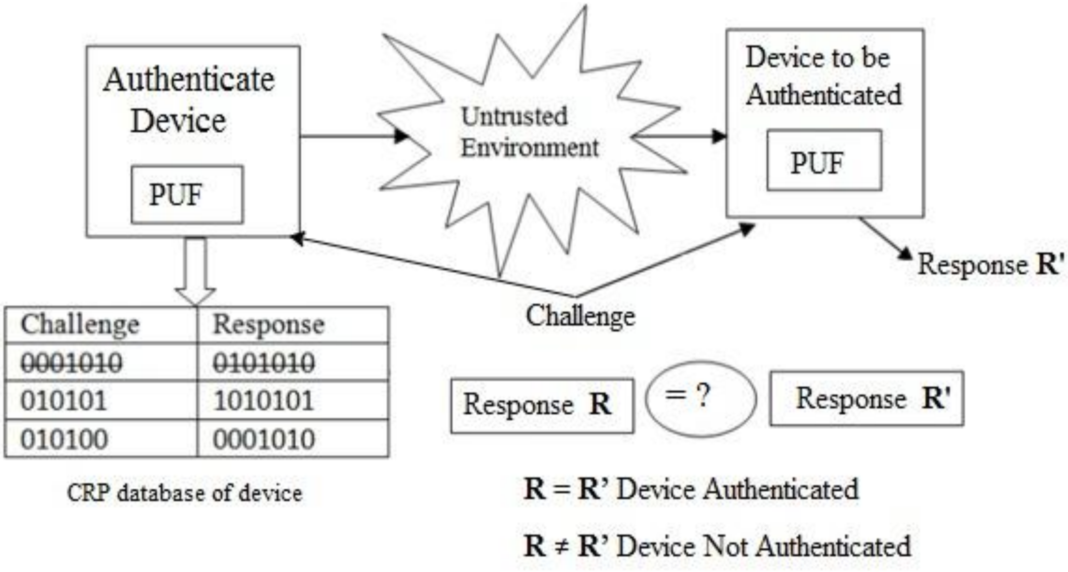
Figure 3.16 ST-PUF Authentication Evaluation

3.6 Applications of PUF

In [109], the author proposes two specialised uses for PUF: inexpensive authentication and cryptographic key creation.

3.6.1 PUF Based Authentication

Secure authentication follows the structure shown in Figure 3.1, wherein the first step is to build a database containing many challenge-response combinations, and the second step is to actually authenticate the user. On account of their non-linear behaviour, PUFs each provide a one-of-a-kind solution to each problem. Recording a PUF output and comparing it to a regenerated one [110] is all an adversary can do, but he has no idea which hardware feature was chosen. To clone the ICs, he will need to remember every possible CRP pair.



Challenge	Response
~~0001010~~	~~0101010~~
010101	1010101
010100	0001010

CRP database of device

Figure 3.17. PUF based Authentication[19]

When products are being made, a huge number of CRP pairs are made and kept in secure locations as part of the production process. The customer receives PUF. It is a challenge to PUF that anyone, even in an untrusted setting, attempts authentication. If the recorded response is very similar to the recorded response, then the IC has been validated. The authentication is successful if and only if the response from the trusted database matches the information on the valid IC. After authentication has taken place, the utilised pair must be deleted to prevent a man in the middle attack.

3.6.2 Cryptographic Key Generation

Like technology improves, we become more reliant on IC to securely handle sensitive data, as when using a smart card for a financial transaction. Importantly, IC guarantees the secrecy of critical information. Keeping a key in non-volatile memory is the standard practise (EEPROM). Storing the key in memory is a security risk since an attacker can access it using invasive and non-invasive attacks. A PUF is a rudimentary security element that serves to generate keys rather than store them. The response cannot be used for secure keys since PUF is susceptible to noise and mistakes. The ECC resolves the discrepancy in the answer. In PUF-based key generation, there are two stages. Figure 3.2 from discusses the process of key generation and key extraction using a symmetric key [111, 112].

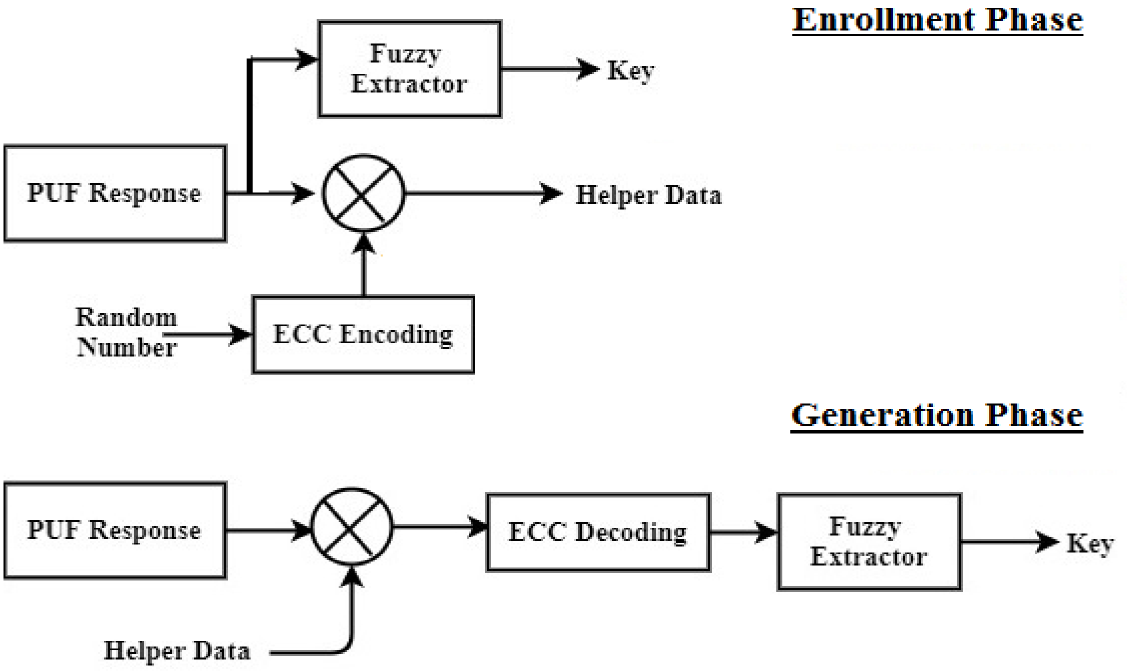


Figure 3.18. Cryptographic Key Generation[109]

Enrolment phase XOR operations on ECC encoder syndrome and PUF's response yield memory-resident aid information. Similar PUF responses (obtained by using the same challenge during enrolment) are XORed with auxiliary data, and then the ECC decoder is used to extract the same key. There is no need to keep a secret key with this method, simply auxiliary information. Even when relaying aid data, the public key's perfect secret is revealed because to the clever use of its physical characteristics for secret storage.

Chapter 4

Mode based Obfuscation

4.1 Introduction

VLSI and embedded systems use DSP processors. Complex DSP processors need a simple mechanism to synchronise data flow. Real-time signal processing multiprocessor scheduling for iterative dataflow methods. Digital signal preparation computations are iterative, therefore data flow graphs (DFGs) describe them. Data flow graphs have nodes, operators, and connections. computational problems and correspondence [139, 140]. Recursive and flexible advanced channels use criticism rings to limit cycle or test time. The cycle time frame or focus bound is used to emphasise the bound of any hub [141,142,143]. Computations require this cycle limitation regardless of technology. When multiple processors are available for recursive computation, it is illogical to anticipate a cycle period that is not the limitation [144]. DFG computations restrict attention timeframes. Representing a calculation may change its meaning. A DFG's cycle bound is essential to a strategy. The strategy struggles to develop with a non-cycle focal time. Rate-ideal booking diagrams for iterative information flow should determine the limit [141,145,146]. Excellent schedules have cycle periods equal to the attention limit. In response to coerced booking, a planning calculation is created using the equipment assets. The number of processors needed to finish computations may exceed available assets. To avoid overlap, these operations should be prioritised or requested. [144,147,148]. A complex Gaussian mixture model technique that consumes a lot of CPU cycles and is hard to apply in real time on embedded devices [148]. To select the priority with the lowest cycle bound, the emphasis bound must be established for each priority. After calculating the emphasis limit, the cycle limit must be estimated quickly. The Longest Path Model (LPM) was recently created to calculate the focus bound [149]. This method still determines sign preparation computation limits. This approach generates lattices from the DFG and analyses components to find the emphasis bound. This study assesses LPM, which is successful but time-consuming, and provides a novel way for controlling figure emphasis bound of limited drive response (FIR) frameworks with inline delay components. This unique LPM-based technique should compute by rearranging network components. Thus, it reduces computation time while maintaining LPM dependability. VLSI research issues include FPGA DSP algorithm computer model implementation [150]. FPGA fault-tolerance analysis requires a complex division module that can handle computationally intensive workloads [151].

As more design companies exit manufacturing, the semiconductor integrated circuit supply chain is becoming increasingly disjointed. As a result, it becomes more difficult to keep high standards of quality and safety throughout all stages of production. Damage to devices, loss of employment, and even death can result from hardware trojans, intellectual property (IP) infringement, overproduction of integrated circuits (IC), reverse engineering, side channel attack, and counterfeiting [152]. Thus, hardware security is crucial. Hardware obfuscation, which hides design functionality, prevents several of the above attacks. This paper discusses hardware obfuscation.

Methods that fall into the passive category include HDL code modifications [153], while methods that fall into the active category include register transfer level (RTL) gate insertions [154], combinational logic modifications [155], and netlist-level obfuscation [156]. DSP circuit obfuscation cannot be overcome by any of the aforementioned methods. This section elaborates on sophisticated hardware obfuscation techniques. This kind of obfuscation is implemented with a counter reset on a Radix-2 real FFT [158]. There is no methodical approach to modifying the control flow provided in this study. DSP circuit obfuscation has a dual purpose. For starters, control-flow must be hidden in DSP circuits since they are so control-driven. Characteristics like as filter taps, FFT length, etc. provide bounds on the performance, size, and power of DSP circuits. This data is likewise concealed via obfuscation. Here, we employ a mode-based obfuscation technique borrowed from [157]. There might be sensible and illogical implementations of the idea. The state of the system's data path and the control-flow output from the control path are both affected by the configuration of a key. A method block diagram is shown in Figure 4.1. Multiple obfuscation strategies present only partial truths.

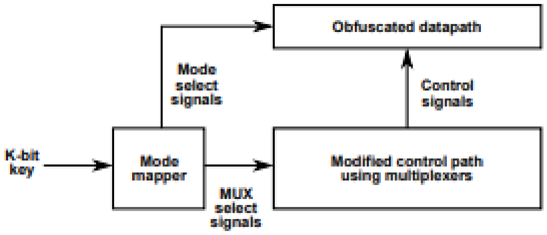


Figure 4.1 Design-based mode-based obfuscation flow

Obfuscation protects the circuit two ways. Since keys are written into the chip after manufacture, locking the design prevents piracy and overproduction. The foundry cannot access the key, making additional circuit copies useless. It also makes reverse-engineering semiconductors harder. The design's control-flow is hidden, and its various modes prevent rivals from using design characteristics to save circuit cost or power. Product life cycle management prioritises gadget security during development and manufacture. Integrated circuit fabrication includes several firms in different countries, making it complicated. Counterfeiting, illegal overproduction, piracy, etc. have cost semiconductor manufacturers a lot of money [159, 160]. Reverse engineering and malicious circuit insertion make circuit reliability and security harder. Obfuscation involves secretly encrypting a system and hiding its design and operation. Hardware obfuscation hides design functionality and prevents black box use with secret keys. Cryptography involves encrypting plaintext and sending information via untrusted channels. The receiving end decodes using the same secret key. The design house keeps the circuits hidden. Foundry doesn't have key. The design house or a trustworthy third party programmes keys after production. Figure 4.2 shows this analogy.

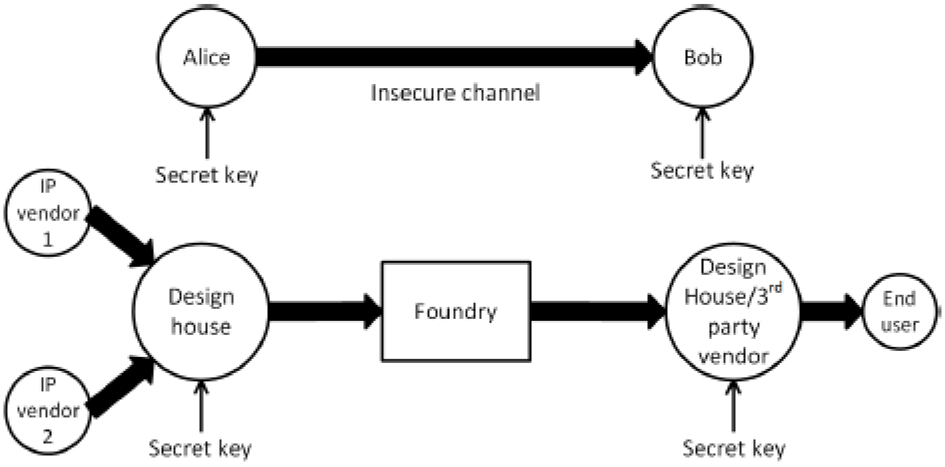


Figure 4.2 Comparison of Encryption Scheme

This work addresses functional obfuscation, where keys govern circuit functioning. Functional obfuscation assumes that physically comparable circuits can compute different functions, whereas physically dissimilar circuits can calculate the same function [161]. Contrast functional obfuscation with logic encryption, which employs keys as inputs to logic gates to conceal arbitrary logic [155]. Despite their diversity, hardware obfuscation methods [155,156,162–165] have downsides. Current technologies are vulnerable to assaults on circuits incorporated with these obfuscation techniques. Some solutions cost space, energy, or time, while others are difficult to develop and implement. Here, we propose dynamic obfuscation. Dynamic obfuscation occasionally malfunctions. With fixed obfuscation, erroneous keys always produce inaccurate results. Dynamic obfuscation improves key size and security. Hardware trojans inspired dynamic obfuscation. The security study details two brute-force and reverse engineering risks. Sequential circuits are used to demonstrate space and power overhead trade-offs. Dynamic and existing obfuscation methods are compared.

4.2 Iteration Bond

4.2.1 Longest Path Matrix Technique

The iteration bound may be found by studying the diagonal elements of the matrices constructed in this fashion [149,166-169]. It was discovered that a polynomial time based technique may be used to calculate the iteration period bound with the applied longest path matrices multiplications [149]. It is possible to calculate the total computational latency for the directed loop in each graph in order to estimate the boundaries. Lev Reyzin has devised an algorithm that reconstructs a degree for a confined tree from a distance matrix that provides a runtime analysis for topological trees [166] and has also investigated a number of graph algorithms. For the examination of run time, a real-time operating system (RTOS) was created. The hardware accelerators were also designed and built on an FPGA using the point-to-point fast simplex link bus to connect it to the Xilinx soft-core processors. The predictability and performance matrices of such hybrid RTOS are superior than software-only design and analysis methodologies [167]. A kernel matching pursuits (KMPs) is utilised to get coefficients encoded with advanced encryption standard (AES) using the listless partitioning embedded block for the secured image compression encoding method [168]. The excellent PSNR values and secure compression of pictures are significant benefits of this method.

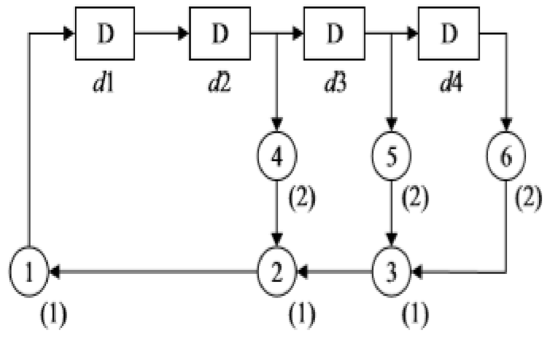


Figure 4.3. Data Flow Graph of finite impulse response system

Adaptive systems were created with a reconfigurable finite impulse response (FIR) filter as a tool for online fault detection and investigation [169]. The proposed design was modelled and verified using Verilog HDL and synthesised on the software tool Xilinx VIVADO. The design was beneficial for online fault modelling and verification with an area-efficient data path, which is currently a major concern in the VLSI design, testing, and verification process. The primary objective of this article is to propose a column shifting algorithm-based iteration bound calculation with fewer delay elements and less computation time. Let d be the total number of delay elements in a DFG. $L^{(m)}$, where m=1,2,..., d, matrices are formed so that the element $l_{i,j}^{(m)}$'s value has the greatest calculation time of all pathways from delay element d_i to d_j that traverse precisely (m-1) delays. If no route exists, $l_{i,j}^{(m)}$= -1 is returned. The layout of the matrix's elements is illustrated below. The longest route matrix approach is presented to determine the iteration bound of a DSP application (3)

$$L^{(m)}=\begin{bmatrix} l_{1,1} & \cdots & l_{1,n} \\ \vdots & \ddots & \vdots \\ l_{n,1} & \cdots & l_{n,n} \end{bmatrix}$$

Consider the DFG illustrated in Figure 4.3. To decide $l_{4,1}^{(1)}$ for the given DFG, all pathways that traverse exactly delay elements from the d_4 delay element to the d_1 element must be examined. To compute element, $l_{4,1}^{(1)}$, there is only one path: d_4 -> n_6 ->n_3->n_2->n_1->d_1. The result of the path's computation time is five, hence $l_{4,1}^{(1)}$= 5. There is no path from delay element d_3 to delay element d_2, hence the expected calculation time for element is $l_{3,2}^{(1)}$ = -1. All the members of the $L^{(1)}$ matrix are calculable as

$$\mathbf{L}^{(1)}=\begin{bmatrix} -1 & 0 & -1 & -1 \\ 4 & -1 & 0 & -1 \\ 5 & -1 & -1 & 0 \\ 5 & -1 & -1 & -1 \end{bmatrix}$$

In order to construct matrices of higher order, such as $L^{(m)}$, where m=2,3,..., d, the DFG need not be found directly. These items may be calculated recursively using the formula (1) as

$$l_{i,j}^{(m+1)} = \max_{k \in K}(-1, l_{i,j}^{(1)} + l_{k,j}^{(m)}) \qquad (1)$$

Where K is the set of numbers in [1,d] for which neither $l_{i,k}^{(1)}$ =-1 nor $l_{m,j}^{(m)}$ =-1 holds. To illustrate, the first step in computing $l_{3,2}^{(2)}$, is to calculate the set K from the set 1,2,3,4. The K has the value 1 since $l_{3,1}^{(2)}$=5 and $l_{3,1}^{(2)}$=0 are both true, and the values k=2,3, and 4 are not present because at least one of the $l_{3,k}^{(1)}$ or $l_{k,2}^{(1)}$ is equal to -1 for each of these values. Using K=1, it is possible to compute the value of $l_{3,2}^{(2)}$ using equation (1) as

$$l_{3,2}^{(2)} = \max_{k \in \{1\}}(-1, l_{3,k}^{(1)} + l_{k,2}^{(1)})$$

With K = 1, the computed element $l_{3,2}^{(2)}$, value is 5. Using the equation (1), it is also possible to calculate the remaining components of the matrix $L^{(2)}$ as

$$\mathbf{L}^{(2)} = \begin{bmatrix} 4 & -1 & 0 & -1 \\ 5 & 4 & -1 & 0 \\ 5 & 5 & -1 & -1 \\ -1 & 5 & -1 & -1 \end{bmatrix}$$

The matrix $L^{(2)}$ is constructed using just $L^{(1)}$, whereas the matrix $L^{(3)}$ uses both $L^{(1)}$ and $L^{(2)}$. To calculate$l_{2,3}^{(3)}$, for instance, K=2 is utilised since $l_{2,3}^{(3)}$=0 and $l_{2,3}^{(3)}$=4 are computed for k=1,3,4 and at least one of $l_{1,k}^{(1)}$ or $l_{1,k}^{(1)}$ is -1. The element $l_{2,3}^{(3)}$ is computed as 4.

Using the equation (1), the successively remaining elements of matrices $L^{(3)}$ and $L^{(4)}$ may be determined.

$$\mathbf{L}^{(3)} = \begin{bmatrix} 5 & 4 & -1 & 0 \\ 8 & 5 & 4 & -1 \\ 9 & 5 & 5 & -1 \\ 9 & -1 & 5 & -1 \end{bmatrix} \quad \mathbf{L}^{(4)} = \begin{bmatrix} 8 & 5 & 4 & -1 \\ 9 & 8 & 5 & 4 \\ 10 & 9 & 5 & 5 \\ 10 & 9 & -1 & 5 \end{bmatrix}$$

Using the $L^{(m)}$ matrices computed, the iteration limit may be established.

The iteration bound for the supplied DFG and L(m) matrices can only be determined by inspecting the diagonal elements from the top left to the bottom right of each matrix, as specified. The studied iteration bound is

$$T_\infty = \max\{\frac{4}{2}, \frac{4}{2}, \frac{5}{3}, \frac{5}{3}, \frac{5}{3}, \frac{8}{4}, \frac{8}{4}, \frac{5}{4}, \frac{5}{4}\}$$

$$= 2 \text{ u.t.}$$

The calculated iteration bound of DFG is 2 u.t.

4.2.2 Proposed Algorithm

In accordance with the previously mentioned longest route matrix approach, the component $l_{i,j}^{(m)}$should be handled by first generating a $L^{(1)}$ lattice and then determining the remaining components via conditional figuring (1). The greatest hindrance to utilising the longest path network strategy is calculation time. Focusing on FIR frameworks with inline delay components, we investigated an alternative network component behaviour. The suggested method can reduce the time required to calculate the emphasis bound. The proposed calculation requires segment relocation; however, this calculation does not need to determine every element of the $L^{(m)}$

framework; it just needs to register the important elements. The suggested Network Component Moving approach needs computing the L$^{(1)}$ framework from DFG and then just three components of the major section for successive higher grids, as opposed to the sixteen components required by the LPM technique for a FIR framework with four defer components. The remaining components of the prior framework may be eliminated by relocating the segment from left to right and removing the most right-most section of the previous network. This just requires the corner-to-corner components of a grid, ensuring that only essential components are calculated.

***Algorithm 1* Column Shifting Algorithm to compute iteration bound of a given DFG**

Here, the requirement is to obtain a suitable DFG to FIR system with inline delay.

1. Compute L$^{(m)}$ matrix from DFG where m=1.

2. Compute $l_{r+1-t,1}^{(m+t)}$ elements using equation (4) where r=1,2,…,d-1 and the value of t increases by t=1,2,3….,d-1 for each higher-order matrix.

$$l_{p,1}^{(m+t)} = max_{k\in K}(-1, l_{p,k}^{(m)} + l_{k+1}^{(m+t-1)}) \qquad (4)$$

Where K is the set of integers k in the interval [1,d] so that $l_{p,k}^{(m)} = -1$ and $l_{k,1}^{(m+t)} = -1$ does not hold.

If the number of elements computed for L$^{(m+t)}$ matrix in step 2 is equal to d-t then, go to step 3

Else

Recheck step 2

3. Shift the elements of L$^{(m+t-1)}$ to the right and discard the last column elements $l_{p,d}^{(m+t)}$ such that $l_{p,q+1}^{(m+t)} = l_{p,q}^{(m+t-1)}$,where p and q =1,2,3,…,d and place these elements directly in L$^{(m+t)}$ matrix.

4. Repeat step 2 and step 3 until the computation of L$^{(d)}$

If L$^{(m+t)}$=L$^{(d)}$ then

Compute iteration bound using equation (2)

Else

Repeat step 4.

4.2.2.1 An Illustration of the Column Shifting Algorithm

Consider the same DFG as that depicted in Figure 1. The iteration bound for the provided DFG has previously been determined using the LPM method mentioned above. Now, the iteration bound is computed and compared to the output of the LPM technique using the column shifting algorithm. The algorithm will only work with FIR systems that contain inline delay elements.

Initial Verification

Due to the fact that the supplied DFG is a FIR system with inline delay elements, the iteration bound can be determined using this approach.

The Initial Step

By inspecting the path of DFG, the elements of matrix $L^{(m)}$ where m=1 can be determined using the technique outlined in the LPM approach. The value of d=4 is decided by the DFG's delay elements. To determine the DFG, all pathways that pass across delay elements from d4 to d1 should be evaluated. To compute element, the sole way is d_4->n_6->n_3->n_2->n_1->d_1. The path's computation time is 5, which is why the element $l_{4,1}^{(1)}$ is equal to 5. For element, it is worth noting that there is no path connecting the delay element d_3 to the delay element d_2, implying that the element's $l_{3,2}^{(1)}$ processing time is equal to -1. All of the $L^{(1)}$ matrix's elements can be found as

$$\mathbf{L}^{(1)} = \begin{bmatrix} -1 & 0 & -1 & -1 \\ 4 & -1 & 0 & -1 \\ 5 & -1 & -1 & 0 \\ 5 & -1 & -1 & -1 \end{bmatrix}$$

Step 2

To compute higher-order matrices like L$^{(2)}$, L$^{(3)}$,…., L$^{(d)}$, equation (4) is used to compute $l^{(m+t)}_{r+1-t,1}$ elements where r=1,2,….,d-1 and the value of t increases by t=1,2,…,d-1 for each higher-order matrix.

In matrix L$^{(2)}$, the $l^{(2)}_{1,1}, l^{(2)}_{2,1}$ and $l^{(2)}_{3,1}$ elements need to be computed and we can simply discard the computation of $l^{(2)}_{4,1}$ as according to the algorithm the maximum elements count in up to $l^{(2)}_{r+1-t,1}$ for r=3 and t=1 because we are not going to use that element for computation. To compute $l^{(2)}_{1,1}$ elements for r=1 and t=1.

$$l^{(2)}_{1,1} = \max_{k\in\{2\}}\left(-1, l^{(1)}_{1,k} + l^{(1)}_{k,1}\right) = 4$$

By using equation (4) the value of remaining elements $l^{(2)}_{1,1}, l^{(2)}_{2,1}$ and $l^{(2)}_{3,1}$ are computed as 4,5 and 5 respectively. The resultant matrix till now have element shown below that is computed using step 2 is

$$\mathbf{L}_a^{(2)} = \begin{bmatrix} 4 & - & - & - \\ 5 & - & - & - \\ 5 & - & - & - \\ * & - & - & - \end{bmatrix}$$

Now it is required to compute the remaining element of matrix L$^{(2)}$ except the element $l^{(2)}_{4,1}$

Step 3

To compute the remaining elements of L$^{(2)}$ matrix, we need to shift the elements of L$^{(m+t-1)}$ to the right and discard the last column elements $l^{(m+t)}_{p,d}$ such that $l^{(m+t)}_{p,q+1} = l^{(m+t-1)}_{p,q}$, where m=1,t=1, p and q =1,2,…,d and place these elements directly in L$^{(m+t)}$ matrix. As an example, the column become $l^{(2)}_{p,4} = l^{(1)}_{p,3}$ and $l^{(2)}_{p,2} = l^{(1)}_{p,1}$. After shifting, the elements of the matrix L$^{(1)}$ will be

$$\mathbf{L}_b^{(2)} = \begin{bmatrix} - & -1 & 0 & -1 \\ - & 4 & -1 & 0 \\ - & 5 & -1 & -1 \\ - & 5 & -1 & -1 \end{bmatrix}$$

In given matrix * represents that the element will not be required to compute iteration bound and hence it's good to decline the computation to save computational time. Combining the elements of $L_a^{(2)}$ and $L_b^{(2)}$ matrix from step 2 and step 3 the resultant L$^{(2)}$ matrix will be computed as

$$\mathbf{L}^{(2)} = \begin{bmatrix} 4 & -1 & 0 & -1 \\ 5 & 4 & -1 & 0 \\ 6 & 5 & -1 & -1 \\ * & 5 & -1 & -1 \end{bmatrix}$$

Step 4

Now we have to repeat step 2 and step 3 to compute L$^{(3)}$ and L$^{(4)}$ matrices as

$$\mathbf{L}^{(3)} = \begin{bmatrix} 5 & 4 & -1 & 0 \\ 8 & 5 & 4 & -1 \\ * & 5 & 5 & -1 \\ * & -1 & 5 & -1 \end{bmatrix} \mathbf{L}^{(4)} = \begin{bmatrix} 8 & 5 & 4 & -1 \\ * & 8 & 5 & 4 \\ * & 9 & 5 & 5 \\ * & 9 & -1 & 5 \end{bmatrix}$$

To compute matrix L$^{(3)}$ all the elements of the column should match as $l_{p,4}^{(3)} = l_{p,3}^{(2)}$ and $l_{p,2}^{(3)}$= $l_{p,1}^{(2)}$ and so on. In the first column of L$^{(3)}$ i.e. in the $l_{p,1}^{(3)}$ the column only the computation of $l_{1,1}^{(3)}$ and $l_{1,2}^{(3)}$ elements are required and hence we need not compute the elements $l_{1,3}^{(3)}$ and $l_{1,4}^{(3)}$. Then we need to shift the elements to the right from matrix L$^{(2)}$ to get all the remaining elements.

In the case of L$^{(4)}$matrix only $l_{1,1}^{(4)}$ the element needs to be computed and thus we need not compute the remaining element of that column and to get the rest of elements of remaining columns we can get from matrix L$^{(3)}$.

Calculation of Iteration Bound

After getting the elements of all four matrices, we can compute iteration bound of the given DFG same we do in the LPM method. Thus we need to examine diagonal elements of every matrix from the top-left element to bottom right element as

$$\mathbf{L}^{(1)} = \begin{bmatrix} \mathbf{-1} & 0 & -1 & -1 \\ 4 & \mathbf{-1} & 0 & -1 \\ 5 & -1 & \mathbf{-1} & 0 \\ 5 & -1 & -1 & \mathbf{-1} \end{bmatrix} \quad \mathbf{L}^{(3)} = \begin{bmatrix} 5 & 4 & -1 & 0 \\ 8 & \mathbf{5} & 4 & -1 \\ & 5 & \mathbf{5} & -1 \\ & -1 & 5 & \mathbf{-1} \end{bmatrix}$$

$$\mathbf{L}^{(2)} = \begin{bmatrix} \mathbf{4} & -1 & 0 & -1 \\ 5 & \mathbf{4} & -1 & 0 \\ 6 & 5 & \mathbf{-1} & -1 \\ & 5 & -1 & \mathbf{-1} \end{bmatrix} \quad \mathbf{L}^{(4)} = \begin{bmatrix} \mathbf{8} & 5 & 4 & -1 \\ & \mathbf{8} & 5 & 4 \\ & 9 & \mathbf{5} & 5 \\ & 9 & -1 & \mathbf{5} \end{bmatrix}$$

Using equation (2) the iteration bound T_∞ can be calculated as given

$$T_\infty = \max \left\{\frac{4}{2},\frac{4}{2},\frac{5}{3},\frac{5}{3},\frac{5}{3},\frac{8}{4},\frac{8}{4},\frac{5}{4},\frac{5}{4}\right\}$$

$$= 2 \text{ u.t.}$$

The iteration bound of the given DFG is 2 u.t. The iteration bound computed using this proposed algorithm matches the iteration bound computed using the Longest Path Matrix algorithm described above. Table 4.1 gives a comparison among the computed number of elements with exiting LPM and the proposed column shifting algorithm.

Table 4.1 Comparison of the computed number of elements using LPM and proposed Column Shifting Method

Matrix (d)	**Matrix Size**	**LPM Method**	**Proposed Algorithm**	**Reduction from LPM**
L$^{(1)}$	4X4	d^2 =16	d^2 =16	0.00%
L$^{(2)}$	4X4	d^2 =16	d-1=03	81.25%
L$^{(3)}$	4X4	d^2 =16	d-2 =02	87.50%
L$^{(4)}$	4X4	d^2 =16	d-3 =01	93.75%
Total Elements		64	22	65.13%

4.2.3 Comparison with the LPM Technique

In this part, we have conducted a comparative analysis of the computational complexity between the LPM methodology and the recommended CS method. The comparison is conducted by considering the solution of the previous example for the DFG as seen in Figure 1. The iteration bound for the given Data Flow Graph (DFG) necessitates the creation of four matrices, each consisting of 4x4 elements, due to the inclusion of four delay components (d=4). The required calculations for both methodologies are presented in Table 4.1. The comparison is focused solely on the computation of components using equation (4), as modifications to elements include the reallocation of memory addresses. Based on the comparison, it can be inferred that the reduction of new element computations may be achieved with the mere rearrangement of elements. The table illustrates the inverse relationship between the value of d and the quantity of calculated elements in the initial column. Instead of the variable "d," it is sufficient to compute the value of only one element.

The quantity of processed components for the major grid $L^{(1)}$ is d X d for both approaches. By examining the pathways and nodes, these components are determined to be lawfully registered with the DFG. For higher-request lattices, the LPM approach necessitates the calculation of d X d components for each framework; for d=4, the total number of calculated components is 64 for all grids. The required number of components for each successive higher request lattice can be resolved using the proposed CS technique as d-1 for $L^{(2)}$, d-2 for $L^{(3)}$, d-3 for $L^{(4)}$, and d-n+1 for $L^{(m)}$, where n is the total number of components in a segment and d=4. These components are specifically required for section. The total number of registered components using the procedures illustrated is 64 when LPM is used and 22 when recommended strategies are used. In comparison to the LPM strategy, just 34.87 percent of components are required to calculate the cycle bound. Components are reduced by 65.13 percent. According to the association shown in Table 1, the drop in computation complexity increases as the demand for framework rises and the amount of components required decreases. For d=6, we must process d2=36 components for each lattice using the LPM strategy, but this technique, the final framework $L^{(6)}$, requires just the calculation of one component because the emphasis bound requires only corner to corner components.

Additionally, the proposed model of iteration-bound computation column shifting method can be expanded for implementation on FPGA in order to do hardware exploration. With the use of

FPGAs, fault tolerance can be added to the existing model, which is a critical worry for VLSI design for testing. Numerous algorithm models are created and implemented on FPGA as a hardware description, taking into account issues such as chip power consumption, associated delays, hardware complexity, failure models, and secure designs.

4.3 Obfuscation Methodology

Here, we will get into the fundamentals of creating a mode-based obfuscated layout. These techniques leverage standard transformations and tunable DSP circuitry characteristics to affect outputs in various ways. A fundamental FFT circuit is used to demonstrate these concepts.

4.3.1 Folding Transformation

Folding is a high-level method that allows designers to create time-multiplexed designs [170]. For a certain folding factor N, the same set of hardware functional units can perform N algorithm operations in N clock cycles. The inputs to the functional units of a folded or time-multiplexed circuit are chosen by control signals. The circuit's performance can be modified by adjusting these control signals

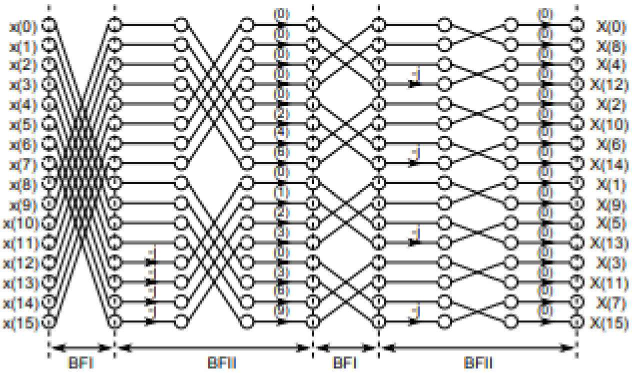


Figure 4.4 Flow graph of Radix -2^2 16 point DIF FFT

As described in [171], elaborate, pipelined FFT structures can be created using folding. For a N size FFT, it is observed that a two- parallel architecture is generated based on N/2 folding factor. Figure 4.4 depicts a 16-point, radix-2^2, decimation-in-frequency (DIF) complicated FFT circuit used to explain folding.

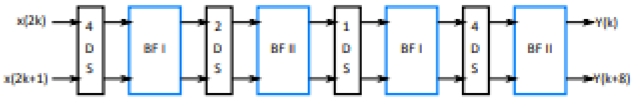


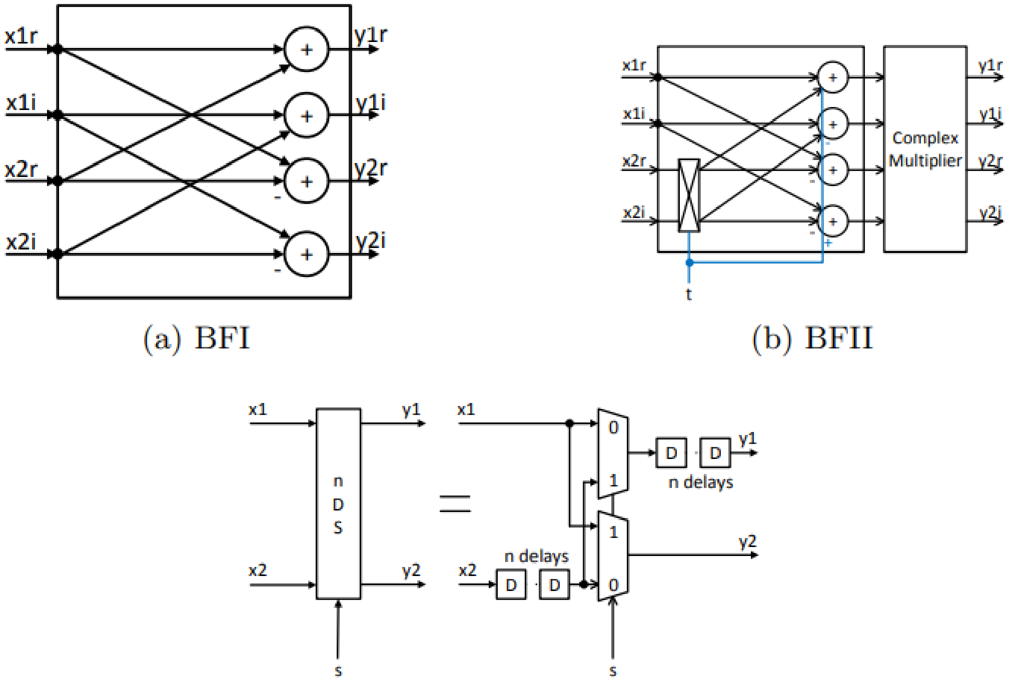
Figure 4.5 Folded Fast Fourier Transform (FFT) structure with 16 points

With the right choice of folding sets, this FFT may be folded into a 2-parallel structure, as shown in Figure 4.5. Figure 4.6 depicts the primary building components of the folded architecture, which include the two-point butterfly (BFI), the two-point butterfly that also performs trivial multiplication (by -j) and twiddle factor multiplication (BFII), and the delay-switch-delay structures of size n (nDS). Alternate folding sets and 2-parallel operation structures can be generated to make the design flexible. These sets of folds are called design modes, and they will be used in various combinations to create a fully hidden system.

4.3.2 Modifications to the control flow for obfuscation

The utilization of folded Fast Fourier Transform (FFT) architectures enables the manipulation of control flow parameters, hence facilitating the construction of functionally incorrect modes subsequent to the implementation of folding techniques and the attainment of satisfactory folded FFT designs. These modes are classified as devoid of significance, and we demonstrate how modifications to the signals generated by the control pathway of the design may be employed to generate them.

The management of the folded Fast Fourier Transform (FFT) components in Figure 4.6 is achieved by the manipulation of BFII's t parameter, which corresponds to multiplication by -j, as well as the switches' s. The expected operation of the delay-switch-delay structures may be achieved by utilising the proper control signals, which can be obtained from a log2N-bit counter. When these delay-switch-delay structures are driven by erroneous patterns, the ensuing modes emit noise. An increase in the number of delay-switch-delay structures accompanied by inadequate control signals leads to heightened levels of unpredictability in the outcomes. The modifications and correct functioning of these structures are depicted in Figure 4.7.



(a) BFI (b) BFII

(c) Delay-switch-delay

Figure 4.6 Blocks of Folded Architecture

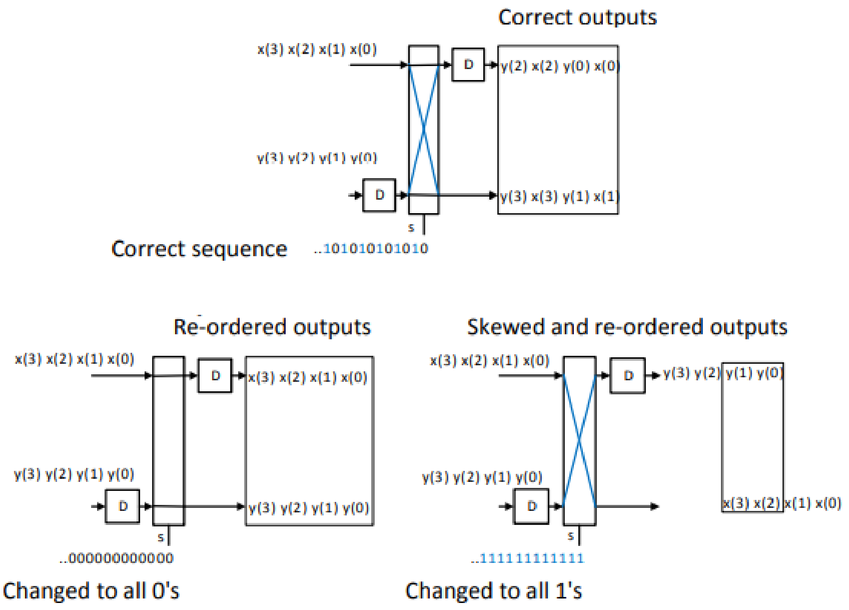


Figure 4.7 Modified Control Logic of Delay-Switch Structure

The process flow for a radix-22 FFT is shown in Figure 4.4, where it can be seen to consist of columns of twiddle factors and multiplication by -j terms. These adjustments are compatible with the BF II structure shown in Figure 4.6 when it is folded. The scheduling order for each step may be derived from the folded set selections. We may use this to learn that changes to twiddle factors or -j multiplication will have a limited impact on the outcomes. This is a critical insight because it explains why nonsensical modes with half-right results are generated. There is not much you can learn about the system's operation from these modes. The ensuing chapters will explain why these modalities are necessary.

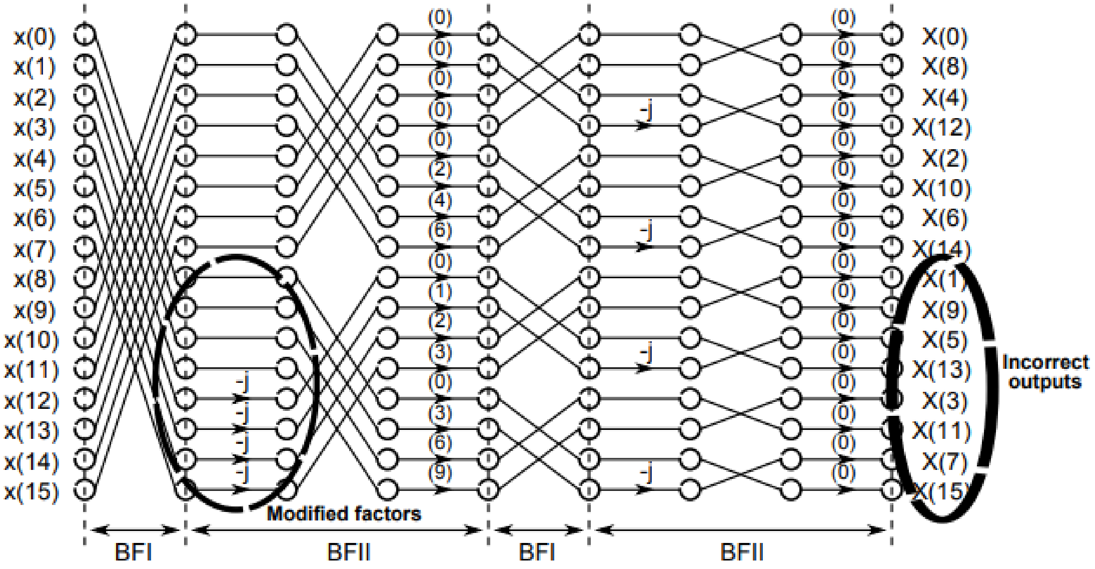


Figure 4.8 Effect of modification of –j factor

As an example, look at the FFT flowchart in Figure 4.8. The outputs are X(0)/X(8), X(2)/X(10), X(1)/X(9), X(3)/X(11), X(4)/X(12), X(6)/X(14), X(5)/X(13), and X(7)/X(15) (15), assuming the system is working in 2-parallel mode. Only the first four outputs (X/X = 0), the second and fourth (X/X = 2 and 4), and the sixth (X/X = 14) are correct; the rest are wrong. This modification is analogous to altering the control signal t of the initial BFII block in the masked FFT design. Figure 4.9 depicts the alterations to the block's activities over time. As a result, we have implemented a mode that is only 50% reliable. Meaningless modes that accurately calculate between 25% and 50% of the outputs may be constructed using many such combinations.

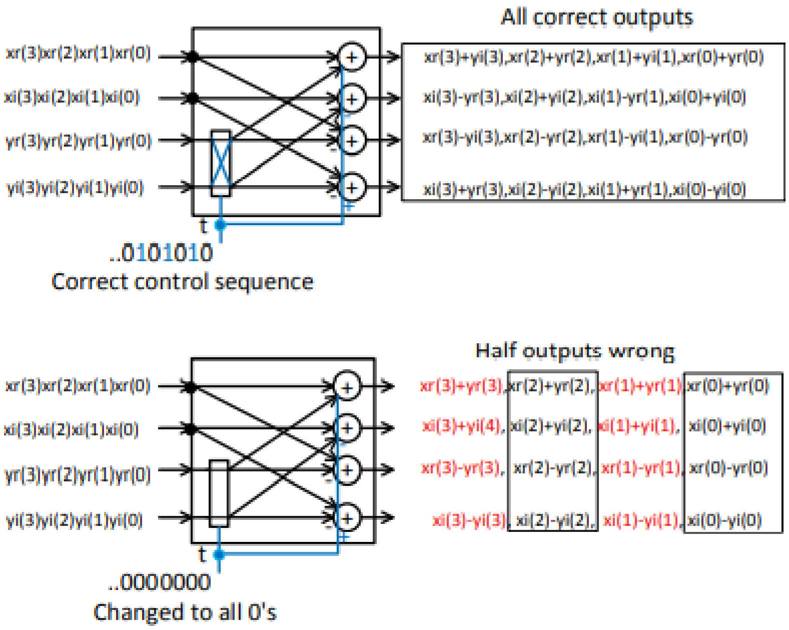

All correct outputs
xr(3)xr(2)xr(1)xr(0)
xi(3)xi(2)xi(1)xi(0)
yr(3)yr(2)yr(1)yr(0)
yi(3)yi(2)yi(1)yi(0)
+
xr(3)+yi(3),xr(2)+yr(2),xr(1)+yi(1),xr(0)+yr(0)
xi(3)-yr(3),xi(2)+yi(2),xi(1)-yr(1),xi(0)+yi(0)
xr(3)-yi(3),xr(2)-yr(2),xr(1)-yi(1),xr(0)-yr(0)
xi(3)+yr(3),xi(2)-yi(2),xi(1)+yr(1),xi(0)-yi(0)
t
..0101010
Correct control sequence
Half outputs wrong
xr(3)xr(2)xr(1)xr(0)
xi(3)xi(2)xi(1)xi(0)
yr(3)yr(2)yr(1)yr(0)
yi(3)yi(2)yi(1)yi(0)
xr(3)+yr(3), xr(2)+yr(2), xr(1)+yr(1), xr(0)+yr(0)
xi(3)+yi(4), xi(2)+yi(2), xi(1)+yi(1), xi(0)+yi(0)
xr(3)-yr(3), xr(2)-yr(2), xr(1)-yr(1), xr(0)-yr(0)
xi(3)-yi(3), xi(2)-yi(2), xi(1)-yi(1), xi(0)-yi(0)
t
..0000000
Changed to all 0's

Figure 4.9 Modification of Control Logic

4.3.3 Time varying and Dynamic sequentially triggered Modes

Incorporating trigger circuits into the design allows the fixed mode obfuscation to be converted into a dynamic one. It's important to remember that the trigger circuits generate signals that activate seldom and at regular intervals. The combination of the mode-based design with the trigger circuits is shown in Figure 4.10. To mask the original control signals, a trigger combination circuit is employed instead of the previously obtained modifications. Multiplexers receive both the disguised and genuine signals. A multiplexer connected to one of the control signals is used as an example in the diagram.

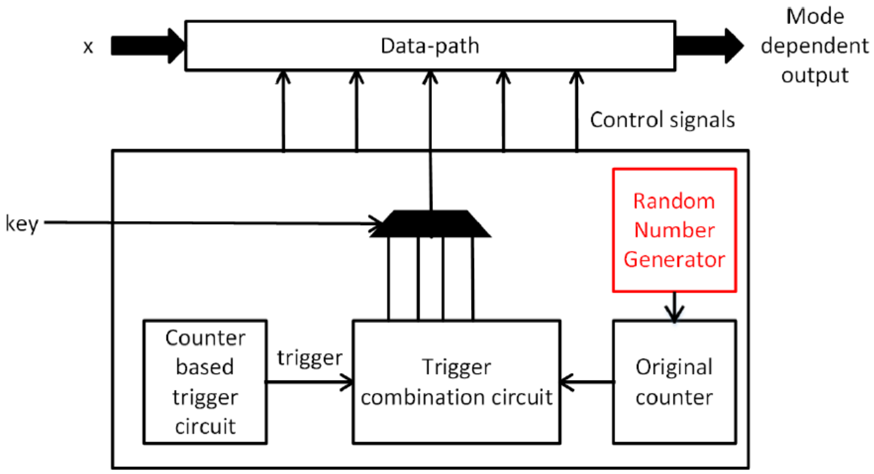


Figure 4.10 Mode Based Obfuscation

The implementation of trigger circuits has altered the behaviour of the system's meaningless states. When the switch is in the on position, the results of calculations are inaccurate. When not in a meaningful mode, the system computes results that are computationally valid. Time-varying modes are the consequence because the system behaves differently over time each time a new key value is introduced. The system's unpredictable behaviour makes any attack that requires traversing key space to determine correct key values more challenging. The robustness of the system is improved by introducing randomness into the modes. In order to break the signal's periodicity, a random number generator's output is fed into the trigger producing circuit. The trigger circuit is not programmed to fire at regular intervals for each incorrect key value but rather to fire intermittently. Thus, the modes become dynamic, resulting in a variety of system behaviours in response to the same key value.

The trigger must occur seldom for each of the undesirable dynamic behaviours. As a result, the time required for the trigger signal must be substantial. In addition, when the trigger is pulled, catastrophic failure of the circuit is expected to occur. The duty cycle of the trigger signal must be high enough for this to occur. It is also important that there aren't too many false positives, as it would make it harder to identify the inappropriate mode. There are criteria that must be met while designing trigger circuits. Another crucial aspect to think about is how well the trigger circuits were included into the initial design. These days, structural inspection of circuits is assumed to be possible in assaults based on reverse engineering. So, the trigger circuits have to be made so that they can't be tracked back to their original inputs or outputs. Combination trigger circuits are meticulously designed to make this a reality. Next, we'll take a closer look at these two crucial ideas.

4.4 Complete Design

Next, we will look at how folding and control-flow tweaks may be used to make a completely veiled design. We can build a reconfigurable and flexible architecture that can be tailored to the required level of security and the tolerable amount of overhead, since these techniques can be applied in a variety of ways.

4.4.1 Meaningful and Non-Meaningful Modes

To construct two parallel FFT structures of varying lengths, we employ the folding idea to radix-22 FFT flow graphs. A system might be built that can use 16-point, 64-point, 256-point, and 1024-point FFTs, for example. For this reason, you may choose from a wide variety of folding sets. One of the four resulting architectures is shown in Figure 4.11. We are now working to combine the four architectural styles into a single platform with four separate operational modes. We pick architectures where the shared building components make up the vast bulk of the total.

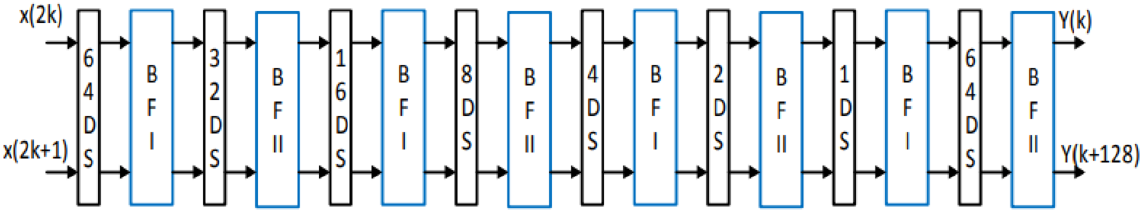


Figure 4.11 2 Parallel Folded FFT

Delay-switch-delay elements and muxes are the only adjustments needed. The resulting 16-bit/64-bit/256-bit/1024-bit obfuscated FFT is shown in Figure 2.9. The inputs c0 through c3 are used to pick one of the four modes by sending the matching choose signals from the key. As a result, we have a design that can be used in four different meaningful modes, each of which is functionally sound. Notably, the masked FFT may be made to function in one of four distinct modes, each with its own set of advantages and disadvantages (which will be discussed in more detail below).

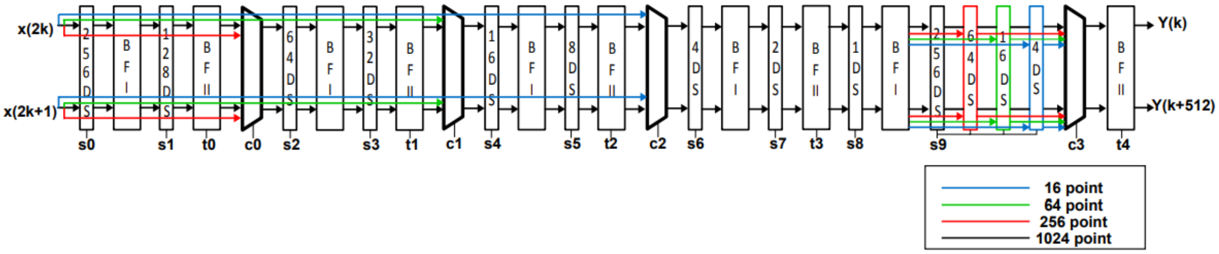


Figure 4.12 Obfuscated Architecture with Meaningful Modes

We use the previously mentioned ideas of switching control flows and selecting butterfly units via simple multiplication. Control signals s0s9 for switches and t0t4 for butterfly units are shown in Figure 4.12. Each signal's proper sequence is calculated using a 10-bit counter and is dependent on the architecture's placement and associated delays, as described in [171]. Obfuscation, however, involves obtaining other sequences from the same counter. We have found that the order of these signals has varying effects on the final result, therefore using a wrong order can mask the design. Figure 4.13 illustrates this idea by showing a 10-bit register denoted by cntr and its i-th bit denoted by cntr[i]. It is important to keep in mind that there is only one correct sequence for s0 in 1024-point FFT mode, namely cntr[8]. The remaining permutations of sequences are produced from random sequences or bits 0 and 1, both of which have an effect on the outputs. There are four possible sequences that may be used to mask any incoming signals. However, it should be noted that the mux at the signal output may be made bigger by increasing the number of sequence derivations for each of these signals. The drawbacks that come with increasing the mux size from 4:1 to 32:1 will be discussed in more depth below.

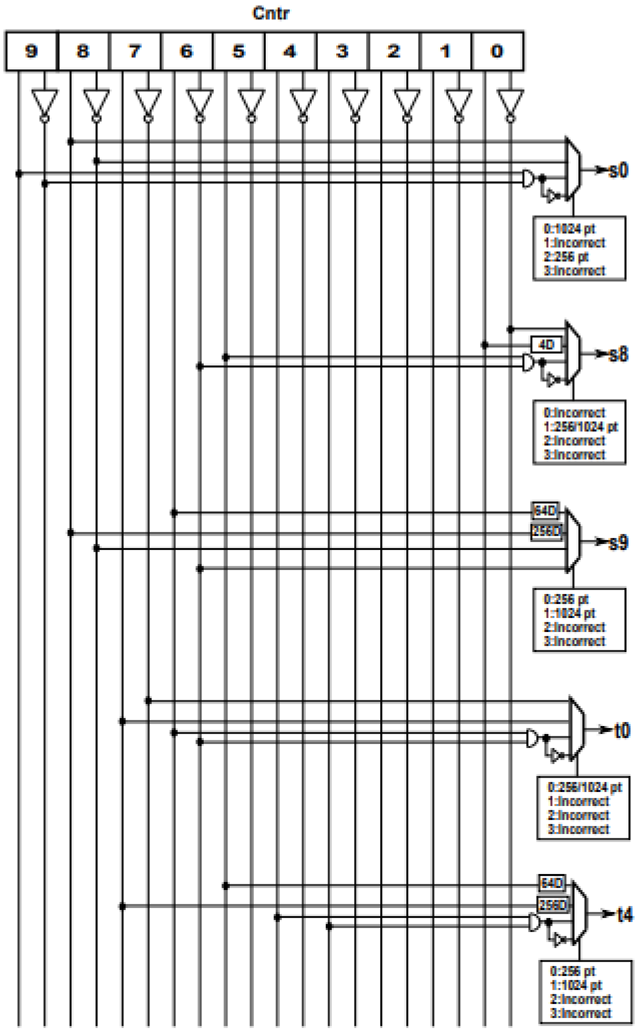


Figure 4.13 Obfuscated Architecture of Control Path

Once these obscured signals are configured, the design key may be used to determine the choose input for each of the muxes. A correct key enables the selection of all valid control signal combinations. Using a variety of additional control signal combinations, the remainder of the keys are mapped to meaningless or partially accurate states. Therefore, depending on the size of the key, the design incorporates meaningless modes. For each of these meaningless modes, it is recommended to deviate the control signal combination by at least 50 percent. For example, it is possible to construct a meaningless mode that uses the right control signals for certain parameters but the wrong ones for others, such as s[0], s[2], s[4], s[6], s[8], t[0], and t[2]. This leads to a state in which no correct outputs may be generated. However, in 256-point 2-parallel mode, if we select proper control for all signals except t[0] and t[4], we may get a half correct, nonsensical mode with 25% accurate outputs. Confidential designer-only control information is used in the creation of obfuscated modes.

4.4.2 Trigger Circuit

New hardware trojans are a threat to modern integrated circuits. Trojans are defined as "malicious modifications to hardware that alter the hardware's normal operational behaviour" [172]. Due to their small size, stealthy nature, and activation under settings that are hard to recreate during testing, these circuits are elusive. When activated, Trojans do harm to the circuit by altering outputs or destroying its infrastructure. Trojans introduce a great deal of variety into a system’s architecture and ways of functioning. Trojan circuits are also difficult to identify if their power and timing are not drastically different from the original design. The detection of hardware trojans and protective measures have been the subject of extensive study. There have been studies and attempts to classify the several trojan circuit varieties. In this paper, we take a look at how a sequential hardware trojan is built and then apply those same methods to our own creation.

.

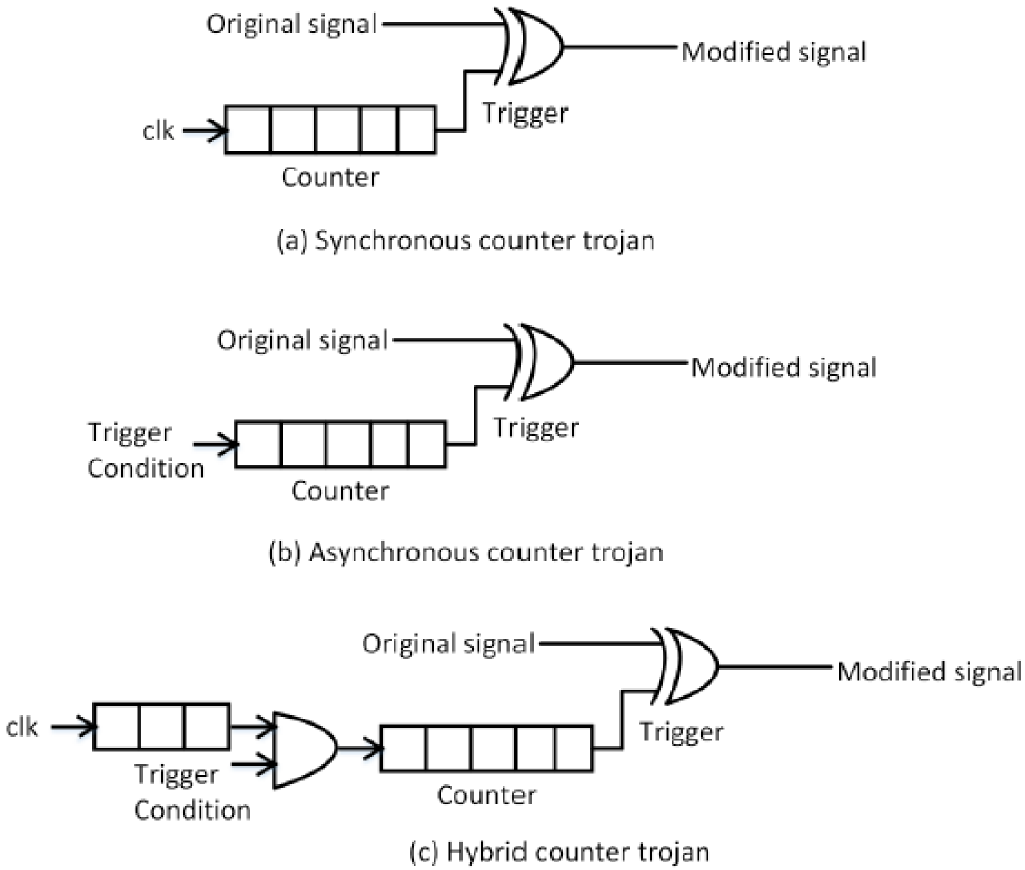


Figure 4.14 Hardware Trojan Circuits

Synchronous, asynchronous, or mixed forms of sequential trojan circuits are all possible, as explained in [173]. These sequential trojans' reduced circuit configurations are shown in Figure 4.14. In a synchronous counter trojan, the trigger occurs when the counter reaches a specific value. This trojan is time-dependent; therefore, the moniker "time bomb." Asynchronous counter malware doesn't add one to the counter with every tick of the clock, but rather the number is incremented in response to predetermined conditions. In the end, a hybrid circuit combining synchronous and asynchronous trojans is possible. A combination of clock-based counters and asynchronous trigger situations activates the trojan. Once activated, the trigger modifies the original signal to produce unintended results. While hardware trojans pose a significant danger to designs, a defence might use them to their advantage by hiding the code in question. The defending team has a leg up on the attacker in terms of understanding the circuit's architecture, making it easier to plant trojan horse-like circuits in otherwise obscure spots. Once the circuits are in place, the obfuscation of the design may be controlled using the appropriate keys. Literature-identified Here, we use Trojan circuits to our advantage. As part of the obfuscation, multiplexers are added to the system, and trigger combination circuits connect the original control signals to their inputs. The following are requirements for the layout of these circuits:

1. The trigger combination circuits must be indistinguishable from the original circuit, making them hard to find and delete even if key-gate muxes are found during netlist investigation.

2. The logic of the trigger combination circuit has to include enough randomization and variations to prevent tracing. As a result, it is unlikely that an adversary could successfully trace from the key input to the counter output.

3. Masking both good and bad signals is necessary to prevent structural distinction.

We build combination circuits out of simple parts while keeping these design goals in mind. The use of several different gates to fuse control and trigger signals is illustrated in Figure 4.15. In this picture, C1 represents the first control signal while C10 represents the second.

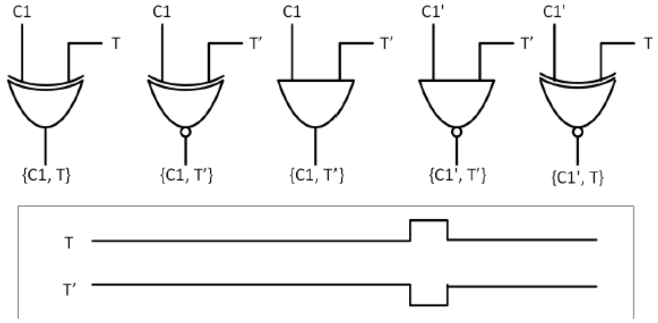


Figure 4.15 Logic gates of the combination trigger circuit

T and T’ variants of the trigger signal may be employed. All of the displayed sample gate combinations can successfully integrate the design's trigger and original signals.

Figure 4.16 shows how the control path of the mode-based design incorporates the logic gate-based trigger combination circuit. The key size in this case is 8 bits, thus we need four 4-input muxes. One valid signal and three null signals are required for each mux.

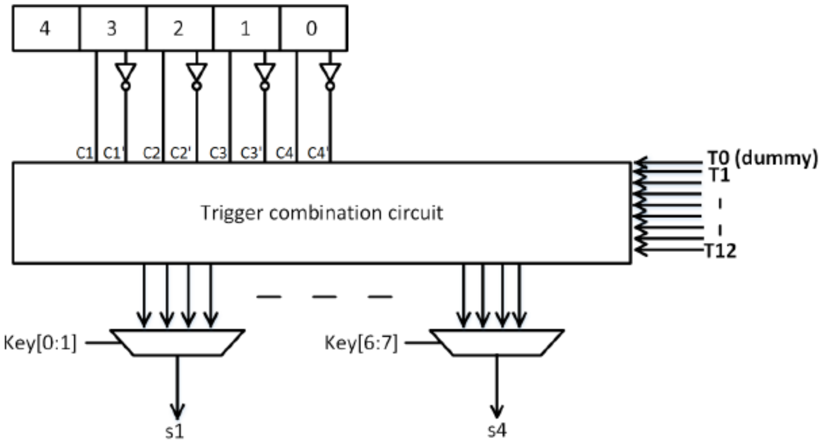


Figure 4.16 Circuit for triggering the multiplexer's key bits using the generated control signals

There are 12 true trigger signals and 1 false trigger signal sent into the trigger combination circuit. The phoney signal mimics the interaction between the correct control signal and the trigger signal. All 1s or all 0s might be part of the bogus signal. As a result, there exist structural similarities between the correct control signal and all other control signals.

4.5 Analysis of Obfuscated Modes

To evaluate the effectiveness of this obfuscation technique, we must first create an attack model and then investigate the function of obfuscation modes. Keep in mind that the attacker might be making use of more sophisticated methods and equipment, and that several types of attack models exist. However, we make use of a typical attack model used to develop obfuscation techniques, one that is tailored to the circumstances of an assault based on IP piracy or reverse engineering [174].

4.5.1 Attack Model

Using the aforementioned design process, an obfuscated FFT with several meaningful and non-meaningful modes is generated. Given the widespread use of the FFT as a submodule, we may deduce that the FFT is fundamental to the design of each unique application. After a netlist is created, the design is sent to unreliable foundries to be manufactured. The key is encoded and stored in the chip's secure memory once the design has been produced and returned to the design firm. By pressing this button, the user can select between many useful circuit modes.

We hypothesise that there are several ways in which an attacker may get the obfuscated netlist and use it to do structural observations and functional simulations of the circuit. The objective of such an attack would be to crack the right code and read the system's authentic master key. The second goal is to make an educated approximation as to the FFT length actually used in the IC layout. An attacker can expect to receive legitimate I/O pairs from a working IC. However, this data is already known to the programme using the FFT, therefore it adds nothing new to the analysis.

4.5.2 Obfuscation of Control Flow

Assume that the design calls for C separate control signals. Each signal is attenuated by a factor of L using the aforementioned techniques. This is achieved by connecting the control signal's output to a L:1 multiplexer. For example, the values of s = 10 and t = 5 in Figure 4.13 and Figure 4.12, respectively, correspond to L = 4 and C = 15, respectively. The power of this strategy lies in the fact that accurate generation of the select signal for this L: 1 bit mux is impossible without first doing a thorough simulation of the system. That's why there are LC possible signal combinations, but only one of them is correct. It is feasible to achieve the same degree of obfuscation with a smaller value of L for a circuit with a high number of control signals.

Next, we'll talk about how to pick the right signals to generate modes. There are M possible incorrect signal selections per mode. If M is equal to zero, then means the mode is applicable. The outputs of both the switches and the butterfly units vary by a factor of 100 for every change in the control signal. We use a value of M C/2 to produce meaningless modes, even when small adjustments to the control signals lead to a respectable Hamming distance between the right and erroneous outputs. This guarantees that the system is at least 50% different from the correct control sequence combination, providing security against assaults. It follows that the mux has the potential to introduce L errors into the signal for any value of M. This results in a sum total of modes for each given value of M. The length of the key may be adjusted by changing the value of M, which in turn affects the number of modes. Therefore, using L and M successfully conceals control information.

4.6 Results and Discussion

4.6.1 Area and Power Overhead

By conducting simulations and synthesising systems with different modes and key sizes, followed by assessing the associated area and power overheads, an estimation of the obfuscation strategy's cost may be derived. Verilog Hardware Description Language (HDL) is employed for the composition of architectural requirements, whilst Design Compiler serves the purpose of synthesis. The circuits are operating at a frequency of 100MHz, and all synthesis processes using a 65nM technology library. The evaluation of costs is conducted with respect to a 1024-point folded Fast Fourier Transform (FFT) architecture that is unobscured. By manipulating several design characteristics, we are able to generate three distinct categories of outcomes. In the initial round of experimentation, the data path is modified to encompass a diverse range of advantageous modes, spanning from 1 to 4. Meanwhile, the switch size in the control path remains constant at 4, and the key size remains fixed at 16. It is important to acknowledge that modifications to the design's data flow are necessary for the appropriate implementation of the relevant modes. However, it is worth mentioning that the inclusion of more modes leads to an increase in associated overheads. The data is shown in Table 4.2. In the majority of applications, it is advisable to maintain the number of modes in this Fast Fourier Transform (FFT) architecture at its default value of 2. In the context of security, critical applications may choose to employ more robust modes.

. Table 4.2 Overhead due to Meaningful Modes

No of meaningful modes	Total area overhead	Total power overhead
1 (1024 points)	0.2 %	0.5%
2 (256/1024 point)	8.5%	10.6%
3 (64/256/1024 point)	38%	15.5%
4 (16/24/256/1024 point)	41.5%	17.3%

We then proceed to flip the switch at the obfuscation-added control path outputs. It was found that these switches are mostly responsible for the security-enhancing control-flow obscurity of the architecture. Overhead increases by 2% for mux sizes between 2 and 16 as seen in Table 2.2. As a result, it may be used as a parameter to adjust obfuscation settings as needed. The mode value was set to 2, and the key size was set to 16.

Table 4.3 Overhead due to MUX in Control path

Mux size of Control path (L)	Control path overhead		Total overhead	
	Area	Power	Area	Power
2	2%	0.7%	8.2%	10.1%
4	5.5%	1.7%	8.35%	10.5%
8	22%	4.2%	8.5%	11.4%
16	41%	6.7%	9.1%	12.1%

Table 4.4 Overhead due to Key Size

Key Size (Mode)	Total Overhead	
	Area	Power
4(16)	8.29%	10.53%
8(256)	8.33%	10.55%
16(65536)	8.35%	10.59%
20(1048576)	8.42%	10.63%
28(268435456)	8.47%	10.66%

For the final set of findings, we maintain two relevant modes, four switch positions, and variable key length. Using signal combinations from the obfuscated design, modes are generated for each key. The number of erroneous signals is nominally maintained at 50%. Observing that the area and power overheads for key sizes up to 28 do not vary significantly, it is straightforward to design an architecture that can tolerate big key sizes.

Since general encryption and obfuscation algorithms are not designed with DSP circuit obfuscation in mind, comparing their overheads would be misleading. This includes preserving critical properties like FFT length. However, a nominal meaningful mode value of 2 yields an 8% space overhead and a 10% power overhead [18-20,32,33,36], which is comparable to conventional methods of obfuscation and encryption. To minimise both space and power consumption, one approach may be to provide inputs for key size, switch size, and the number of relevant modes.

Chapter 5

Power Attack

5.1 Introduction

Determining the private key and finding security holes in a cryptographic method are both possible through a process called cryptanalysis. In this case, an attacker doesn't need to know anything about algorithms in order to try to acquire access over devices by exploiting flaws in the hardware itself. The EDA tool's default standard cell library is configured for CMOS-based low-power circuit design [175, 176]. Figure 5.1 depicts the secondary information, or side-channel information, that devices release in addition to their principal outputs in the forms of power, time, electromagnetic radiation, fault, etc. The side-channel attack examines this data in an effort to discover the secret. Since the cryptographic algorithm, is public knowledge, the input, plain text message and the cypher text message, may or may not be known, but the secret key, must be protected. The objective of the attack is to deduce the secret key using only the input message and any contextual information available.

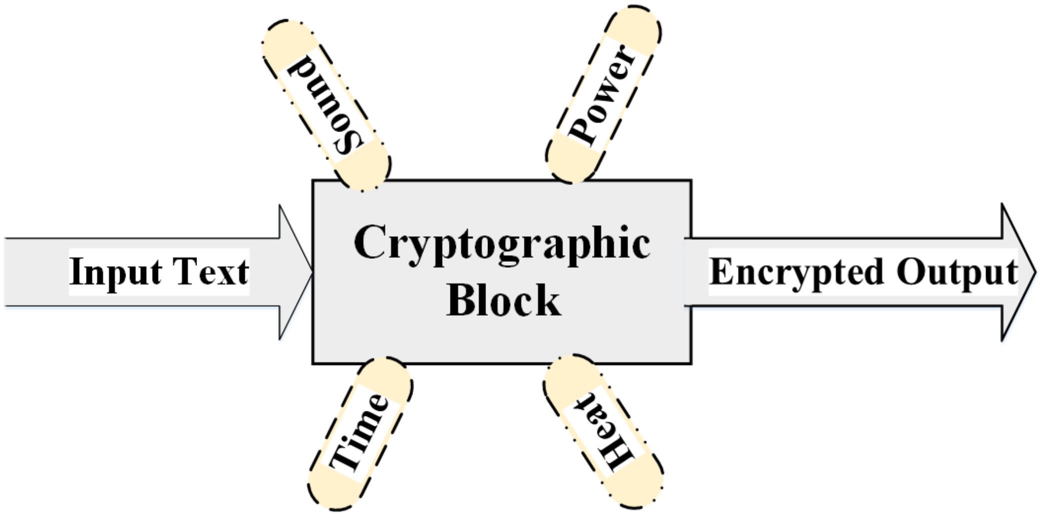


Figure 5.1 Side Channel information source

The most studied form of side-channel assault is power attacker analysis. Power attack analysis, a subset of side-channel analysis, examines the hardware's external power measurements during a cryptographic operation. The objective of the attack is to acquire information that will compromise security. Leakage power rises when MOS transistor sizes decrease towards the nanoscale range. Depending on the information being processed, the CMOS transistor's switching/dynamic power will vary. As it processes data, a device uses energy. Each input has its own power profile, which is related to the message it received. The [177] input and output power consumptions are recorded in an SBOX, making it a desirable object to acquire. Adversaries attempt to guess the input of SBOX based on the SBOX output and their power value, and the secret key can be predicted when XORed with the input message. Keeping track of the power trace for each input option is a difficult process. One 50-ohm resistor wired in series with the power source. A waveform analyzer or oscilloscope is used to measure and evaluate the current flowing through the resistor. The first part of a power assault analysis is data collection, during which a database of input and output power consumption is built, and the second is data analysis, during which information about SBOX's input and output is gleaned from traces of the latter's power use.

5.2 Consumption of Power in CMOS Circuits

CMOS implementations of digital circuits are highly preferred due to their minimal power consumption. When performing a computation, the circuit consumes a certain amount of electricity. The author of [178] describes the power consumption profile presented as data-dependent. Three kinds of CMOS power consumption (P_{sc}) include static power (P_s), dynamic power (P_{dyn}), and short circuit power. It is disclosed that CMOS, not gate is composed of PMOS and NMOS connected in series [72, 93]. For minimal input PMOS on and NMOS cutoff, the load capacitor is charged to its utmost value, allowing it to store energy. When the gate input is high, the NMOS conducts and the PMOS stops conducting, releasing the capacitor's stored energy.

5.2.1 Static Power

A circuit's "static power" is the amount of energy it takes to maintain a constant voltage or current under a certain set of operating circumstances. When a circuit is not doing any switching activity, it draws static power from the power source and ground rail. Power loss is another name for this. Perfectly, P_s=0.

$$P_s = I_s \times V_{DD} \qquad (5.1)$$

Where I_s is caused by a drain-to-source bias current in the substrate.

Static current is represented as

$$I_s = K \frac{W}{L} e^{(\frac{V_{GS} - V_{TH}}{\eta V_T})} (1 - e^{\frac{V_{DS}}{V_T}})(e^{\frac{V_{SB}}{V_T}} - 1) \qquad (5.2)$$

The operational voltage at a junction is given by V_{GS}, V_{DS}, and V_{SB}, where V_{TH} is the threshold voltage and V_T is the thermal voltage. The terminal static voltage and circuit architecture determine the device's static current. [179].

5.2.2 Short Circuit Power

The input level causes a transition from the saturation region of PMOS to the cutoff region of NMOS. Since a MOS transistor's ability to switch regions quickly offers non-zero rise and fall times, they are not ideal. PMOS and NMOS both conduct for a little time when changeover. A direct current (DC) channel between the power and ground rail was discovered, with a brief period of high current flow. This current's power consumption is a result of a short circuit.

$$\textit{Energy consumed per switching activity period } E_{sc} = V_{DD} I_{peak} t_{sc} \qquad (5.3)$$

$$\textit{Power consumption per cycle } P_{sc} = V_{DD} I_{peak} t_{sc} f_{clk} \qquad (5.4)$$

5.2.3 Dynamic Power

Information exposures caused by dynamic power are the core of the attack vector for power analysis. Peak power and the number of data transmitted have a direct relationship with dynamic power. The transitional power requirements of a circuit are described in [100]; when the output node transitions, the high node capacitor charges to VDD, requiring power from the circuit. The PMOS loses half of its energy as heat, while the capacitor retains the remaining half.

The CL discharges and the stored energy is dissipated through the NMOS and into ground when the output level is switched from high to low.

$$\text{Energy is drawn from the supply } E_{V_{DD}} = \int_0^\infty i_{V_{DD}}(t)V_{DD}(t)dt$$

$$= V_{DD}\int_0^\infty C_L \frac{dvout}{dt} dt$$

$$= C_L V_{DD}\int_0^{V_{DD}} dvout$$

$$= C_L V_{DD}^{\ 2} \qquad (5.5)$$

$$\text{Energystoredincapacitor} E_c = \int_0^\infty i_{v_{DD}}(t)V_{out}(t)dt$$

$$= \int_0^\infty C_L \frac{d_{vout}}{dt} v_{out} dt$$

$$= C_L \int_0^{V_{DD}} v_{out} dv_{out}$$

$$= \frac{C_L V_{DD}^{\ 2}}{2} \qquad (5.6)$$

It consumes $C_L V_{DD}^2$ watts of power for one switching cycle. The dynamic power is denoted by the equation $P_{dyn}= C_L V_{DD}^2$ where is the switching activity at the output terminal. This is because the probability of multiple transitions results in energy dissipation proportionate to the number of transitions. Its dynamic power changes as seen in equation at its highest input frequency of f_{clk} (5.7)

$$P_{dyn} = \alpha f_{clk} C_L V_{DD}^{\ 2} \qquad (5.7)$$

The number of times an input is switched each second is represented by the switching activity factor. The activity parameter necessitates the provision of input data. During the brief period that an input is changing states, dynamic power is pulled from the power supply, but for the other input combinations, only static power or leakage power is drawn. There are 2n possible input combinations for an n-input circuit. Power dissipation for each of a CMOS inverter's four transitions is reported in table 5.1. The change from 0 to 1 involves a negligible amount of static power loss [80]. It is PMOS that charges the load capacitor during the 0-1 transition, and NMOS that discharges it during the 1-0 transition. Capacitor charging and discharging events use up a lot of kinetic energy. The dynamic power ranges from P_{0-1} to P_{1-0}, with P_{0-0} and P_{1-1} being lower.

$P_{0\text{-}1} > P_{1\text{-}0}$ denotes that the energy needed to go from 0 to 1 is greater than that needed to go in the other direction.

Table 5.1 CMOS Inverter's Power consumption pattern with respective bit

Input	Output	Power
0	0	P_s
0	1	P_s+P_d
1	0	P_s+P_d
1	1	P_s

5.2.4 Power Delay Product

With PDP, we can quantify the joules of energy expended in each switching occurrence. Assuming a maximum switching frequency of $f_{max}=1/2t_p$,

$$PDP = C_L V_{DD}{}^2 f_{\max} t_p = \frac{C_L V_{DD}{}^2}{2} \tag{5.8}$$

5.2.5 Energy Delay Product

By lowering the supply voltage, the switching energy measured by PDP can be kept to a minimum. Since energy consumption and delay are inversely proportional to supply voltage, increasing the voltage will decrease delay while increasing energy consumption and vice versa.

$$EDP = PDP \times t_p = \frac{C_L V_{DD}{}^2}{2} \times t_p \tag{5.9}$$

5.3 Power Analysis Attack

Depending on the complexity of the attack, researchers divide up the data into one of three categories: Differential Power Analysis (DPA), or Correlation Power Analysis (CPA), Simple Power Analysis (SPA), differential power analysis (DPA) [180, 181].

5.3.1 Simple Power Analysis

An SPA actively takes part in deciphering data from power traces that have been gathered by a cryptography module. The power trace is a time-varying, two-dimensional waveform that measures power consumption. It is a compilation of power readings from various points in the cryptography process. Image 5.2 depicts an AES power trace. It is challenging to extract useful information from a visual evaluation of an AES curve since the core section of the curve is repeated for 10 cycles. A SPA is not the same as a full-on assault. A random technique is used to determine the key in cryptography.

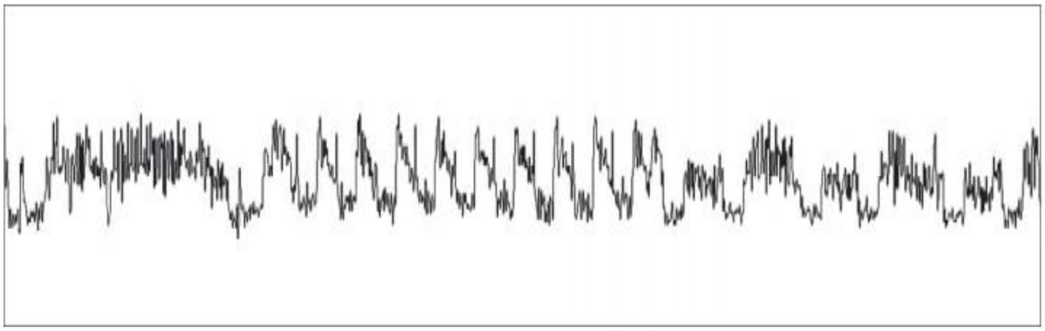


Figure 5.2 Power Trace in SPA

5.3.2 Differential Power Analysis

DPA has proposed a more robust attack similar to SPA, which would require a significant amount of power traces. Data-dependent power analysis (DPA) makes advantage of the power consumption pattern shown by the power trace. Even in a loud setting, DPA can disclose the hidden key. The circuit's power consumption is proportional to the voltage at the circuit's internal state. A power trace, or 2D time-related waveform, displays the power amplitude fluctuation for various inputs. DPA successfully reveals the connection between power lines and the voltage at an internal node as a result of the input. DPA assumes that the plain text input or SBOX output is already known, and proceeds to acquire and analyse the data in two stages. Information gathering and processing are depicted graphically in Figure 5.3. Several simulations must be run in order to collect enough data to record numerous power traces.

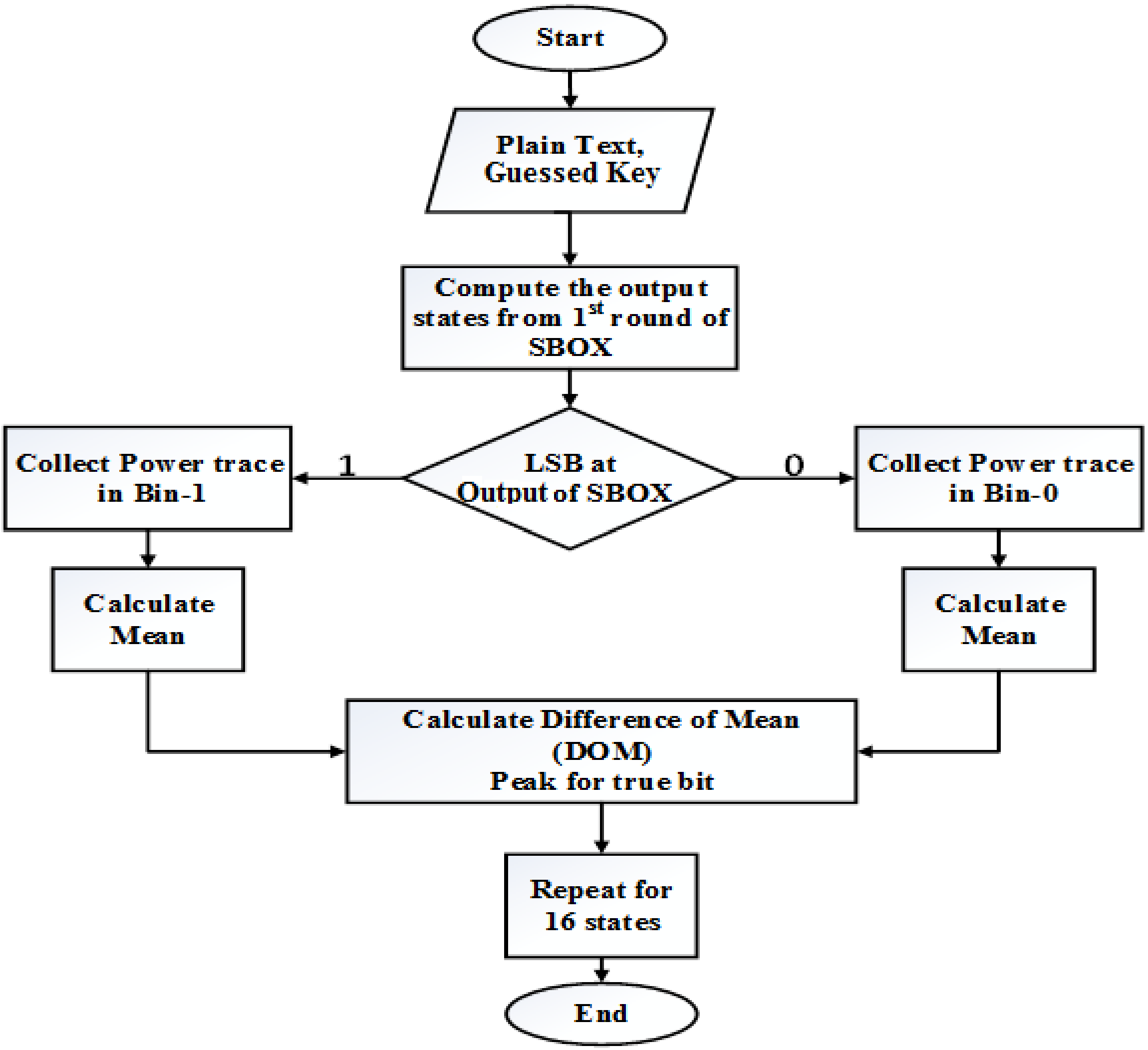

Start
Plain Text,
Guessed Key
Compute the output states from 1st round of SBOX
LSB at Output of SBOX
1
0
Collect Power trace in Bin-1
Collect Power trace in Bin-0
Calculate Mean
Calculate Mean
Calculate Difference of Mean (DOM) Peak for true bit
Repeat for 16 states
End

Figure 5.3 Flow chart of DPA[185]

During encryption [182, 183], a large quantity of plaintext data is fed into the SBOX circuit with a static secret key, and the power waveform is examined with an oscilloscope or waveform analyzer. An opponent can learn the voltage of the internal state by analysing the data and comparing it to a secret key in the range 00-FFH. Due to the impossibility of saving and evaluating power traces on each round if the attack is executed on the previous round, the DPA assaults described in [184] can only be applied to SBOX in either the first or last round. In the first phase of an SBOX, it is easy to implement a DPA assault. The collected trace was split in half, one half corresponding to the internal state voltage, during the analysis phase. Each group's average waveform is then compared to the other group's to determine the disparity. The information can be gleaned from the differential curve of the average power waveform if a spike is noticed; if the waveform remains constant, then the presumed key is erroneous [71]. This process, which here focuses on an 8-bit subset of a 128-bit key, is repeated 256 times to cover all potential keys.

5.3.3 Correlation power Analysis

The Pearson correlation coefficient is utilised in place of the mean difference in this variation of the attack first described in [71,83, 186]. Correlation power attack (CPA) is more accurate and exact than differential power analysis (DPA) because it employs a mathematical power model to estimate energy consumption. By comparing these estimates with measured power usage, we can uncover the power model's hidden secret weapon: Hamming weight and hamming distance. Using a postulated power model, such as hammering weight or hamming distance, CPA determines the relationship between real power consumption and the model. The hamming weight power model calculates a circuit's power consumption on the basis that 0 results in no power consumption and 1 results in a large amount of power. Here, the power consumption scales linearly with the precision of the supplied data. The HD value is the total number of ones that enter the SBOX, whereas the HW value is the total number of ones (both 0 and 1) that enter the SBOX. Dependence on HD is demonstrated by the classification of the waveform based on the HD obtained from the key. Higher HD values indicate greater power usage, whereas lower HD values indicate less energy is used. The correct key was identified by testing all possible combinations until the largest discrepancy between actual and expected power consumption was discovered. If the observed power trace closely matches the projected power value, then it is more likely that the secret key can be predicted. As seen in Figure 5.4, data was continuously flowing throughout the CPA trial.

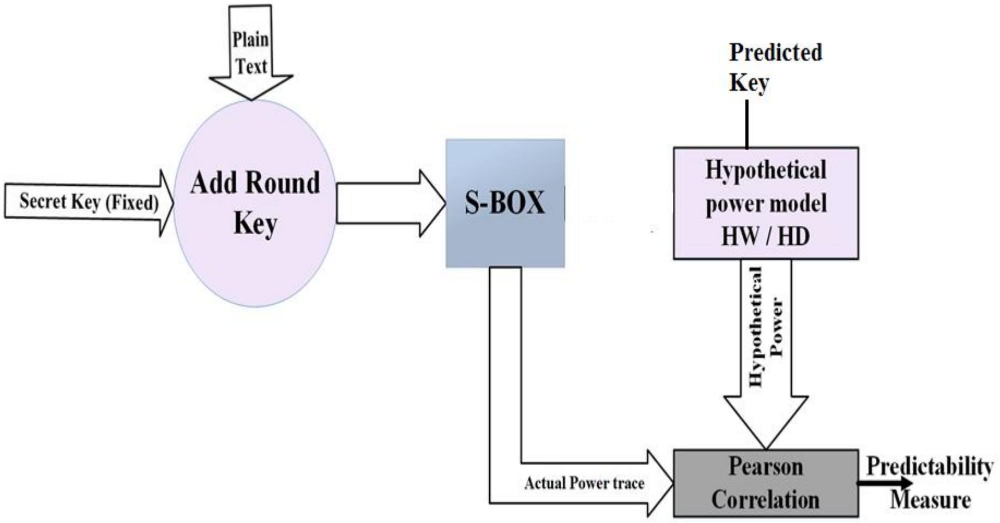


Figure 5.4 Flow Chart of CPA

The equation (5.10) is the result of calculating the correlation coefficient between actual and expected power consumption.

$$\rho(P_{actual}, P_{predicted}) = \frac{\mathrm{cov}(P_{actual}, \mathrm{P}_{predicted})}{\sqrt{\mathrm{var}}(P_{actual})\sqrt{\mathrm{var}}(P_{predicted})} \qquad (5.10)$$

The variation of the actual power value and the variance of the anticipated power value for all inputs are denoted by var(Pactual) and var(Ppredicted), respectively. The correlation coefficient, which evaluates how closely Pactual and anticipated agree with one another, might be anywhere from minus one to plus one. Values of = -1 indicate a negative link between Pactual and predicted, = 0 indicate no correlation between Pactual and Ppredicted, and = +1 indicate a positive association between Pactual and Ppredicted; a large value of is to be expected for a correct key guess. An 8-bit key's correlation is halved if just one bit is incorrect, therefore picking the erroneous key is much more unlikely with this method. Anticorrelation is seen whenever something goes wrong. A stronger relationship is seen when the value is positive.

5.4 Power Model

Differential power analysis attacks are briefly discussed in [84, 187], and they rely on the mathematical model's ability to roughly calculate the power requirement of the circuit. The effectiveness of a hypothesis power model is crucial to the outcome of a power attack. The power consumption of the simulated circuit may be accurately predicted because the cryptographic architecture is already known. The power consumption of each input is plotted in an ADEL browser window; a software environment must be created to display the power consumption of each input and output. The hammering weight and hammering distance power models are two popular options.

5.4.1 Power Model for Hamming Weights

From the information provided, HWPM is able to perform an initial calculation of the amount of power that is required by the circuit. This approach helps to approximate the power value of a circuit when the sequential input data is unknown. This is because power is proportional to the number of high bis in the data presented in figure 5.5. As a result, this method is helpful. In this particular scenario, low bits in the input data do not consume any power, but high bits contribute to the overall consumption of power.

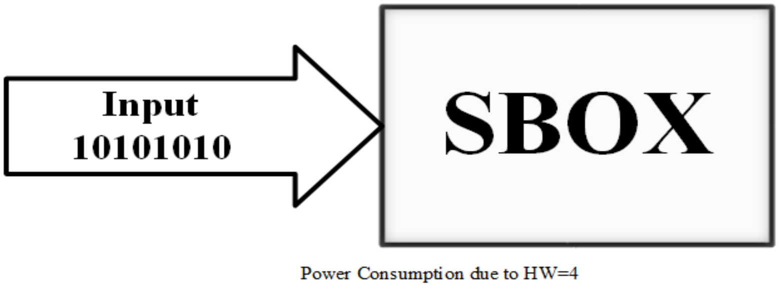


Figure 5.5 Estimating Power Using the Hamming Weight

Power consumption is lowest when all bits in the input data are zero, and highest when all bits are high. The HW model just pulls the most essential information about the circuit, and it only does it once per week. Let R be the information coming in through the input terminal if the power consumption of the circuit can be calculated with an equation. (5.11)

$$W = aHW(R) + b \tag{5.11}$$

5.4.2 Modelling Power at a Hamming Distance

The hamming distance power model is a useful approximation for estimating power consumption. The hamming distance is a more advanced form of the hamming weight power model that predicts power consumption based on the bit-switching rate at the input terminal.

According to the HD model of [82, 97], the circuit's power consumption is generally proportional to the logic transitions 0-1 and 1-0. The XOR of two values' hammering weight, used to measure the rate of change. According to the hamming distance power model, bit transitions from 0 to 1 and 1 to 0 are considered to use the same amount of energy, whereas bit transitions from 0 to 1 and 1 to 1 are assumed to consume no energy at all. For an n-bit circuit, the hamming weight is the number of bits that are pulled to 1 and the average is n2 with a variance of m4.

Register R0's current state is assumed to be constant (and not zero) and other input is used to XOR the bits in register R1, as shown in Figure 5.6. Power usage is approximately proportional to the hammering weight of the XOR of the beginning and final values when the register value is updated.

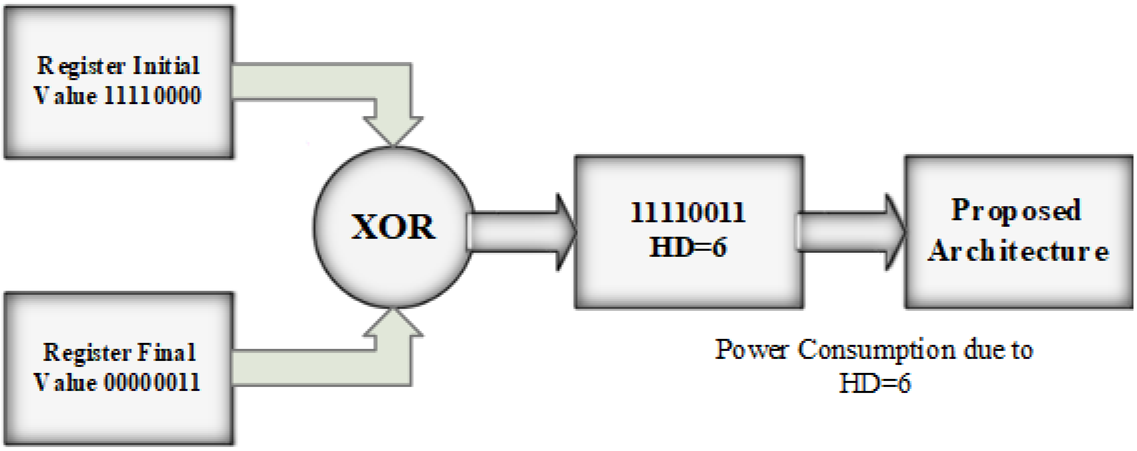

Register Initial Value 11110000
Register Final Value 00000011
XOR
11110011 HD=6
Proposed Architecture
Power Consumption due to HD=6

Figure 5.6 The Hamming-Distance Power Approximation

The input power is approximated by the following equation if R0 and R1 are two consecutive values at the circuit's terminals (5.12)

$$W = HD(R_0, R_1) = aHW(R_0 \oplus R_1) + b \qquad (5.12)$$

Input hammering weight is modelled as a function of power or current consumption in the hamming distance power model. The power consumption of the chip as a whole is not represented by this model, only the power consumed by the data-dependent subsystem. Term b represents static power, often known as power due to noise, crosstalk, or wire isolation; this constant value generally balances out the other components of power in CMOS circuits when the input is varied. If all bits in R0 or R1 are 0, then the noise in the circuit determines the constant bis, and the gain of the scaler between HD and power consumption is an. The hamming weight power model is comparable to the hamming distance model if and only if a=1 and b=0. This simplifies to the following equation: (5.13)

$$W = HW(R_0 \oplus R_1) \qquad (5.13)$$

Power consumption by a CMOS device as a function of hammering weight and hamming distance is depicted in Figure 5.7 and Figure 5.8 [91]. When no switching is taking place, the CMOS cell consumes essentially no power. As the quantity of bits of input data (HW) and the rate at which bits are switched (HD) are both increased, the resulting current increases. Since HW=8 only happens once and 8-bit changes only happen once per combination, the curves for HW and HD values of eight are decreasing.

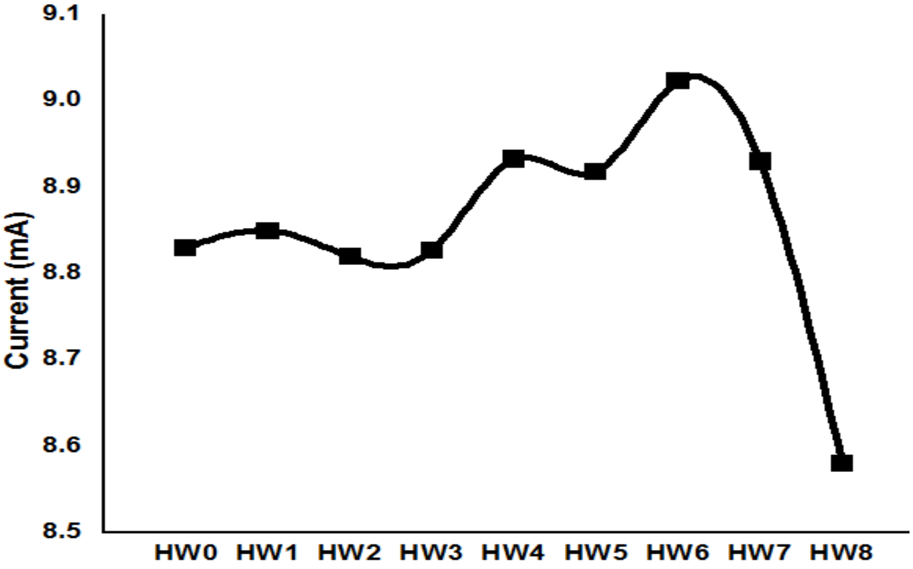


Figure 5.7 Current vs Hamming weight (HW)

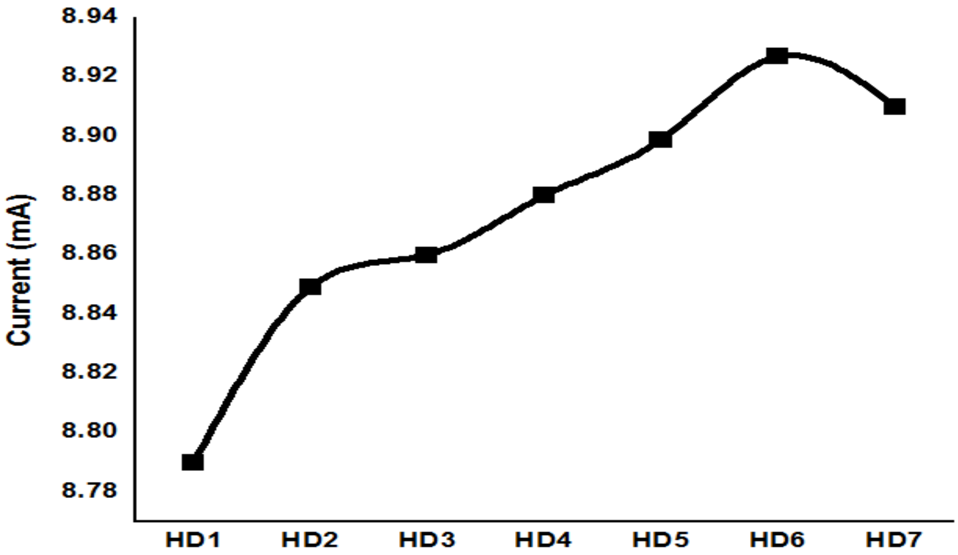


Figure 5.8 Current vs Hamming Diatance (HD)

5.5 Experimental Setup

5.5.1 Differential Power Analysis

The AES subsystem can be attacked either in the first or the last round. Round-key analysis and SBOX power-trace capture are performed on the plain text for the initial attack. On the other hand, the prior round attack necessitates more work than just decrypting the previous round using the key and the previous SBOX. The power trace from the previous round to the first round SBOX is computed using the decoded cipher text [75, 188, 189]. The initial round of SBOX is used to launch a DPA assault, as described in [190]. Time spent in simulations is another factor in favour of going in for the kill in the opening round. The entire encipherment procedure must be simulated in the first round. Add round key and SBOX schematic developed in Cadence virtuoso schematic composer using CMOS 90nm technology and a 1-volt supply voltage.

For the DPA attack, a Cadence spectre was used for simulation, and a waveform analyzer was used to examine the 2056 power traces that were necessary. The key and plain text are each 128 bits long and are separated into 16 SBOX. By using a single byte for each SBOX, we can cut the power consumption down from 2128 to 162 8=4096. This means that the security of an SBOX can be compromised with merely a trace with a power of 4096. The power trace is analysed using the statistical method of difference of mean in a DPA assault. CMOS circuits' power consumption grows in step with the number of high bits in the input. Assume the plain text was selected at random from 00-FFH, and run simulations of every potential key for SBOX. Save the power history of each input and its matching output. An adversary's attempt to guess the secret key by statistical means occurs during the data analysis phase.

The AddRound key stage is subjected to all 256 potential input text combinations (00H-FFH) under the assumption that the secret key is 25H. Each of the 16 bytes of AddRoundkey's output is then fed into SBOX in turn. SBOX modifies the power trace and byte at the end of the transmission chain in the SubByte step. Power trace of the first SBOX using secret key 25H and plaintext 00-FF is shown in Figure 5.9. For each plain text, the simulation takes 2.56 s, with a 10 ns delay between inputs. The Clip feature of the cadence virtuoso calculator allows you to choose a specific instant in the waveform. Each power trace was processed through the clip function 256 times from 0 to 10 nanoseconds in order to extract the matching plaintext. The voltage profile of the SBOX is shown in Figure 5.10 for the 0-FH plain text value, which is the first 16 bytes.

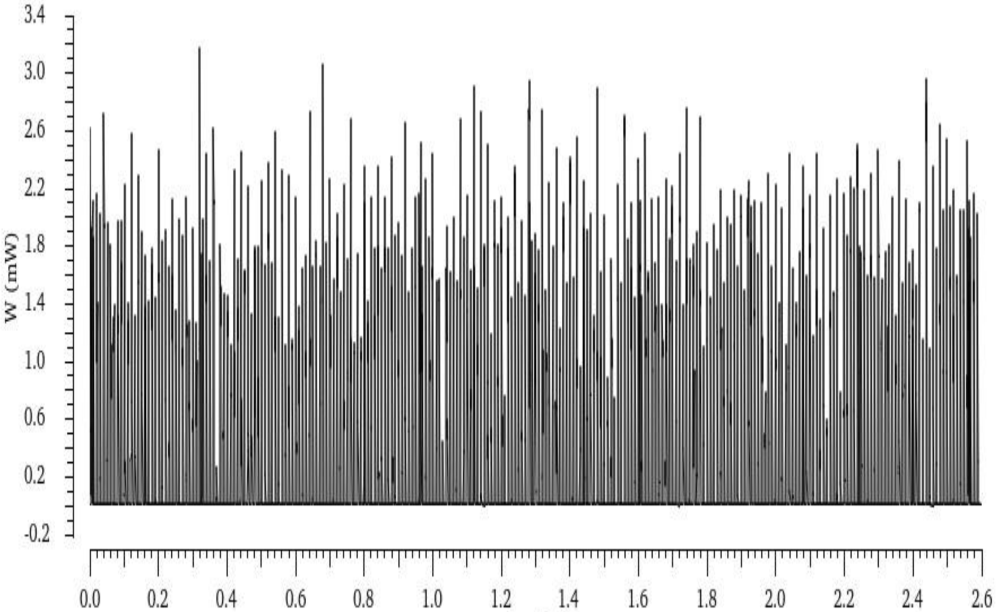


Figure 5.9 DPA Power Flow Diagram for Proposed Design

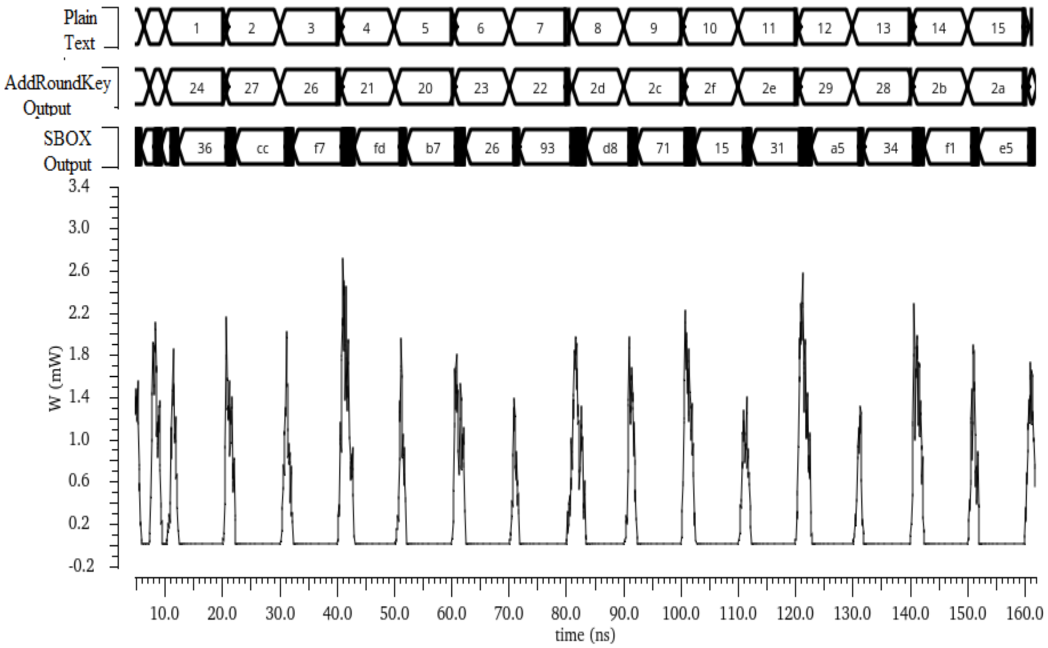


Figure 5.10 DPA power trace for the first 16 inputs of the proposed design

The DUT has a greater power need for LSB 1 output than for LSB 0. The LSB of power traces must be used to sort them into one of two categories. Simple text where the leading zero is located in bin-0 and the leading one in bin-1. Find the median power trace value for each bin. Figures 5.11 and 5.12 depict the average power trace for each bin. The average waveform was calculated, power traces were made uniform in length, and inputs were averaged across a 10 ns simulation time step. Both Bin 0 and Bin 1 are organised in a regular fashion. Unlike the original power traces, which basically followed a straight line, the DPA shows a spike in the waveform if the successful key was a guess (see Figure 5.13). The availability of critical evidence in support of a hypothesis may be inferred via differential power traces. A spike in the observed waveform indicates that the key is correct, whereas a straight line in the difference of the mean curve indicates that the key is incorrect.

.

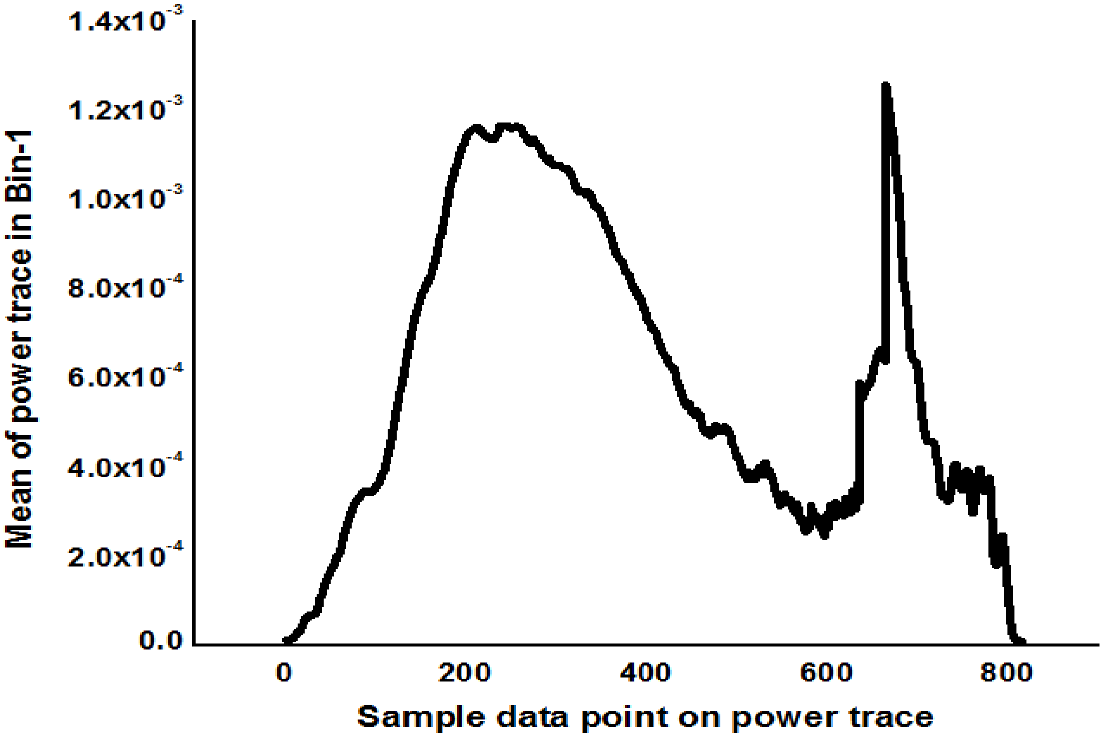


Figure 5.11 Sum of bin1's power traces

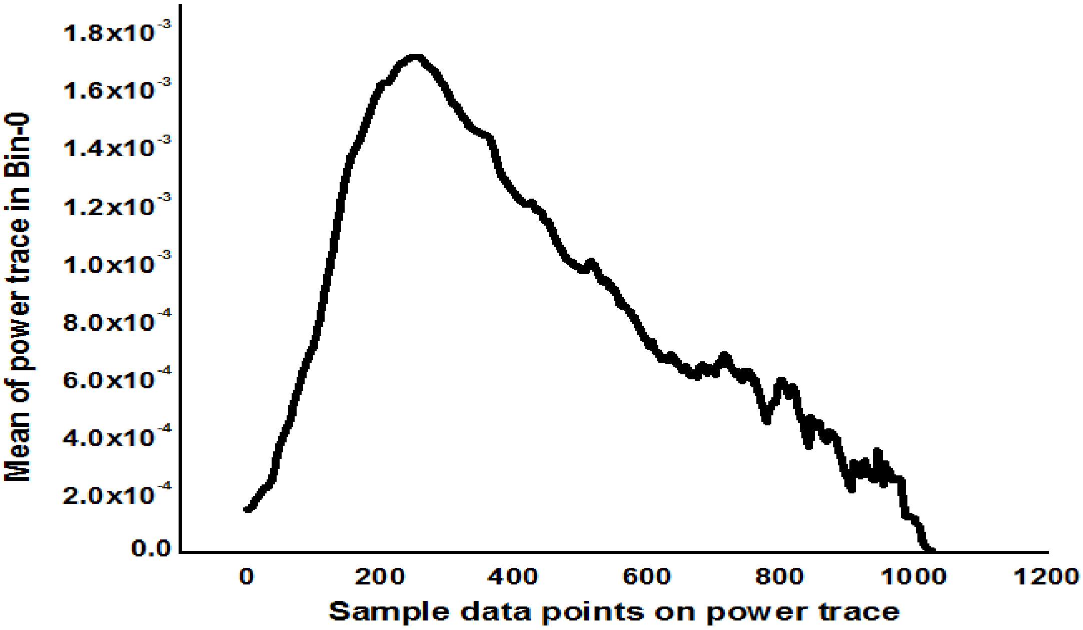


Figure 5.12 Power curve average in bin 0

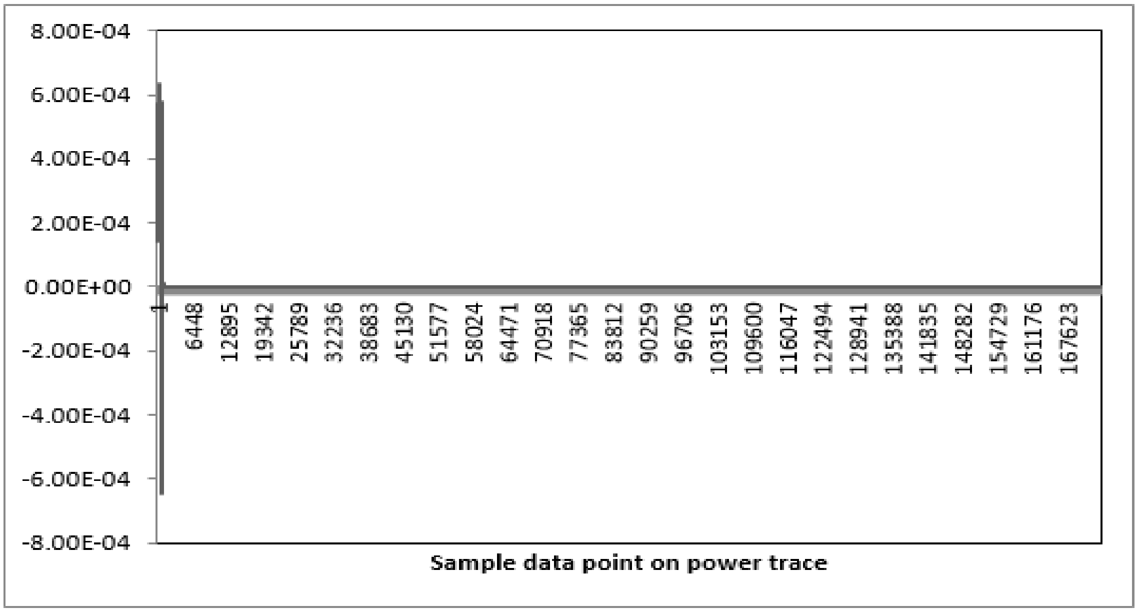


Figure 5.13 LSB1 and LSB0 Average Power Differential Curve and Trace

Mean power traces have similar shapes. In this case, 25H is the key, and the input range for plain text is 00 to FFH. We applied each input for 10 ns, for a total transient time of 2560 ns. For each input, we collected 1000 samples and trimmed the power traces so that the average power differential between them was no more than 10ns. Figure5.13 does not show any spike, but by scaling the power trace from lower Sample Points, we can detect the spike at Sample Point 495 in Figure5.14's guessed-key plot. Any peak less than 3 ns cannot be ascribed to the DPA experiment since the ideal circuit is not functioning in the simulation. Even yet, noise at this moment may cause a surge.

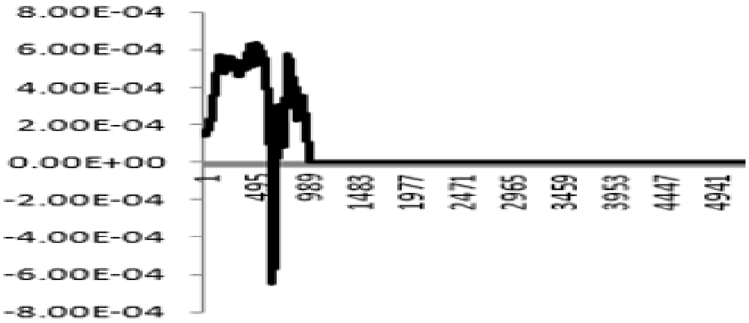


Figure 5.14 The differential curve, magnified for a lower sampling value

The sample point and the predicted key must be correlated between the time it takes for the input vector to switch and the delay introduced by the AddRoundkey stage. By making guesses at each conceivable key, DPA can then use the resulting differential key to determine which supposed key yielded the largest spike hits. In the first round of a DPA attack, the following limitations of the DPA attack were observed [191].

a. Curve 0 is created by the non-target bit in a way that is unrelated to Curve 1 created by the Target Bit.

b. It is not the case that the correct key guess is reflected in the expected value of the target bit for the wrong key.

c. Noise causes a peak in the DOM waveform, known as a ghost peak.

It can be difficult for the attacker to make an accurate prediction if the correct peak is smaller than the ghost peak.

5.5.2 Correlation power Analysis

The success of a CPA attack on SBOX hinges on the number of traces necessary to reliably forecast the key used in the AddRoundkey inputIn a CPA attack, the attacker uses hamming weight and hamming distance to construct a hypothesis power model, and then checks the simulated power trace for indications of a link to that model [192]. Determine the data-dependent dynamic power usage with a 25H estimate and the plain text 00-FFH technique. If we assume the key is 0, we can apply the equation defining the input to SBOX to get the power value of each of the plain texts 00-FFH. Repeat the preceding process 256 times while assuming the key. The next step is to determine whether or not the 256-hypothesis power value is significantly different from the simulated power value using Pearson correlation. Figures 5.15 and 5.16 depict the Pearson correlation coefficient between actual power values and their predicted values based on the hammering weight model and the hammering the distance power model, respectively. When the wrong key is entered, the curve flattens down, whereas when the right one is entered, a peak appears. In the case of the postulated key 25H, the peak correlation coefficients found using HWPM and HDPM are 0.3245 and 0.39644, respectively. When the hypothesis model and the simulation model are in agreement with the guessed key, the correlation coefficient reaches its maximum value. An increase in HDPM over HWPM during the CPA indicates a more precise method of obtaining the attack key.

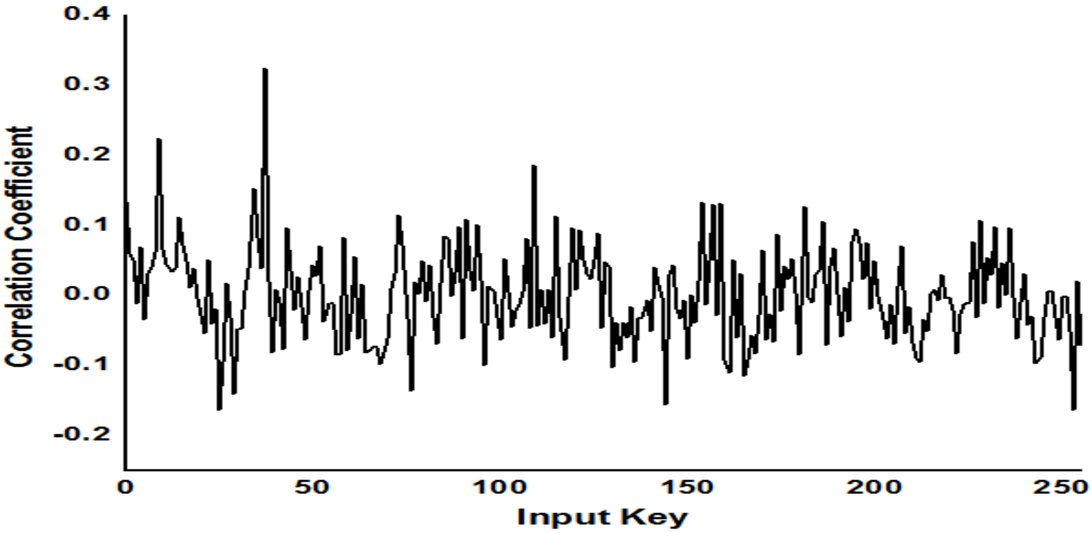


Figure 5.15 The suggested mask design's correlation to the right guess key at 25 HWPM.

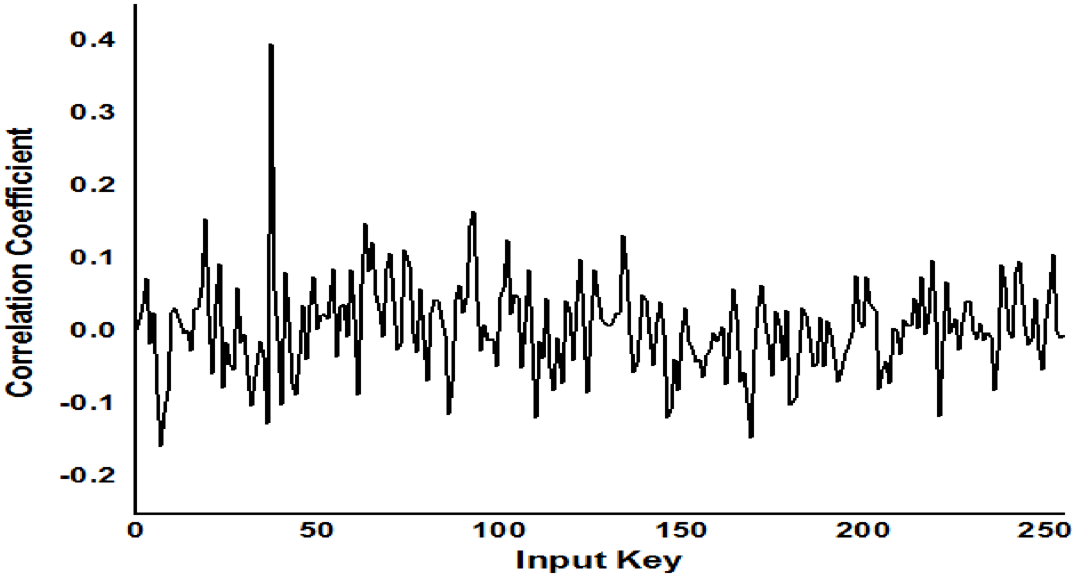


Figure 5.16 The proposed design's correlation with HDPM's ability to correctly estimate 25H keys

Power traces for the HWPM and HDPM models were acquired using the secret key 25H, while the hypothesised key was 12H. This resulted from an additional experiment using CPA. The correlation coefficient between the real power trace and the HWPM and HDPM power value with incorrect key 12H is depicted in Figure 5.17, and the same coefficient is depicted in Figure 5.18 for the 256 possible combinations of plain text 00-FF. The curve of the correlation coefficient looks like a ghost at the summit who can't decide which key to use. Assuming a bad key, the attacker tried another guess until the curve reached its top. Since this is the case, the CPA attack can successfully deduce the secret key. CPA-resistant SBOX that conceals the key during a power attack was made possible by this discovery.

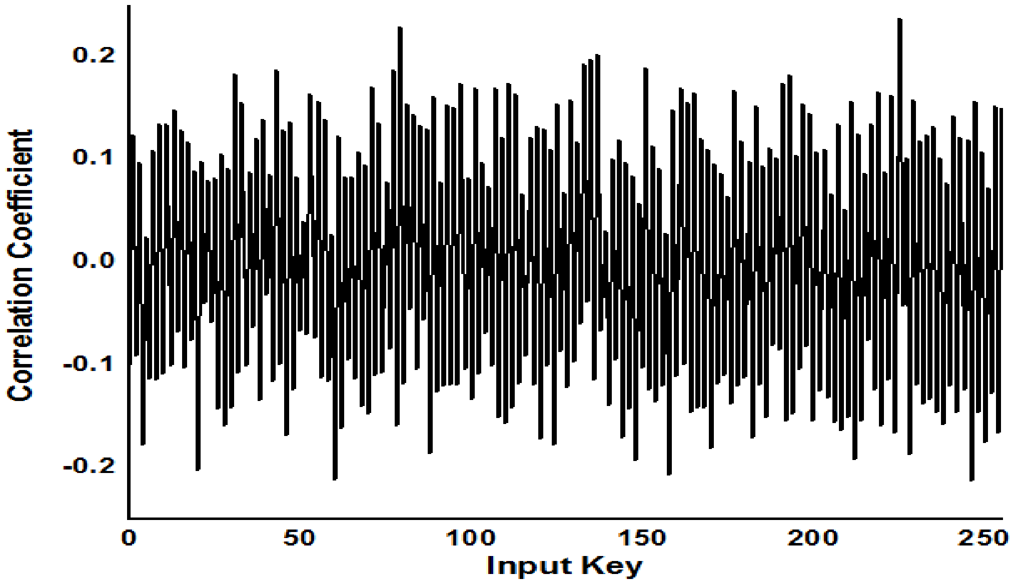


Figure 5.17 Proposed architecture with HWPM's correlation to wrong guesses of the key

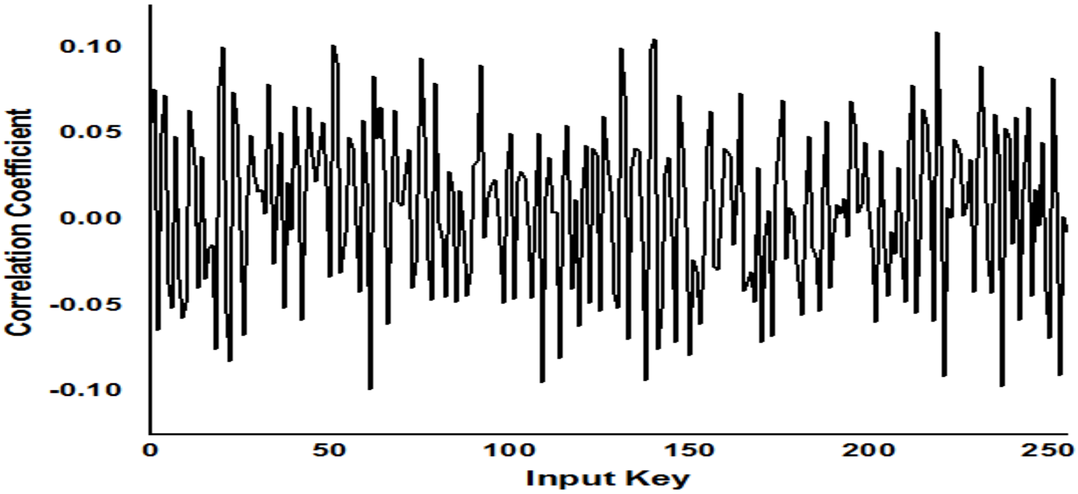


Figure 5.18 Proposed mask design based on a correlation coefficient with HDPM's ability to incorrectly guess keys

Hamming Weight and Hamming Distance can detect key-dependent processes in cryptographic algorithms. The attacker can detect key-dependent power usage during these procedures. An attacker uses correlation analysis to link power usage with the Hamming Weight or Hamming Distance of intermediate values or key bits during a CPA attack. Higher correlation at some periods suggests a breach, allowing the attacker to recover important data. Template Attacks: Attackers may develop templates based on projected power consumption patterns for given Hamming Weight or Distance values. Key-dependent patterns can be found by correlating these templates with cryptographic power usage. Analysing the statistical distribution of Hamming Weight and Hamming Distance might help attackers pinpoint power consumption zones of interest. changes from statistical expectations may suggest key-dependent changes. These attacks can be prevented by randomising processes, masking, or employing secure hardware to decrease the link between power use and sensitive data. Designing safe cryptographic systems resistant to side-channel attacks like CPA requires understanding power consumption, Hamming Weight, and Hamming Distance.

Chapter 6

Defence against Side-Channel Attacks

6.1 Introduction

Scaling has a benefit in terms of reduction in area, but continual scaling increases the need for power [193, 194, 195]. A new class of security attacks is produced as a result of the data-dependent component of power usage. Adding security at the algorithmic level to the IC is insufficient. Attack resistant IC is protection that is in line with various design levels throughout design and manufacture. Low power VLSI circuit design follows the CMOS based cell library standard by default [196]. The two types of countermeasure approaches that are currently in use, hiding [104, 197] and masking at the cell level, have both been put into practise. The criteria for concealing countermeasures make the power requirements of the cryptography module dependent on both, the required operation and the intermediate values. The correlation coefficient for all hypothesis keys is near to zero when using the hiding strategy, making it inconclusive to draw the right conclusion. Three-phase dual-rail precharge logic, wave dynamic differential logic (WDDL) [198, 199], and sense-amplifier based logic (SABL) have all implemented hiding techniques (TDPL). The concealment method alters the circuit's power usage on different scales. The input signals are randomised by the masking approach to avoid input-output data dependence. The dependence between actual consumption of power and hammer weights or input’s hammer distance is used in the power analysis attack. The final output terminal's hammering weight determines the cryptographic circuit's output. Multiple internal nodes are created by the power attack resistant mask circuit; instead of relying on only one node, the power value depends on all of the internal nodes. The purpose of a masking countermeasure is to mask the input to SBOX before any cryptographic operations are performed. The overwhelming amount of data that must be handled appears random from the outside. The output of SBOX depends on mask data rather than the real data that needs to be processed (unmasked). After masking or scrambling the input to SBOX using a random bit, make the substitution as indicated in figure 6.1.

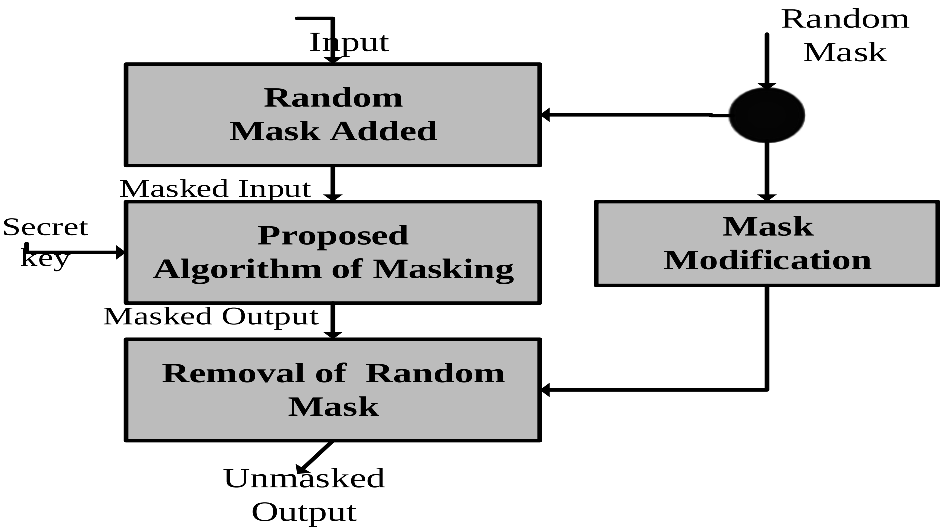


Figure 6.1 Process of Boolean Masking

The mask bit is once more unmasked or descrambled at the output terminal using an internal or external circuit to generate random bit. Removing or lessening the association is the aim of resistance to a power attack. Practically speaking, the data dependency cannot be completely eliminated. In order to lessen the correlation between actual and expected power, noise or a lower power value may be added. An algorithmic level masking necessitates rewriting of the algorithm, which adds more work, and masking at the algorithmic level is implemented without affecting the characteristics of consumption of power for cryptographic circuits. Masking is a technique used at the algorithmic or gate level to randomise the internal outcome.

Using the masked logic gate to create the cryptographic circuits depicted in [200, 90] is an alternate approach of masking. It was deduced from the SBOX diagram that an XOR-AND ASIC cell is needed. Lower correlation between hypothesis and real power is ensured by SBOX design with mask cell. In order to prevent wire from storing a value correlated with the algorithmic amount, mask cells randomise the intermediate value.

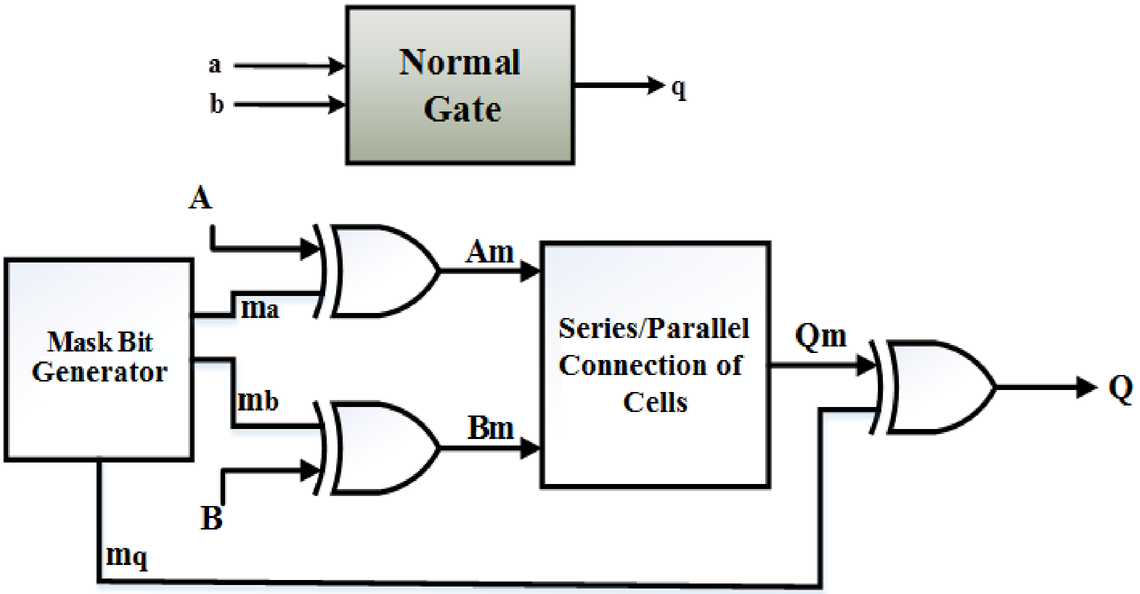


Figure 6.2 Unmask and mask gate

The cell's security analysis is a further argument in favour of employing the masquerade cell to target resistant cells. The security computation was flawed since it assumed the mask cell's input never fluctuates throughout a single clock cycle. Multiple output transitions per clock cycle and transitions that occur before output stability to the final value are examples of glitches. [87, 201].

6.2 Masked Cell

Figure 6.2 [79, 86, 202] explains the normal and masks gate architecture, where by using mask bit generator circuit the normal gate is modified. Mask generators create a mask bit that is XORed with the input signal to alter the signal. An externally or internally generated mask bit can be used in the same way to de-mask the output of a mask gate. In [186], q is the standard output of a gate and is shown to be a function of inputs a and b. (a,b). The mask bit ma is applied to input a, changing it to am, and the mask bit mb is applied to input b, changing it to bm. Similarly, if we use mask bit mq to unmask mask output qm, we get q instead of m. There are five inputs to a mask gate.

$$q_m = f(a_m, m_a, b_m, m_b, m_q)$$ [54, 59].

$$\begin{aligned} a_m &= (a \oplus m_a) \\ b_m &= (b \oplus m_b) \\ q_m &= (q \oplus m_q) \end{aligned} \tag{6.1}$$

The study of mask cells was carried out on the assumption that the value of any given cell does not change by more than one every clock cycle and that the propagation delay of any given gate is zero, which means that glitches were not considered during design.. There is no need to supply any power when the voltage stays the same. The voltage at the principal input/output or intermediate terminal in a digital circuit represents the logic level. As was previously said, the data being processed has a huge impact on the amount of energy required to advance the logic a level in a digital circuit. On the other hand, according to [203], when the level is held constant, no energy is consumed. The amount of power needed to flip a logical 0 to a logical 1 is For (0-1) switching, E_{0-1} displays the transition energy required. Similarly, the E_{1-0} is the 1-0 transition energy. Energy consumption when charging the load capacitor is denoted by E_{0-1}, while energy dissipation during discharge is denoted by E_{1-0}. Given that $E_{1-0} > E_{0-1}$. Keeping your mind at the same level of reasoning doesn't cost anything, hence $E_{1-1} = E_{0-0} = 0$.

6.2.1 Security Analysis of Unmasked Cell

Assuming that the input to the gate is uniformly distributed and arrives at the terminal at the same time as the output, this work explores the power-assault countermeasure of unmasking an XOR or AND cell. According to Table 6.1, there are sixteen distinct transitions that can occur in a 2-input gate. Complete Input and Transition Energy Output from XOR and AND Gates. The 16 permutations can be broken down into two groups, the first of which contains transitions that result in q=0 and the second of which results in q=1. The first set consists of changes with scores of 0-0 and 1-0, while the second set consists of changes with scores of 1-1 and 0-1.

In XOR gate 8 times q =1 and 8-time q=0.average power consumption for q=0 $E_{(y=0)}$ and average power requirement for q=0 $E_{(y=1)}$

$$E_{(y=0)} = \frac{4E_{(0-0)} + 4E_{(1-1)}}{8} \;\; and \;\; E_{(y=1)} = \frac{4E_{(0-1)} + 4E_{(1-0)}}{8}$$

Similarly, for AND gate q=1, 4 times and q=0, 12 times.

$$E_{(y=0)}=\frac{9E_{(0-0)}+3E_{(1-1)}}{12}\ \ and\ \ E_{(y=1)}=\frac{3E_{(0-1)}+E_{(1-0)}}{4}$$

Average energy requires to compute XOR output high output $E_{(y=1)}$ is not equal to compute low output $E_{(y=0)}$ presented in equation (6.2).

For XOR gate

$$\begin{aligned}
&E_{(y=0)}=4E_{(0-0)}+4E_{(0-1)}=13.18\,fJ\\
&E_{(y=1)}=4E_{(0-1)}+4E_{(1-1)}=11.79\,fJ\\
&Thus\ \ E_{(y=0)}\neq E_{(y=1)} \qquad (6.2)\\
&Average\,E_{(y=0)}=1.6475\,fJ\\
&Average\,E_{(y=1)}=1.47\,fJ
\end{aligned}$$

Table 6.1 Transition Energy for unmasked XOR-AND cell

Input Transition		XOR			AND		
A	B	Y	Energy (fW)	Energy to settle output node Y	Y	Energy (fW)	Energy to settle output Node Y
0-0	0-0	0-0	0.06	E_{0-0}	0-0	0.07	E_{0-0}
	0-1	0-1	1.93	E_{0-1}	0-0	0.08	E_{0-0}
	1-0	1-0	1.92	E_{1-0}	0-0	0.017	E_{0-0}
	1-1	1-1	0.07	E_{1-1}	0-0	0.004	E_{0-0}
0-1	0-0	0-1	1.5	E_{0-1}	0-0	0.09	E_{0-0}
	0-1	0-0	2.04	E_{0-0}	0-1	2.1	E_{0-1}
	1-0	1-1	2.52	E_{1-1}	0-0	2	E_{0-0}
	1-1	1-0	1.3	E_{1-0}	0-1	2.01	E_{0-1}
1-0	0-0	1-0	1.78	E_{1-0}	0-0	0.01	E_{0-0}
	0-1	1-1	2	E_{1-1}	0-0	0.11	E_{0-0}
	1-0	0-0	1.9	E_{0-0}	1-0	2.17	E_{1-0}
	1-1	0-1	1.8	E_{0-1}	1-0	1.92	E_{1-0}
	0-0	1-1	0.13	E_{1-1}	0-0	0.01	E_{0-0}
	0-1	1-0	2.5	E_{1-0}	0-1	2.1	E_{0-1}

1-1	1-0	0-1	2.1	E_{0-1}	1-0	2.07	E_{1-0}
	1-1	0-0	0.1	E_{0-0}	1-1	0.01	E_{1-1}

Average energy requires to compute AND output high output $E_{(y=1)}$ is not equal to compute low output $E_{(y=0)}$ presented in equation (6.3).

For AND gate

$$\begin{aligned} &E_{(y=0)} = 9E_{(0-0)} + 3E_{(1-0)} = 8.58\,fJ \\ &E_{(y=1)} = E_{(0-1)} + E_{(1-1)} = 6.22\,fJ \\ &Thus\,E_{(y=0)} \neq E_{(y=1)} \\ &Average\,E_{(y=0)} = 0.715\,fJ \\ &Average\,E_{(y=1)} = 1.555\,fJ \end{aligned} \tag{6.3}$$

6.2.2 Power Analysis of Masked Cell

If the average of $E_{(y=0)}$ and $E_{(y=1)}$ is more than 1, then the leakage can be measured by the attacker alongside the side channel data. The cell is deemed to be resistant to attack if the average of $E_{(y=0)}$ and $E_{(y=1)}$ is zero. If the Average $E_{(y=0)}$ minus the Average $E_{(y=1)}$ for the gate in use is zero, then the mask cell's goals have been met [55]. Getting there is as simple as making sure that all transitions have the same amount of energy, i.e., E_{00}=E_{01}=E_{10}=E_{11}. The threshold for the number of bits required and the accompanying demands on storage space and energy production is raised when first-order masking is expanded to multiple order bits. There is only one element of security in use in first-order masking. Separate mask bits m0 and m1 are applied to the inputs a and b, resulting in the two values a_0= (a $\oplus$ m_0), a_1=a, b_0= (b $\oplus$ m_1), b_1=b and b1=b. You're free to use the same or a different mask bit. Not all four possible permutations of m0m1 can be handled by the same mask gate, thus the mask generator must provide a mask bit that is compatible with the chosen mask gate. As the truth-table of an XOR gate is always the same, the input is XORed with the mask bit. The mask gate improves the XOR gate's architecture by adding to the gate's number at the input and output terminals. Two new XOR gate architectures and one new AND gate architecture have been suggested. The truth table for the mask gate demonstrates that the hammering weights at the intermediate terminal and the output terminal are equal for all four possible configurations.

6.2.2.1 Proposed Masked Cell (XOR-1)

Mask XOR cells with the suggested architecture are built using a combination of a 4-XOR and a 1-AND gate. The intermediate signal is given at the computed unmask output from the AND gates T0, T1, T2, and T3.

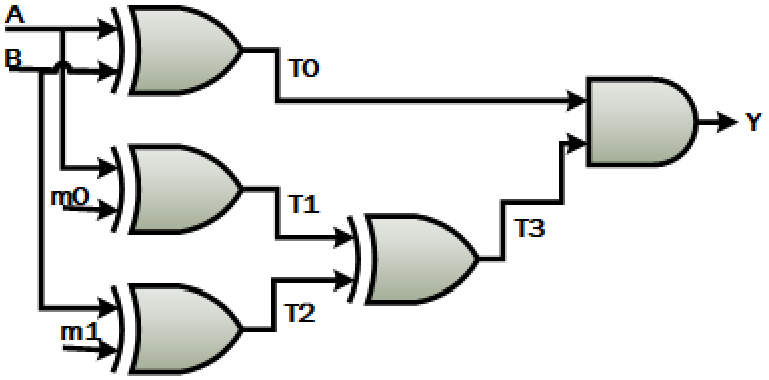


Figure 6.3 Mask XOR-1 proposed cell

$$Boolean\, \exp ression\, of\;\; masked\;\; XOR - 1$$
$$T0 = A \oplus m0$$
$$T1 = B \oplus m1$$
$$T2 = A \oplus m1$$
$$T3 = T1 \oplus T2$$
$$Y = T0 \bullet T3$$

Table 6.2 Mask XOR-1 Cell Truth Table with Mask bit

Input		Mask bit 11					Mask bit 00				
A	B	T0	T1	T2	T3	Y	T0	T1	T2	T3	Y
0	0	0	1	1	0	0	0	0	0	0	0
0	1	1	1	0	1	1	1	0	1	1	1
1	0	1	0	1	1	1	1	1	0	1	1
1	1	0	0	0	0	0	0	1	1	0	0
HW		2	2	2	2	2	2	2	2	2	2

6.2.2.2 Proposed Masked Cell (XOR-2)

Three XOR gates are used for the mask input, two XOR gates are used for the intermediate signals (T0-T4), and the unmask output Y is generated by an XNOR gate.

$$Boolean \exp ression\ of\ mask\ XOR-2$$
$$T0 = A \oplus m0$$
$$T1 = m0 \oplus m1$$
$$T2 = B \oplus m1$$
$$T3 = T0 \oplus T1$$
$$T4 = T1 \oplus T2$$
$$Y =\sim (T3 \oplus T4)$$

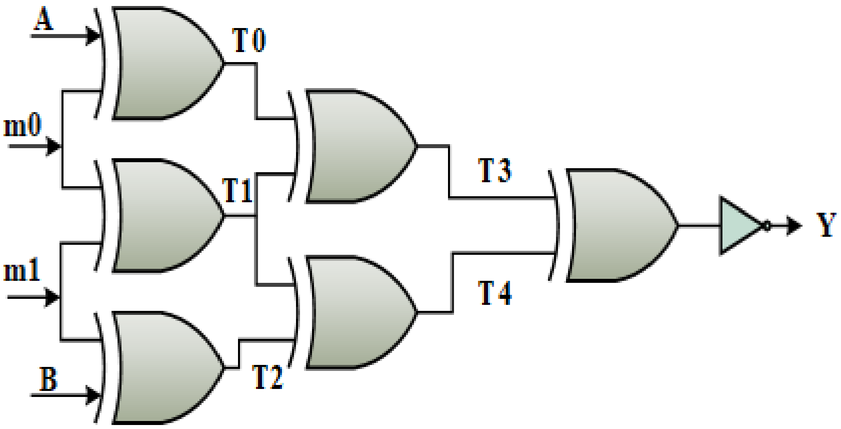


Figure 6.4 Proposed mask XOR-2 cell

Table 6.3 Mask XOR-2 Truth Table with Mask bit

Input		Mask bit 10						Mask bit 01				
A	B	T0	T1	T2	T3	T4	Y	T0	T1	T2	T3	Y
0	0	1	1	0	1	1	0	0	0	0	0	0
0	1	1	1	1	0	0	1	1	0	1	1	1
1	0	0	1	0	1	1	1	1	1	0	1	1
1	1	0	1	1	0	0	0	0	1	1	0	0
HW		2	2	4	2	2	2	2	2	2	2	2

6.2.2.3 Proposed Masked Cell (AND)

Specifically, a 5-XOR gate and a 3-AND gate are needed to implement the proposed mask AND cell. The current inputs A and B are hidden by the mask bits m0 and m1, and it is the intermediate stage signals T0 through T6 that control power distribution within the cell's internal terminal. The final result is calculated and sent out through Y.

Boolean expression of mask AND cell

$$
\begin{aligned}
&T0 = A \oplus m0 \\
&T1 = m0 \oplus m1 \\
&T2 = B \oplus m1 \\
&T3 = T0 \oplus T1 \\
&T4 = T1 \oplus T2 \\
&T5 = T0 \bullet T3 \\
&T6 = T2 \bullet T4 \\
&Y = T5 \bullet T6
\end{aligned}
$$

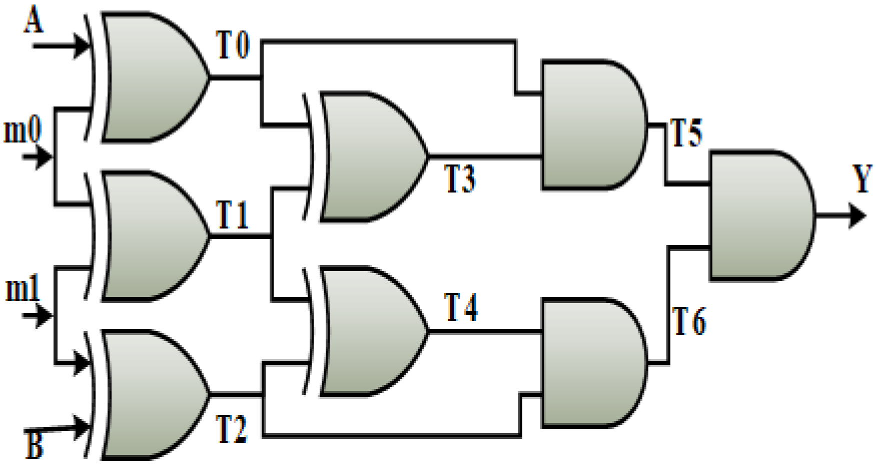


Figure 6.5 Proposed mask AND cell

Table 6.4Truth Table of Mask AND with Mask bit

Input		Mask bit m0m1= 00							
A	B	T0	T1	T2	T3	T4	T5	T6	Y
0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	0	1	0	1	0
1	0	1	0	0	1	0	1	0	0
1	1	1	0	1	1	1	1	1	1
HW		2	0	2	2	2	2	2	2

Each input to the proposed gate is masked using another XOR gate. The mask XOR-1, mask XOR-2, and mask AND from Figures 6.3, 6.4, and 6.5 are all represented in the Boolean equation at the same time. Bits 00 and 11 of the mask are processed by the XOR gate, whereas bits 01 and 10 are processed by the XNOR gate in mask XOR-1. Bits 01 and 10 can be masked to create an XOR gate, while bit 00 can be masked to create an AND gate. Unmasking the mask gate's output may be done in a few different ways: using a newly generated bit, by using the AND gate on the mask XOR-1 or XOR-2 gate's output, or by using the XNOR gate on the mask AND gate's output. Tables 6.2, 6.3, and 6.4 show the truth tables for mask XOR-1 cells, mask XOR-2 cells, and mask-AND cells, respectively, with a uniformly distributed "2" at the internal terminal. Even if the value of the wire at the internal terminal is unaffected by the equal hammering weight contained in the wire, the attacker still has to know the value of each intermediate terminal in order to crack the code using a mask gate. At the same time, the unmask gate can cut off its connection to the terminal-less output.

6.2.3 Security of Masked Gate

However, the improved security provided by mask gates comes at the expense of higher gate counts and higher energy requirements. For a gate to be resistant to attack, the intermediate outcome must be evenly distributed and decoupled from the gate's primary input. Since the hammering weight is distributed uniformly, the output is unrelated to the input, making it more secure against attacks that rely on predicting secret data, such as that displayed in [204] of the circuit. The hammering weight of a masked AND gate is more unevenly distributed than that of a masked XOR gate. These are potential entry points for an assault that could compromise sensitive data. According to Hamming's weight, the quantity of energy needed to change levels is quite high. Transitioning from low to high in a standard gate requires more power than the opposite. It's preferable to have a small average energy gap between the masked gates. To determine whether the masked XOR-1 output terminal is high or low, we use the formula E(y=1) - E(y=0) = 0:15fJ, the same formula for the masked XOR-2 output terminal yields 1.5625fJ, and the same formula yields 5.6fJ for the masked AND cell. In comparison to the static unmask circuit, the switching energy required by the masked version of XOR-1 is reduced by 82.9%. That's why it follows that the proposed disguised XOR gates are promising potential defences against attacks. Both the normalised energy deviation (NED) and the normalised standard deviation (NSD) are useful metrics for gauging power attack resistance, as demonstrated in equation (10). The minimum and maximum energy differential, in percentage, are shown by NED for each possible transition. When NED is low, there is little to no energy fluctuation, requiring more sophisticated countermeasures from the attacker.

When the average energy is known, the NSD is calculated as (E / EAvg), where E is the standard deviation of the energy. The NSD indicates the spread of consumed energy around the mean; a large NSD number suggests that the energy is widely dispersed; a small NSD value implies that it is clustered around the mean. Analysis of the NED and NSD at a frequency of 1GHz; a smaller number for each indicates a more secure system; chapter 4 covers the attacker's need for more precise measuring and resistance to power attacks. The performance of a masked XOR-AND gate is compared in Table 6.5 to that of a static unmask CMOS design using the same mask bit. Using the suggested masked XOR-1 with mask bit 00 yields a 2.3% improvement in NED, whereas using mask bit 11 yields a 1.5% improvement. There is a 1.7% and 1.89% increase in NED when using mask XOR-2 with mask bits 01 sand 10. The masked XOR gate's performance is improved by NSD by 6.95% for mask bit 00 and by 3.29&percnt; for mask bit 11. At a 1GHz input frequency, the masked AND gate proposed to replace the mask bit 00 gate increases NED by 0.03% and NSD by 25.75 %. The masked XOR-AND gate does not show a notable drop in NED because of the unequal distribution of hamming weight at the internal node. Since the AND gate has a non-uniform hamming weight at the intermediate terminal, the NSD is more effectively reduced in the masked AND gate than in the masked XOR gate. The suggested masked XOR-2 and AND cell reduce NSD by 20% and 18.94%, respectively, compared to that of a static unmask CMOS circuit running at 1 GHz, despite the fact that cryptanalysis requires complex measurements. Finding your way to the heart of the circuit and learning its secrets will be just as challenging. You can see the percentage drop in Pearson's correlation coefficient between the masked XOR-AND gate and the static unmask design in Figure 6.6. Table 6.5 shows that the person coefficient is significantly lower in the masked gate compared to the static unmask cell. Normal XOR has a Pearson coefficient of 0.134, while mask XOR-1 and 2 have values of 0.0053 and 0.0239, and mask XOR-3 and 4 have values of 0.0369 and 0.0416, respectively. The mask AND cell has a Pearson value of 0.3, whereas the normal cell has a coefficient of 0.372. Reduced mask cell counts demonstrate input data independence for power patterns. Thus, the security level of the method can be improved at the expense of an increased hardware cost by implementing the cryptographic circuit with a masked gate. In contrast to the four AND and two XOR cells needed for the masking approach of an AND cell in [181], the suggested mask AND cell only needs two XOR and one AND cell.

Table 6.5 Mask gate energy correlation and energy parameter

Energy Parameter							Pearson Coefficient (ρ)
	E_{Min}	E_{Max}	NED	E_{Avg}	σ_E	NSD	
Static unmask XOR	0.06	2.52	0.9762	0.1538	1.5748	0.6402	0.134
Mask XOR-1 (00)	0.01	7.9	0.9987	0.4931	5.4026	0.6847	0.0053
Mask XOR-1 (11)	0.1	12.1	0.9917	0.75	7.9841	0.6653	0.0239
Mask XOR-2 (01)	0.1	15.4	0.9935	0.9563	7.8316	0.5119	0.0369
Mask XOR-2 (10)	0.09	16.9	0.9947	1.0506	8.7227	0.5189	0.0416
Static unmask AND	0.004	2.17	0.9982	0.1354	1.2715	0.587	0.372
Mask AND (00)	0.06	40	0.9985	2.4963	17.401	0.4357	0.3

The SVM supervised learning model is a classification approach that relies on the power of support vector machines. When the bit mask is applied, the performance of the SBOX is predicted using the support vector machine's cost and epsilon. The price tag reflects how much leeway SVM gives the user in terms of data manipulation. In general, wider margins and more easily made decisions correspond to lower cost values, while narrower margins and more categorizations of points correspond to higher cost values. The larger the amount of epsilon, the more room there is for inaccuracy without negative consequences being imposed on the model. When using a mask with C=4 and epsilon=0.97, the implemented SBOX achieves C=8.

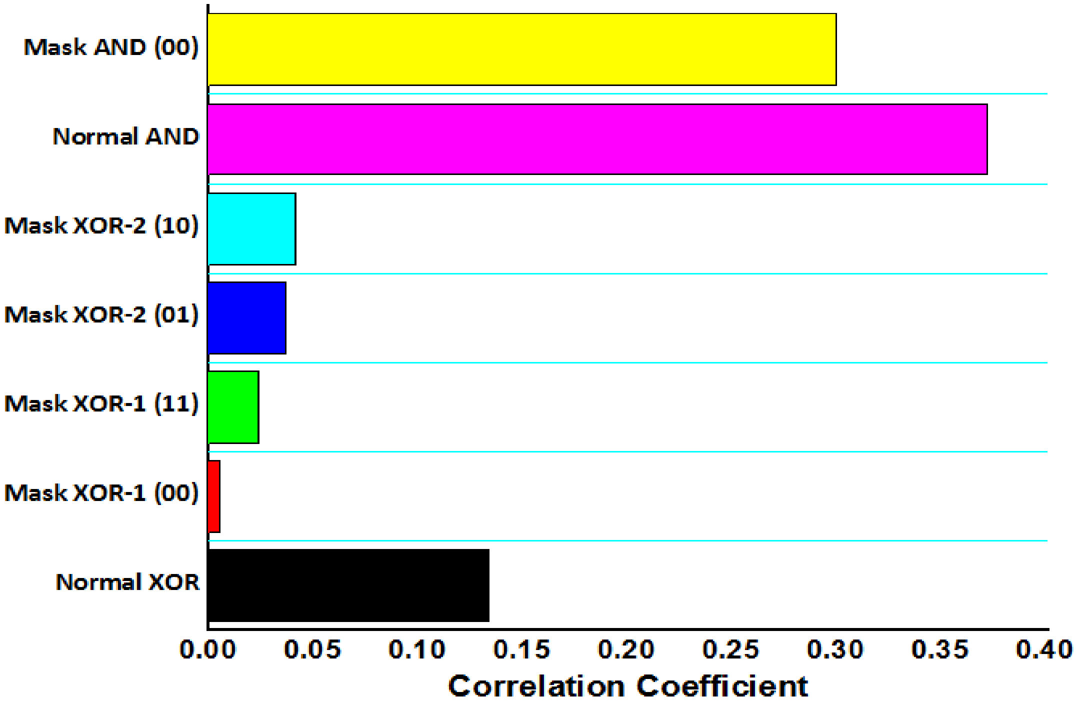


Figure 6.6 Normal and Mask cell Pearson correlation coefficient

6.2.4 Masked Cell with Biased Bit

One common method of defending against a power attack is the use of secret sharing, for which masking is well-known. Discreet data is divided into (d+1) equal parts, where d is the masking order. The underlying cell of SBOX is masked in this work using only 2 random bits. Using the XOR gate, the first-order mask can be decomposed into its component halves. Application of a really random value for the mask bit is required when masking a nonlinear block like SBOX. The ideal mask would have a probability of 0.5, where both the occurrence of '0' and '1' are equally likely (p0=p1). After passing sensitive data through an XOR gate with a random mask, the resulting masked values, sm = (s⊕m), are statistically indistinguishable from the original data. When comparing s and sm, there is no mutual information. Under imperfect masking, the chance of an event happening is (0.5 + ε), where ε≠0. One value's likelihood of occurrence increases by, while the other value's likelihood lowers by. When secret data is created randomly with a bias mask bit leakage, it is possible to recover the variable even if countermeasures are taken [205]. The distribution of power has become unequal. According to the equation given below, there is a link between information leakage analysis (ILA), power tracing, probability following, and likelihood (6.4).

When a bitmask is skewed, the amount of data that can leak out depends on how much of a bias there is. A plot of ILA versus p is shown in Figure 6.7; the mask bit flips between 0 and 1 as seen by the symmetry of the curve about the x-axis. A circuit with no masking protection, as indicated by p = 0 or p = 1, leaks the most possible information. A significant number of power traces are needed to collect enough data when p = 1/2, when the circuit is protected by complete masking. If the masking is faulty or biassed (p≠0), more information is leaked as p gets closer to the axis's terminals. This is due to the fact that a power attack requires a negative attack (ILA) proportionate to the number of power traces. The assault can be launched with less power tracking.

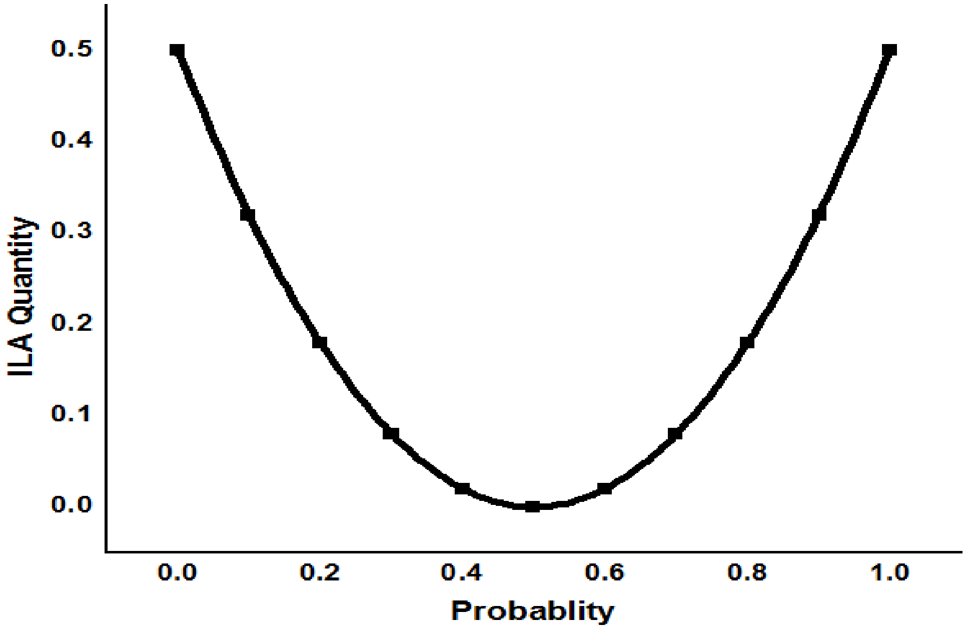


Figure 6.7 ILA vs mask bit occurrence likelihood

6.3 Attack model on Masked Cell Design

The suggested mask XOR-1 and mask AND gate have been implemented in a functional implementation of SBOX. The SBOX computing algorithm shown in this chapter substitutes mask XOR-1 mask AND for the more basic XOR-AND. By guessing the secret key, simulated power traces of an SBOX are obtained, allowing the correlation power attack to be examined. The range of possible values for 12H plain text is from 00 to FFH. Figure 6.7 shows the association between hammer weight and power trace, while Figure 6.8 shows the same relationship for hammer distance and power trace. In contrast to the SBOX mask, the correlation coefficient that reveals the true power consumption in connection to several plain texts shows many peaks for SBOX. This makes it impossible to successfully guess the correct key. SBOX with the suggested mask requires 13260 BSIM3V3 MOS cells, an increase of 5.7% compared to SBOX with static cells. Since the number of cells in the Mask SBOX has grown, the leakage current has built up, resulting in a significant amount of wasted energy (55.96µW). There is a total propagation delay of 1.567ns and a dynamic power usage of 964.3µW for the mask SBOX.

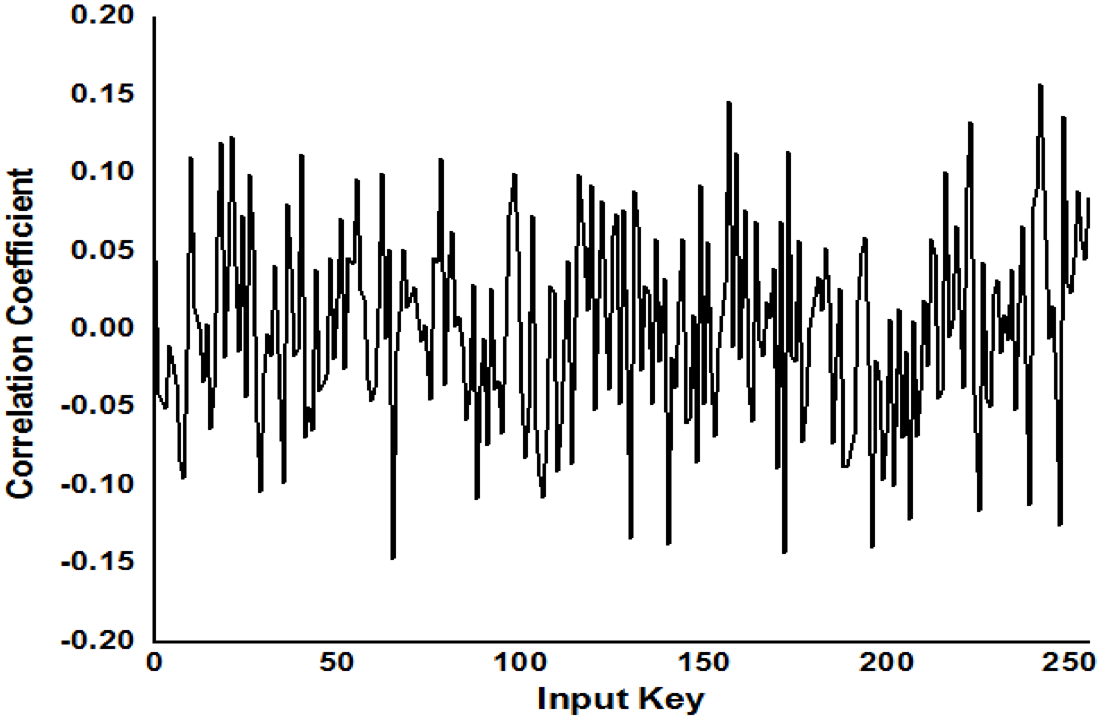


Figure 6.8 Pearson's correlation between the accurate guess and the actual key for the SBOX mask and HWPM

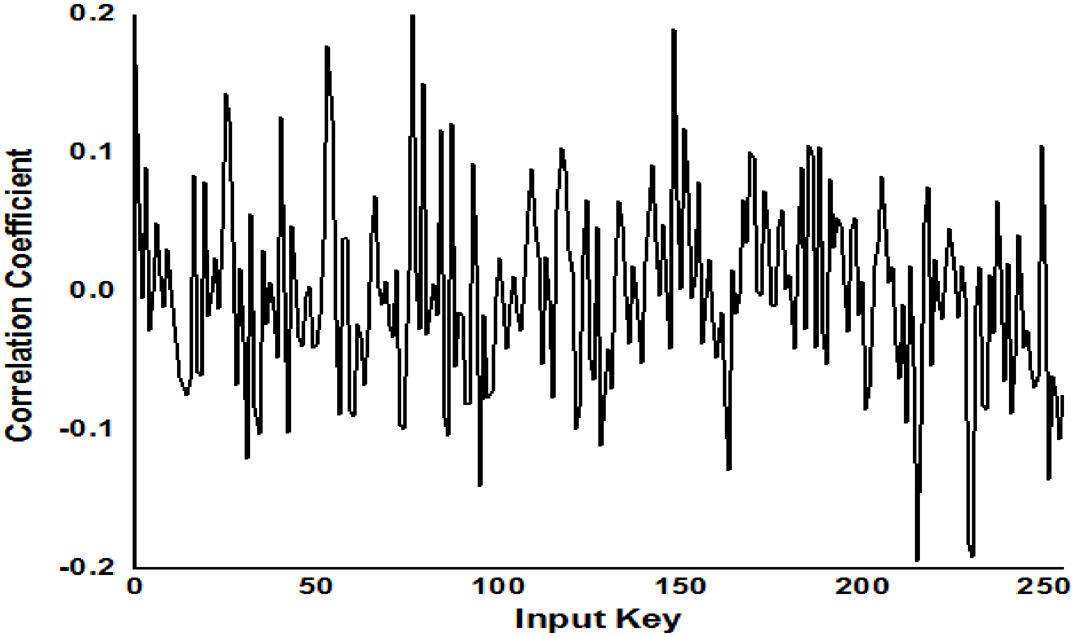


Figure 6.9 Proposed mask design with HDPM correlation with correct key guesses

Figures 6.10 and 6.11 display the Pearson correlation () with different plaint text between the simulated power consumption and the hamming weight (HW) and the hamming distance (HD) power model in comparison to the unmask cell SBOX. The correlation value is reduced for cells in an SBOX mask. The power consumption of a masked SBOX implementation is inferred to be less text-dependent than that of an unmasked counterpart. The correlation coeffcient for plain text 12H is enhanced by 57.86% when using HWPM, and by 49.08% when using HDPM. Correlation coefficients for HWPM and HDPM implementations of SBOX using mask cells with various assumed keys. With the correlation coefficient drastically down, we may conclude that the amount of energy used by a mask cell is unrelated to the data being processed.

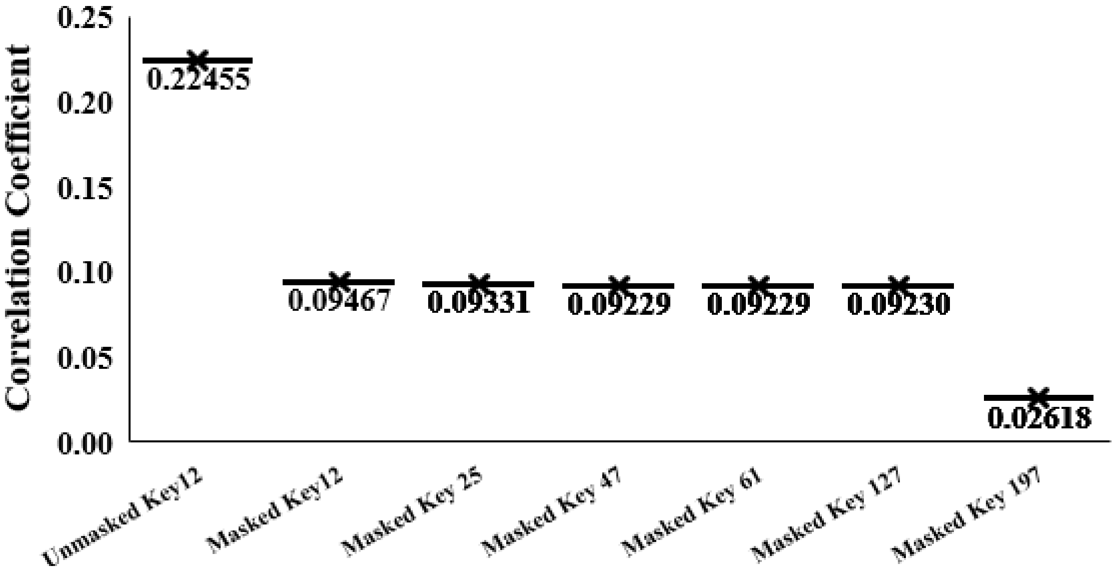


Figure 6.10 The right key's HWPM correlation coefficient is compared to that of an SBOX under a CPA attack.

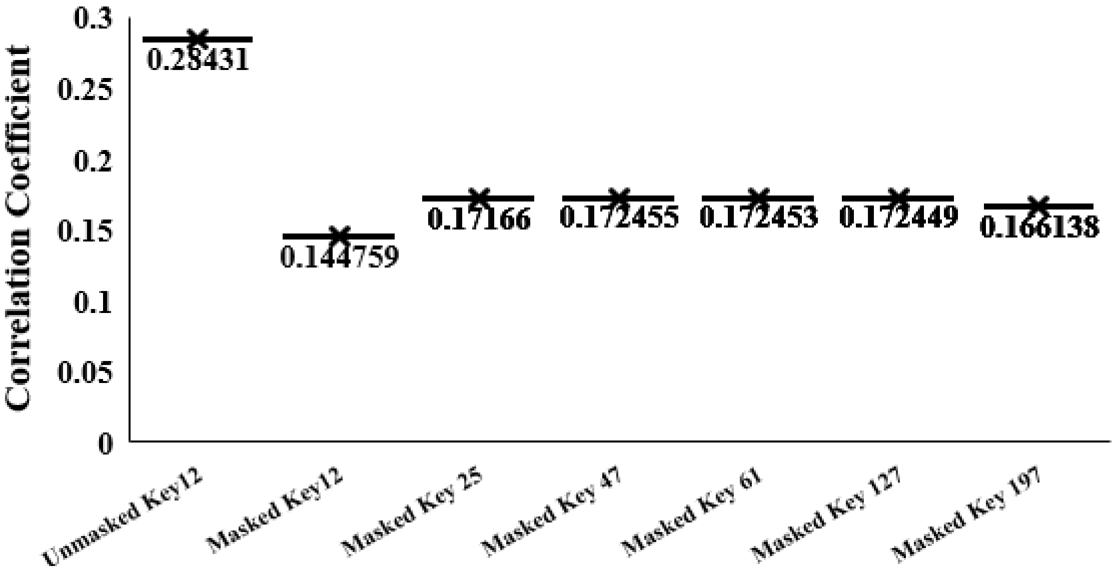


Figure 6.11 SBOX under CPA attack: comparing correlation coefficient (ρ) to HDPM's "right" key

6.4 Results and Discussion

SBOX has been built in this study using the CMOS 90nm technology node because to the fact that the power trace always gives off some type of detectable side-channel information. With the help of a resistor with a relatively low value and a power source, we were able to gather the data we needed. Each input to the SBOX elicits a different response and uses a different amount of power. The amount of energy used by an SBOX is based on the number of high bits and the number of changes made to the input. A high bit in the most significant bit (MSB) consumes more energy than a high bit in the least significant bit (LSB) (MSB), and vice versa.

The preceding section dissected three distinct power attack models: the single-point attack (SPA), the double-point attack (DPA), and the central-point attack (CPA). The hamming weight (HW) and hamming distance (HD) power models form the basis for the DPA and CPA attack methods, respectively. An attack of power based on the realisation that a certain numerical number is correlated with actual power. An attacker trying a DPA attack will assume that they know the secret key and will try every possible combination from 00 to FFH. Input power trace for SBOX with LSB set to 1 against input power trace for SBOX with LSB set to 0. If you don't know the location of the key peak, you can use the DOM waveform as a guide, as the waveform will flatten out toward the horizontal axis otherwise. The highest point on the DOM curve is at position 495. When an attacker uses a DPA attack, they see the correct key as the peak of the curve and end up with their best estimate as the correct one. The presence of noise in the circuit is a major limitation of DPA attacks; unexpected oscillations in the curve are occasionally noticed due to noise, and the peak is sometimes discovered on the wrong key. In order to guess the key, CPA uses fewer traces than DPA, making it a more efficient method. The power model based on hammering weight and hamming distance has been utilised to find a correlation between the measured power and the calculated power using the Pearson correlation coefficient. When launching a CPA attack, the attacker first attempts to guess the input text key by trying every conceivable key between 00 and FFH.

Determine the power consumption correlation coefficient between the user's guesses and the actual usage. Plot the correlation coefficient with regard guessed key. If the key was guessed correctly, a distinct peak will be seen, and if not, a ghost waveform will be seen. The CPA attack allows the secret key to be guessed with a high degree of accuracy using imagination. In order to increase the safety of the currently utilised encryption technique, it was necessary to employ attack countermeasure cells in the development of the requisite hardware architecture. As can be seen in the correlation coefficients (0.2905 for hamming weight and -0.273 for hamming distance) between input data and power consumption, the hardware design of SBOX, achieved using standard XOR & AND gates, exhibits a significant relationship between the two. Although the incorporation of extra components in the design increases the cost, masking is a powerful countermeasure against spower attacks. To de-mask the input of SBOX after an algorithm-level masking operation, one must compute a nonlinear mask byte. Masking on a cellular level requires less machinery. In this case, the entire SBOX is meant to hide the XOR and AND gates.

Table 6.6 Evaluating the suggested designs' topologies

	Masked SBOX	**[181]**	**Unmasked SBOX**	**[181]**
Technology	CMOS 90nm	180nm CMOS	CMOS 90nm	180nm CMOS
Static Power	55.96 µW	51.151nW	8.729 µW	22.973nW
Dynamic Power	964.3 µW	16.2mW	216µW	4.187mW
Delay	1.567ns	9.44ns	0.005ns	7.02ns

Both the mask XOR and mask AND cells proposed here demonstrate that power is consumed uniformly across the internal terminal, rather than being concentrated on a single signal. Model output truth table demonstrating mask XOR-AND cell with hammering weight of 2. To reveal the output terminal, mask gates need an extra XOR gate, mask XOR cells need an AND/XNOR gate, and mask AND gates need an extra AND gate. Hardware requirements and characteristics for both unmasking and masking SBOX are listed in Table 6.6.

The number of cells is reduced by 41 relative to SBOX, which was presented in [181]. Although the static power consumption of an unmasked SBOX is somewhat high at 8.729 W, it is significantly reduced at 216 W compared to the dynamic power consumption of 4.187 mW, thanks to the design at CMOS 90nm. In contrast to the 2061 cells in [181], the number of cells needed for SBOX with a mask cell is just 635. The mask SBOX has a delay of 1.567 ns, however the static power is significantly boosted. The proposed SBOX value is compared to previous findings in Table 6.7. Due to masking, the total number of gates needed for an SBOX is reduced to 2173 from 5478 for a balanced pipeline and 3628 for multiplicative masking in the [206] gate count. Using Boolean masking as done in [207] reduces the number of gates needed to 2051 while increasing the delay. Increases in gate count and latency are proportionate when masking is performed just on a subset of a circuit, as in [207], where only AND cells are masked, and [206], where only the multiplicative unit is masked. Using the masking function, as indicated in table 6.7, SBOX can be unpredictable.

The power trace is only weakly associated with the processed data when the Pearson, correlation coefficient value, is low. To emphasise, during a CPA attack, the attacker cannot guess a sufficiently hidden secret. The supplied SBOX, when used in conjunction with Boolean masks at the cellular level, obtains a correlation coefficient of 0.0946 for HWPM and 0.1447 for HDPM, both of which are much lower than [74, 208-210]. HWPM has less data leaks than HDPM due to its smaller value. Figure 6.12 displays an increase in unpredictability from 42.13% for HWPM to 50.89% for HDPM when using SBOX with a Boolean mask at the cell, as compared to SBOX with a regular cell.

Table 6.7 Evaluation of the hammering weight/distance power model's correlation coefficient

	(ρ) HDPM	**(ρ) HWPM**
[208]		0.59
[209]	-0.43	
[74]		0.997
[210]	0.1838	
This Work SBOX with unmask cell	0.2843	0.2245
This Work SBOX with mask cell	0.1447	0.0946

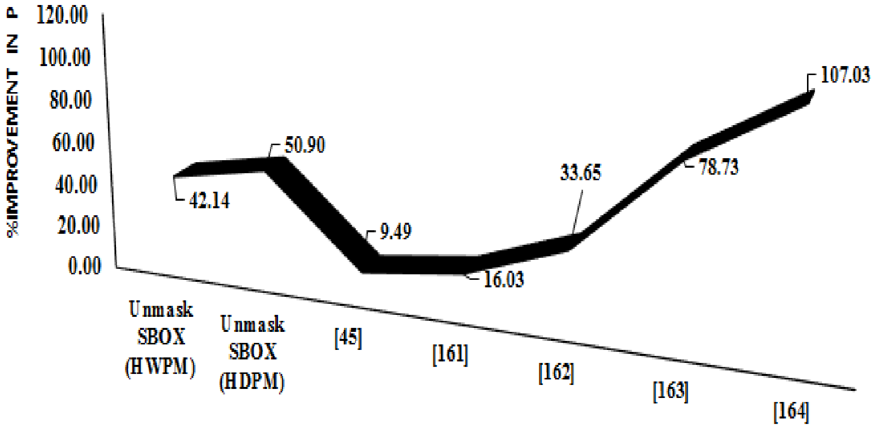


Figure 6.12 Improvement in ρ of proposed design with mask cell

Table 6.8 Parameters of the suggested design are compared to the fabrication process.

References	**Technology**	**Delay (ns)**	**Area (Gate Counts)**	**Masking Method**
[186]	CMOS 180nm		13000	Gate level masking of AND
[203]	CMOS 90nm	8.33	5478	Balanced Pipelining
[203]	CMOS 90nm	59.13	3628	Multiplicative Inverse
[181]	CMOS 180nm	14.299	2061	Boolean Masking
This Work	CMOS 90nm	1.567	2173	Gate level masking of XOR & AND

Table 6.8 compares the established SBOX implementation to the proposed masked cell's performance characteristics. In [186, 203, 181], the schematic of an SBOX with a masked XOR-AND gate is shown. This implementation has a smaller value of delay than the actual result. More gates are needed than in [181], but less than in [186, 203].

Chapter 7

Conclusion and Future Work

7.1 **Conclusion**

This study evaluates the cost and effectiveness of several countermeasures against power assaults on a secure 16-bit FFT implemented in 90nm CMOS technology. Both the DPA and the CPA were presented after the initial round of discussions had concluded. The suggested design analyses the power profile in relation to the data being processed, and the underlying cell is made up of static, TG, and PTL CMOS circuitry. Power attack analysis is another restriction on the information leakage and power consumption patterns of static CMOS circuitry. Although the number of gates is decreased using hybrid CMOS logic, the processing time is lengthened. NED and NSD are measures of the difficulty of an attack on a computer system. Energy consumption remains proportionate to input data, as shown by the design's NED 0.85 and NSD 0.0266 ratings for static CMOS logic. The suggested revision, which makes use of hybrid CMOS logic, brings the NED value down to 0.4391 and the NSD value down to 0.0208, suggesting reduced data leakage and fewer power traces.

Our research led us to build a new, better-correlating box that can withstand power attacks. The mask gate countermeasure increases gates and power measurements while introducing low-correlation data leaks and power traces. Input masked value, mask input bit, and power consumption pattern are all interconnected in a nonlinear fashion in the proposed masked AND and XOR cell. Compared to the unmasked cell correlation coefficient of 0.372 and 0.134, the mask AND-XOR cell correlation coefficient is just 0.3 and 0.0053. Increase of 8.49% in the correlation coefficient, which evaluates the unpredictability of independence of power consumption patterns from processed data, is based on the power consumption pattern of a mask cell. For authentication and key generation on a device, the PUF hardware module comes highly recommended. The PUF reaction has to be completely one of a kind. With PUF, the designer may choose which of their circuit's properties will be used in the computation. Two devices, one with a route delay and the other with hysteresis width, and the Schmitt trigger, a key unit for the design of PUF whose response is a function of input, are shown to exemplify the concepts presented in this study. Under normal working circumstances, the proposed ST PUF has a uniqueness of 44.71 percent.

7.2 Future Work

Any electrical property with a variable nature, from a constant to a steady value, can be used to make a PUF. By choosing different device-specific properties, PUF circuits may be reclassified. Finding novel electrical characteristics to develop a novel, robust PUF structure. The key objective is to streamline the PUF structure and reactivity in such a way that it cannot be predicted by a learning environment. The examination of CMOS circuitry to eliminate the reliance between primary output and secondary information would be an additional field of study towards the construction of an attack-resistant cryptography module. Forgo the need for further protections by building attack resistance into your IC from the start.

Bibliography

1. Pedro Garcia Lopez, Alberto Montresor, Dick Epema, Anwitaman Datta, Teruo Higashino, Adriana Iamnitchi, Marinho Barcellos, Pascal Felber, and Etienne Riv iere. Edge-centric computing: Vision and challenges. ACM SIGCOMM Computer Communication Review, 45(5):37–42, 2015.

2. Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. IEEE Internet of Things Journal, 3(5):637–646, 2016.

3. G. Moore, "Cramming more components onto integrated circuits," Electronics, vol. 8, pp. 114–117, Apr. 1965.

4. Horan, Amanda, Christopher Johnson, and Heather Sykes. "Foreign infringement of intellectual property rights: Implications for selected US industries." No. 1505-2016-130785. 2005.

5. T. Xu and M. Potkonjak, "The digital bidirectional function as a hardware security primitive: Architecture and applications," IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), Rome, Italy, 2015, pp. 335-340: IEEE.

6. Z. Cherif, J.-L. Danger, F. Lozac'h, Y. Mathieu, and L. Bossuet, "Evaluation of Delay PUFs on CMOS 65 nm Technology: ASIC vs FPGA," proceeding of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy, Tel-Aviv, Israel, 2013.

7. J. P. Garcia, "Electronic logic circuit with physically unclonable function characteristics," U.S. Patent No. 8,274,306. Washington, DC: U.S. Patent and Trademark Office, September 25, 2012.

8. R. Maes, Physically unclonable functions: Constructions, properties, and applications, 1st ed. Netherlands: Springer Science & Business Media, 2013, pp. XVII, 193.

9. R. Nithyanand and J. Solis, "A theoretical analysis: Physical unclonable functions and the software protection problem," IEEE Symposium on Security and Privacy Workshops, San Francisco, CA, USA, 2012, pp. 1-11: IEEE.

10. Nagra, Jasvir, and Christian Collberg. Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection: Obfuscation, Watermarking, and Tamperproofing for Software Protection. Pearson Education, 2009.

11. Zhu, William Feng. "Concepts and techniques in software watermarking and obfuscation." PhD diss., ResearchSpace@ Auckland, 2007.

12. Ghosh, Santosh, Debdeep Mukhopadhyay, and Dipanwita Roychowdhury. "Secure dual-core cryptoprocessor for pairings over barreto-naehrig curves on fpga platform." IEEE Transactions on Very Large Scale Integration (VLSI) Systems21, no. 3 (2013): 434-442.

13. Mazumdar, Bodhisatwa, Debdeep Mukhopadhyay, and Indranil Sengupta. "Construction of RSBFs with improved cryptographic properties to resist differential fault attack on grain family of stream ciphers." Cryptography and Communications 7, no. 1 (2015): 35-69.

14. Li, Li, and Hai Zhou. "Structural transformation for best-possible obfuscation of sequential circuits." In Hardware-Oriented Security and Trust (HOST), 2013 IEEE International Symposium on, pp. 55-60. IEEE, 2013.

15. Lao, Yingjie, and Keshab K. Parhi. "Protecting DSP circuits through obfuscation." In Circuits and Systems (ISCAS), 2014 IEEE International Symposium on, pp. 798-801. IEEE, 2014.

16. Suh, G. Edward, and Srinivas Devadas. "Physical unclonable functions for device authentication and secret key generation." In Design Automation Conference, 2007. DAC'07. 44th ACM/IEEE, pp. 9-14. IEEE, 2007.

17. Koteshwara, Sandhya, Chris H. Kim, and Keshab K. Parhi. "Key-Based Dynamic Functional Obfuscation of Integrated Circuits Using Sequentially Triggered Mode-Based Design." IEEE Transactions on Information Forensics and Security 13, no. 1 (2018): 79-93.

18. B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas, "Silicon physical random functions," proceeding of 9th ACM conference on Computer and communications security, Washington, DC, USA, 2002, pp. 148-160: ACM

19. G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," proceeding of 44th ACM/IEEE Design Automation Conference, San Diego, CA, USA, 2007, pp. 9-14: IEEE

20. H. Kang, Y. Hori, T. Katashita, A. Satoh, and K. Iwamura, "PUF Evaluation with post-processing and Modified Modeling Attack,"International Journal of Security Its Applications, vol. 7, no. 4, pp. 231-242, 2013.

21. Horan, Amanda, Christopher Johnson, and Heather Sykes. "Foreign infringement of intellectual property rights: Implications for selected US industries." No. 1505-2016-130785. 2005.

22. S. Zhang and W. Zhong, "A New Type of Countermeasure against DPA in Multi-Sbox of Block Cipher," Wireless Communications Mobile Computing, vol. 2018, no. Article ID 5945312, p. 11 Pages, 2018

23. D. Kamel, F.-X. Standaert, and D. Flandre, "Scaling trends of the AES S-box low power consumption in 130 and 65 nm CMOS technology nodes, "IEEE International Symposium on Circuits and Systems, Taipei, Taiwan, pp. 1385-1388, IEEE, 2009.

24. N. Mentens, L. Batina, B. Preneel, and I. Verbauwhede, "A systematic evaluation of compact hardware implementations for the Rijndael S-box, "Cryptographers' Track at the RSA Conference, San Francisco, CA, USA, vol. 3376, pp. 323-333, Springer, 2005.

25. M. Mozaffari-Kermani and A. Reyhani-Masoleh, "Efficient and high-performance parallel hardware architectures for the AES -GCM," IEEE Transactions on Computers, vol. 61, no. 8, pp. 1165-1178, 2011.

26. S. Yang, W. Wolf, N. Vijaykrishnan, D. N. Serpanos, and Y. Xie, "Power attack resistant cryptosystem design: A dynamic voltage and frequency switching approach," proceeding of Design, Automation, and Test in Europe, Munich, Germany, pp. 64-69, IEEE, 2005.

27. A. Maiti and P. Schaumont, "Improved ring oscillator PUF: An FPGA-friendly secure primitive," Journal of cryptology, vol. 24, no. 2, pp. 375-397, 2011.

28. S. Eiroa, M. I. Baturone Castillo, A. J. Acosta Jiménez, and J. Dávila, "Using physical unclonable functions for hardware authentication: A survey," proceeding of XXV Conference on Design of Circuits and Integrated Systems (DCIS), 2010.

29. A. Kumar and R. S. Mishra, "Challenge-Response Pair (CRP) Generator Using Schmitt Trigger Physical Unclonable Function," proceeding of Advanced Computing and Communication Technologies, Panipat, India vol. 702, pp. 213-223, Springer, 2019.

30. S. S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, "The butterfly PUF protecting IP on every FPGA," IEEE International Workshop on Hardware-Oriented Security and Trust, Anaheim, CA, USA, pp. 67-70, IEEE, 2008.

31. S. Katzenbeisser, Ü. Kocabaş, V. Rožić, A.-R. Sadeghi, I. Verbauwhede, and C. Wachsmann, "PUFs: Myth, fact or busted? A security evaluation of physically unclonable functions (PUFs) cast in silicon," International Workshop on Cryptographic Hardware and Embedded Systems, Leuven, Belgium, vol. 7428, pp. 283-301, Springer, 2012.

32. A. Kumar, J. Pathak, and S. L. Tripathi, "Frequency-Based RO-PUF," in AI Techniques for Reliability Prediction for Electronic Components: IGI Global, pp. 252-261, 2020.

33. M. Majzoobi, M. Rostami, F. Koushanfar, D. S. Wallach, and S. Devadas, "Slender PUF protocol: A lightweight, robust, and secure authentication by substring matching, "IEEE Symposium on Security and Privacy Workshops, San Francisco, CA, USA, pp. 33-44, IEEE, 2012.

34. H. Handschuh, G.-J. Schrijen, and P. Tuyls, "Hardware intrinsic security from physically unclonable functions," Towards Hardware-Intrinsic Security, pp. 39-53 Springer, 2010.

35. B. Barak, R. Shaltiel, and E. Tromer, "True random number generators secure in a changing environment," International Workshop on Cryptographic Hardware and Embedded Systems, Cologne, Germany, 2003, vol. 2779, pp. 166-180: Springer, 2003.

36. M. Bhargava, C. Cakir, and K. Mai, "Attack resistant sense amplifier based PUFs (SA-PUF) with deterministic and controllable reliability of PUF responses," IEEE international symposium on hardware-oriented security and trust (HOST), Anaheim, CA, USA, pp. 106-111, IEEE, 2010.

37. NXP Semiconductors N.V, "PUF-physical unclonable functions-protecting next-generation smart card ics with sram based pufs," http: //www.nxp.com/documents/other/75017366.pdf, Feb. 2013.

38. F. S. Hossain and M. L. Ali, "A Novel Byte-Substitution Architecture for the AES Cryptosystem," PloS one, vol. 10, no. 10, 2015.

39. A. Maiti, R. Nagesh, A. Reddy, and P. Schaumont, "Physical unclonable function and true random number generator: a compact and scalable implementation," 19th ACM Great Lakes Symposium on VLSI, Boston Area, MA, USA, pp. 425-428: ACM, 2009.

40. Kong, Joonho, and Farinaz Koushanfar. "Processor-based strong physical unclonable functions with aging-based response tuning." IEEE Transactions on Emerging Topics in Computing, vol. 2, no. 1,pp. 16-29, 2013.

41. Maiti, Abhranil, Vikash Gunreddy, and Patrick Schaumont. "A systematic method to evaluate and compare the performance of physical unclonable functions." In Embedded systems design with FPGAs, pp. 245-267. Springer, New York, NY, 2013

42. S. Zeitouni, Y. Oren, C. Wachsmann, P. Koeberl, and A.-R. Sadeghi, "Remanence decay side-channel: The PUF case," J. IEEE Transactions on Information Forensics Security vol. 11, no. 6, pp. 1106-1116, 2015.

43. C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," Proceedings of the IEEE, vol. 102, no. 8, pp. 1126-1141, 2014.

44. S. Meguerdichian and M. Potkonjak, "Device aging-based physically unclonable functions," 48th ACM/EDAC/IEEE Design Automation Conference (DAC), New York, NY, USA, pp. 288-289, IEEE, 2011.

45. Z. C. Jouini, J.-L. Danger, and L. Bossuet, "Characterization of Physically Unclonable Functions at Design Stage, "Colloque du GDR SoC-SiP, Paris, France< hal-00753222>, C. Universite de Lyon, Saint-Etienne, France, Ed., ed. France, 2012.

46. R. Pegu and R. Mudoi, "Design and analysis of Mux based physical unclonable functions," IEEE Trans. Comput.-Aided Design Integr. Circ. Syst, vol. 33, no. 5, pp. 649-662, 2015.

47. E. Ozturk, G. Hammouri, and B. Sunar, "Physical unclonable function with tristate buffers, "IEEE International Symposium on Circuits and Systems, Seattle, WA, USA, pp. 3194-3197, IEEE, 2008.

48. S. U. Hussain, S. Yellapantula, M. Majzoobi, and F. Koushanfar, "BIST-PUF: Online, hardware-based evaluation of physically unclonable circuit identifiers," proceeding of IEEE/ACM International Conference on Computer-Aided Design (ICCAD), San Jose, CA, USA, pp. 162-169, IEEE, 2014.

49. Y. Yao, M. Kim, J. Li, I. L. Markov, and F. Koushanfar, "ClockPUF: Physical Unclonable Functions based on clock networks," proceeding of Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, pp. 422-427, IEEE, 2013.

50. Rajendran, J., Pino, Y., Sinanoglu, O., & Karri, R. "Security analysis of logic obfuscation". In Proceedings of the 49th Annual Design Automation Conference, 83-89, 2012

51. Zhang, J. "A practical logic obfuscation technique for hardware security". IEEE Transactions on very large scale integration (VLSI) systems, *24*(3), 1193-1197, 2015

52. Yasin M, Sinanoglu O, "Evaluation of logic locking", IFIP/IEEE International Conference on Very Large Scale Integration, 1-6, 2017

53. Kamali, H. M., Azar, K. Z., Gaj, K., Homayoun, H., & Sasan, A. "Lut-lock: A novel lut-based logic obfuscation for fpga-bitstream and asic-hardware protection", IEEE Computer Society Annual Symposium on VLSI, 405-410, 2018

54. Patnaik, S., Rangarajan, N., Knechtel, J., Sinanoglu, O., & Rakheja, S., "Advancing hardware security using polymorphic and stochastic spin-hall effect devices", IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE), 97-102, 2018.

55. Sengupta, A., & Mohanty, S. P, "Functional obfuscation of DSP cores using robust logic locking and encryption", IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 709-710, 2018.

56. Kolhe, G., PD, S. M., Rafatirad, S., Mahmoodi, H., Sasan, A., & Homayoun, H., "On custom lut-based obfuscation", Great Lakes Symposium on VLSI, 477-482, 2019.

57. Kumar, A. P., Bharathi, S., Meghana, C., Anusha, K., & Priyatharishini, M., "Toggle count based logic obfuscation", IEEE International conference on Electronics, Communication and Aerospace Technology ICECA, 809-814, 2019.

58. Rajendran, J., Pino, Y., Sinanoglu, O., & Karri, R, "Security analysis of logic obfuscation", In Proceedings of the 49th Annual Design Automation Conference, 83-89, 2012.

59. M.M.Shihab et.al., “Design Obfuscation through selective Post-Fabrication Transistor Level Programming”, Design, Automation and Test in Europe Conference and Exhibition (DATE), 528-533, 2019.

60. Batabyal, S., & Rai, A. B, “Design of A Ring Oscillator Based PUF with Enhanced Challenge Response pair and Improved Reliability”, IEEE International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), 1370-1374,2019.

61. Rezaei, A., Shen, Y., & Zhou, H, “Rescuing logic encryption in post-sat era by locking & obfuscation”, IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE),13-18, 2020.

62. Park, B., & Maghari, N., “Logic Obfuscation Through Enhanced Threshold Voltage Defined Logic Family”, *IEEE Transactions on Circuits and Systems II: Express Briefs*, *67*(12), 3407-3411, 2020.

63. Patil, V. C., & Kundu, S., “On leveraging multi-threshold FinFETs for design obfuscation”, *IEEE Computer Society Annual Symposium on VLSI (ISVLSI),*108-113,2020.

64. Cheng, C., & Parhi, K. K, “Hardware efficient fast DCT based on novel cyclic convolution structures”, *IEEE Transactions on signal processing*, *54*(11), 4419-4434, 2006

65. Cheng, C., & Parhi, K. K., “A novel systolic array structure for DCT”; IEEE Transactions on Circuits and Systems II: Express Briefs, 52(7), 366-369, 2005.

66. Chung, R. L., Chen, C. W., Chen, C. A., Abu, P. A. R., & Chen, S. L., “ VLSI Implementation of a Cost-Efficient Loeffler DCT Algorithm with Recursive CORDIC for DCT-Based Encoder”, *Electronics*, *10*(7), 862, 2021.

67. M Perera, S., Madanayake, A., Dornback, N., & Udayanga, N., “Design and digital implementation of fast and recursive DCT II–IV algorithms”, *Circuits, Systems, and Signal Processing*, *38*(2), 529-555, 2019.

68. Banerjee, N., Karakonstantis, G., & Roy, K., “Process variation tolerant low power DCT architecture”, IEEE Design, Automation & Test in Europe Conference & Exhibition, 1-6, 2007.

69. Chiper, D. F., Cotorobai, L. T., “A New Approach for a Unified Architecture for Type IV DCT/DST with an Efficient Incorporation of Obfuscation Technique”, *Electronics*, *10*(14), 1656, 2021.

70. Swaminathan, G., Murugesan, G., Sasikala, S., & Murali, L., “A novel implementation of combined systolic and folded architectures for adaptive filters in FPGA”, *Microprocessors and Microsystems*, *74*, 103018, 2020.

71. H. Mestiri, N. Benhadjyoussef, M. Machhout, and R. Tourki, "A comparative study of power consumption models for CPA attack," International Journal of Computer Network and Information Security, vol. 5, no. 3, p. 25, 2013.

72. T. Fujino, "Tamper-resistant memory integrated circuit and encryption circuit using same," U.S. Patent 8,861,720, Washington, DC: U.S. Patent and Trademark Office. October 14, 2014.

73. M. Alioto, L. Giancane, G. Scotti, and A. Trifiletti, "Leakage power analysis attacks: A novel class of attacks to nanometer cryptographic circuits," IEEE Transactions on Circuits Systems I: Regular Papers, vol. 57, no. 2, pp. 355-367, 2009.

74. J. Liu, D. Gu, and Z. Guo, "Correlation power analysis against stream cipher mickey v2, "International Conference on Computational Intelligence and Security, Nanning, China, pp. 320-324, IEEE, 2010.

75. E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model, "International workshop on cryptographic hardware and embedded systems, Cambridge, MA, USA, vol. 3156, pp. 16-29, Springer, 2004.

76. A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A compact Rijndael hardware architecture with S-box optimization," proceeding of International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, QLD, Australia, 2001, vol. 2248, pp. 239-254, Berlin, Heidelberg: Springer, 2001.

77. C. Canovas and J. Clédière, "What do S-boxes say in differential side-channel attacks?," IACR Cryptology ePrint Archive, p 311, 2005.

78. D. Bellizia, S. Bongiovanni, P. Monsurro, G. Scotti, and A. Trifiletti, "Univariate power analysis attacks exploiting static dissipation of nanometer CMOS VLSI circuits for cryptographic applications," IEEE Transactions on Emerging Topics in Computing, vol. 5, no. 3, pp. 329-339, 2016.

79. Y.-J. Baek and M.-J. Noh, "Differential power attack and masking method," Trends in Mathematics, vol. 8, no. 1, pp. 53-67, 2005.

80. O. Lo, W. J. Buchanan, and D. Carson, "Power analysis attacks on the AES -128 S-box using differential power analysis (DPA) and correlation power analysis (CPA)," Journal of Cyber Security Technology, vol. 1, no. 2, pp. 88-107, 2017.

81. P. Saravanan and P. Kalpana, "An energy-efficient XOR gate implementation resistant to power analysis attacks," J. Eng. Sci. Technol, vol. 10, pp. 1275-1292, 2015.

82. L. Zhang, L. Vega, and M. Taylor, "Power side channels in security ICs: hardware countermeasures," arXiv preprint arXiv: 00681," University of California, San Diego2016.

83. S. Mangard, T. Popp, and B. M. Gammel, "Side-Channel Leakage of Masked CMOS Gates," Cryptographers' Track at the RSA Conference, San Francisco, CA, USA, vol. 3376, pp. 351-365: Springer Berlin Heidelberg, 2004

84. M. Jayakumar, FPGA-Masked S-Box Implementation for AES Engine (VLSI Design: Circuits, Systems, and Applications, no. 469). pp. 223-233, Springer,2018.

85. K. Tiri et al., "AES -based cryptographic and biometric security coprocessor IC in 0.18-/spl mu/m CMOS resistant to side-channel power analysis attacks," proceeding of 42nd Design Automation Conference, 2005., Anaheim, CA, USA, pp. 216-219, IEEE, 2005.

86. M. Masoumi, "A highly efficient and secure hardware implementation of the advanced encryption standard," Journal of Information Security Applications, vol. 48, p. 102371, 2019.

87. E. Tena-Sanchez, J. Castro, and A. J. Acosta, "A methodology for optimized design of secure differential logic gates for DPA resistant circuits," IEEE Journal on Emerging Selected Topics in Circuits Systems, vol. 4, no. 2, pp. 203-215, 2014.

88. T. Popp and S. Mangard, "Masked dual-rail pre-charge logic: DPA-resistance without routing constraints, "International Workshop on Cryptographic Hardware and Embedded Systems, Edinburgh, United Kingdom, vol. 3659, pp. 172-186, Springer, 2005.

89. Y.-h. ZENG, X.-c. ZOU, Z.-l. LIU, and J.-m. LEI, "Low-power clock-less hardware implementation of the Rijndael S-box for wireless sensor networks," The Journal of China Universities of Posts and Telecommunications, vol. 14, no. 4, pp. 104-109, 2007.

90. S. Zhang and W. Zhong, "A New Type of Countermeasure against DPA in Multi-Sbox of Block Cipher," Wireless Communications Mobile Computing, vol. 2018, no. Article ID 5945312, p. 11 Pages, 2018.

91. S. N. Dhanuskodi, S. Keshavarz, and D. Holcomb, "LLPA: logic state based leakage power analysis, "IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Pittsburgh, PA, USA, pp. 218-223, IEEE, 2016.

92. C. Teegarden, M. Bhargava, and K. Mai, "Side-channel attack resistant ROM-based AES S-Box, "IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), Anaheim, CA, USA, pp. 124-129, IEEE, 2010.

93. B. Halak, J. Murphy, and A. Yakovlev, "Power balanced circuits for leakage-power-attacks resilient design, "Science and Information Conference (SAI), London, UK, pp. 1178-1183, IEEE, 2015.

94. C. Monteiro, Y. Takahashi, and T. Sekine, "Resistance against power analysis attacks on adiabatic dynamic and adiabatic differential logics for smart card," International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS), Chiang Mai, Thailand, pp. 1-5, IEEE, 2015.

95. K. Tiri and I. Verbauwhede, "Securing encryption algorithms against DPA at the logic level: Next-generation smart card technology, "International Workshop on Cryptographic Hardware and Embedded Systems, Cologne, Germany, 2003, vol. 2779, pp. 125-136: Springer, 2003.

96. S. Mangard, E. Oswald, and F.-X. Standaert, "One for all–all for one: unifying standard differential power analysis attacks," IET Information Security, vol. 5, no. 2, pp. 100-110, 2011.

97. J. J. Fournier, S. Moore, H. Li, R. Mullins, and G. Taylor, "Security evaluation of asynchronous circuits, "International Workshop on Cryptographic Hardware and Embedded Systems, Cologne, Germany, 2003, vol. 2779, pp. 137-151: Springer, 2003.

98. N. Mentens, L. Batina, B. Preneel, and I. Verbauwhede, "A systematic evaluation of compact hardware implementations for the Rijndael S-box, "Cryptographers’ Track at the RSA Conference, San Francisco, CA, USA, vol. 3376, pp. 323-333, Springer, 2005.

99. F. A. Khan, J. Ahmed, J. S. Khan, J. Ahmad, M. A. Khan, and S. O. Hwang, "A new technique for designing 8× 8 substitution box for image encryption applications, "9th Computer Science and Electronic Engineering (CEEC), Colchester, UK, pp. 7-12, IEEE, 2017.

100. D. Kamel, F.-X. Standaert, and D. Flandre, "Scaling trends of the AES S-box low power consumption in 130 and 65 nm CMOS technology nodes, "IEEE International Symposium on Circuits and Systems, Taipei, Taiwan, pp. 1385-1388, IEEE, 2009.

101. E. M. Mahmoud, A. Abd, T. A. E. El Hafez, and T. A. El Hafez, "Dynamic AES -128 with key-dependent S-box," International Journal of Engineering Research and Applications, vol. 3, no. 1, pp. 1662-1670, 2013.

102. A. Pammu, K.-S. Chong, and B.-H. Gwee, "Secured low power overhead compensator look-up-table (LUT) substitution box (S-Box) architecture," proceeding of IEEE International Conference on Networking, Architecture and Storage (NAS), Long Beach, CA, USA, pp. 1-7, IEEE, 2016.

103. S. Dey and R. Ghosh, "A review of existing 4-bit crypto S-box cryptanalysis techniques and two new techniques with 4-bit Boolean functions for cryptanalysis of 4-bit crypto S-boxes," Advances in Pure Mathematics, vol. 8, no. 03, p. 272, 2018.

104. K. Tanimura and N. Dutt, "A standard cell-based DPA attack countermeasure using homogeneous dual-rail logic (HDRL)," J University of California, pp. 12-01, 2012.

105. Lao, Yingjie, and Keshab K. Parhi. "Statistical analysis of MUX-based physical unclonable functions." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 33, no. 5, pp. 649-662, 2014.

106. S. Baur and H. Boche, "Robust secure storage of data sources with perfect secrecy," IEEE Workshop on Information Forensics and Security (WIFS), Rennes, France, pp. 1-6, IEEE, 2017.

107. A. Maiti, I. Kim, and P. Schaumont, "A robust physical unclonable function with enhanced challenge-response set," IEEE Transactions on Information Forensics Security, vol. 7, no. 1, pp. 333-345, 2011.

108. A. Mutlu, K. J. Le, M. Celik, D.-s. Tsien, G. Shyu, and L.-C. Yeh, "An exploratory study on statistical timing analysis and parametric yield optimization, "8th International Symposium on Quality Electronic Design (ISQED'07), San Jose, CA, USA, pp. 677-684, IEEE, 2007.

109. S. Devadas, E. Suh, S. Paral, R. Sowell, T. Ziola, and V. Khandelwal, "Design and implementation of PUF-based "unclonable" RFID ICs for anti-counterfeiting and security

applications," proceeding of IEEE international conference on RFID, Las Vegas, NV, USA, pp. 58-64, IEEE, 2008.

110. Y. Wen and Y. Lao, "Efficient fuzzy extractor implementations for PUF based authentication," proceeding of the 12th International Conference on Malicious and Unwanted Software (MALWARE), Fajardo, Puerto Rico, 2017.

111. M. Mozaffari-Kermani and A. Reyhani-Masoleh, "Efficient and high-performance parallel hardware architectures for the AES -GCM," IEEE Transactions on Computers, vol. 61, no. 8, pp. 1165-1178, 2011.

112. Z. Paral and S. Devadas, "Reliable and efficient PUF-based key generation using pattern matching, "IEEE International Symposium on Hardware-Oriented Security and Trust, San Diego CA, USA, pp. 128-133, IEEE, 2011.

113. A. Kumar, R. S. Mishra, and K. Kashwan, "PUF Based challenge Response Pair For Secured Authentication," International Journal of Control Theory and Applications, vol. 9, no. 41, pp. 115-121, 2017.

114. M. Rostami, F. Koushanfar, and R. Karri, "A primer on hardware security: Models, methods, and metrics," Proceedings of the IEEE, vol. 102, no. 8, pp. 1283-1295, 2014.

115. A. Kumar, R. S. Mishra, and K. Kashwan, "Challenge-response generation using RO-PUF with reduced hardware," proceeding of International Conference on Advances in Computing, Communications and Informatics (ICACCI), Jaipur, India, pp. 1305-1308, IEEE, 2016.

116. G. T. Becker and Systems, "On the pitfalls of using arbiter-PUFs as building blocks," IEEE Transactions on Computer-Aided Design of Integrated Circuits, vol. 34, no. 8, pp. 1295-1307, 2015.

117. X. Zhang and K. K. Parhi, "High-speed VLSI architectures for the AES algorithm," IEEE transactions on very large scale integration systems, vol. 12, no. 9, pp. 957-967, 2004.

118. R. Kumar, V. C. Patil, and S. Kundu, "Design of unique and reliable physically unclonable functions based on current starved inverter chain," IEEE Computer Society Annual Symposium on VLSI, Chennai, India, pp. 224-229, IEEE, 2011.

119. Z. Cherif, J.-L. Danger, S. Guilley, and L. Bossuet, "An easy-to-design PUF based on a single oscillator: the loop PUF," proceeding of 15th Euromicro Conference on Digital System Design, Izmir, Turkey, pp. 156-162, IEEE, 2012.

120. H. Bin, S. Goto, and Y. Tsunoo, "A multiple bits output ring-oscillator physical unclonable function," proceeding of International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS), Chiang Mai, Thailand, pp. 1-5: IEEE, 2011.

121. J.-L. Zhang, G. Qu, Y.-Q. Lv, and Q. Zhou, "A survey on silicon PUFs and recent advances in ring oscillator PUFs," Journal of computer science technology, vol. 29, no. 4, pp. 664-678, 2014.

122. Y. Cao, L. Zhang, C.-H. Chang, and S. Chen, "A low-power hybrid RO PUF with improved thermal stability for lightweight applications," IEEE Transactions on computer-aided design of integrated circuits systems, vol. 34, no. 7, pp. 1143-1147, 2015.

123. M. Platonov, "SRAM-Based Physical Unclonable Function on an Atmel ATmega Microcontroller," Doctoral dissertation, Master’s Thesis, Czech Technical University of Prague, Prague, 2013.

124. V. Rožić, W. Dehaene, and I. Verbauwhede, "Design solutions for securing SRAM cell against power analysis," proceeding of IEEE International Symposium on Hardware-Oriented Security and Trust, San Francisco, CA, USA, pp. 122-127, IEEE, 2012.

125. Y. Shifman, A. Miller, O. Keren, Y. Weizmann, and J. Shor, "A Method to Improve Reliability in a 65-nm SRAM PUF Array," IEEE Solid-State Circuits Letters, vol. 1, no. 6, pp. 138-141, 2018.

126. X. Xu, A. Rahmati, D. E. Holcomb, K. Fu, and W. Burleson, "Reliable physical unclonable functions using data retention voltage of SRAM cells," IEEE Transactions on Computer-Aided Design of Integrated Circuits Systems, vol. 34, no. 6, pp. 903-914, 2015.

127. F. Arith, M. I. Idris, M. N. S. Zainudin, M. Chachuli, and S. Amaniah, "Low voltage CMOS Schmitt trigger in 0.18 μm technology," IOSR Journal of Engineering, vol. 3, no. 2, pp. 8-15, 2013.

128. L. A. P. Melek, A. L. da Silva, M. C. Schneider, and C. Galup-Montoro, "Analysis and design of the classical CMOS Schmitt trigger in subthreshold operation," IEEE Transactions on Circuits Systems I: Regular Papers, vol. 64, no. 4, pp. 869-878, 2016.

129. T. Aoki, Y. Kurokawa, and M. Kozuma, "Semiconductor device having Schmitt trigger NAND circuit and Schmitt trigger inverter," U.S. Patent No. 9,245,589. Washington, DC: U.S. Patent and Trademark Office, January 26, 2016.

130. S. B. Ramakrishna, S. Madhusudhan, B. Nikshep, B. Naveen, and H. Teerthaprasad, "Power and delay optimization of domino Schmitt trigger configurations with enhanced hysteresis voltage," Analog Integrated Circuits Signal Processing, vol. 102, no. 1, pp. 1-9, 2019.

131. J. Park, "Secure hardware design against side-channel attacks," Doctoral thesis, computer Engineering, Iowa state university, Ames, Iowa, 2016.

132. J. B. Kuang, "SOI CMOS Schmitt trigger circuits with controllable hysteresis," U.S. Patent No. 6,441,663. Washington, DC: U.S. Patent and Trademark Office, August 27, 2002.

133. D. S. Barlow, "Self-adjusting Schmitt trigger," U.S. Patent No. 7,167,032. Washington, DC: U.S. Patent and Trademark Office, January 23, 2007.

134. W. Lew and R. Cadotte Jr, "Gate pulsing gate ladder," U.S. Patent No. 9,577,628. Washington, DC: U.S. Patent and Trademark Office, February 21, 2017.

135. V. Chauhan and P. Garg, "Compensated Schmitt trigger circuit for providing monotonic hysteresis response," U.S. Patent Application No. 11/148,947, January 26, 2006.

136. A. Kumar, S. L. Tripathi, and R. S. Mishra, "METAPUF: A challenge response pair generator," Periodicals of Engineering Natural Sciences, vol. 6, no. 2, pp. 58-63, 2018.

137. K. Yelamarthi, "Timing-driven variation-aware partitioning and optimization of mixed static-dynamic CMOS circuits," Circuits and Systems, vol. 4, no. 2, p. s7 pages, Art. no. 29874, 2013.

138. J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection, "International workshop on cryptographic hardware and embedded systems, Vienna, Austria, vol. 4727, pp. 63-80, Springer, 2007.

139. Kavi K.M, Buckles B.P, Bhat U N, “A formal definition of data flow graph models”, Computers, IEEE Transactions on 100(11), 940-948, 1986.

140. Parhi KK, Messerschmitt D, “Look ahead computation: Improving iteration bound in linear recursions. In: Acoustics”, Speech and Signal Processing, IEEE ICASSP’87, 12, 1855-1858, 1987.

141. Parhi KK, Messerschmitt DG, “Static rate- optimal scheduling of iterative data –flow programs via optimum unfolding”, Computers, IEEE transaction on 40(2),178-195, 1991.

142. Ali Shatnawi, “Computing the loop bound in iterative Data Flow Graphs using natural token flow”, World Academy of Science, Engineering and Technology, 908-913, 2007.

143. S Madhavan, G Yamuna, “DWT based grey scale image watermarking using area of best fir equation and cuckoo search algorithm”, International Journal of Computational Science and Engineering, 15(3/4), 236-247, 2017.

144. Ito K, Parhi KK, “Determining the minimum iteration period of an algorithm. Journal of VLSI signal processing systems for signal”, image and video technology, 11(3), 229-244, 1995.

145. X.Ni, H.Zhang, D.Wang, J. Luo, “Implementation of dynamic reconfigurable interpolator for open architecture CNC by using FPGA”, International Journal of Embedded System, 9(1), 45-53, 2017.

146. N U Bhanu, A Chilambuchelvan, “VLSI architecture for high speed and low power implementation of 5/3 lifting discreet wavelet transform”, International Journal of Computational Science and Engineering, 12(2/3), 254-263, 2016.

147. Girard P, “Survey of low- power testing of VLSI circuits” Design and Test of Computers, IEEE 19(3),80-90, 2002.

148. Z.Tang, Z.Lin, B Li, L Chen, "The embedded real time detection system of moving object based on improved Gaussian mixture model", International Journal of Embedded System, 8(2/3), 119-124, 2016.

149. Gerez, S.H, Heemstra de Groot, S.M, Herrmann, O.E, "A polynomial-time algorithm for computation of the iteration- period bound in recursive data flow graph" IEEE Transaction on Circuit and System I: Fundamental Theory and Applications, 39(1),49-52, 1992.

150. Parhi KK, "VLSI Digital Signal processing system/; design and implementation", John Wiley & Sons, 2007.

151. A A Varghese, C Pradeep, "FPGA implementation of area efficient single precision floating point complex divider with fault detection", International Journal of Computational System Engineering, 2(3), 177-181, 2016.

152. Masoud Rostami, Farinaz Koushanfar, Jeyavijayan Rajendran, and Ramesh Karri, "Hardware security: Threat models and metrics", In Proceedings of the International Conference on Computer-Aided Design, pages 819-823, 2013.

153. Maciej Brzozowski and Vyacheslav N Yarmolik, "Obfuscation as intellectual rights protection in vhdl language", In Proceedings of the 6th International Conference on Computer Information Systems and Industrial Management Applications (CISIM), pages 337-340, 2007.

154. Rajat Subhra Chakraborty and Swarup Bhunia, "RTL hardware IP protection using key-based control and data ow obfuscation", In Proceedings of the 23rd International Conference on VLSI Design, pages 405-410, 2010.

155. Jarrod A Roy, Farinaz Koushanfar, and Igor L Markov, " EPIC: Ending piracy of integrated circuits", In Proceedings of the conference on Design, automation and test in Europe, pages 1069-1074. ACM, 2008.

156. Rajat Subhra Chakraborty and Swarup Bhunia, “HARPOON: an obfuscation based SoC design methodology for hardware protection”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 28(10), pp 1493-1502, 2009.

157. Yingjie Lao and Keshab K Parhi, “Obfuscating DSP circuits via high-level transformations”, IEEE Transactions on VLSI Systems, pages 819-830, May 2015.

158. Goutham NC Shanmugam, Yingjie Lao, and Keshab K Parhi, “An obfuscated radix-2 real FFT architecture”, In 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1056-1060, IEEE, 2015.

159. Masoud Rostami, Farinaz Koushanfar, and Ramesh Karri, “A primer on hardware security: Models, methods, and metrics”, Proceedings of the IEEE, 102(8), page 1283-1295, 2014.

160. John Villasenor and Mohammed Tehranipoor, “Chop shop electronics”, IEEE Spectrum, 50(10), page 41-45, 2013.

161. Keshab K Parhi, “Verifying equivalence of digital signal processing circuits”, In Signals, Systems and Computers (ASILOMAR)”, IEEE Conference on Record of the Forty Sixth Asilomar Conference, pages 99-103, 2012.

162. Rajat Subhra Chakraborty and Swarup Bhunia, “Hardware protection and authentication through netlist level obfuscation”, In Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design, pages 674-677, IEEE Press, 2008.

163. Jeyavijayan Rajendran, Youngok Pino, Ozgur Sinanoglu, and Ramesh Karri, “Logic encryption: A fault analysis perspective”, In Proceedings of the Conference on Design, Automation and Test in Europe, pages 953-958, EDA Consortium, 2012.

164. Jiliang Zhang, “A practical logic obfuscation technique for hardware security”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 24(3), page 1193-1197, 2016.

165. Sophie Dupuis, Papa-Sidi Ba, Giorgio Di Natale, Marie-Lise Flottes, and Bruno Rouzeyre, "A novel hardware logic encryption technique for thwarting illegal overproduction and Hardware Trojans", IEEE 20th International On-Line Testing Symposium (IOLTS), pages 49-54, 2014.

166. Reyzin L, Srivastava N, "On the longest path algorithm for reconstructing trees from distance matrices", Information Processing Letters, 101(3), pages 98-100, 2007.

167.Gomes, J.Pereria, P.Garcia, F.Salgado,V.Silva, S.Pinto,M.Ekpanyapong,A.Tavares, "Hybrid real time operating system: Development of critical FreeRTOS features on FPGA", International Journal of Embedded System, 8(5), pages 483-492, 2016.

168. S P Raja, A Suruliandi, "Secure image compression using AES in bandlet domain", International Journal of Computational Science and Engineering, 17(2), pages 208-219, 2018.

169. R Saranya, C Pradeep, R Radhakrishnan, "Design and implementation of a reconfigurable finite impulse response filter for adaptive system", International Journal of Computational System Engineering, 3(1/2), pages 82-90, 2017.

170. Keshab K Parhi, "VLSI digital signal processing systems: design and implementation", Wiley, New York, 1999

171. Manohar Ayinala, Michael Brown, and Keshab K Parhi, "Pipelined parallel FFT architectures via folding transformation", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 20(6), pages 1068-1081, 2012.

172. Swarup Bhunia, Michael S Hsiao, Mainak Banga, and Seetharam Narasimhan, "Hardware Trojan attacks: threat analysis and countermeasures", Proceedings of the IEEE, 102(8), page 1229-1247, 2014.

173. Rajat Subhra Chakraborty, Seetharam Narasimhan, and Swarup Bhunia, “Hardware Trojan: Threats and emerging solutions”,IEEE International High Level Design Validation and Test Workshop, pages 166-171, 2009.

174. Jeyavijayan Rajendran, Youngok Pino, Ozgur Sinanoglu, and Ramesh Karri, “Security analysis of logic obfuscation”, In Proceedings of the 49th Annual Design Automation Conference, pages 83-89, 2012.

175. A. J. L. Martin, "Cadence design environment, Tutorial paper," New Mexico State University 2002.

176. A. Singh, P. Agarwal, and M. Chand, "Analysis of Development of Dynamic S-Box Generation," Comput. Sci. Inf. Technol, vol. 5, no. 5, pp. 154-163, 2017.

177. Mangard, Stefan, Thomas Popp, and Berndt M. Gammel. “Side-Channel Leakage of Masked CMOS Gates.” proceedings of Topics in Cryptology CT-RSA, Lecture Notes in Computer Science, vol 3376, Berlin, Heidelberg:springer, 2005.

178. V. Konstantakos, K. Kosmatopoulos, S. Nikolaidis, and T. Laopoulos, "Measurement of power consumption in digital systems," IEEE Transactions on Instrumentation measurement, vol. 55, no. 5, pp. 1662-1670, 2006.

179. T. Lopez and R. Elferich, "Measurement technique for the static output characterization of high-current power MOSFETs," IEEE Transactions on Instrumentation Measurement, vol. 56, no. 4, pp. 1347-1354, 2007.

180. P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," proceeding of Annual International Cryptology Conference, Santa Barbara, CA, USA, vol. 1666, pp. 388-397, Springer, Berlin, Heidelberg: Springer, 1999.

181. J. Zeng and C. Z. Xu, "An improved masked S-box for AES and hardware implementation," Journal of Convergence Information Technology, vol. 7, no. 10, pp. 338-344, 2012.

182. Standard, N. F. "Announcing the advanced encryption standard (AES)." Federal Information Processing Standards Publication, 197(1-51), pp 3-3, 2001

183. F.-X. Standaert, “Introduction to side-channel attacks, Secure Integrated Circuits, and Systems (Integrated Circuits and Systems)”. US: Springer, 2009.

184. M. Alioto, S. Bongiovanni, M. Djukanovic, G. Scotti, and A. Trifiletti, "Effectiveness of leakage power analysis attacks on DPA-resistant logic styles under process variations," IEEE Transactions on Circuits Systems I: Regular Papers, vol. 61, no. 2, pp. 429-442, 2013

185. K. Kumar, D. Mukhopadhyay, and D. Roy Chowdhury, "Design of a differential power analysis resistant masked AES S-Box," proceeding of International Conference on Cryptology in India, Chennai, India, vol. 4859, pp. 373-383, Springer, 2007.

186. J. Boyar and R. Peralta, "A new combinational logic minimization technique with applications to cryptology," proceedings of Experimental Algorithms SEA. Lecture Notes in Computer Science, vol 6049, pp. 178-189, Naples, Italy, Springer, 2010.

187. K. Tiri and I. Verbauwhede, "A VLSI design flow for secure side-channel attack resistant ICs," in Design, Automation, and Test in Europe, Munich, Germany, Germany, pp. 58-63, IEEE, 2005.

188. W. Yu and S. Köse, "A voltage regulator-assisted lightweight AES implementation against DPA attacks," IEEE Transactions on Circuits Systems I: Regular Papers, vol. 63, no. 8, pp. 1152-1163, 2016.

189. W. Yu and S. Köse, "A lightweight masked AES implementation for securing IoT against CPA attacks," IEEE Transactions on Circuits Systems I: Regular Papers, vol. 64, no. 11, pp. 2934-2944, 2017.

190. Y. HAN, X.-c. ZOU, Z.-l. LIU, and Y.-c. CHEN, "The research of DPA attacks against AES implementations," The Journal of China Universities of Posts and Telecommunications, vol. 15, no. 4, pp. 101-106, 2008.

191. W. Schindler, K. Lemke, and C. Paar, "A stochastic model for differential side-channel cryptanalysis," proceeding if international Workshop on Cryptographic Hardware and Embedded Systems, Edinburgh, United Kingdom, vol. 3659, pp. 30-46, Springer, 2005.

192. C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," National Taiwan University, Taiwan 2003.

193. F. Regazzoni et al., "A simulation-based methodology for evaluating the DPA-resistance of cryptographic functional units with application to CMOS and MCML technologies," proceeding of International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation, Samos, Greece, pp. 209-214, IEEE, 2007.

194. Z. Toprak, A. Verma, Y. Leblebici, P. Ienne, and C. Paar, "Design of low-power DPA-resistant cryptographic functional units," proceeding of the Cryptographic Advances in Secure Hardware, 2005.

195. D. Yamamoto, G. Hospodar, R. Maes, and I. Verbauwhede, "Performance and security evaluation of AES s-box-based glitch PUFs on FPGAs," proceeding of International Conference on Security, Privacy, and Applied Cryptography Engineering, Chennai, India, vol. 7644, pp. 45-62, Springer, 2012.

196. P. Hamalainen, T. Alho, M. Hannikainen, and T. D. Hamalainen, "Design and implementation of low-area and low-power AES encryption hardware core," proceeding of 9th EUROMICRO conference on digital system design (DSD'06), Dubrovnik, Croatia, pp. 577-583, IEEE, 2006.

197. M. Nassar, Y. Souissi, S. Guilley, and J.-L. Danger, "RSM: a small and fast countermeasure for AES, secure against 1st and 2nd-order zero-offset SCAs," proceeding of Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, pp. 1173-1178, IEEE, 2012.

198. R. P. McEvoy, C. C. Murphy, W. P. Marnane, M. Tunstall, and Systems, "Isolated WDDL: a hiding countermeasure for differential power analysis on FPGAs," ACM Transactions on Reconfigurable Technology, vol. 2, no. 1, pp. 1-23, 2009.

199. Y. Li, K. Ohta, and K. Sakiyama, "Revisit fault sensitivity analysis on WDDL-AES," proceeding of IEEE International Symposium on Hardware-Oriented Security and Trust, San Diego CA, USA, pp. 148-153, IEEE, 2011.

200. H. Kim, S. Hong, and J. Lim, "A fast and provably secure higher-order masking of AES S-box," proceeding of International Workshop on Cryptographic Hardware and Embedded Systems, Nara, Japan vol. 6917, pp. 95-107, Springer, 2011.

201. T. S. Messerges, "Power analysis attacks and countermeasures for cryptographic algorithms," Doctoral Thesis, Elect. Eng./Computer Science, the University of Illinois at Chicago, the University of Illinois at Chicago, United States, 2000.

202. H. Gross, K. Stoffelen, L. De Meyer, M. Krenn, and S. Mangard, "First-Order Masking with Only Two Random Bits," proceeding of the Proceeding of ACM Workshop on Theory of Implementation Security Workshop, London, United Kingdom, 2019.

203. J. D. Golic and R. Menicocci, "Universal masking on logic gate level," Electronics Letters, vol. 40, no. 9, pp. 526-528, 2004.

204. J.-F. Lin, Y.-T. Hwang, M.-H. Sheu, and C.-C. Ho, "A novel high-speed and energy-efficient 10-transistor full adder design," IEEE Transactions on Circuits Systems I: Regular Paper, vol. 54, no. 5, pp. 1050-1059, 2007.

205. Zhang, Liwei, A. Adam Ding, Yunsi Fei, and Pei Luo. "A unified metric for quantifying information leakage of cryptographic devices under power analysis attacks." proceeding of International Conference on the Theory and Application of Cryptology and Information Security, pp. 338-360. Springer, Berlin, Heidelberg, 2015.

206. M. Alam, S. Ghosh, M. Mohan, D. Mukhopadhyay, D. R. Chowdhury, and I. Gupta, "Effect of glitches against masked AES S-box implementation and countermeasure," IET Information Security, vol. 3, no. 1, pp. 34-44, 2009.

207. Y. Zhou, G. Qian, Y. Xing, H. Liu, S. Goto, and Y. Tsunoo, "An approach of using different positions of double registers to protect AES hardware structure from DPA," proceeding of third International Symposium on Electronic Commerce and Security, Guangzhou, China, pp. 223-227, IEEE, 2010.

208. O. Adegbite and S. R. Hasan, "A novel correlation power analysis attack on PIC based AES -128 without access to crypto device," proceeding of 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA, pp. 1320-1323, IEEE, 2017.

209. J. M. Padilla, U. Meyer-Baese, and S. Foo, "Security evaluation of Tree Parity Re-keying Machine implementations utilizing side-channel emissions," EURASIP Journal on Information Security, vol. 2018, no. 1, pp. 1-16, 2018.

210. H. Mestiri, F. Kahri, B. Bouallegue, and M. Machhout, "A CPA attack against cryptographic hardware implementation on SASEBO-GII," proceeding of International Conference on Green Energy Conversion Systems (GECS), Hammamet, Tunisia, pp. 1-5, IEEE, 2017.

List of Publications

In this thesis, we have designed a novel architecture using Schmitt-triggered PUF and the Folding technique for Logical Obfuscation in order to enhance device security. The obtained results will be published in the "Journal of Enggering Research" and the "International Journal of Embedded System." The results have been published in the following journal/conference/patent/copyright and book chapter:

Objective 1

1. Jyotirmoy Pathak, Suman Lata Tripathi, "Novel Architecture for Authentication Based Reliable Hardware Security Model", International Conference on Data and Information sciences, Springer LNNS, Springer, Agra, May 21.

2.Jyotirmoy Pathak, Kumar, Abhishek and Suman Lata Tripathi. "A system to generate key for hardware authentication." The Patent Office, Journal No. 51/2019, Application No.201911049739 A.

3. Kumar, Abhishek, Jyotirmoy Pathak, and Suman Lata Tripathi. "Frequency-Based RO-PUF." AI Techniques for Reliability Prediction for Electronic Components, pp. 252-261. IGI Global, 2020.

Objective 2

1. Jyotirmoy Pathak, Suman Lata Tripathi, "Column Shifting algorithm to compute iteration bound of FIR systems having inline delays", International Journal of Embedded Systems, volume-14(5), pp- 443-450, 2021.

2. Jyotirmoy Pathak, Abhishek Kumar, Suman Lata Tripathi, "High Level Transformation Technique for designing Reliable and Secure DSP Architecture", AI Techniques for Reliability Prediction for Electronic Components, pp. 164-174, IGI Global Publication, 2020.

3. Jyotirmoy Pathak, Suman Lata Tripathi, "Handbook for Frontend Design", Copyright office Government of India, Registration No. -L-98676/2021

Objective 3

1. .Jyotirmoy Pathak, Suman Lata Tripthi, “Novel obfuscated secure architecture for Baugh Wooley multiplier”, International Conference on Advances in Computer Engineering & Communication Technology, Journal of Physics: Conference Series, IOP, Oct. 21

2. Jyotirmoy Pathak, Suman Lata Tripathi, “Hardware Protection through Logic Obfuscation”, Advance VLSI Design and TestabilityIssues, pp. 339-350 ,CRC Press, 2020.

Objective 4

1. Jyotirmoy Pathak, Suman Lata Tripathi, “A Novel model for resisting side channel attack by masking of gates”, Journal of Engineering Research, DOI: doi.org/10.36909/jer.ICMET.17165.

2. Jyotirmoy Pathak, Abhishek Kumar, Suman Lata Tripathi, “A Novel Obfuscated Processor Architecture for Hardware Security and Optimization”, The Patent Office, Journal No. 51/2019, Application No – 201911049369-A