Power Analysis of Sub-threshold Logics for Security Applications

by

Farhad Haghighizadeh

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2012

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Requirements of ultra-low power for many portable devices have drawn increased attention to digital sub-threshold logic design. Major reductions in power consumption and frequency of operation degradation due to the exponential decrease of the drain current in the sub-threshold region has made this logic an excellent choice, particularly for ultra-low power applications where performance is not the primary concern. Examples include RFID, wireless sensor networks and biomedical implantable devices. Along with energy consumption, security is another compelling requirement for these applications. Power analysis attacks, such as Correlation Power Analysis (CPA), are a powerful type of side channel attacks that are capable of performing a non-invasive attack with minimum equipment. As such, they present a serious threat to devices with secret information inside. This research analyzes sub-threshold logics from a previously unexplored perspective, side channel information leakage.

Various transistor level and RTL circuits are implemented in the sub-threshold region as well as in the strong inversion region (normally the standard region of operation) using a 65 nm process. Measures, such as Difference of Mean Energies (DME), Normalized Energy Deviation (NED) and Normalized Standard Deviation (NSD) are employed to evaluate the implemented architectures. A CPA attack is also performed on more complex designs and the obtained correlation coefficients are used to compare sub-threshold and strong inversion logics.

This research demonstrates that sub-threshold does not only increase the security against side channel attacks, but can also decrease the amount of leaked information. This research also shows that a circuit operating at sub-threshold consumes considerably less energy than the same circuit operating in strong inversion and the level of its instantaneous power consumption is significantly lower. Therefore, the noise power required to cover the secret information decreases and the attack may be dramatically more difficult due to major increase in the number of required power traces and run time. Thus, this research is important for identifying sub-threshold as a future viable technology for secure embedded applications.

Acknowledgements

I would like to express my most sincere thanks and deepest gratitude to my parents, Zohreh Amini and Hossein Haghighizadeh, for their infinite love and continuous support and dedication. I would never have had the chance to follow my dreams without their support. To them, I owe my entire achievements. I would also like to thank my grandma, Monir Naseri, and my brother, Ali Haghighizadeh, for their unconditional love and inspiration.

I would like to thank my supervisor, Prof. Cathy Gebotys, for her guidance, kind support and encouragement throughout my graduate studies. This thesis would have not been possible without her help and support.

My thanks and appreciation also go to Dr. Amir Khatibzadeh for his invaluable help throughout this research.

As well, I am grateful to the administrative and technical staff of the University of Waterloo, especially Phil Regier, for his assistance with the tools.

Additionally, I would like to thank my colleagues in the Laboratory for Side Channel Security of Embedded Systems at the University of Waterloo, including Dr. Marcio Juliato, Dr. Edgar Mateos Santillan, Masoumeh Dadjou and Dr. Patrick Longa for their friendship and encouragement.

Last but not least, I wish like to thank all of my dear friends, Alborz Rezazadeh, Rozhin Yousefi, Mohammadreza Fakhari Moghaddam Arani and Sasan Taghizadeh who enriched my life beyond my studies. I have been lucky to have such good friends by my side.

To my parents

for their unconditional love, support and encouragement

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Objectives and Motivations

In the mid 1990's, power consumption limitations, especially for portable devices commenced a new era in sub-threshold circuit design. Specifically, all the transistors in a sub-threshold circuit operate in sub-threshold region. Applying a power supply voltage less than the threshold voltage of the MOS transistor ensures this functionality. The drain current of an MOS transistors illustrate an exponential dependence on the gate voltage in the sub-threshold region in contrast to the linear/quadratic dependency in the standard region of operation, also known as strong inversion.

This exponential decrease of drain current with the gate voltage causes a major reduction in power consumption; however, it also degrades the frequency of operation. Therefore, sub-threshold logics are an excellent choice for ultra-low power applications where performance is not the primary concern. Despite the research efforts into sub-threshold circuits, sub-threshold chips have not yet been commercially available [1]. Some recent commercial chips utilize near-threshold circuitry [2] but sub-threshold design and chips remain in academic and industrial labs. Some examples of research applications of sub-threshold include RFID[3], wireless sensor networks, biomedical implantable devices[4], and others [5][6][7]. Clearly security is an important aspect in these applications, yet limited research has been done in this area.

Side channel analysis attacks utilizing the power measurement of circuits were published in 1999 [8]. These attacks exploit the secret information of cryptographic devices by observing physical characteristics of the device, such as power consumption [8], electromagnetic radiation [9] and run time [10]. Power analysis attacks such as differential power analysis (DPA) and correlation power analysis (CPA) are an important type of side channel analysis attack. Instantaneous power consumption of a cryptographic device is recorded during its operation and subsequent analysis may reveal the secret information by exploiting the dependency of power consumption on the handled data within the device. In general the attacker does not need to know detailed information about the implementation of the cryptographic device in order to launch the attack. Hence, their capability to perform a non-invasive attack with minimum equipment and implementation knowledge has made them a serious threat for cryptographic devices.

Security is a compelling requirement for most applications in which sub-threshold operation is necessary. A crypto-processor is a common component in many devices. Thus it is important to secure the secret key of crypto-processors against power analysis attacks, especially DPA and CPA attacks. However, side channel analysis of this logic scheme is generally an unexplored aspect. Previous research has either not launched a CPA on sub-threshold or focused on only a smaller sub-circuit of a standardized cipher, known as AES [11].

The ultimate objective of this thesis is to study the estimated power analysis of a sub-threshold circuit. However, other side channel information leakage measures, such as difference of mean energies (DME), frequency of observation, normalized energy deviation (NED) and normalized standard deviation (NSD) are also used for comparison purposes. Simple architectures are

implemented at the transistor level and an AES core is implemented at the register transfer level. An application-specific integrated circuits (ASIC) design methodology for sub-threshold design is used to implement and study such architectures. Preliminary simulations demonstrate the reduction of power consumption correlation with input data by lowering the supply voltage. In light of the above mentioned concerns, motivations and desirable preliminary results, a detailed study on the power analysis of sub-threshold logics is performed.

In the first step, a power consumption analysis of an exclusive-or (XOR) gate is performed using DME measure to provide insight into the side channel information behavior in the sub-threshold region. Further investigation on basic gates is performed by analyzing averaged power consumption of NAND, NOR and XOR gates in various transitions by means of frequency of observation, NED and NSD measures. A final architecture designed at the transistor level, named parallel XORs is analyzed with DME and CPA measures. Afterward, measurements move to a more complex architecture, an important component of AES, the S-Box, and a crypto-AES processor. Correlation power analysis is performed on power consumption traces obtained from electronic design automation (EDA) tools. The challenges involved in a register transfer level (RTL) design for a sub-threshold operation are also described.

There is no default judgment about the side channel security of circuits operating at sub-threshold in this thesis. Moreover the purpose is not to prove that sub-threshold logic is either more secure or less secure, but to study the side channel information leakage in that region and compare it to strong inversion as the standard region of operation.

1.2 Thesis Overview

The thesis is organized as follows. Chapter 2 presents a literature survey on sub-threshold logics, side channel attacks and side channel resistant logics followed by an overview of previous research proposed on side channel analysis of sub-threshold logics. Methodologies used for sub-threshold circuits design is described in Chapter 3 which starts with an explanation of transistor level design and continues with details of RTL level design and digital ASIC design flow for sub-threshold. Chapter 4 introduces power analysis metrics used to evaluate the architectures proposed in Chapter 3. Simulation results and information leakage comparisons between strong inversion, sub-threshold and two side channel resistant logics is presented in Chapter 5. Finally, Chapter 6 summarizes and concludes the thesis and provides recommendations for future work.

Chapter 2
Background and Previous Research

The main components of this research are digital sub-threshold logic and side channel analysis. In this chapter, background information required to understand later chapters is described. The first section explains the behavior of a transistor operating in a sub-threshold region and introduces current and power models of a transistor at sub-threshold. Side channel analysis attacks are explained in section 2.2, which begins with a brief review of simple power analysis and differential power analysis. The section continues with a correlation power analysis introduction and ends with a discussion on side channel analysis measures and proposed side channel resistant logics. Previous research work accomplished in the area of side channel analysis of sub-threshold circuits is investigated in the final section of this chapter.

2.1 Digital Sub-threshold Logic

In the mid 1960's, microwatt power consumption limitations of electronic watches drew attention to an unexplored aspect of newly proposed CMOS technology- the sub-threshold current. In 1972, Barron [12] presented a model that showed the exponential dependent of the sub-threshold current on the surface voltage, but did not propose any simple relationship with the gate voltage [13]. In the same year, Swanson and Meindl [14] explained the relationship between surface voltage and gate voltage. They applied their model to find the transfer characteristic of a CMOS inverter in weak inversion and for the first time showed that CMOS logic circuits can operate at a supply voltage as low as 8kT/q [13].

Despite all of the research undertaken since the mid 1960's, the application of sub-threshold circuits was completely ignored until the mid 1990's, when power consumption limitations especially for portable devices initiated a new era in sub-threshold circuit design.

A sub-threshold circuit is defined as a circuit in which all of the transistors operate in the sub-threshold region. Applying a power supply voltage less than the threshold of the MOS transistor ensures this functionality. Nowadays, sub-threshold CMOS logic has made its way into applications for which energy consumption is the key-metrics. The exponential relationship between the drain current and the gate to source voltage of an MOS transistor in the sub-threshold region gives an exponential reduction in power consumption, but also exponentially increases the delay [15].

Several research projects have tried to find the minimum energy point at which the energy consumption of a circuit is less than any other point of the parameter space [16-20]. Calhoun shows in [17] that the minimum energy operation occurs in the sub-threshold region and [19] proposes an analytical solution to find the optimum power supply voltage, V_{DD}, and threshold voltage, V_T, to minimize energy for a given frequency in the sub-threshold region. Figure 2.1 shows the energy per operation and delay as a function of V_{DD}. The minimum energy point, quadratic decrease in energy

and exponential increase in delay can be observed in this figure. Choosing the operating point in a circuit is a trade-off between energy and delay that can also be observed in Figure 2.1. One can see that sub-threshold circuits minimize energy consumption at a cost of slower speed.

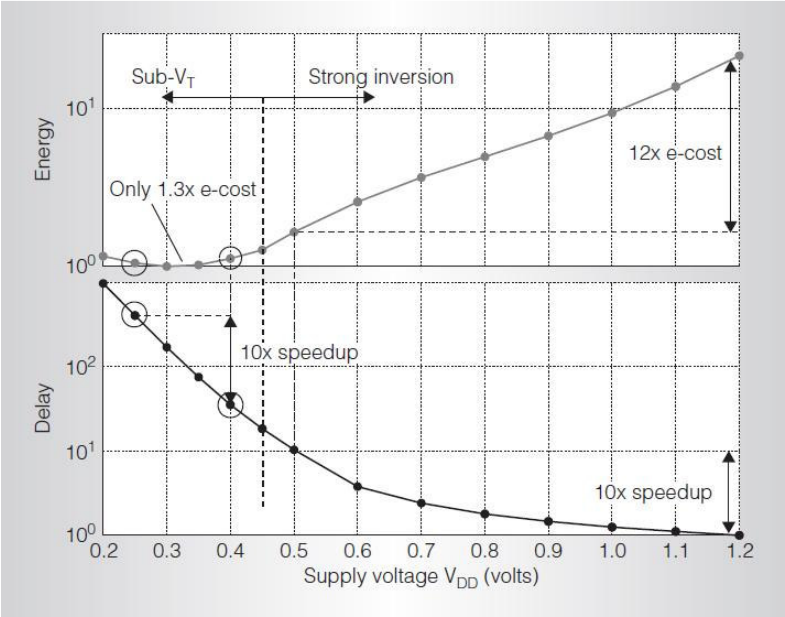


Figure 2.1: Normalized energy per operation (top) and normalized delay (bottom) of a digital circuit as a function of VDD [1].

This section starts with a description of a transistor model at sub-threshold voltage and then provides a power consumption model based on the presented transistor model.

2.1.1 MOS Transistor Model for Sub-threshold Operation

Analyzing a circuit in a sub-threshold region requires an accurate MOS transistor model adapted for low-voltage and low-current applications. In 1995, Enz, Krummenacher and Vittoz presented a fully analytical MOS transistor model dedicated to the design and analysis of low-voltage and low-current analog circuits, known as EKV [21].

Figure 2.2 shows the cross section view and symbol of an n-channel MOS transistor. The value of drain and source voltages with respect to the pinch-off voltage divides the operation modes of the MOS transistor into four modes: conduction (strong inversion), blocked (weak inversion), forward saturation and reverse saturation. The pinch-off voltage, V_P, is the gate to source voltage for which the channel width is reduced to zero.

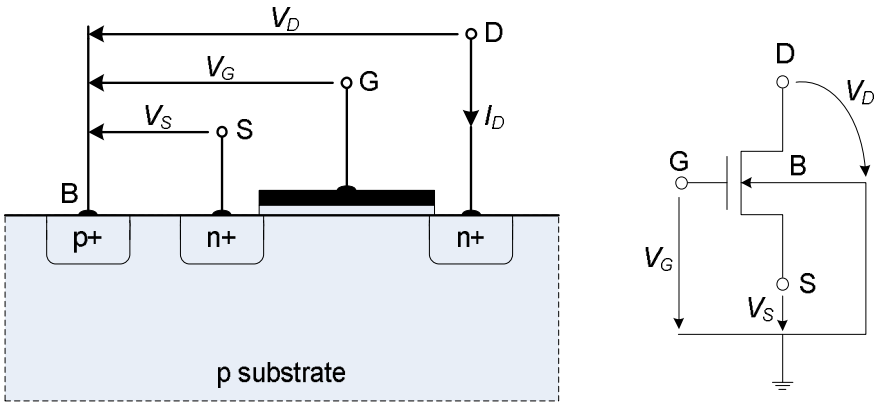


Figure 2.2: Cross section view and symbol of an n-channel MOS transistor

Figure 2.3 illustrates the aforementioned modes. If V_D and V_S are both less than V_P, the channel is in strong inversion and the transistor works in conduction mode. Either forward saturation or reverse saturation occurs, depending on the sign of $V_D - V_S$, when the channel is pinched-off from the drain end or source end, respectively. If both V_D and V_S are larger than V_P, the whole channel is pinched off and the device operates in blocked mode. If either of V_D or V_S is still close to the pinch-off voltage, the device is weakly inverted.

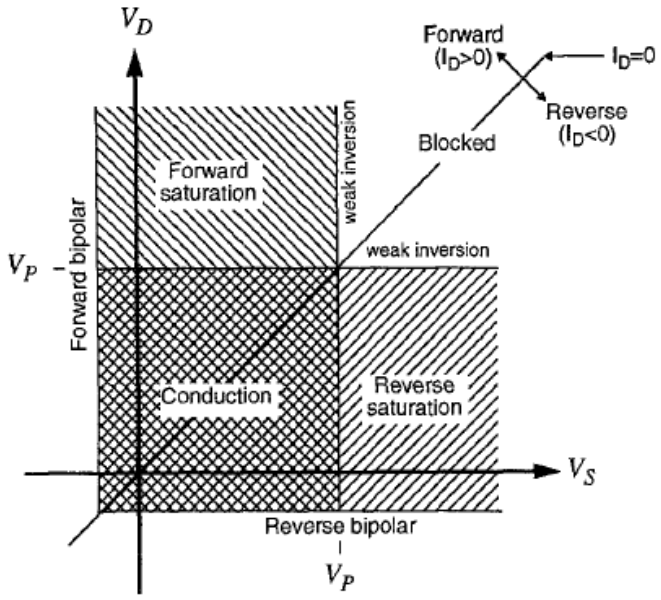


Figure 2.3: Modes of operation of a MOS transistor [21].

According to the EKV model, the drain current can be decomposed into a forward current, I_F, which depends only on the difference voltage $V_P - V_S$, and a reverse current, I_R, which depends only on $V_P - V_D$. Expression 2.1 shows the equation for I_D. Equations 2.2 and 2.3 evaluate the expressions for the forward and reverse currents in strong inversion and weak inversion, respectively. Derivations of these formulas can be found in [21].

$$I_D = I_F - I_R \tag{2.1}$$

$$I_F = \begin{cases} \frac{n\beta}{2}(V_P - V_S)^2 & for: V_S < V_P \\ 0 & for: V_S \geq V_P \end{cases} \tag{2.2}$$

$$I_R = \begin{cases} \frac{n\beta}{2}(V_P - V_D)^2 & for: V_D < V_P \\ 0 & for: V_D \geq V_P \end{cases}$$

$$I_F = K_w \beta V_{th}^2 e^{\frac{V_P - V_S}{V_{th}}} \tag{2.3}$$

$$I_R = K_w \beta V_{th}^2 e^{\frac{V_P - V_D}{V_{th}}}$$

where:

$$\beta = \mu_n C'_{ox} \frac{W}{L} \tag{2.4}$$

$n = 1 + \frac{C_d}{C_{ox}}$, n is called the sub-threshold slope factor (2.5)

$$K_w = (n-1) e^{\frac{\Psi_0 - 2\Phi_F}{V_{th}}} \tag{2.6}$$

Other parameters include μ_n is the mobility of electrons, C'_{ox} is the gate oxide capacitance per unit area, W is the width and L is the length of transistor. Also, V_{th} is the thermodynamic voltage of MOS transistor, Ψ_0 is the surface potential constant, and Φ_F is the Fermi potential. Their expressions and definitions are not important for this work and can be found in [21].

An accurate model like EKV is necessary for careful analysis of analog circuits or gate level circuit design, especially when the circuit operates at the weak inversion edge. However, more basic models exist which are able to propose a reasonable estimate of circuit behavior. Since, in the sub-threshold region, the sub-threshold current (which is a diffusion current) dominates other components of the drain current, such as gate leakage and Gate-Induced Drain Leakage (GIDL), the total drain current can be equated to the sub-threshold current. Equation 2.7 represents this model of a sub-threshold current [13].

$$I_D = I_0 e^{\frac{V_{GS} - V_T}{nV_{th}}} \tag{2.7}$$

where V_T is the threshold voltage of MOS transistor and I_0 is the drain current when $V_{GS} = V_T$, as given in Equation 2.8:

$$I_0 = \beta\,(n-1)\,V_{th}^2 \tag{2.8}$$

To model low V_{DS} roll-off and Drain-Induced Barrier Lowering (DIBL), Equation 2.7 can be upgraded to Equation 2.9.

$$I_D = I_0\, e^{\frac{V_{GS}-V_T+\eta V_{DS}}{nV_{th}}}\left(1-e^{\frac{-V_{DS}}{V_{th}}}\right) \tag{2.9}$$

shere, η is the DIBL coefficient. The sub-threshold current derivation can be found in [22].

Figure 2.4 shows the drain current versus the gate to source voltage. As Equation 2.9 predicts and Figure 2.4 represents, the drain current, I_D, varies exponentially with V_{GS} in the sub-threshold region. As will be mentioned later, this exponential relationship in the sub-threshold region causes an exponential reduction in power consumption.

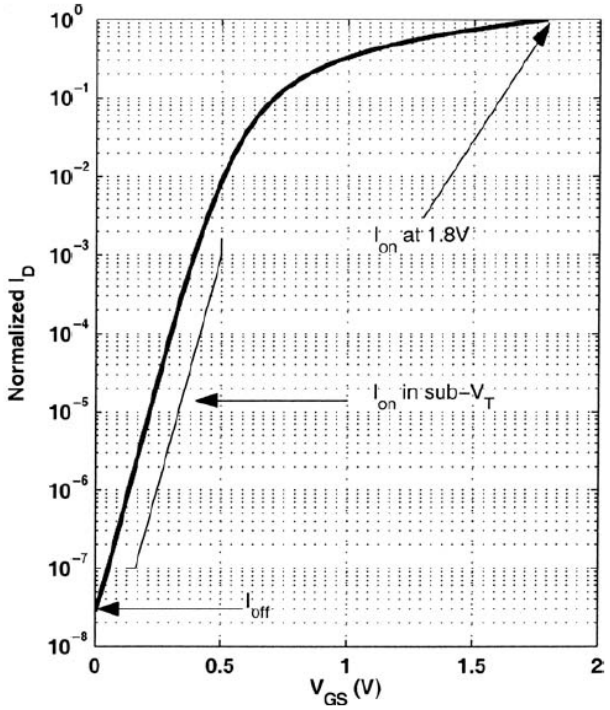


Figure 2.4: I_D versus V_{GS} for a MOS transistor in 0.18µm process with nominal V_{DD} of 1.8V [13].

Figure 2.5 compares the variation of I_D with respect to V_{DS} in strong inversion against a sub-threshold. Sub-threshold curves show the exponential dependence on V_{GS}, but they appear quite

similar to the strong inversion curves in their shape. The quasi-linear region comes from the roll-off of the current at low V_{DS}. Unlike strong inversion, the onset of this roll-off depends only on V_{DS} and not on V_{GS}. In strong inversion, the V_{DS} dependence on the velocity saturation region results from channel length modulation and is commonly modeled with the early voltage. Early voltage is the voltage where a tangent to I_D-V_{DS} curve intercepts the voltage axis. In sub-threshold, the V_{DS} dependence in the quasi-saturation region results from DIBL and can be modeled with a DIBL coefficient.

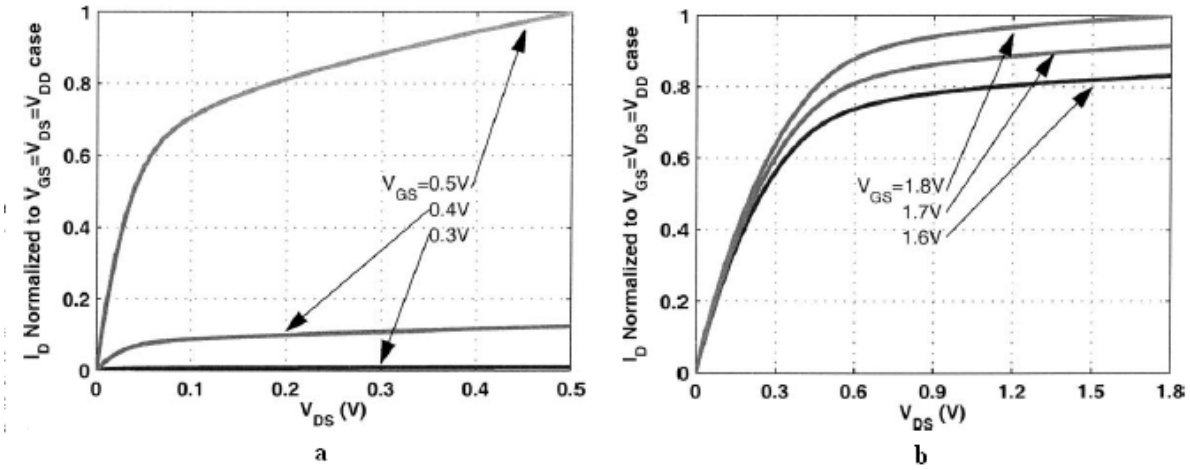


Figure 2.5: I_D versus V_{DS} for three values of V_{GS} in a 0.18µm process. a. sub-threshold, b. strong inversion [13].

2.1.2 MOS Power Model for Sub-threshold Operation

The total power consumption of a digital circuit is the sum of its dynamic, static and short-circuit power, as given in Equation 2.10 [23].

$$P_{total} = P_{Dynamic} + P_{Static} + P_{Short\text{-}Circuit} \tag{2.10}$$

The dynamic power is evaluated by Equation 2.11.

$$P_{Dynamic} = \alpha \times f \times C_{eff} \times V_{DD}^2 \tag{2.11}$$

Where α is activity factor, f is the switching frequency and C_{eff} is the effective capacitance.

Decreasing V_{DD}, lowers the dynamic power quadratically. Therefore, a circuit operating in the sub-threshold region consumes much less power than the same circuit in strong inversion with an identical activity factor and frequency. For instance, in 65 nm technology with a V_{DD} of 0.9 V, a 95% reduction in dynamic power results by moving the V_{DD} to 0.2 V.

Static or leakage power is the power consumed by the system while in steady state and is given by Equation 2.12.

$$P_{Static} = I_{Leakage} \times V_{DD} \tag{2.12}$$

With lower supply voltage during sub-threshold operation, V_{DS} is less than the V_{DS} in strong inversion which results in lower leakage current and therefore lower leakage power. Depending on the technology and V_{DD} scaling, the leakage power is reduced by 4 to 90 times [23]. Since the frequency is lower in sub-threshold operation, leakage power is integrated over a larger time, which makes the leakage energy increase in the sub-threshold region. Thus, dynamic energy, which is the dominant portion of the total energy in strong inversion is no longer dominant at sub-threshold. In fact, it is even less than the leakage energy close to the minimum energy point [19].

Short circuit power is the power dissipated by the short circuit that flows directly between V_{DD} and V_{SS} during a switching transition. Short circuit power is shown in Equation 2.13.

$$P_{Short\text{-}Circuit} = I_{Short\text{-}Circuit} \times V_{DD} \tag{2.13}$$

Due to slower operation at sub-threshold, the period of short circuit in CMOS cells is increased. Even so, the short circuit power is a factor of the supply voltage, and thus is reduced [23].

2.2 Side Channel Analysis Attacks

Secret information hidden in cryptographic devices can be extracted by passive observation of the device's functional behavior or active manipulation of the device to behave abnormally and extract secret information from the intentional abnormality. The first category is generally referred to as side channel attacks (SCA), and the latter as fault injection attacks.

Side channel attacks break a cryptographic device by exploiting information from the physical characteristics of the device, such as power consumption [8], electromagnetic radiation [9] and run time [10]. Recording many samples of instantaneous power consumption (known as a power trace) of a device and analyzing it to exploit secret information, such as a key, is called a power analysis attack.

Power analysis of side channel signals from smartcards was first presented in 1999 by Kocher et al. [8] on a DES algorithm. Instantaneous power consumption of a cryptographic device normally depends on the data it processes and the operation it performs. Power analysis attacks are based on exploiting these dependencies, and various methodologies are introduced to achieve this goal. Simple power analysis (SPA), differential power analysis (DPA) and correlation power analysis (CPA) are the most common types of power analysis methodologies.

This section reviews SPA and DPA as the first proposed methods of power analysis attack. It then provides a brief introduction to CPA, as the method used in this research. Next, measures for analyzing the side channel information leakage are briefly listed. Finally, a few of the side channel resistant logics presented thus far will be introduced and SABL and WDDL (the ones which are employed in this research for comparison) are explained.

2.2.1 Overview of Simple Power Analysis and Differential Power Analysis

SPA attacks exploit information by simply measuring the instantaneous power consumption of a device and correlating its fluctuations with the different rounds of cryptographic algorithms or operations and key values. In other words, an attacker tries to find the key by directly interpreting the single available trace to find patterns or matched templates. On the other hand, DPA attacks extract the key by performing statistical analysis on a large number of power traces to find out how power consumption, at fixed moments in time, depends on the processed data. DPA attacks are thus based on the data dependency of power traces [24].

While SPA attacks require detailed knowledge about the implementation of the cryptographic algorithm, DPA attacks only need to know the algorithm itself. Although several SPA attacks on algorithms like AES have been reported [25, 26], it is much easier to prevent the threat of these attacks compared to DPA attacks. However, SPA is the sole possible method when only one power trace is available. DPA attacks require a large number of traces and their runtime is significantly longer than that of SPA's.

DPA works based on the difference between power consumption of 0 to 1 and 1 to 0 transitions. It collects N power traces corresponding to N plaintexts, P_i ($i = 1 \dots N$), and chooses a selection function, f, which operates on P_i, K_s (the guess key), and bit b, the examined bit (e.g. a bit of the S-Box output). The output of the selection function can be either 0 or 1. The next step is to compute a differential trace $\Delta_f(b)$, which is the difference between the average of traces with f equals 1 and with f equals 0. Expression 2.14 summarizes the described methodology. $W(P_i)$ is the power trace corresponding to the plaintext P_i.

$$\Delta_f(b) = \frac{\sum_{i=1}^{N} f(P_i,b,K_s)\, W(P_i)}{\sum_{i=1}^{N} f(P_i,b,K_s)} - \frac{\sum_{i=1}^{N} (1-f(P_i,b,K_s))\, W(P_i)}{\sum_{i=1}^{N} (1-f(P_i,b,K_s))} \tag{2.14}$$

If calculated bits during the cryptographic algorithm are uniformly distributed and the number of power traces are sufficient, $\Delta_f(b)$ corresponding to a wrong K_s will be zero. Thus, the only $\Delta_f(b)$ which gives the value of nonzero reveals the correct key [27].

In order to enhance DPA, extended attacks, such as higher order DPA (HO-DPA) [8, 28], multi-bit DPA [29, 30] and correlation power analysis (CPA) [31, 32], are proposed. The difference between these methods is mainly in the complexity of the statistical analysis. Correlation power analysis, as an effective and commonly used approach, is chosen as the attack methodology in this research.

2.2.2 Correlation Power Analysis

Correlation power analysis (CPA) attacks are based on the correlation between the power consumption of the cryptographic device and the Hamming weight or Hamming distance of the handled data. In power analysis attacks, it is necessary to have a power model that maps data values processed by the device to the power consumption traces. Specifically, in a DPA attack, the selection function performs the required mapping, which assumes dissimilar power consumption for 0 to 1 and

1 to 0 transitions. However, the power models used in CPA, Hamming weight [33] and Hamming distance [31] assume that 0 to 1 and 1 to 0 transitions contribute equally to power consumption and that 0 to 0 and 1 to 1 transitions also lead to the same power consumption. It is assumed in CPA that information leakage through power consumption depends on the number of bits switching from one state to another at a given time, not the type of transition.

The Hamming weight of a vector input v_0, HW(v_0), is defined as the number of set bits in v_0, and it is assumed that the power consumption is proportional to HW(v_0). The Hamming distance of two vector inputs v_0 and v_1, HD(v_0, v_1), is the number of flipping bits to go from v_0 to v_1. In a Hamming distance model, the power consumption of the device is modeled with the number of bits switching from one state to either its preceding or succeeding state, while in a Hamming weight model the knowledge of the current state is sufficient. Therefore, a Hamming distance model requires more details of the device and may not be possible to mount in all applications.

Brier et al. proposed a model for power consumption based on the Hamming distance model in [31]. The Brier's model can be seen in Expression 2.15.

$$W = a\ HD(D,R) + b \tag{2.15}$$

This model assumes a linear relationship between power consumption, W, and the Hamming distance between D, a uniform random variable, and a reference state, R. This model only represents the data-dependent part of power consumption, which does not seem unrealistic because the majority of a cell's power is consumed within the bus lines [31]. Constant b models enclose offsets, time-dependent components, and noise.

The correlation factor ρ_{WH} between power consumption and the Hamming distance can be calculated as follows:

$$\rho_{WH} = \frac{cov(W,H)}{\sigma_W \sigma_H} \tag{2.16}$$

where σ^2 is variance, we have:

$$\sigma_W^2 = a^2\ \sigma_H^2 + \sigma_b^2 \tag{2.17}$$

So, expression 2.15 can be further simplified to expression 2.18.

$$\rho_{WH} = \frac{a\sigma_H}{\sigma_W} = \frac{a\sigma_H}{\sqrt{a^2\sigma_H^2 + \sigma_b^2}} = \frac{a\sqrt{m}}{\sqrt{ma^2 + 4\sigma_b^2}} \tag{2.18}$$

where m is the number of bits in D as a uniform random variable and m/2 is the mean and m/4 is the variance of *HD(D+R)* as a uniform random variable. The correlation factor is a value between -1 and +1, which ±1 means perfect correlation and the sign depends on the linear gain, a.

Expression 2.18 shows that a minimized noise variance, σ_b^2, maximizes the correlation factor, which helps to determine the reference state, *R*. The process is to scan all possible values of *R* and rank them by the correlation factor. The one with the maximum correlation factor is the correct reference value.

The above claim is proved in [31]. Suppose a correct reference, *R*, and an incorrect guess of the reference value, *R'*, which has *k* bits different from *R*. This would give, *HD(R+R')=k*. Since *b* is independent from other variables, the correlation factor is:

$$\rho_{WH'} = \frac{cov(aH+b,H')}{\sigma_W \sigma'_H} = \frac{a}{\sigma_W}\frac{cov(H,H')}{\sigma'_H} = \rho_{WH}\rho_{HH'} = \rho_{WH}\frac{m-2k}{m} \tag{2.19}$$

Expression 2.19 shows that even a 1 bit difference between *R* and *R'* reduces the correlation factor by 1/4.

In summary, in order to perform a CPA attack, a power model based on the Hamming weight or Hamming distance must first be chosen. In the next step, the device needs to be run with all possible values of a reference state or input plaintext, and a power consumption trace has to be measured. In the final step, the correlation factors between the predetermined values from the power model and measured power traces are calculated and the maximum correlation factor corresponds with the correct key of the device.

2.2.3 Measures for Side Channel information leakage

Various measures exist to evaluate side channel information leakage of cryptographic devices. CPA can be considered as an effective and accurate measure that not only reveals the secret key but also analyzes the amount of information leakage using obtained correlation coefficients. Other than CPA, simpler measures are also introduced that can be employed to study the data and operation dependency of instantaneous or averaged power consumption of a device. The following paragraph gives a brief overview of some of most commonly used measures which will be discussed in further detail in Chapter 4.

Difference of mean energies (DME), suggested by [34], is a measure which highlights the difference between power traces that process 1 compared to those that process 0. The frequency of observation measure used in [35] visualizes the closeness of average power consumption values for all possible inputs. This measure is based on the fact that attacking a system whose average power consumption values for various combinations of inputs are aggregated in a small range requires more effort and sample traces than a system with a wider range of averaged power consumption values. Normalized energy deviation (NED) and normalized standard deviation (NSD), first used by [35], form another type of measure that provides a simple yet effective security evaluation for many logic schemes. They quantize the previous measure, frequency of observation, and determine the dispersion of averaged power consumption values of a cryptographic device for various data transitions.

2.2.4 Side Channel Resistant Logics

The important key to designing a secure cryptographic device is to eliminate data dependency and operation dependency of power consumption. A device which consumes constant energy in all clock cycles of an operation can be a perfectly secured design. A circuit with constant energy consumption and power trace for all types of transitions-0 to 1, 1 to 0, 1 to 1 and 0 to 0-satisfies our desired goal.

Figure 2.6 shows an Inverter in Static Complementary CMOS logic (scCMOS), which is the default logic scheme available in existing standard cell libraries. This cell consumes power from the power supply only during 0 to 1 and 1 to 0 transitions. Since the load capacitance is different in these two cases, the consumed power during these two transitions is different. No power is consumed during 0 to 0 and 1 to 1 transitions. Thus, this logic scheme leaks high amounts of information. A secure logic scheme must have output switching independent of input switching as well as constant load capacitance for all transitions [35].

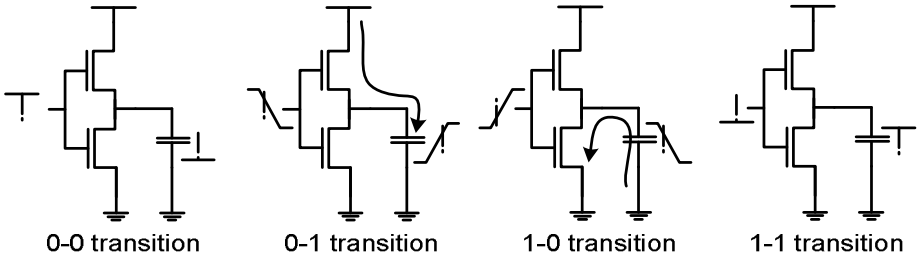


Figure 2.6: Output transitions in a static complementary CMOS logic [35].

Various logics are proposed to provide the above characteristic. Sense Amplifier Based Logic (SABL) [35], Wave Dynamic Differential Logic (WDDL) [36], Dynamic Current Mode Logic (DyCML) [37], Low-Swing Current Mode Logic (LSCML) [38] and Masked Dual-Rail Pre-charge Logic (MDPL) [39] are examples of the presented logics. Mace et al. summarizes the characteristics of mentioned logics in [40], as shown in Table 2.1.

Logic styles	Dual-Rail	Masked	Pre-Charged	Standard Cell
CMOS				✓
SABL	✓		✓	
WDDL	✓		✓	✓
DyCML	✓		✓	
LSCML	✓		✓	
MDPL	✓	✓	✓	✓

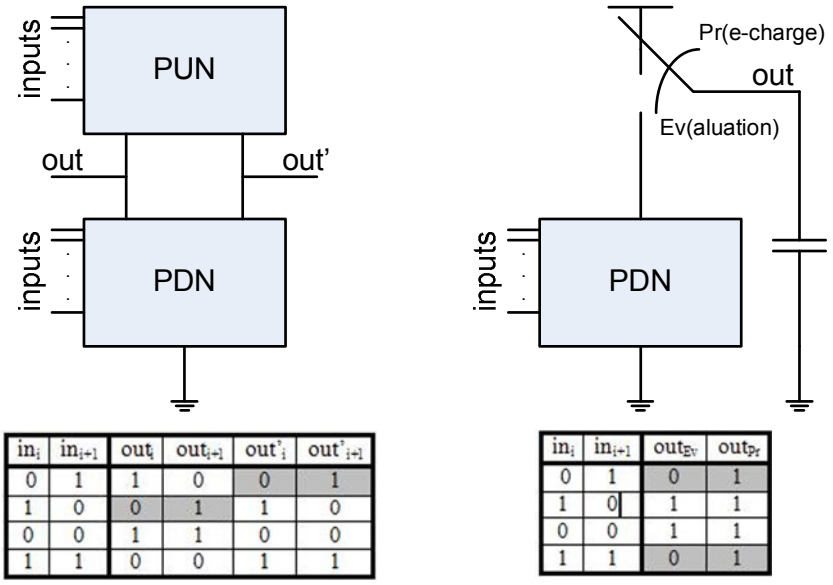
Table 2.1: Summary of characteristics for side channel resistant logics [40].

According to Table 2.1, all side channel resistant logics are dual-rail and pre-charged. SABL, DyCML and LSCML are full custom logic styles, while WDDL and MDPL are compatible with standard cell libraries. In this research, SABL and WDDL are chosen for study and comparison. Following is a brief overview of these two logic schemes.

Figure 2.7.a illustrates a differential network. This network provides true and false values of the output signal with the help of De-Morgan's law. De-Morgan's law generates a false output using false inputs. The truth table of this network is shown in Figure 2.7.a. Since both *out* and *out'* signals flip in each transition of 1 to 0 and 0 to 1, the total power consumption of this differential network is the same for 1 to 0 and 0 to 1.

Figure 2.7.b demonstrates a dynamic network. Here, the clock period is divided into two phases: pre-charge and evaluation. In the first phase, the output signal is pre-charged to the pre-charge value. In the latter phase, output is evaluated based on inputs. This modification in the circuit makes the output flip in 0 to 0 and 1 to 1 transitions; hence, it always consumes power. As the truth table of Figure 2.7.b demonstrates, power consumption for 0 to 1 and 1 to 1 transitions are identical, as is power for 0 to 0 and 1 to 0.

Combining differential and dynamic schemes into one dynamic and differential logic scheme provides almost the same power consumption for all clock cycles. SABL and WDDL are two dynamic and differential logics. The implementation of NAND, NOR and XOR gates in these two schemes is explained in the following sections.



in_i	in_{i+1}	out_i	out_{i+1}	out'_i	out'_{i+1}
0	1	1	0	0	1
1	0	0	1	1	0
0	0	1	1	0	0
1	1	0	0	1	1

a. Differential logic

in_i	in_{i+1}	out_{Ev}	out_{Pr}
0	1	0	1
1	0	1	1
0	0	1	1
1	1	0	1

b. Dynamic logic

Figure 2.7: Power characteristics in the form of a truth table for a) Differential logic b) Dynamic logic [35].

2.2.4.1 SABL

A generic n-gate in SABL logic is shown in Figure 2.8. The differential pull-down network along with the output pre-charge circuit provides the same power consumption for all transitions. SABL is also designed to have a constant load capacitance for all transitions. Implemented NAND, NOR and XOR gates in the SABL logic scheme are shown in Figure 2.9. DPDN is designed such that the conducting path in all possible paths has the same resistance, which is ensured by having the same number of identical transistors in each conducting path of the DPDN network [24].

SABL logic can only be used in custom design. It is impossible to use standard cell libraries and available digital design tools to design a SABL circuit. This issue, together with the significantly large power consumption caused by almost doubling the total number of transistors, is a major drawback of this logic scheme.

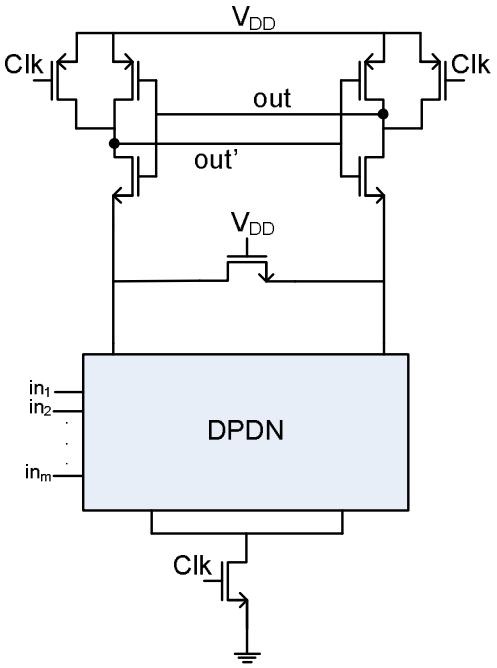


Figure 2.8: Generic n-gate SABL logic [35].

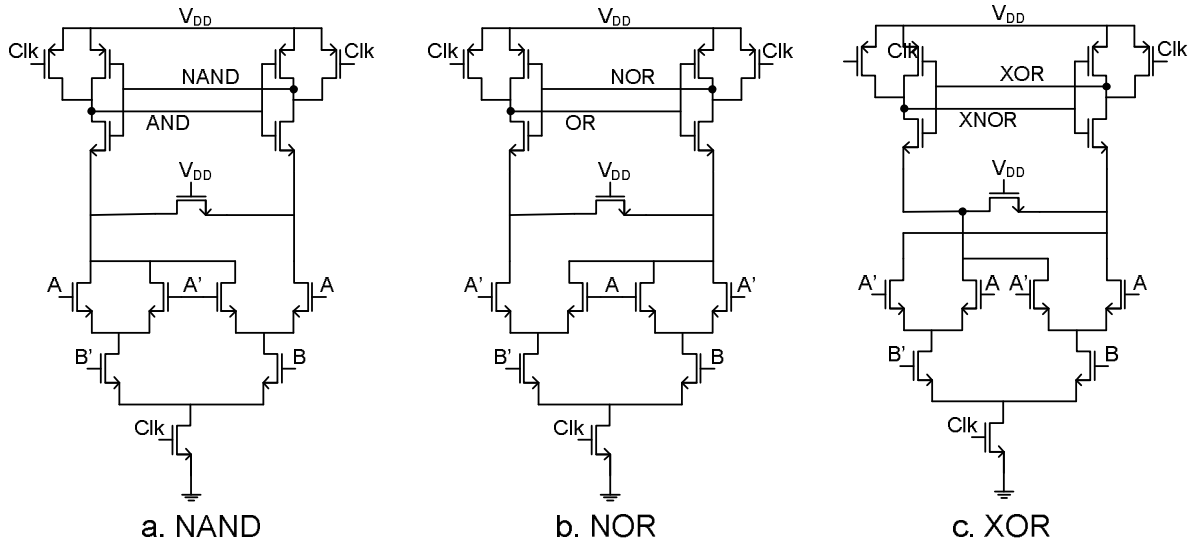


Figure 2.9: SABL gates a) NAND b) NOR b) XOR.

2.2.4.2 WDDL

Figure 2.10 represents NAND, NOR and XOR gates in a WDDL logic scheme. The first difference between WDDL and SABL is that available standard cell libraries can be employed in designing a circuit in WDDL logic. Therefore, this logic scheme provides the capability of an RTL design of a secure circuit in WDDL logic, using the available digital design tools.

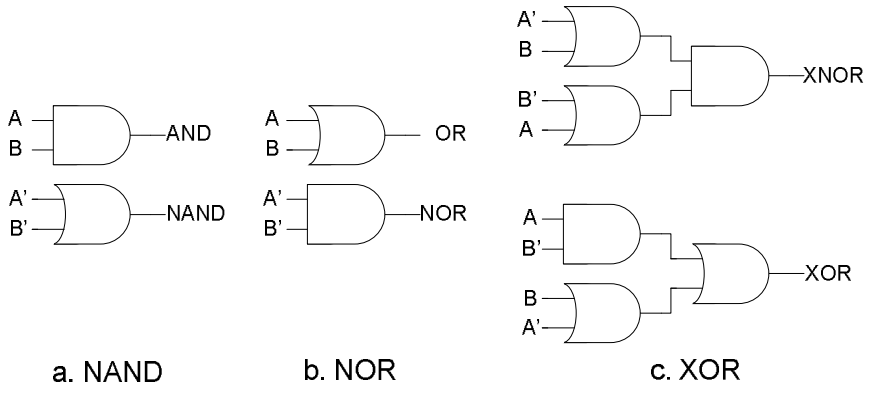


Figure 2.10: WDDL gates a) NAND b) NOR c) XOR.

The second difference is the pre-charging methodology. Figure 2.11 shows the pre-charge circuit for combinatorial WDDL gates. In WDDL, inputs are pre-charged and the pre-charge signals at inputs ripple all the way through the combinatorial circuit to the output, where they pre-charge the output. Having a pre-charge circuit for the inputs eliminates the necessity of having this circuit for all gates. Hence, the area and power consumption of a WDDL gate is lower than the area and power consumption of the same gate in SABL logic.

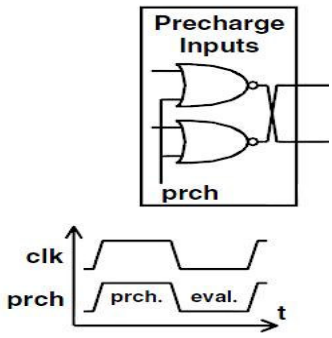


Figure 2.11: WDDL pre-charge circuit [36].

2.3 Previous Side Channel Research of Sub-threshold Circuits

Extensive research has been carried out in the area of sub-threshold circuits. However, side channel information leakage of this logic scheme is still relatively unexplored. To the best of our knowledge, only three papers, [41-43], all published in 2008, have focused on studying information leakage of sub-threshold against differential power analysis attacks.

Alstad and Aunet implemented a static CMOS 8-bit ripple carry adder in [41] and a compact 4-staged pipelined and asynchronous S-Box in [42]. Both circuits were simulated at the transistor level using 90 nm CMOS technology. They used normalized standard deviation of the supply current as the measure of security and claimed that sub-threshold operation reduces the standard deviation with a factor of 2500. Normalized standard deviation was introduced in section 2.2.3. It is not a very strong measure of security especially when it is used individually and it will be discussed in more detail later in Chapter 4.

Haider and Nazhandali mounted a DPA attack on an S-Box in which SPICE-level simulation is performed on a transistor level design implemented in 45 nm technology [43]. Their focus is on the signal-to-noise ratio (SNR) of the sub-threshold circuit versus strong inversion. Figure 2.12 represents their simulation results, showing in part “a” the minimum average noise power that hides the secret key and defeats the DPA attack. According to their results, there are four orders of magnitude difference between the noise power required to cover the secret in sub-threshold and strong inversion. Figure 2.12.b demonstrates the SNR at which a DPA attack can be successful. Larger values of SNR

for sub-threshold mean that less noise is needed to cover the secret information at sub-threshold compared to strong inversion.

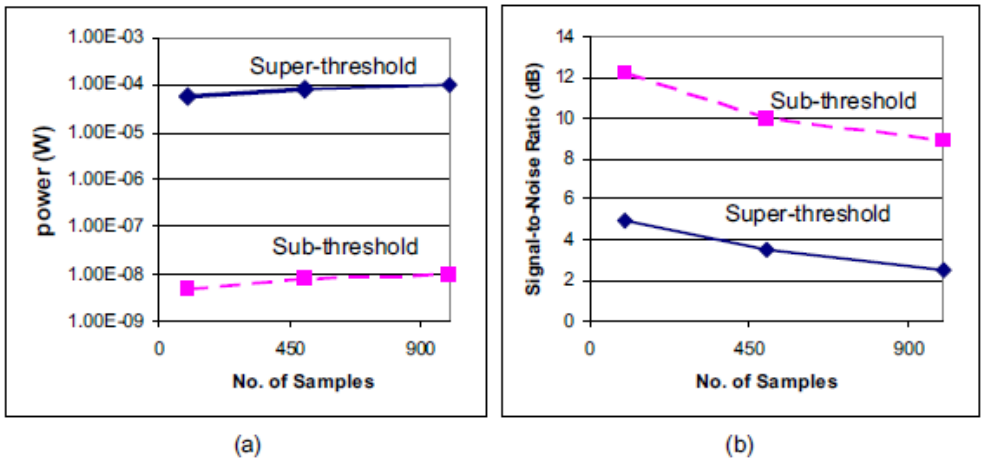


Figure 2.12: Analysis of a secure circuit resistivity toward power analysis attacks [43].

In summary, side channel analysis of sub-threshold circuits has been previously studied in three papers, in which normalized standard deviation and signal-to-noise ratio of a transistor level AES S-Box and an 8-bit ripple carry adder are the main focus. However, further studies on sub-threshold logics are required. A closer look into instantaneous power consumption of a circuit operating in a sub-threshold region can demonstrate the information leakage behavior of this logic scheme. Moreover, correlation power analysis attack is also a serious threat that sub-threshold logics vulnerability against this attack needs to be studied. Finally, side channel analysis of register transfer level implementation of sub-threshold circuits is another unexplored aspect of this logic that is considered in this research.

The next chapter will study the circuits used for power analysis in this research and propose the design methodology for sub-threshold circuit in both transistor level and RTL.

Chapter 3
Sub-threshold Circuits and Design Methodology

This chapter discusses the methodology and challenges of designing a circuit for sub-threshold operation. In section 3.1 a list of challenges which a designer will confront in designing a circuit for sub-threshold operation is presented. In section 3.2, which focuses on transistor level design, the characteristics of an inverter gate at sub-threshold provide a basis for transistor level design which is used in the subsequent design of other gates, such as NAND, NOR and XOR. Afterwards, register transfer level design is presented in section 3.3. The digital design flow of a sub-threshold circuit is explained in that section followed by a performance analysis of standard cell libraries and the Advanced Encryption Standard (AES) crypto-processor used in this research.

3.1 Design Challenges at Sub-threshold

While a circuit operating in sub-threshold region consumes little power, low current level and an exponential dependence of current to voltage introduce a group of deficiencies and challenges which need to be considered by designers. This section briefly describes some of the more important design challenges.

1. Performance

The weak current flow in sub-threshold circuits results in longer delays due to the longer time required to charge and discharge capacitances in the circuit. As mentioned earlier in this section, there is a trade-off in choosing the value of V_{DD} between delay and energy. Choosing the value of V_{DD} higher than corresponding value for minimum energy point can benefit the performance at only a slight cost in energy.

2. Minimum Operational Voltage

The supply voltage of 3-4 V_{th} (V_{th}, the thermal voltage equal to kt/q) is the minimum possible V_{DD} for circuits operating at sub-threshold [23]. The Voltage Transfer Characteristic (VTC) of a minimum-sized inverter at 25°C in a 65 nm TSMC process is sketched for different values of supply voltage in Figure 3.1.

3. Variability

Exponential *I-V* characteristics in a sub-threshold region can cause a large variance in transistor behavior, including process, voltage and temperature variability. Hence, a designer needs to use effective techniques to design more robust and reliable circuits [23, 44].

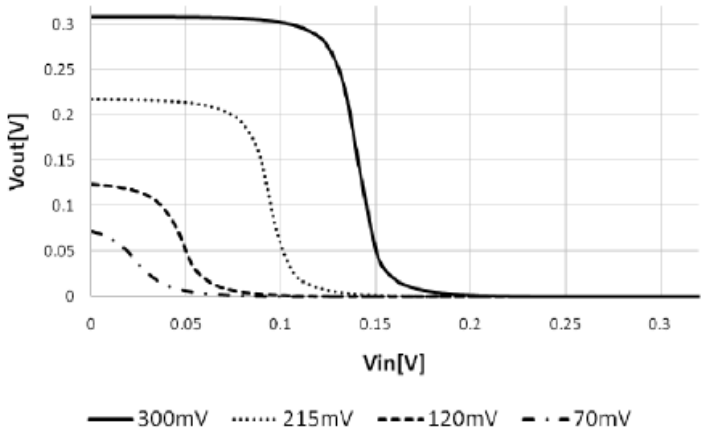


Figure 3.1: VTC as a function of supply voltage [23]

4. Device Optimization

Devices optimized for strong inversion may not give optimal results for sub-threshold operation. The optimization of devices for sub-threshold operation can thus help to achieve higher frequencies [44].

5. Robustness of logic families

Low V_{DD} results in a reduced I_{ON}/I_{OFF} ratio that can reduce robustness. Static CMOS gates function correctly at sub-threshold; however, other logic families may suffer from variations due to low I_{ON}/I_{OFF} ratio [44].

6. Standard Cell Libraries

The simulation and synthesis of sub-threshold circuits is a great challenge due to the unavailability of standard cell libraries designed specifically for sub-threshold operation. Current standard cell libraries are characterized for a specific voltage which is in the strong inversion region. Thus, the synthesis of circuits at sub-threshold requires re-characterization and modification of libraries.

3.2 Transistor Level Design

This section describes the design methodology for transistor level sub-threshold circuits. An exponential dependency of the drain current on V_{DS} alters the transistor's behavior and introduces a new design methodology. This section studies the characteristics of an inverter circuit (e.g. sizing, speed, frequency of operation and minimum operational voltage) to provide a general idea of a transistor level circuit in a sub-threshold region. The section also provide a design explanation of the most important components of every digital circuit (NAND, NOR and XOR gates). All implementations and simulations are performed in Cadence Virtuoso Analog Environment using 65 nm TSMC technology. The transistor level design flow used in this research is shown in Figure 3.2.

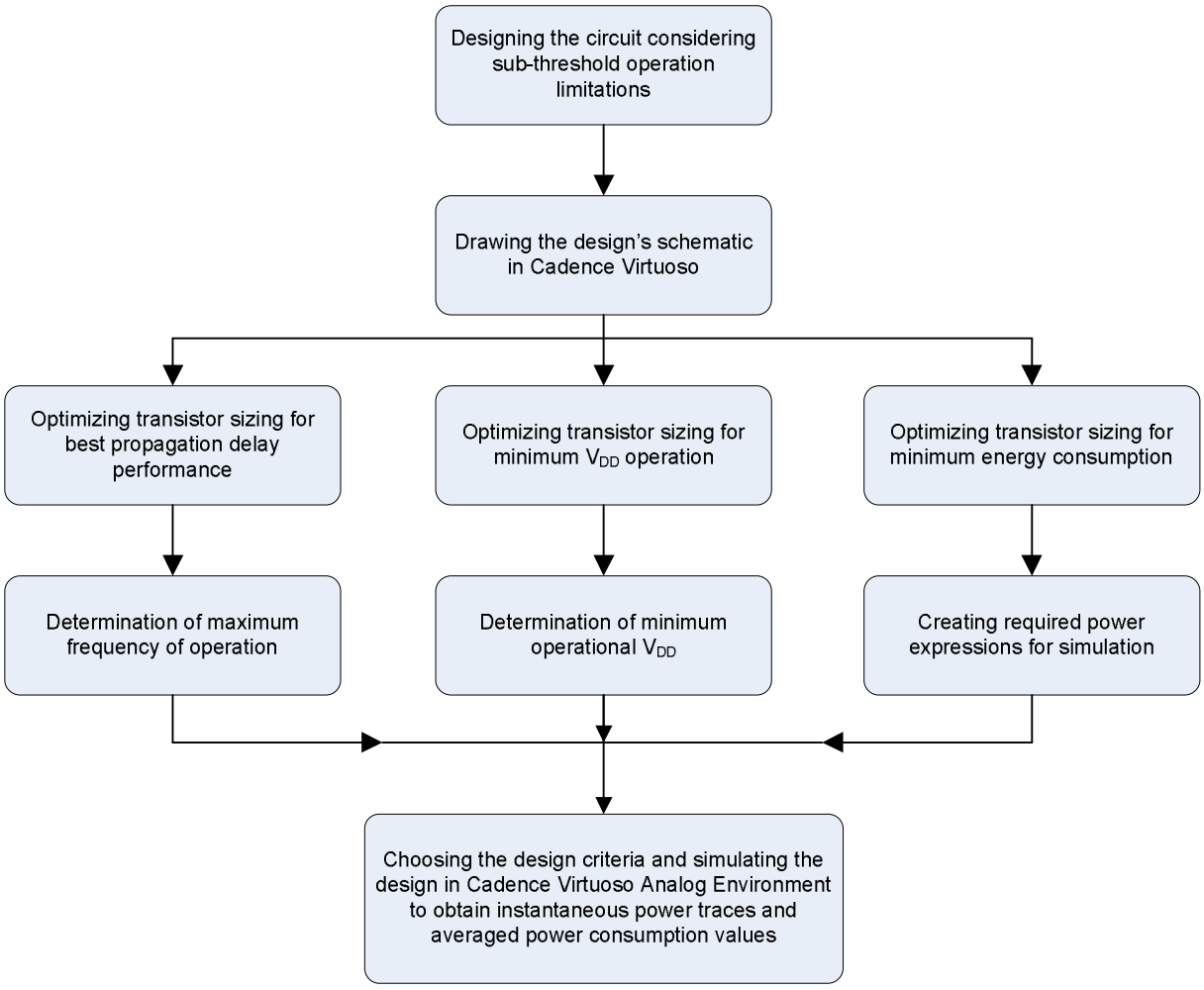


Figure 3.2: Sub-threshold transistor level design flow.

3.2.1 Inverter Operation in Sub-threshold Region

The first design element to be discussed is transistor sizing which can be studied from three points of view. One is, the ratio of PMOS width to NMOS width, at which the circuit operates with the minimum V_{DD}. The ratio W_P/W_N that obtains the same current for both PMOS and NMOS transistors provides the minimum V_{DD} operation. Reference [13] suggests the value of 12 for W_P/W_N to achieve the minimum voltage of 50 mV. However, based on the available transistor model for this research, which is the general purpose model of the 65 nm TSMC process, a lower W_P/W_N value gives better functionality in sub-threshold voltages. Therefore, a minimum supply voltage operation occurs for the equal width size of PMOS and NMOS.

The next factor that impacts sizing is the propagation delay. Based on experiments done in this research on inverters for different sub-threshold voltages in the range of 0.1 V to 0.3 V, the ratio of 2.7 was observed as the best ratio of W_P/W_N to provide equal rising and falling propagation delay. Inverter sizing to achieve the minimum energy is the last criterion that occurs with minimum transistor sizing. Hence, the smallest sized NMOS transistor should be chosen and, depending on the application the ratio of PMOS to NMOS transistor can be set.

The inverter delay in both strong inversion and sub-threshold, in case of symmetrical PMOS and NMOS transistors is given by Equations 3.1 and 3.2, respectively [13].

$$t_d = \frac{K\, C_g V_{DD}}{(V_{DD} - V_T)^{\alpha}} \tag{3.1}$$

$$t_{d,sub} = \frac{K\, C_g V_{DD}}{I_o exp\,(\frac{V_{DD} - V_T}{n\, V_{th}})} \tag{3.2}$$

As Equation 3.2 presents, an exponential decrease of I_{on} in the sub-threshold region leads to an exponential dependency of delay on V_{DD}, which is a stronger dependency compared to the strong inversion delay presented in Equation 3.1. Figure 3.3 shows the normalized inverter speed across the full range of supply voltage. The inverter speed degrades slowly in strong inversion and drops fast for V_{DD} below 0.4 V.

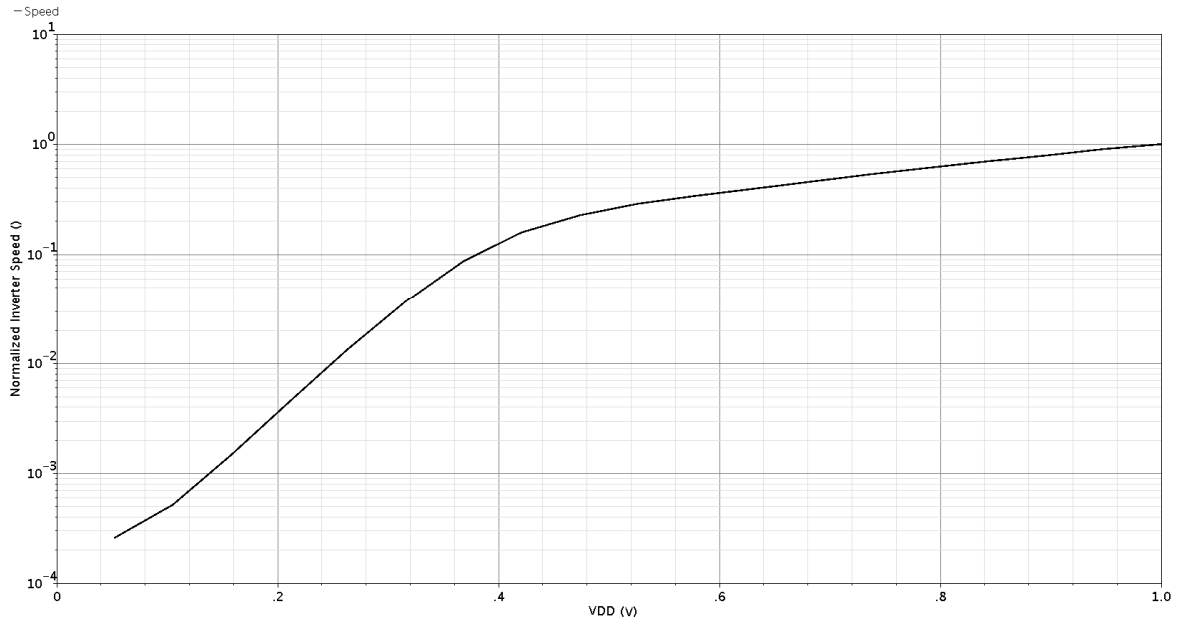


Figure 3.3: Normalized inverter speed versus supply voltage.

Taking into consideration the minimum size for an NMOS transistor and the value of 2 for W_P/W_N, the inverter is tested for various supply voltages in the sub-threshold region. The output voltage of the inverter for V_{DD} = 200 mV, 150 mV, 75 mV and 50 mV is shown in Figure 3.4. For supply voltages of 200 mV and 150 mV, the circuit maintains a full 10% - 90% output swing. Although this swing is degraded for V_{DD} = 75 mV, the output voltage still covers the 10% - 90% swing. From a supply voltage of 60 mV, the output swing degrades drastically and the inverter can no longer be considered operational. The degraded output swing of the inverter for a supply voltage of 50 mV can be observed in Figure 3.4.d. Therefore, the minimum operational voltage using the available models in this research is 60mV.

Now that the sizing, speed and minimum operational voltage are determined, the maximum frequency of operation can be obtained. The output voltages of the inverter for the frequencies of 10 MHz, 33.33 MHz, 50 MHz and 66.67 MHz are shown in Figure 3.5. The operational voltage is 150 mV. We can observe that the output voltage swing starts to degrade from 50 MHz and then cannot maintain the 10% - 90% swing at a frequency of 66.67 MHz (the period of 15 ns). Thus we can see that, the limited range of operational frequency is a significant drawback of sub-threshold circuits. Based on the available models in this research, the maximum frequency is around 70 MHz.

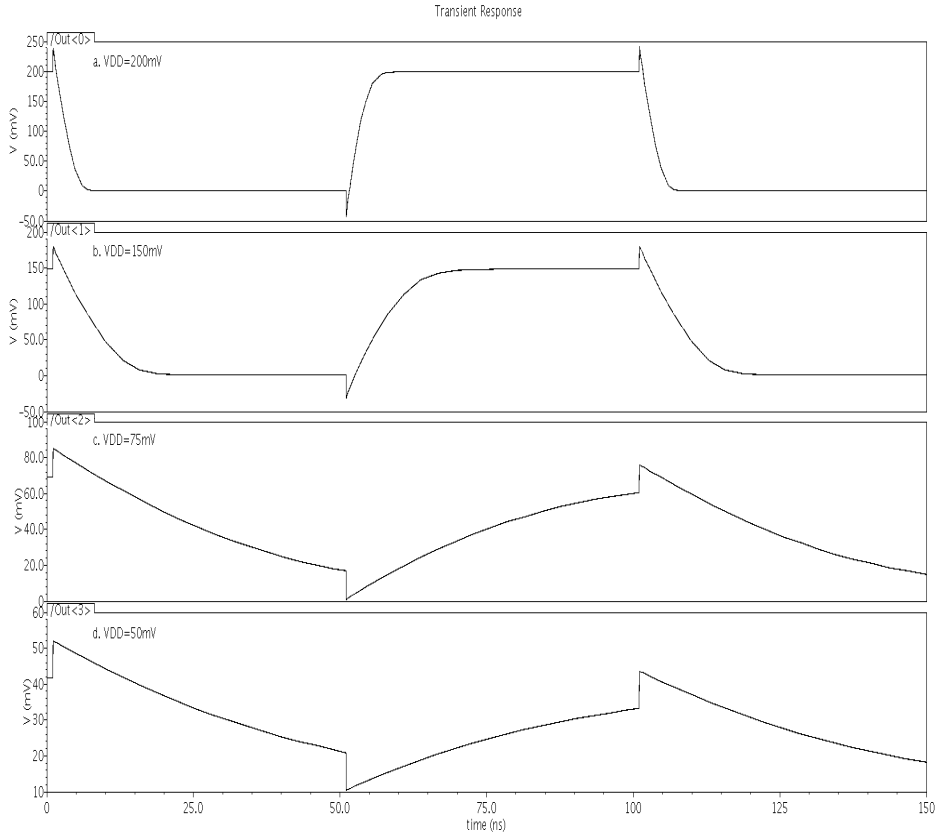


Figure 3.4: Inverter output at the frequency of 10 MHz and supply voltages of a) V_{DD} = 200 mV, b) V_{DD} = 150 mV, c) V_{DD} = 75 mV and d) V_{DD} = 50 mV.

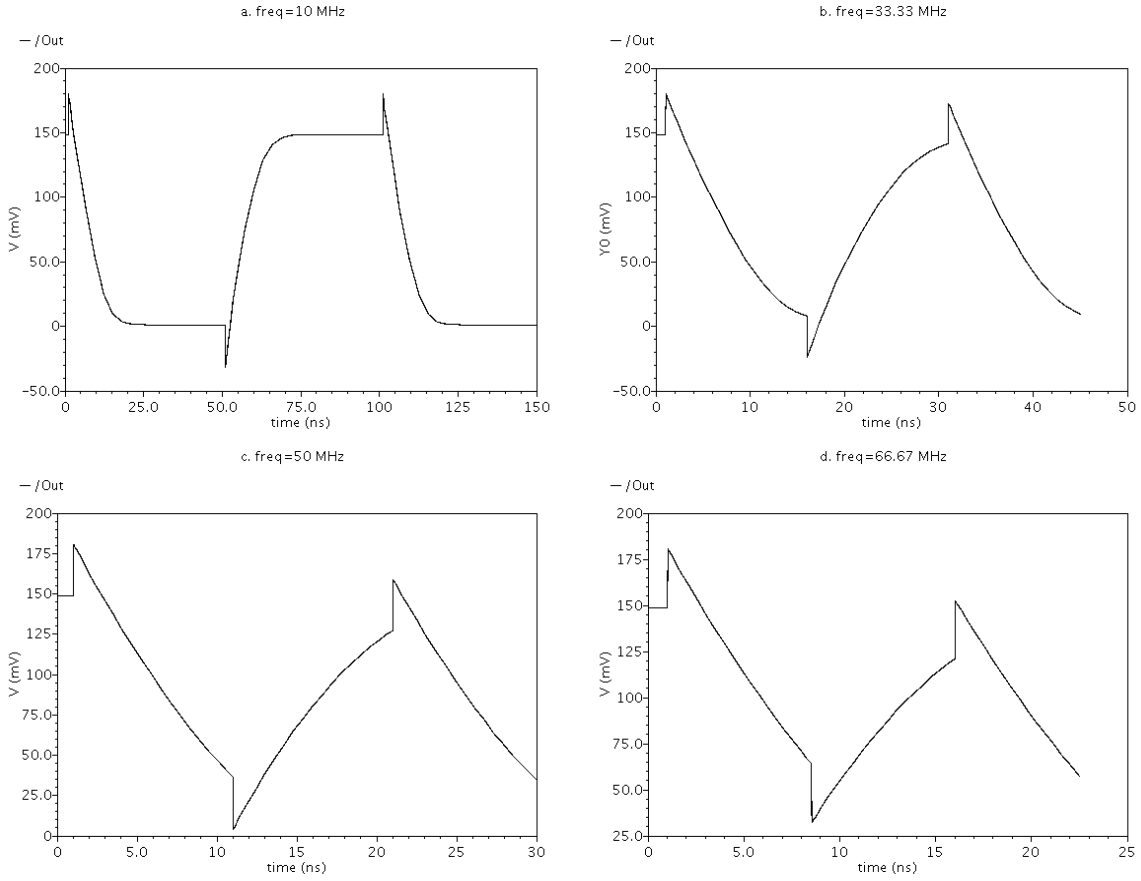


Figure 3.5: Inverter output for the supply voltage of 150 mV and frequencies of a) f = 10 MHz, b) f = 33.33 MHz, c) f = 50 MHz and d) f = 66.67 MHz.

3.2.2 NAND Gate Operation in Sub-threshold Region

The architecture of the NAND gate is the same as the static CMOS architecture for strong inversion NAND gate, the only difference being the transistor sizing. In strong inversion, it is better to set the W_P/W_N ratio around two to equalize the low-to-high and high-to-low propagation delays. However, as mentioned in the previous section, this ratio has to be one to achieve the minimum operational voltage.

Table 3.1 summarizes the characteristics of the NAND gate in the sub-threshold region. The minimum V_{DD} at the frequency of 10 MHz is 100 mV, and the maximum frequencies of operation at voltages of 200 mV and 150 mV are 125 MHz and 40 MHz, respectively. The drastic decrease in speed with the reduction of the supply voltage can be observed in these results.

Minimum supply voltage @ 10 MHz	100 mV
Maximum frequency @ 200 mV	125 MHz
Maximum frequency @ 150 mV	40 MHz

Table 3.1: Characteristics of a static CMOS NAND gate in sub-threshold.

3.2.3 NOR Gate Operation in Sub-threshold Region

Similar to a NAND gate, the architecture of a NOR gate in sub-threshold region is the same as a static CMOS NOR gate in strong inversion, transistor sizing being the only difference. The ratio of PMOS to NMOS width is minimized to obtain the minimum operational voltage.

Table 3.2 summarizes the characteristics of a NOR gate in sub-threshold region. Maximum frequencies at 200 mV and 150 mV remain the same as for a NAND gate. However, the minimum supply voltage is a bit higher than for a NAND gate and occurs at 120 mV.

Minimum supply voltage @ 10 MHz	120 mV
Maximum frequency @ 200 mV	125 MHz
Maximum frequency @ 150 mV	40 MHz

Table 3.2: Characteristics of a static CMOS NOR gate in sub-threshold.

3.2.4 XOR Gate Operation in Sub-threshold Region

The XOR architecture used in this research is shown in Figure 3.6. Eight transistors are used to form the XOR gate. Following the aforementioned rule for W_P/W_N, this ratio is set to minimum. Table 3.3 provides an overview of the XOR gate characteristics. Since this gate includes more transistors than the previous two gates and has a more complex architecture, the output voltage swing starts to drop sooner. The minimum supply voltage at which the 10% - 90% output swing is achieved is 130 mV. The maximum frequency at supply voltage of 200 mV is 90 MHz, which decreases to 40 MHz for a supply voltage of 150 mV.

Minimum supply voltage @ 10 MHz	130 mV
Maximum frequency @ 200 mV	90 MHz
Maximum frequency @ 150 mV	40 MHz

Table 3.3: Characteristics of a static CMOS XOR gate in sub-threshold.

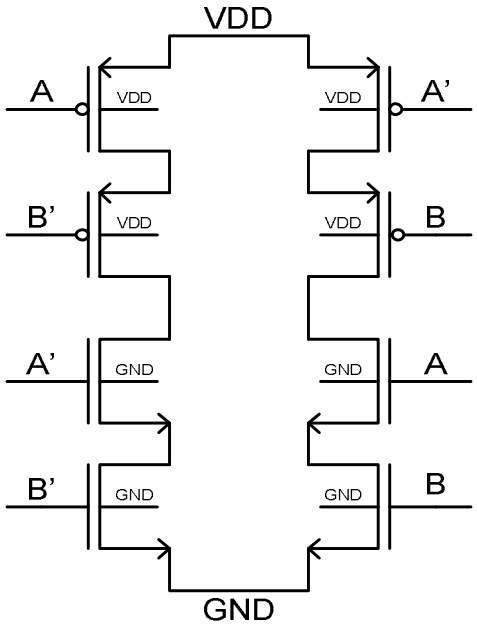


Figure 3.6: Static CMOS XOR architecture.

3.2.5 Parallel XORs

Since XOR is one of the most important components of a cryptographic hardware, further investigations on this specific gate are provided. In addition to a single XOR, another architecture which is used in this research for side channel analysis is a set of eight parallel XOR gates that takes an 8-bit input and produces an 8-bit output using an 8-bit key. This architecture is shown in Figure 3.7. Differential and correlation analysis of this gate is presented in Chapter 5.

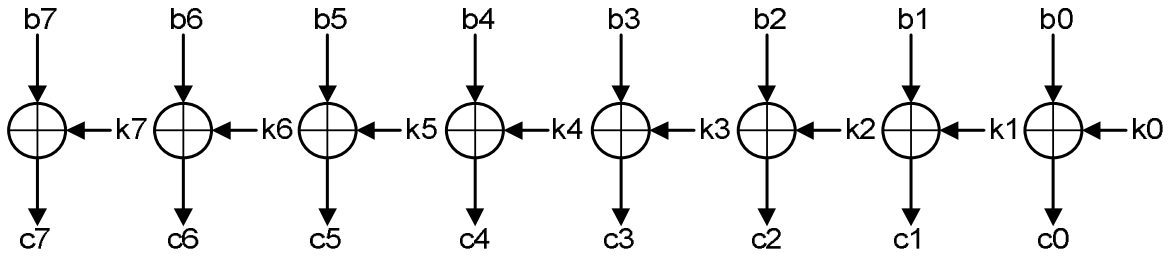


Figure 3.7: Parallel XORs architecture.

3.3 RTL Design

Designing a complete crypto-processor for encryption algorithms, such as AES, at the transistor level that includes thousands of transistors requires an enormous effort. Computer Aided Design (CAD) tools have reduced both the design effort and time required for large designs. The design flow starts by describing the architecture using an HDL language, such as Verilog HDL. After simulating and verifying the functionality of the design, RTL design is synthesized to a gate level design using a logic-synthesis tool like Synopsys Design Compiler. The logic-synthesis tool utilizes a standard cell library to synthesize an RTL circuit into a gate level circuit and also provides timing information. The next step is post-synthesis simulation and timing analysis. Depending on the application, power measurements can be performed using a switching activity file and the gate level netlist. The last step involves placement, routing and post-layout simulation.

Sub-threshold digital design flow, including the details on the tools, is discussed in section 3.3.1, followed by a discussion of the performance of standard cell libraries at a sub-threshold voltage in section 3.3.2. AES and S-Box architectures used in this research are explained in last two sections.

3.3.1 Sub-threshold Digital Design Flow

The digital design flow of a sub-threshold design is the same as the one for strong inversion. However, standard cell libraries used to synthesize RTL into gate level require some modifications. Figure 3.8 presents an overview of the sub-threshold digital design flow used in this research.

The major difference of this flow with the strong inversion design flow is in the left branch of Figure 3.8. As standard cell libraries are characterized for a specific voltage which is in the strong inversion region, the operation voltage of these cells needs to be changed. The Cadence Encounter Library Characterizer (ELC) [45] is a tool that characterizes a standard cell library for user-defined setups. ELC inputs are:

- A SPICE-format sub-circuit file which includes all the details of transistor devices, resistances and capacitances of each standard cell.
- A SPICE-format device model file that describes the transistor device models.
- A setup file which defines device parameters for different process corners, supply voltages, temperatures, input slew rates and output loads.
- A configuration file (elccfg) which contains environment variables or setup directives which will be used during the run.

ELC receives the mentioned inputs and analyzes the SPICE-format sub-circuit, recognizes the function or logic structure and generates the function or logic model. It then defines the characterization environment by user-defined parameters in the setup file. In the last step, it invokes and executes SPICE, summarizes the results, and generates an ALF file which can be converted to library formats. Following are the commands to perform the above steps. Additional details can be found in [46]. All italic names in this chapter represent user files.

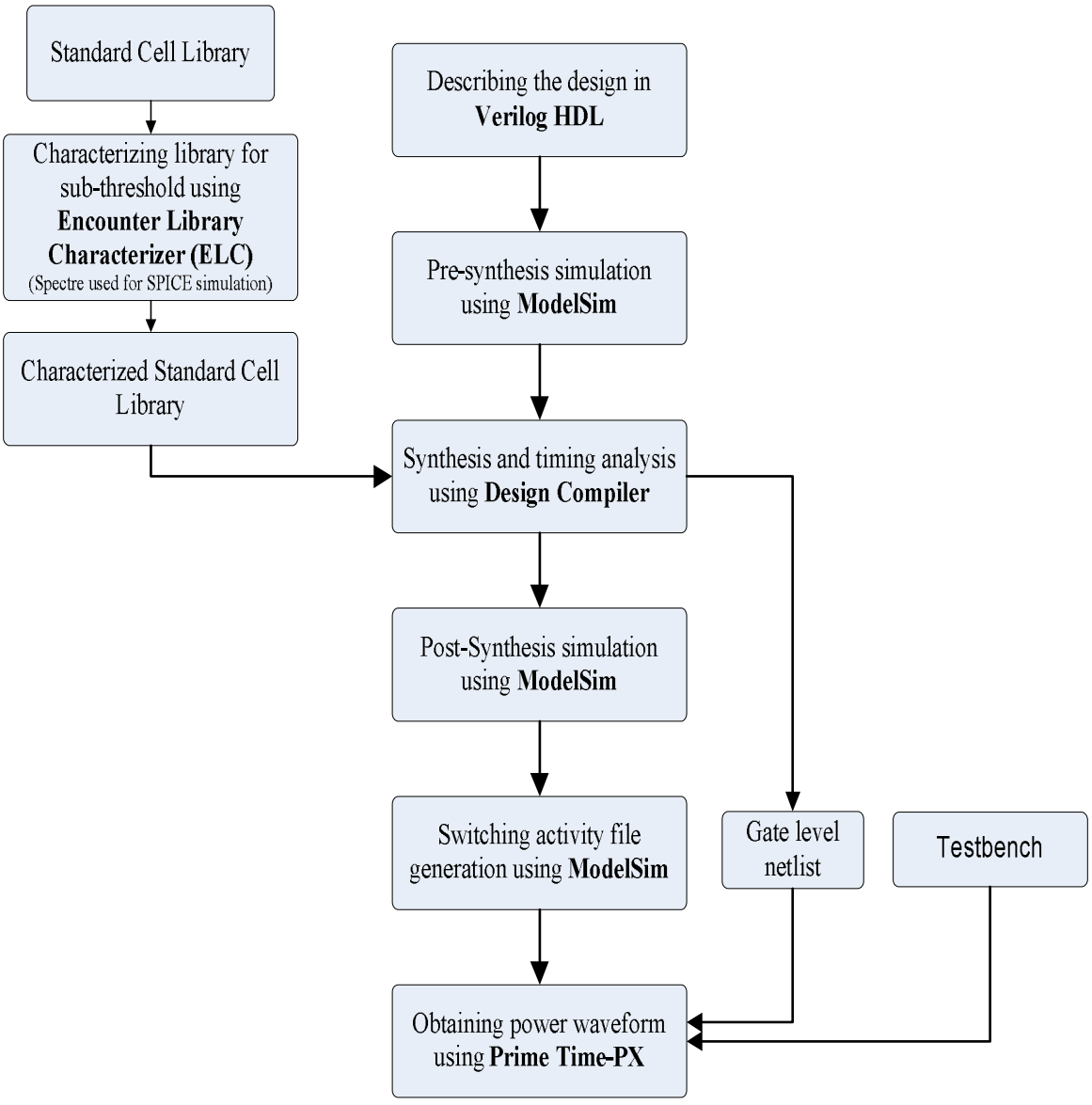


Figure 3.8: Sub-threshold RTL design flow.

```
db_open foo                                              #opens a database
db_prepare -force    #creates environment for running SPICE simulation
db_gate                    #recognizes the design and creates gate file
db_spice -s spectre -p typical -keep_log          #SPICE simulation
db_output -r foo.alf.rep -alf foo.alf -p subth    #outputs alf file
db_verilog -r foo.v            #generates a Verilog logic description
db_close
alf2lib -alf foo.alf -lib subth.lib      #converts .alf to .lib file
```

The output library database contains the timing, power, and noise model for each of the standard cells, based on the parameters defined in the setup file. Modification of the standard cell library occurs in this first step, and the setup file used has to be modified for the sub-threshold region. Next, the process corner, supply voltage, input slew rate and output load have to be defined for sub-threshold voltage.

Synopsys Design Compiler (DC) [47] is the logic-synthesis tool utilized in this research. DC's inputs are standard cell library and behavorial description of the design in Verilog or VHDL. As will be seen in section 3.3.2, some cells in standard cell libraries do not exhibit acceptable performance in the sub-threshold region; hence, these cells need either to be avoided or modified to be able to work in a circuit operating in sub-threshold. The Design Compiler converts a behavioral Verilog file to a structural file using determined standard cells. Design Compiler is capable of using only some specified cells or avoid using some specified cells in conversion as long as the circuit is achievable with allowed cells.

In order to forbid the Design Compiler from using *Cell1* and *Cell2*, we can use the following commands.

```
set_dont_use my_lib/Cell1
set_dont_use my_lib/Cell2
```

In order to authorize the Design Compiler to only use *Cell3* and *Cell4*, we can use following commands.

```
set_dont_use my_lib/*
remove_attribute my_lib/Cell2 dont_use
remove_attribute my_lib/Cell3 dont_use
```

PrimeTime PX [48] is an add-on to Synopsys PrimeTime that obtains an accurate power dissipation analysis based on the circuit connectivity, switching activity, net capacitance and cell-level power behavior data in the Synopsys database format (.db) library. PrimeTime PX supports two modes of power analysis: the averaged and the time-based. In this research, the latter mode is of interest.

PrimeTime PX reads in the gate level netlist, Synopsys design constraint file, parasitic file, switching activity file and the defined testbench. Then it uses the technology library file to perform

power analysis and generate a power waveform. The power waveform can be used later for correlation power analysis. Following are commands used in PrimeTimw PX to obtain the power waveform.

```
set power_enable_analysis TRUE
set power_analysis_mode time_based

##########################################################################
#
#       link design
##########################################################################
#
set search_path       "search_path1 search_path2 search_path3"
set link_library " * my_lib.db"

read_verilog          gate_level_netlist.v
current_design  top_module
link

##########################################################################
#
#       set transition time / annotate parasitics
##########################################################################
#
read_sdc         sdc_file.sdc
read_parasitics parasitic_file.spef

##########################################################################
#
#       check/update/report timing
##########################################################################
#
check_timing
update_timing
report_timing

##########################################################################
#
#       read switching activity file
##########################################################################
#
read_vcd "switching_activity.vcd" -strip_path "testbench/DUT"

##########################################################################
#
```

```
#       check/update/report power
#########################################################################
#
check_power
set_power_analysis_options -waveform_format out -waveform_output
     Output_Waverform
update_power
report_power
```

Mentor Modelsim [49] is used to generate a switching activity file. This file contains the switching activities on all nodes through the circuit and is a key file in the power estimation process. Following are commands used to generate this file. These commands should be called after a simulation is started in Modelsim.

```
vcd file   switching_activity.vcd
vcd add -r       testbench/DUT/*
run        Run_Time
vcd2wlf -nocase switching_activity.vcd     switching_activity.wlf
wlf2vcd -o Output_switching_activity.vcd switching_activity.vcd
```

The last two commands are necessary due to incompatibilities between Synopsys and Mentor Graphics in naming signals. In order to resolve this issue, we can convert the vcd file to a wlf file and then convert the wlf file back to the vcd format.

3.3.2 Standard Cell Library Performance in Sub-threshold

As mentioned earlier, standard cell libraries are not specifically designed to operate in the sub-threshold region. The characterization of libraries for sub-threshold was the first required modification described. Nevertheless, not all cells in a library perform well in sub-threshold. The authors in [13] evaluated the performance of a 0.18 µm standard cell library in a sub-threshold operation, as shown in Figure 3.9. It shows the lowest operational V_{DD} for various cells in the library at typical, fast-slow and slow-fast corners.

If all cells in a standard cell library function properly at sub-threshold, the functionality of a circuit synthesized using this library is guaranteed [13]. Thus, a cell that fails to operate at that region has to be isolated from the library used by the logic-synthesis tool to produce a functional circuit for sub-threshold. This is the second required modification on standard cell libraries. Cells that exhibit the worst performance below the threshold voltage of a transistor are logic gates with a stack of series devices (e.g. And/Or/Invert (AOI)), logic gates with multiple devices in parallel, and flip-flops [13]. These cells have to be modified to be able to operate in sub-threshold region or be isolated from the list of synthesizable cells.

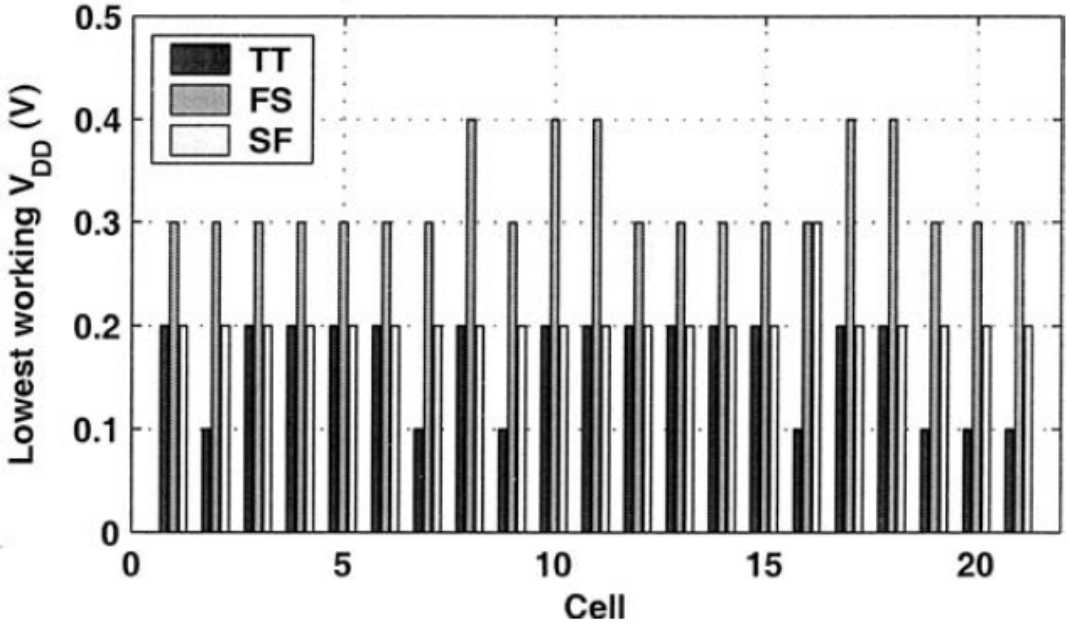


Figure 3.9: Standard cell functionality in synthesized FIR filter using normal cell selection over process corners (simulation) (© 2005 IEEE) [13].

3.3.3 AES Crypto-Processor

An Advanced Encryption Standard [11] core used in this research is a compact 8-bit AES core proposed in [50]. The high-level architecture of this core is shown in Figure 3.10. All data paths are 8-bit wide. This core is implemented in Verilog HDL and synthesized using 65 nm TSMC standard cell library.

The interface unit provides the requisite handshake signals to enable the core to perform as a co-processor. The control unit is responsible for producing control signals and clocking all registers at the proper time. The ShiftRows unit is a series of shift registers whose inputs are controlled by multiplexers. The MixColumn unit contains a few registers whose input is the XOR of the previous register and a multiple of the input. ShiftRows and MixColumn operate on one row and one column of data, respectively, which is 32 bits; however, the data path width of this architecture is 8 bit. Employing multiple-shift registers and a parallel to a serial converter introduces multiple layers of pipeline that enable the processor to receive 8-bit inputs at a time and to process them.

The most complex block of this core is SubByte, which is also called S-Box. In order to reduce the area and achieve efficiency, elements of $GF(2^8)$ are mapped into a smaller field of $GF(2^4)^2$ and further mapped to $GF(2^2)^2)^2$. The XOR gate is the most frequently used gate in this block and in other blocks of this design also studied individually in this work.

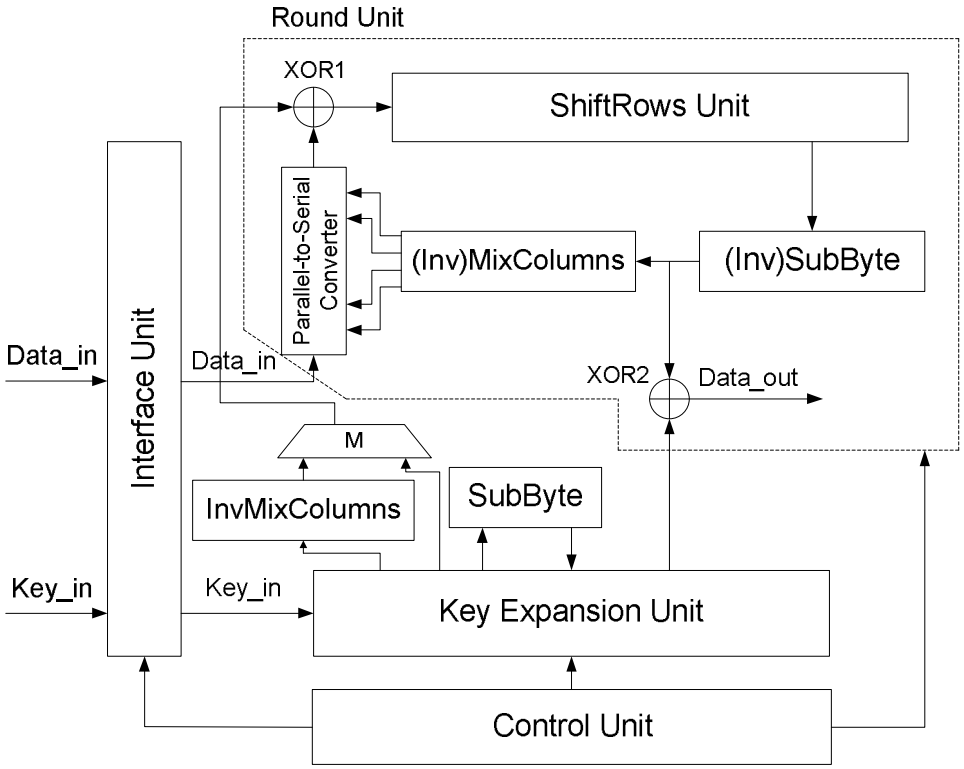


Figure 3.10: High-level architecture of the AES crypto-processor [50].

Figure 3.11 presents the data path of the AES core. The core is able either to encrypt or decrypt the input plaintext. Pipelining enables the core to receive 128-bit plaintext and key in 16 consecutive clock cycles through the input ports, data_in and key_in. After 10 rounds of AES encryption, which takes 160 clock cycles, the ciphertext comes out of the data_out port in 16 cycles. Thus, the total encryption period is 176 cycles. Moreover, due to the 16 levels of pipeline, a new plaintext and key can be uploaded into the cores in the last 16 cycles while the previous ciphertext is arriving at the output. This feature improves the performance of the design in long runs and decreases the number of total cycles from 176 to 160.

During encryption, the key expands parallel to the encryption process. However, since AES is a symmetric key algorithm and the same key is used for decryption, the first 160 clock cycles of the decryption process is devoted to the key expansion. Once the fist byte of the expanded decryption key is produced, decryption starts and takes 176 clock cycles. Hence, the total decryption happens in 336 clock cycles as opposed to 176 cycles of encryption. More detailed information about each block can be found in [50]. The CPA attack, which will be described in Chapter 4, occurs at the data_out node in the first round of the AES algorithm.

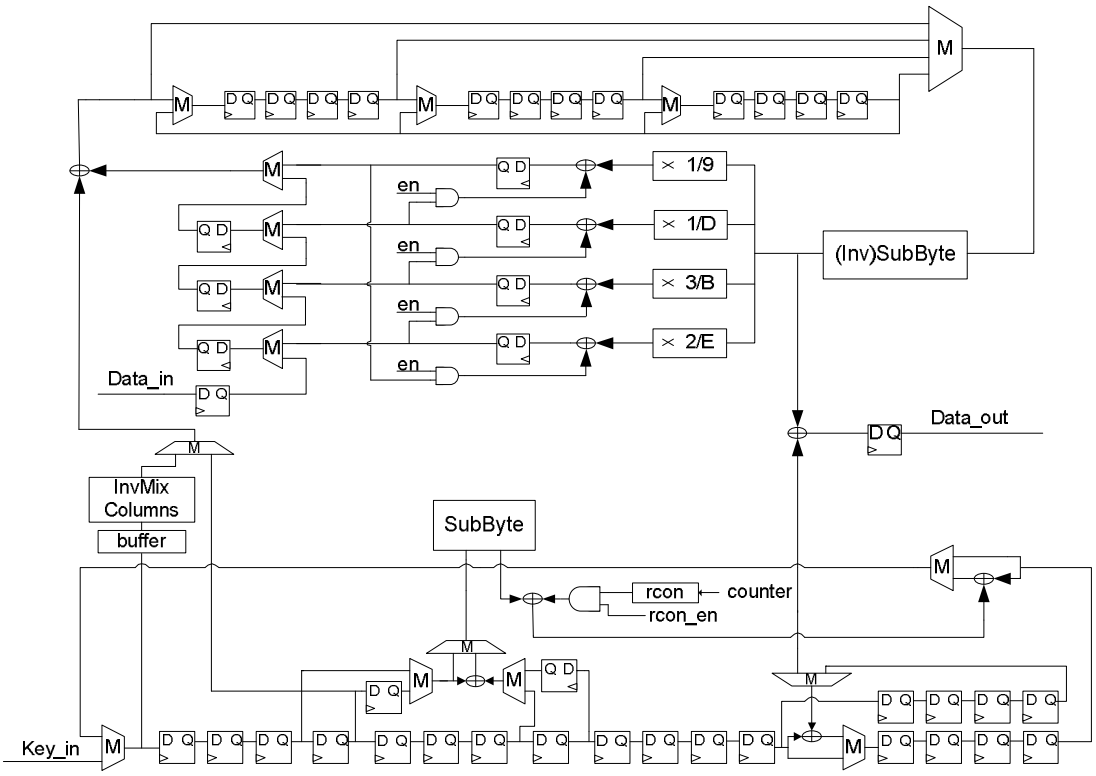


Figure 3.11: AES core data path [50].

3.3.4 S-Box Block

The substitution block of AES is also studied individually. The architecture is shown in Figure 3.12. In this architecture, the 8-bit plaintext is first exclusive-ored with an 8-bit key and the result sent to an S-Box. The output of the S-Box is latched by the register, and the attack occurs at the output of the register.

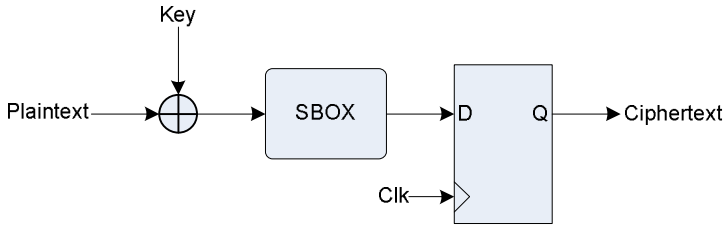


Figure 3.12: S-Box test architecture for attack.

Chapter 4
Side Channel Information Leakage Measurements and Analysis

In order to perform a successful side channel attack on a system, the attacker has to be able to exploit information about the secret key through side channel measurements and recover the secret key from the extracted information [51]. Power consumption is used in this research as the system characteristic that leaks information. The instantaneous power consumption of the design under study is assumed to be known by the attacker. The attacker's job is to recover the secret key with possession of power traces and knowledge about the encryption algorithm.

This chapter defines mathematically and illustrates the measures introduced briefly in section 2.2.3, to evaluate the side channel information leakage of circuits proposed in Chapter 3. Section 4.1 explains difference of mean energies (DME), highlighting the difference between power traces that process 1 compared to those that process 0. Frequency of observation is explained in section 4.2 to demonstrate how the security level of a circuit can be visualized in histograms. The chapter continues by providing more details about normalized energy deviation (NED) and normalized standard deviation (NSD), in section 4.3. As mentioned earlier these measures quantize the dispersion of average power consumption values of a cryptographic device for various data transitions. The last measure is correlation power analysis (CPA), which is an extension of the differential power analysis introduced by Kocher et al. [8]. This is the most effective measure that investigates the actual possibility of finding the secret key from the power trace using correlation analysis. Section 4.4 describes the usage of CPA in this research.

4.1 Difference of Mean Energies (DME)

As mentioned earlier, power consumed in a circuit changes with respect to inputs due to variant current paths and output capacitance loads. This fact forms the basis of DME measure.

Suppose that n different plaintexts are applied to the system shown in Figure 4.1 and their corresponding power traces, E_0 to E_n, are recorded. The attacker chooses a partitioning function, f, in order to separate the recorded traces into two groups. The partitioning function might be the lsb of some expected intermediate data. However, an attacker does not have access to the key and the intermediate data value is calculated based upon a guess of the key. The parameters T_0 and T_1 shown in Expression 4.1 are generated using the partitioning function f. Values f_0 to f_n are the outputs of function f.

$$T_0 = \{E_i | f_i = 0\} \tag{4.1}$$

$$T_1 = \{E_i | f_i = 1\}$$

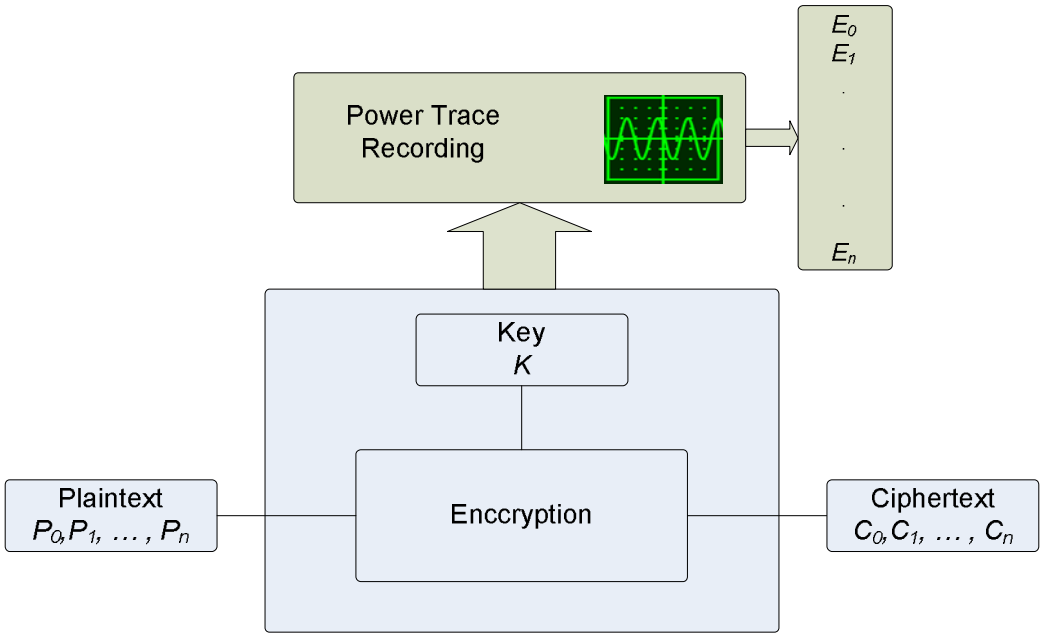


Figure 4.1: Crypto-system with fixed key and *n* different plaintexts.

Now that two sets of power traces, T_0 and T_1, are generated, the mean of each set has to be calculated. M_0 and M_1 are the means of T_0 and T_1, respectively. DME represented in Equation 4.2 is the difference between M_0 and M_1. Please note that DME itself is a trace over time.

$$DME = | M_0 - M_1 | \tag{4.2}$$

For an ideally secure system, a DME signal must be always 0. The highest peak in this signal represents the point in time where the system's behavior is most correlated to the key. Hence, having a group of DME signals obtained by various key guesses and consequently multiple partitioning function values, the DME signal with the highest peak corresponds to the correct key.

In chapter 5, DME is used to evaluate the XOR gate and parallel XOR architecture (from Chapter 3). In the case of XOR gate, power traces for input transitions of 0 to 1 and 1 to 0, having a fixed key of 0, are measured. The same experiment is repeated with a fixed key of 1. Obtained power traces from two cases of sub-threshold and strong inversion are used to compare the system correlation on input's type of transition. For the parallel XORs architecture, 8-bit input varies from 0 to 255 and power traces are recorded having K_0 (lsb of the key) equals to 0 once and again with K_0 equal to 1. The mean of the traces in each group is measured to find M_0 and M_1 signals. Comparing the DME signal of sub-threshold against strong inversion is one way to compare their correlation to the secret key and results are presented in Chapter 5.

4.2 Frequency of Observation

Frequency of observation is a visual measure that helps to observe the dispersion of average power consumption values generated by different inputs combinations. This measure is used in some papers, such as [35]. This measure is applied on three basic gates in chapter 5, NAND, NOR and XOR, in sub-threshold, strong inversion, SABL and WDDL logic schemes. They are provided with all possible combinations of inputs transitions that produce 0 to 1, 1 to 1, 1 to 0, and 0 to 0 transitions. For instance, for a 0 to 1 transition in a 2-input gate, there are four combinations of inputs transitions.

Assuming each transition happens in one clock cycle, power consumption is averaged over each cycle. To obtain the frequency of observation histogram, averaged power consumption values corresponding to all possible combinations of inputs are obtained and the frequency of occurrence of the obtained values are plotted. Figure 4.2 demonstrates the frequency of observation for two systems, one completely secure and the other completely insecure. The horizontal axis shows the averaged power value and the vertical axis represents frequency of observation. It is assumed that the number of possible combinations of inputs is 10. For the insecure system, the averaged power value for each transition is different from other transitions, and hence the frequency of each averaged power value is one; such a system is easily breakable by analyzing power consumption. On the other hand, if a system consumes the same value of power in all clock cycles, the power consumption of this system does not reveal any information about the transition occurring inside the system and makes it impossible to break it. The system shown in grey in Figure 4.2 is an example of a secure system.

Figure 4.3 shows frequency of observation for two real systems (to be detailed in Chapter 5). One can see that, for System 1, averaged power values are spread over a smaller range compared to System 2. Thus, power consumption of System 2 is more correlated to its input transitions, and observing its power consumption reveals more information about the secret key.

Therefore, while frequency of observation is not a measure to accurately evaluate a system's vulnerability against power analysis, it does provide a simple and effective visual tool to estimate the behavior of a system with a limited number of input transition combinations. This measure is effective for comparison of basic gates.

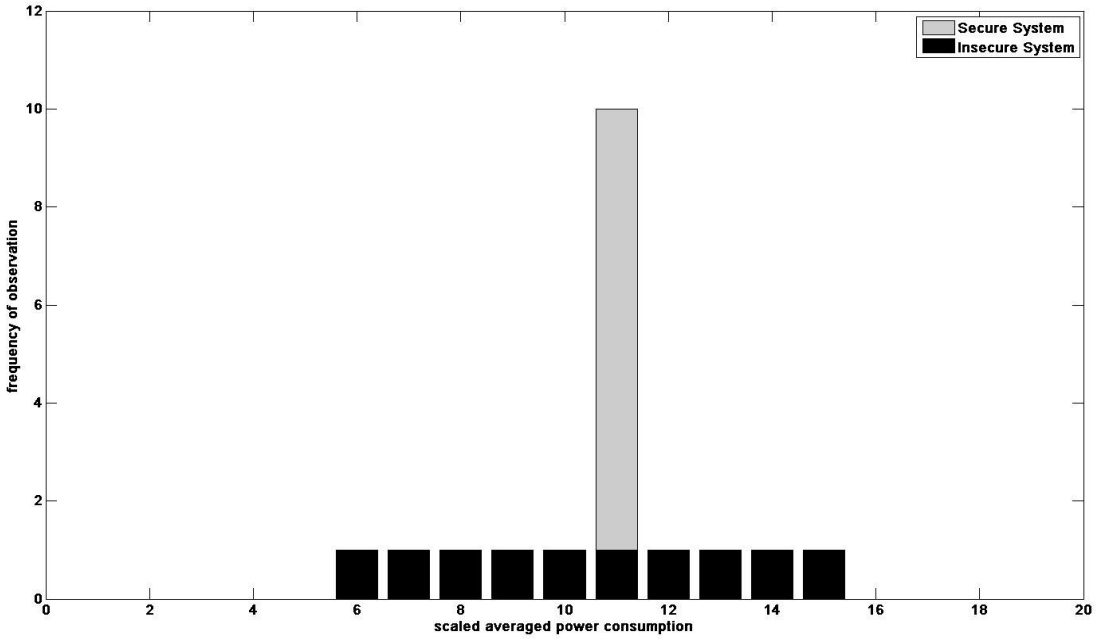


Figure 4.2: Frequency of observation, secure system vs. insecure system.

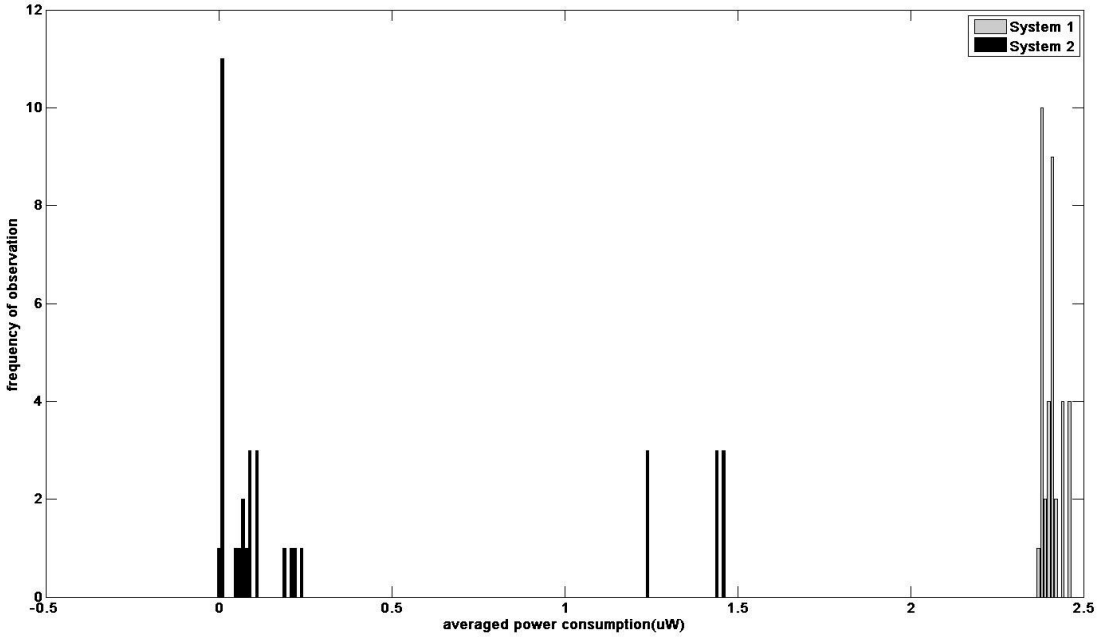


Figure 4.3: Comparison between frequencies of observation of two real systems.

4.3 Normalized Energy Deviation (NED) and Normalized Standard Deviation (NSD)

Formulating frequency of observation leads to the definition of normalized energy deviation and normalized standard deviation. As in the previous section, these two measures take averaged power consumption per cycle as input and indicate the variance of the power values. NED is defined in Formula 4.3.

$$NED = \frac{Max(energy/cycle) - Min(energy/cycle)}{Max(energy/cycle)} \tag{4.3}$$

NED produces a value between 0 and 1. The smaller the variation in power values, the smaller the value of NED. This makes the attack more complex and required more measurements. A narrow range of power values in the frequency of the observation diagram results in a small NED value. Although NED has been used in many papers thus far, it may result in an unfair comparison. Figure 4.4 presents two systems with the same value of NED. However System 1 can be considered a more secure system, since its power values are located in a narrower range.

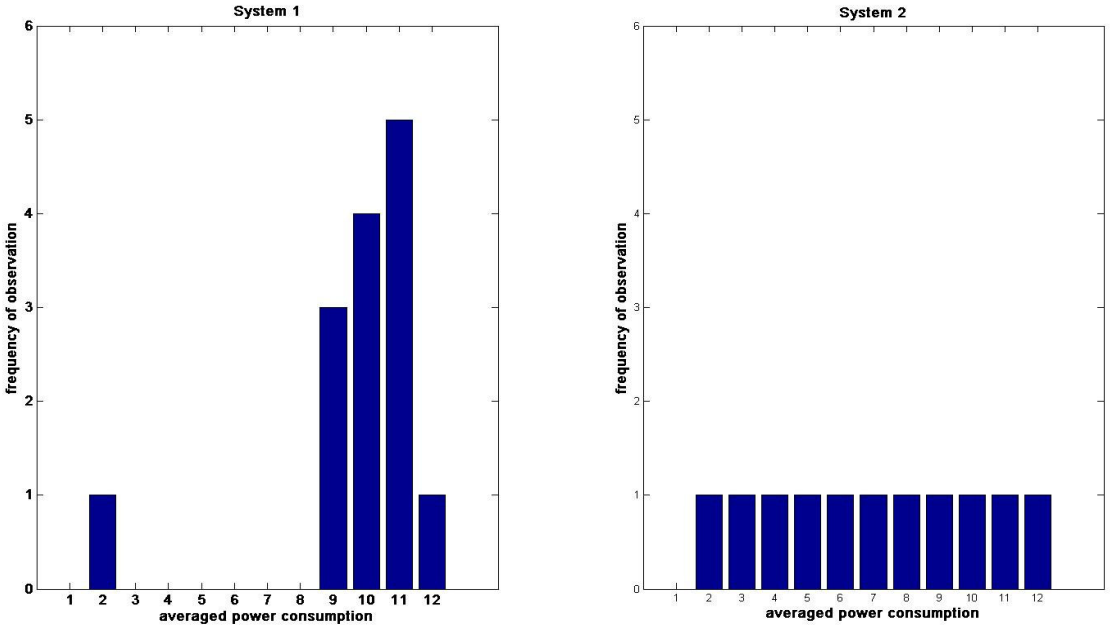


Figure 4.4: Systems with same NED value but different levels of security.

In order to address issues like the one in Figure 4.4, another measure, called NSD, is defined in Formula 4.4.

$$NSD = \frac{Standard\ Deviation}{Mean} \tag{4.4}$$

NSD measures how averaged power values are located around their mean. If they are spread widely around the mean, NSD is larger, and if they are close to the mean, NSD is smaller. In the latter instance, the system is more secure. The NED value for both systems in Figure 4.4 is 0.83; however, the NSD value is 0.25 for System 1 and 0.47 for System 2, signaling the higher security level of System 1.

NSD is not a completely accurate measure. Figure 4.5 shows two systems with the same level of power variation and thus the same level of security; however, NSD for System 1 is about half of the NSD for System 2. The reason is that while both systems have almost the same amount of standard deviation, the mean for System 1 is about twice that of System 2. NED is equal for both systems, so it provides a more precise measure in this situation.

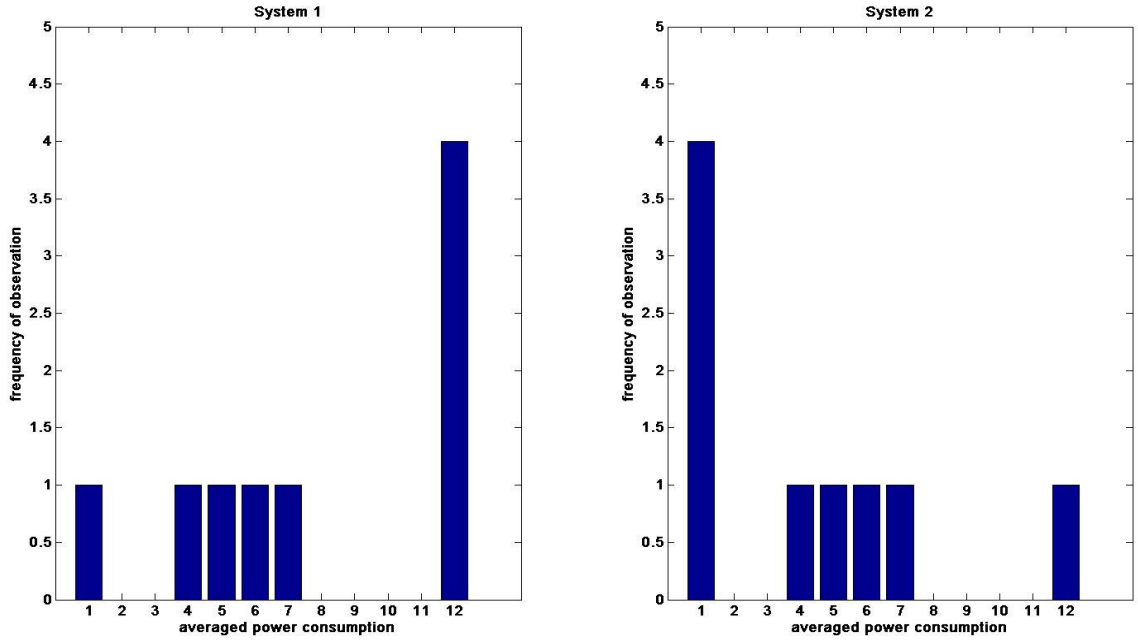


Figure 4.5: Systems with same level of security but different values of NSD.

As a result, NED and NSD are two simple but effective measures that provide security evaluation of systems with limited number of input transition combinations. Although they have been used in various papers, they are not completely accurate and they must be used carefully. In this research, we have used these two measures along with frequency of observation to evaluate three basic gates, NAND, NOR and XOR, in sub-threshold, strong inversion, SABL, and WDDL logic schemes.

4.4 Correlation Power Analysis (CPA)

Correlation power analysis is the most accurate measure among the presented measures. It not only provides correlation coefficients which can be used to compare various systems, but also reveals the secret key. We have used CPA to attack the S-Box block of the AES algorithm and the AES architecture itself. We have also used CPA to study the correlation between key and power consumption in the parallel XORs architecture shown in Figure 3.7.

The CPA process is built in three stages: 1. Power consumption matrix acquisition; 2. Predicted power consumption matrix generation; and 3. Correlation matrix generation. Each step is described in detail in the following sections.

4.4.1 Power Consumption Matrix Acquiring

This stage includes writing a suitable testbench and running a simulation on the design under test to acquire all power traces corresponding to a fixed key and all possible plaintexts.

An S-Box block under attack is shown in Figure 3.12. The key is fixed and the plaintext takes all possible values from 0 to 255. The testbench is written in such a way that the ciphertext C_i, corresponding to the plaintext P_i, is generated in one clock cycle. In the next clock cycle, output is set to 0. Figure 4.6 provides an overview of this testbench.

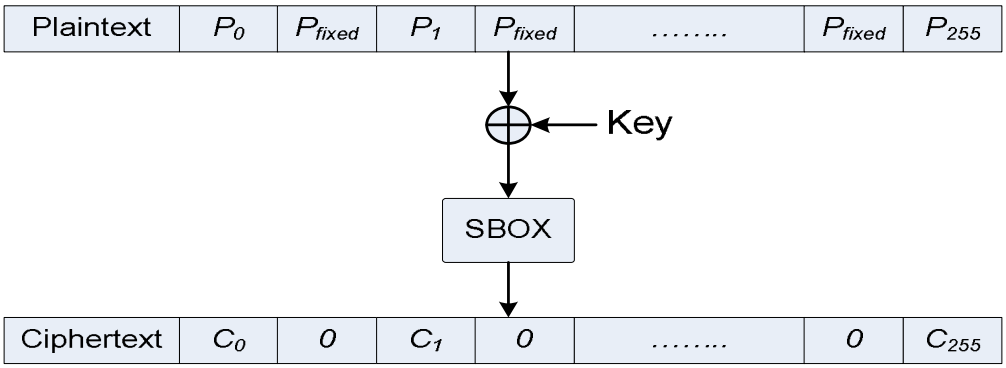


Figure 4.6: S-Box testbench for generating power traces.

The next step is producing power traces using the procedures described in Chapter 3. Generated power trace is a long file that has to be processed in MATLAB to extract 256 traces corresponding to 256 different plaintexts. The final matrix is shown in Figure 4.7. Each row of this matrix is a power trace for one fixed plaintext, and each column shows instantaneous power consumption at time t_i for all possible plaintexts.

P_0	P_{0,t_0}	P_{0,t_1}	. . .	P_{0,t_T}
P_1	P_{1,t_0}	P_{1,t_1}	. . .	P_{1,t_T}
. . .	. . .			
P_{255}	P_{255,t_0}	P_{255,t_1}	. . .	P_{255,t_T}
	t_0	t_1	. . .	t_T

Figure 4.7: Power traces matrix generated from simulation.

The testbench for AES is different, because an encryption process takes 175 clock cycles rather than just one clock cycle in S-Box. Also, the plaintext and key are both 128-bit wide instead of 8-bit wide. The key in the AES testbench is fixed (the same as the S-Box testbench) but instead of varying the plaintext over the whole possible range (which includes 2^{128} different values) only the lowest 8 bits of the plaintext vary from 0 to 255. Each plaintext remains unchanged for 175 clock cycles to let the output ciphertext appears, after which it proceeds to the next value. Thus, 256 encryption operations occur, consecutively. Applying PrimeTime PX to AES using the mentioned testbench generates a very long power trace that requires careful manipulation in MATLAB to generate the power trace matrix. The matrix is the same as that in Figure 4.7; however, total time, *T*, for AES is much longer than the one for S-Box.

The same methodology is used for parallel XORs architecture to acquire the power traces for all 256 possible plaintexts. Despite S-Box and AES, simulation and power measurements are performed at the transistor level in Cadence Virtuoso Analog Environment. The testbench runs the architecture with 256 plaintexts consecutively, which takes 256 clock cycles to complete. As with S-Box and AES, the rest of the procedure manipulates the long power trace obtained from Cadence to create a power traces matrix the same as the one in Figure 4.7.

4.4.2 Predicted Power Consumption Matrix Generation

In this stage, one needs to set up the attack point and generate a predicted power consumption matrix based on either the Hamming distance model or the Hamming weight model. As mentioned in Chapter 3, the attack point in AES architecture is at the Data_out node in the first round. The output of the register is also chosen for the attack point in the S-Box block. Hence, the Hamming weight of

the XORed value of plaintext and key can be used to generate the desired matrix. The matrix is a 256×256 matrix, for which each element can be calculated using Expression 4.5.

$$PredictedPower_{i,j} = Hamming\_Weight(P_i + K_j) \tag{4.5}$$

where P_i is the i^{th} plaintext and K_j is the j^{th} key. Both P_i and K_j are assumed to be 8-bit. Therefore, each element of the matrix can take a value between 0 and 8. The models used are Hamming weight, however, since the testbench of S-Box generates a 0 between each pair of plaintext, the Hamming distance model also has the same formula as the one in Expression 4.5.

The attack point in the parallel XOR architecture is the output of XORs. The model used for this architecture is the Hamming distance one. Expression 4.6 presents the formula used to generate the predicted power matrix.

$$PredictedPower_{i,j} = Hamming\_Distance(P_i + K_j\ ,\ P_{i-1} + K_c) \tag{4.6}$$

where K_c is the correct key. It is assumed that the correct key is known because the purpose of this case is not revealing the secret key and we are interested in studying the correlation between the key and power consumption using the proposed model.

4.4.3 Correlation Matrix Generation

In this stage, the correlation between the power traces matrix obtained by simulation and the predicted power matrix generated in MATLAB will be calculated. The output correlation matrix's size is 256×T, with each row corresponding to a correlation trace over time for a key guess. The MATLAB code to generate this matrix is as follows.

```
for i=1:T
    for j=1:256
      Correlation=corrcoef(power_traces(:,i),predicted_power(:,j));
      Corr_trace(j,i)=Correlation(1,2);
    end
end
```

The inner loop correlates column *i* of the power_trace matrix with all 256 columns of the predicted_power matrix and saves them in column *i* of the Corr_trace matrix. Since columns of power_trace represent time and columns of predicted_power represent keys, column *i* of Corr_trace contains the correlation coefficients of all keys at time *i*.

After 256 iterations, when one time slot is completed in the inner loop, the outer loop moves to the next time slot in the power_trace matrix. This operation repeats for T iterations. At the end, each row in the Corr_trace matrix represents the correlation coefficients of a key over time. Figure 4.8 illustrates the process showing one iteration of the outer loop at time 0.

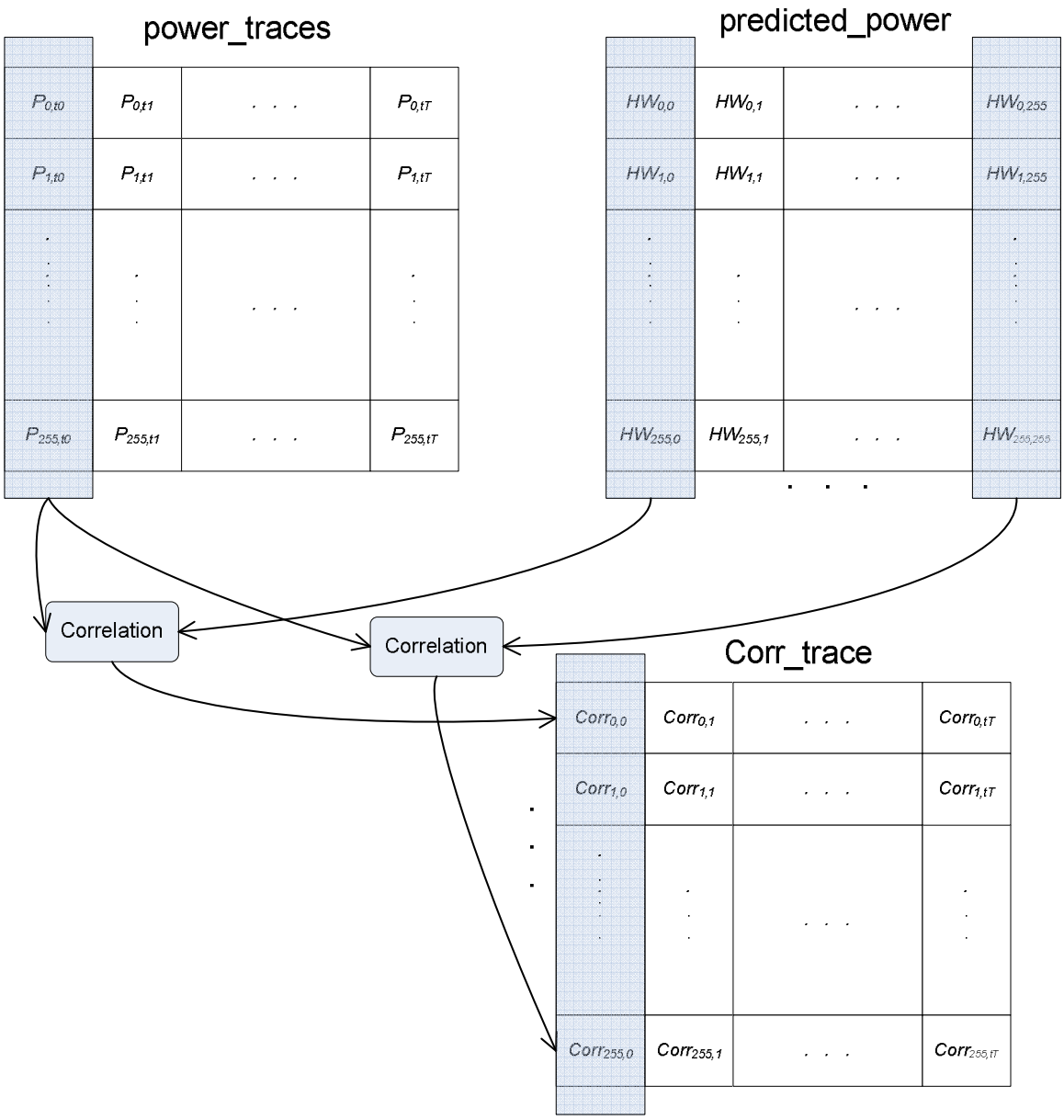


Figure 4.8: One iteration of outer loop to calculate the correlation matrix.

Once the correlation matrix is completed, we have 256 correlation traces. In order to find the correct key, the highest peak of each trace must be determined. If the maximum peak value occurs for the key *k*, it means that *k* has the highest correlation with obtained power traces; hence, the secret key of the system is *k*.

Figure 4.9 demonstrates an example of a correlation trace for a correct key compared to one for a wrong key. Electromagnetic radiation of a synthesized ASIC during a DES operation is measured in [27] and led to the traces in Figure 4.9. We can observe that the maximum peak occurs in a trace that corresponds to the correct key.

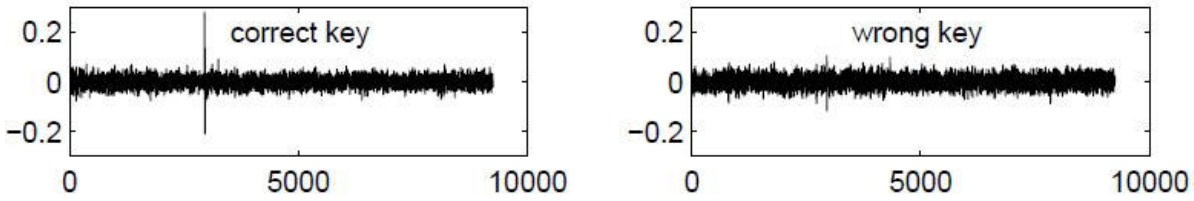


Figure 4.9: Correlation traces for correct key and an example of a wrong key in a CPA attack on a DES operation [27].

Chapter 5
Simulation Results

This chapter evaluates the architectures discussed in Chapter 3 based on the measures introduced in Chapter 4 and presents the simulation results. It also compares side channel information leakage in different logic schemes including SABL, WDDL, strong inversion and sub-threshold. The chapter opens by analysis of the power consumption behavior of an XOR gate in sub-threshold versus strong inversion in transition of output from 0 to 1 and 1 to 0, using the DME measure. This study is the first glance at differences between sub-threshold and strong inversion. In order to further highlight the difference between the behavior of power consumption in sub-threshold and strong inversion confronting various transitions, the parallel XORs architecture is evaluated based on the DME measure from Section 2.

The chapter continues with comparisons of NAND, NOR and XOR gates in SABL, WDDL, sub-threshold and strong inversion logic schemes. Frequency of observation is the first applied measure that provides a visual overview of the security performance of the various logic schemes. Then, NED and NSD are calculated to create a numeric scale for comparison. In the previous section, utilizing XOR and parallel XORs, instantaneous power consumption was employed; in this section, however, averaged power consumption per clock cycle is used for evaluation.

In the last section, correlation power analysis is performed on parallel XORs, S-Box and AES. Comparison of correlation coefficients between sub-threshold and strong inversion is the area of interest for architectures in the last section.

5.1 XOR Gate Analysis with DME Measure

The power consumption behavior of an XOR gate in sub-threshold region is compared to the same gate in strong inversion in this section. First, power waveforms generated using different transitions are provided. Spikes in the power waveform which contain the main information hidden in the trace can be observed. In the next part, the difference between power traces generated during the transition of data from 0 to 1 and from 1 to 0 with either key value of 0 or 1 is studied. This difference forms the main tool to compare sub-threshold and strong inversion. In the last part of this section, the difference of mean energies measure is again used. However, difference signals in the last part come from the difference between power traces of identical data transitions but with different keys.

5.1.1 Power Waveforms at Sub-threshold and Strong Inversion

The XOR gate studied in this section is a two input gate, with one input called *Key* and the other called *Data*. Apart from minor changes in load capacitance, frequency and supply voltage, testbenches for sub-threshold and strong inversion are similar. The load capacitance is 10 fF in strong

inversion and 0.1 fF at sub-threshold. Frequency of operation is considered 1 GHz for strong inversion and 20 MHz for sub-threshold. The circuit is powered with a 1 V supply voltage in strong inversion and a 200 mV supply voltage at sub-threshold.

The testbench keeps the *Key* fixed at 1 and raises *Data* to 1 from 0. It then drops it to 0 and repeats this operation with the *Key* fixed at 0. The testbench also performs this experiment with the reverse transition of *Data*, i.e. from 1 to 0 and 0 to 1. Figures 5.1 to 5.4 show the power waveforms for strong inversion in the left window and input and output signals in the right windows. Figures 5.5 to 5.8 show the same signals for sub-threshold.

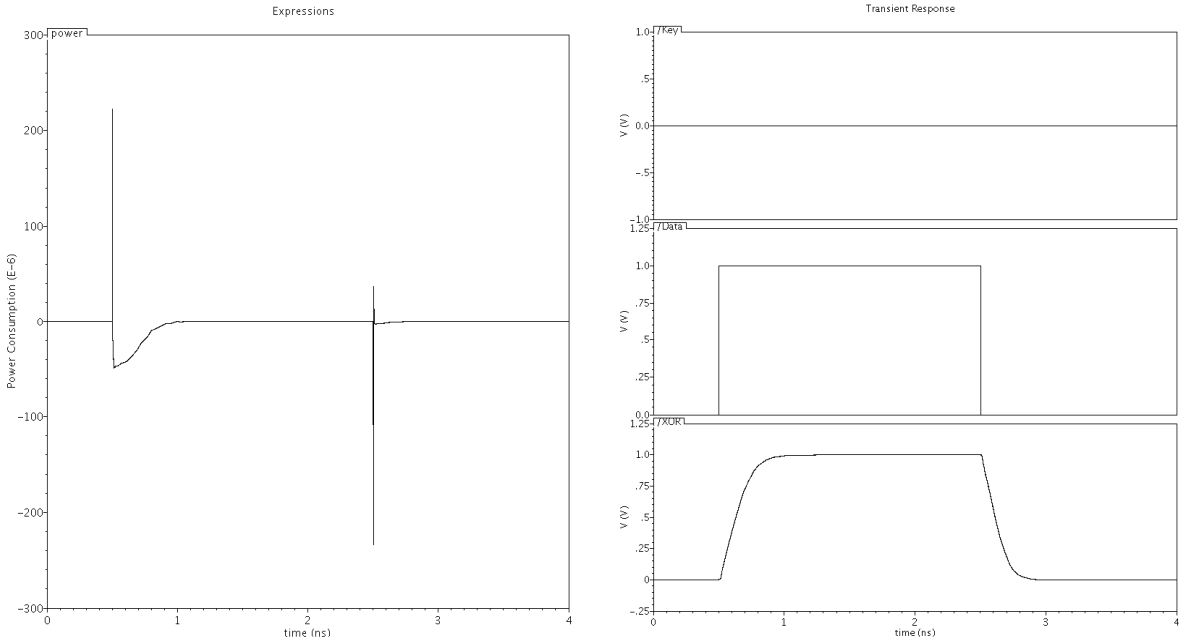


Figure 5.1: Strong inversion power waveform at left for XOR circuit, with plots on the right (top to bottom) of , key, input data and output data for *Key=0* and *Data=0->1->0*.

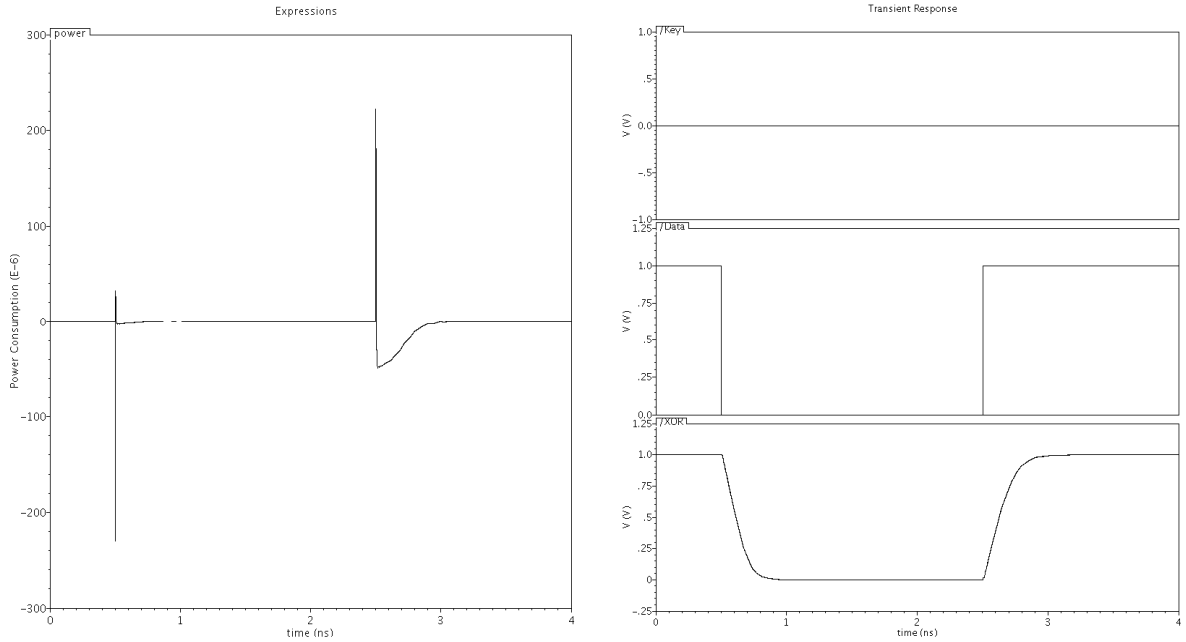


Figure 5.2: Strong inversion power waveform at left for XOR circuit, with plots on the right (top to bottom) of , key, input data and output data for *Key=0* and *Data=1->0->1*.

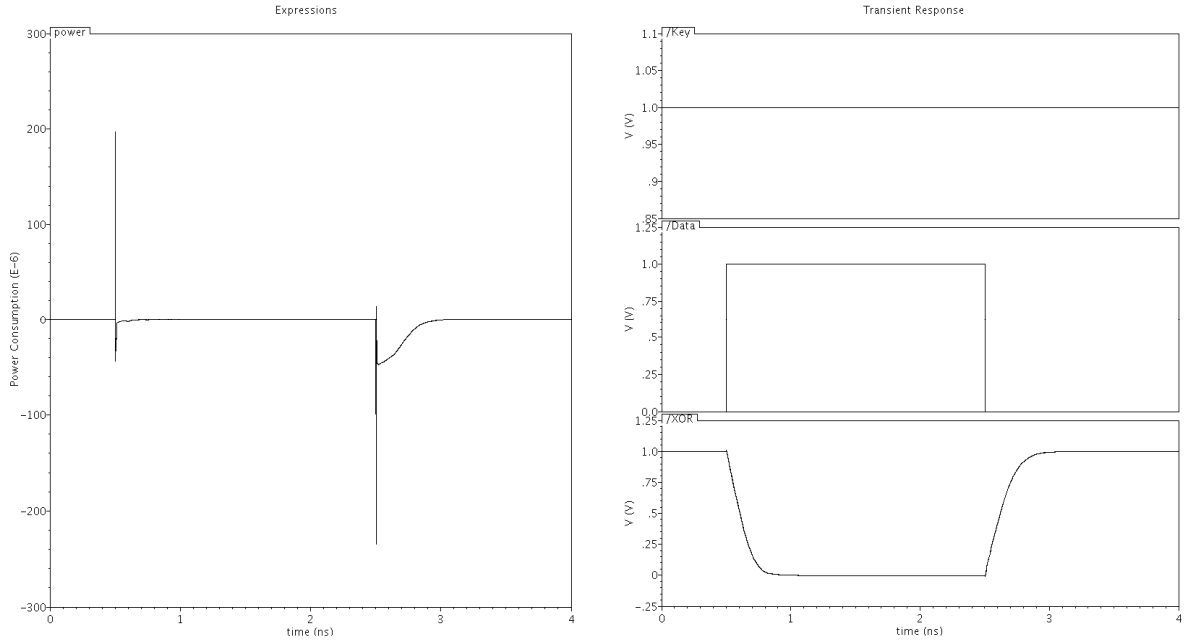


Figure 5.3: Strong inversion power waveform at left for XOR circuit, with plots on the right (top to bottom) of , key, input data and output data for *Key=1* and *Data=0->1->0*.

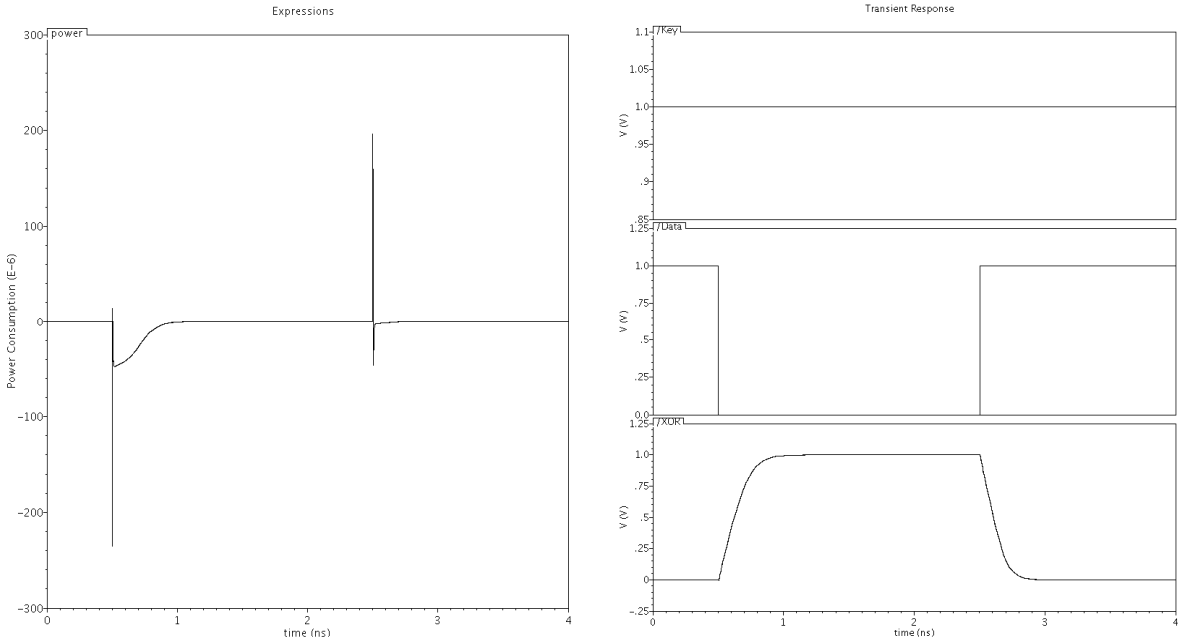


Figure 5.4: Strong inversion power waveform at left for XOR circuit, with plots on the right (top to bottom) of , key, input data and output data for *Key=1* and *Data=1->0->1*.

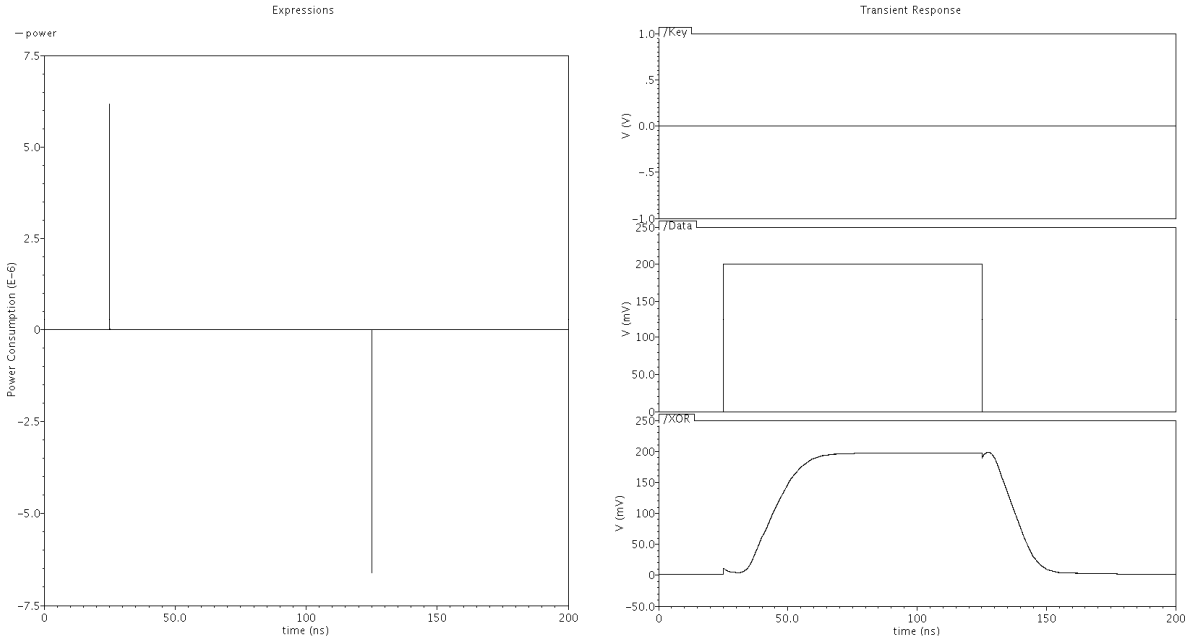


Figure 5.5: Sub-threshold power waveform at left for XOR circuit, with plots on the right (top to bottom) of , key, input data and output data for *Key=0* and *Data=0->1->0*.

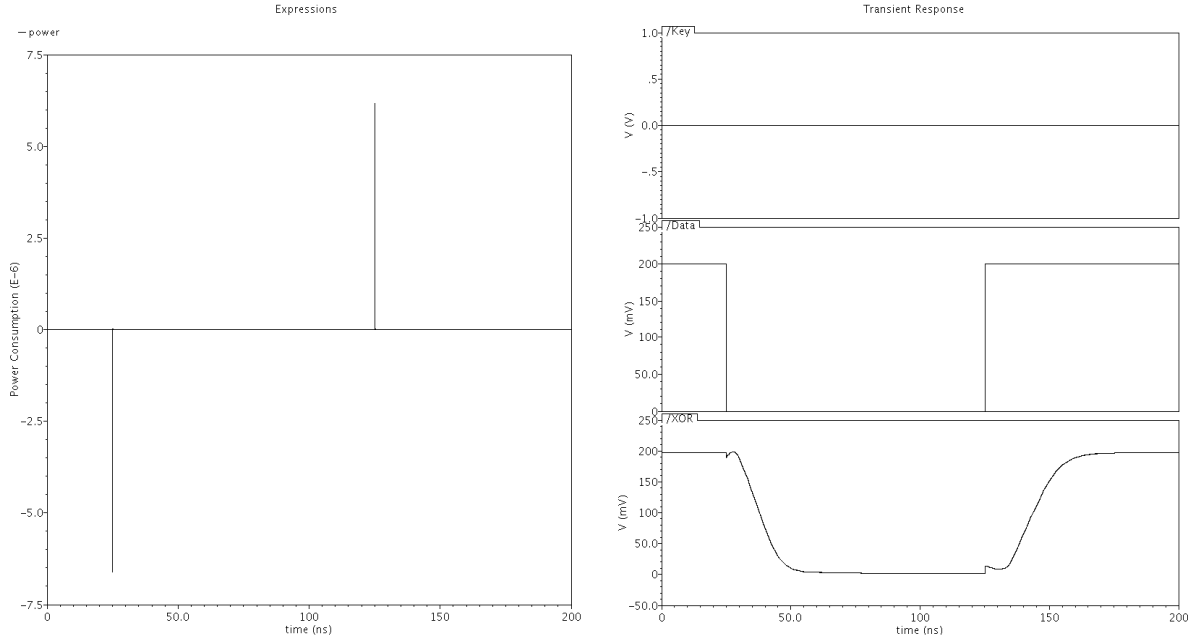


Figure 5.6: Sub-threshold power waveform at left for XOR circuit, with plots on the right (top to bottom) of , key, input data and output data for *Key=0* and *Data=1->0->1*.

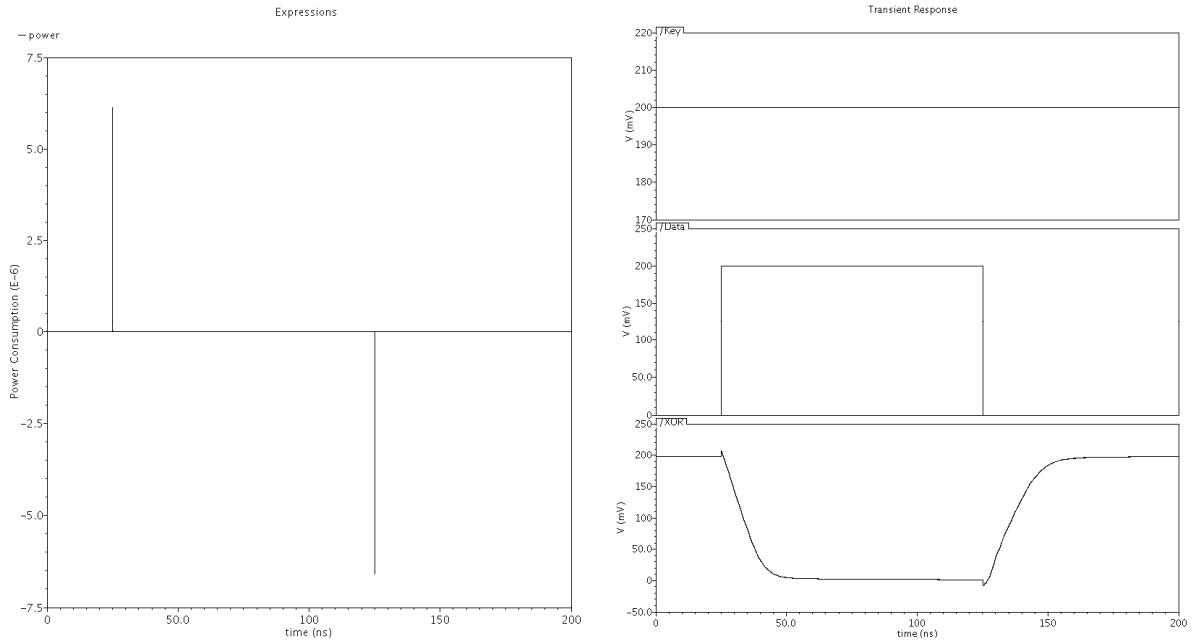


Figure 5.7: Sub-threshold power waveform at left for XOR circuit, with plots on the right (top to bottom) of , key, input data and output data for *Key=1* and *Data=0->1->0*.

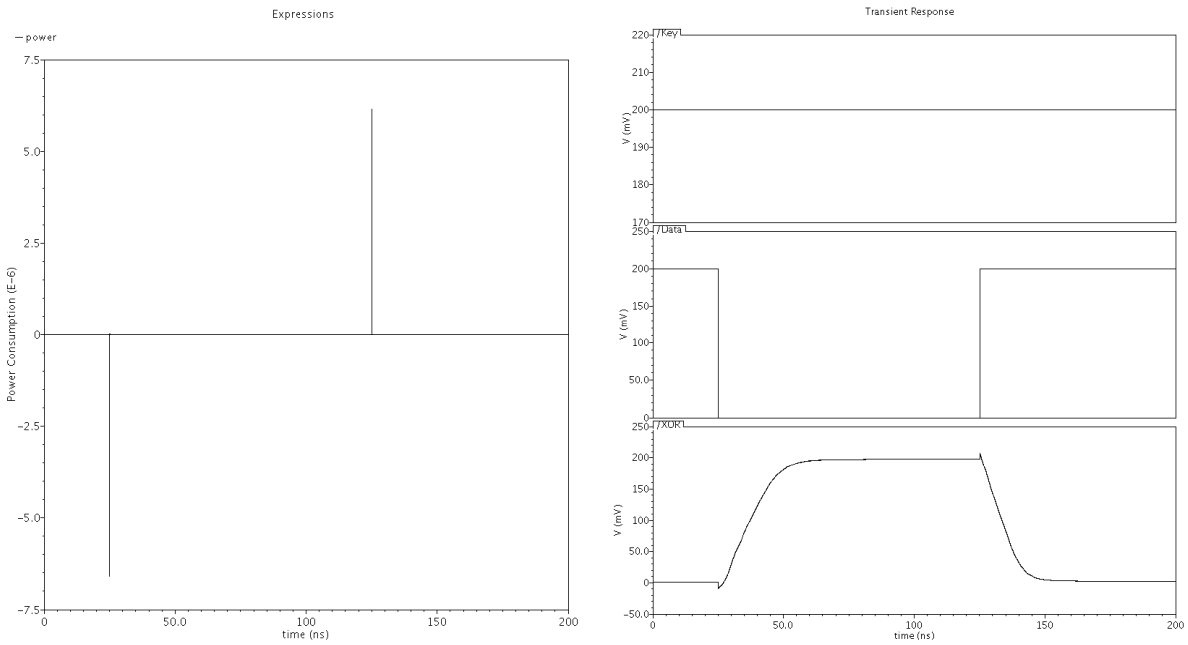


Figure 5.8: Sub-threshold power waveform at left for XOR circuit, with plots on the right (top to bottom) of , key, input data and output data for *Key=1* and *Data=1->0->1*.

The presented power waveforms provide an overview of the XOR gate's behavior in sub-threshold and strong inversion. Nevertheless, a closer look into the close range around the spike signals is necessary and provides a more realistic image. The main reason is that the period of signals in Figures 5.1 to 5.4 is 1ns, while the period is 50 ns for sub-threshold in Figures 5.5 to 5.8. Power traces are sketched over sample numbers to provide the required view. Figure 5.9 shows power waveforms in each case of 0 to 1 and 1 to 0 *Data* transitions for *Key* equal to 1 and *Key* equal to 0. The same sketches are given in Figure 5.10 for sub-threshold.

The first point to notice here is that the higher level of similarity in power traces of sub-threshold compared to strong inversion. As mentioned earlier, power traces for various transitions in an ideally secure system must completely match. Thus, although the XOR gate operating in sub-threshold illustrates a higher quality in power traces at first glance, later sections analyze these traces more precisely.

Although side channel information leakage measurements may occur in instantaneous power traces, averaged power consumption per clock cycle can also be used to compare averaged power required for a 0 to 1 transition versus a 1 to 0 transition. Table 5.1 demonstrates the averaged power values for strong inversion and Table 5.2 shows these values for sub-threshold. Each transition takes two clock cycles. For instance, in the first row of Table 5.1, first two columns are the averaged power values to complete a 0 to 1 transition in two clock cycle and the last two columns show the values to complete a 1 to 0 transition. Values related to a 0 to 1 transition in output are highlighted in grey.

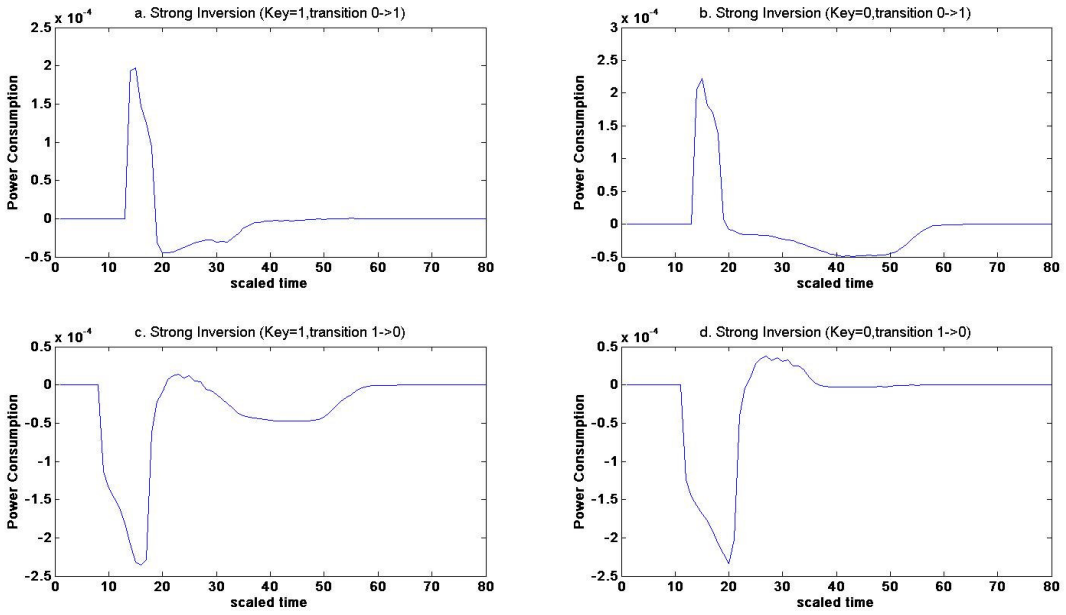


Figure 5.9: A Closer look at power spikes for strong inversion for a) *key=1,* 0-1 transition b) *key=0,* 0-1 transition c) *key=1,* 1-0 transition d) *key=0,* 1-0 transition.

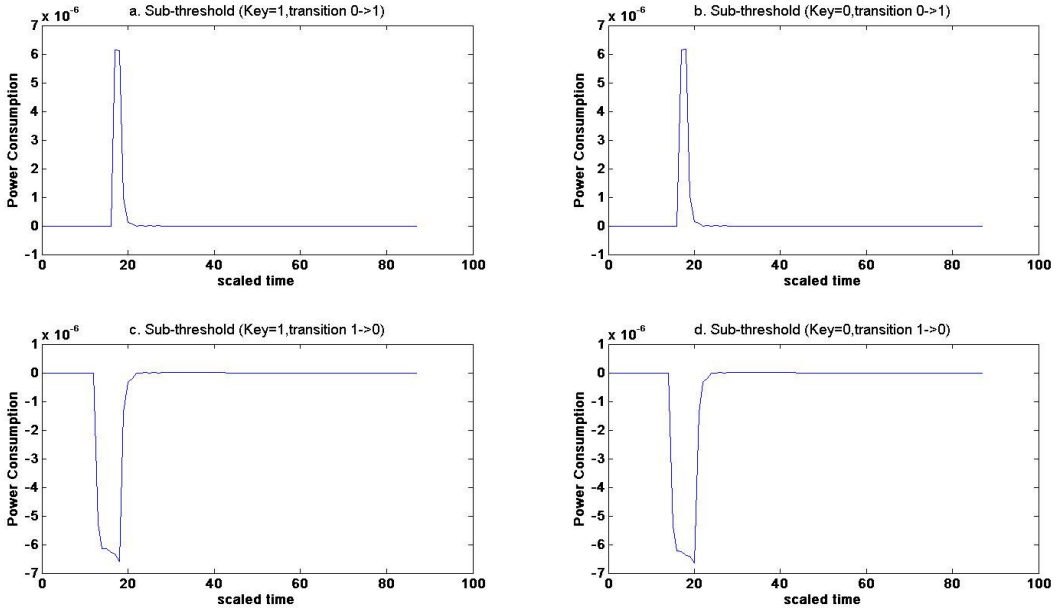


Figure 5.10: A Closer look at power spikes for sub-threshold for a) *key=1,* 0-1 transition b) *key=0,* 0-1 transition c) *key=1,* 1-0 transition d) *key=0,* 1-0 transition.

	Cycle 1	Cycle 2	Cycle 3	Cycle 4
Key=1,*Data*=1→0→1	10.6	0.0811	0.5038	0.0144
Key=1,*Data*=0→1→0	0.516	0.0158	10.45	0.0786
Key=0,*Data*=1→0→1	0.487	0.0143	10.49	0.0838
Key=0,*Data*=0→1→0	10.63	0.0871	0.4745	0.0146

Table 5.1: Averaged power values for strong inversion (grey cells are related to 0 to 1 transition in output), with values in μW.

	Cycle 1	Cycle 2	Cycle 3	Cycle 4
Key=1,*Data*=1→0→1	564.5	151.2	484.7	117.3
Key=1,*Data*=0→1→0	488.7	117.3	547.1	145.8
Key=0,*Data*=1→0→1	440.2	130.8	528.5	226.0
Key=0,*Data*=0→1→0	533.5	233.9	442.9	130.8

Table 5.2: Averaged power values for sub-threshold (grey cells are related to 0 to 1 transition in output), with values in pW.

Based on the values of Tables 5.1 and 5.2, the average power consumed in a 0 to 1 transition of the output is 5.31 μW for strong inversion and 366 pW for sub-threshold. The value for a 1 to 0 transition on the output is 0.260 μW for strong inversion and 294 pW for sub-threshold. These values demonstrate that the average power consumed at sub-threshold region to raise the output signal to 1 from 0 is almost the same as the power consumed to drop it back to 0. On the other hand, these average values for strong inversion are starkly different. The ratio of the average power for 1 to 0 transition to the average power for 0 to 1 transition is 80% for sub-threshold but, only 5% for strong inversion.

The level of the operation voltage is the most important factor that makes the situation worse for strong inversion. However, the frequency of operation and output load capacitance are also involved. The testbench has tested each circuit in its real operational point to make a judgment between sub-threshold and strong inversion with the required characteristics of each region. Therefore, although testing a circuit in strong inversion with a very low frequency and low load capacitance may result in much better results, it would not be a fair experiment.

5.1.2 DME Signals for Fixed Keys

Power traces for all combinations of fixed *Key* and 0-1-0 and 1-0-1 pulses on *Data* were obtained in the previous section. In this section, the difference between a power trace for a 0 to 1 transition and 1 to 0 transition with a fixed *Key* of either 0 or 1 is studied. Expression 5.1 shows the calculation of DME signal in this section.

$$DME0 = | \, |power\_0to1| - |power\_1to0| \, | \qquad for\ key = 0 \qquad (5.1)$$

$$DME1 = | \, |power\_0to1| - |power\_1to0| \, | \qquad for\ key = 1$$

Figures 5.11 to 5.14 show the difference signals of the same case for sub-threshold and strong inversion, plotted in one figure.

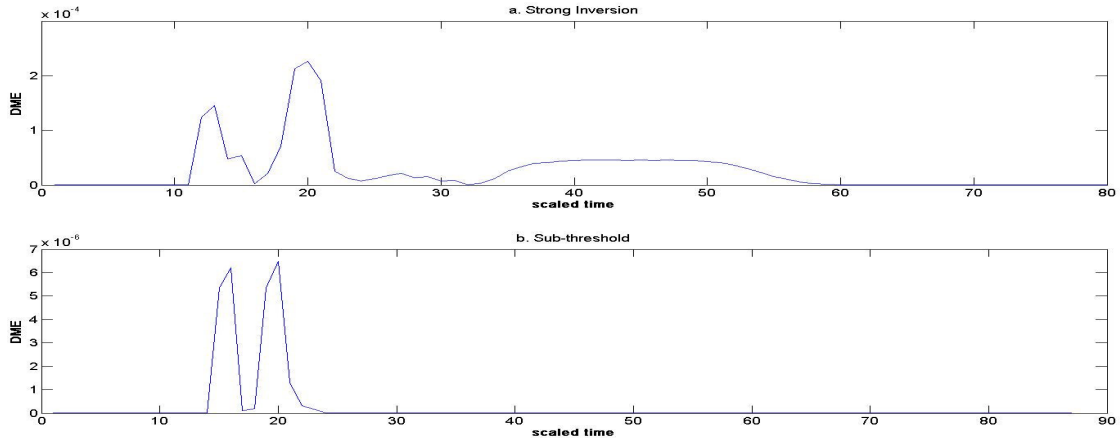


Figure 5.11: DME0 signals for *Key=0* and *Data=0->1->0* for a) Strong Inversion b) Sub-threshold.

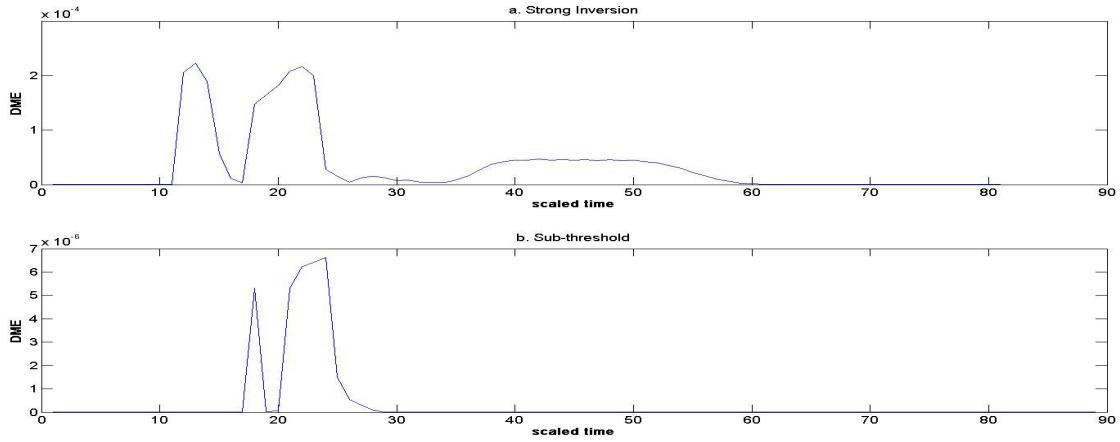


Figure 5.12: DME0 signals for *Key=0* and *Data=1->0->1* for a) Strong Inversion b) Sub-threshold.

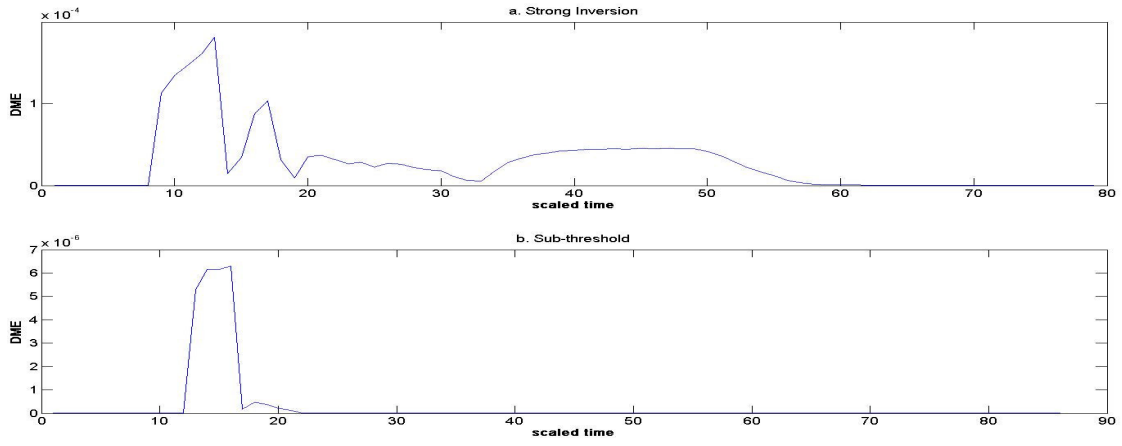


Figure 5.13: DME1 signals for *Key=1* and *Data=0->1->0* for a) Strong Inversion b) Sub-threshold.

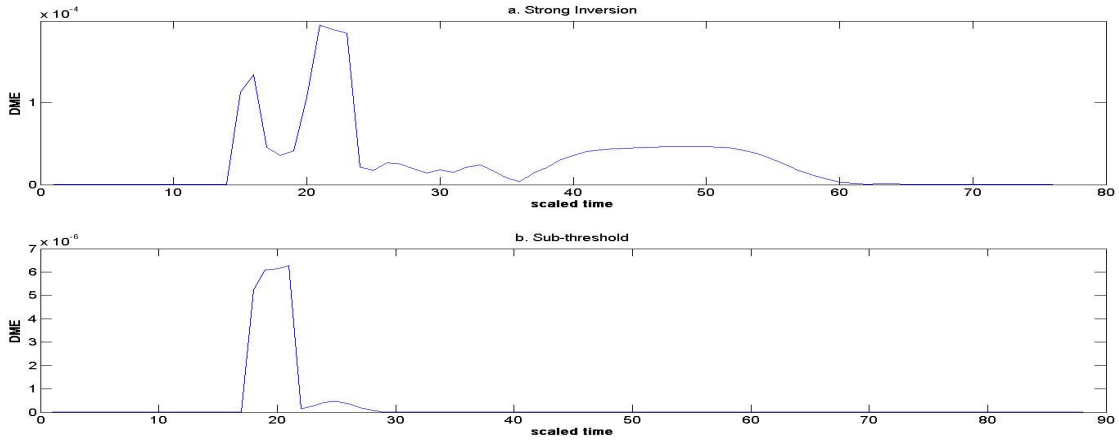


Figure 5.14: DME1 signals for *Key=1* and *Data=1->0->1* for a) Strong Inversion b) Sub-threshold.

Since the absolute value is considered for DME, it is expected that difference signals for the *Key* of 0 will have reasonable similarity. Comparing Figures 5.11 and 5.12 confirms our expectation. The same similarity also exists between Figure 5.13 and Figure 5.14 for *Key* of 1.

The first point to note in these signals is that the average of strong inversion's DME is higher than the average of sub-threshold, which means that the difference signals for sub-threshold are closer to zero most of the time. Sub-threshold has either one or two spikes in DME, which results from the mismatched spike times in power traces. Since the power trace's peaks for 0 to 1 and 1 to 0 transitions are almost the same, matched timing could result in a closer to ideal DME signal.

The second point to note to be made in relation to the presented signals is peak value. Table 5.3 summarizes the results using the ratio shown in Expression 5.2 for sub-threshold and strong inversion. The lower the value of *PeakRatio*, the better the security characteristics of the circuit.

$$PeakRatio = \frac{Maximum\ Peak\ in\ DME}{Maximum\ Peak\ in\ Power\ Consumption} \tag{5.2}$$

It can be observed that *PeakRatio* of sub-threshold, for the *Key* equal to 1, is about half of the ratio for strong inversion, indicating fewer information leaks from the power trace at sub-threshold. The ratio for *Key* equals to 0 is almost the same for both logics with a slight advantage for strong inversion. Another interesting point in Table 5.3 is that all of the ratios for sub-threshold are almost equal.

	Strong Inversion	Sub-threshold
Key=1,*Data*=1→0→1	17%	6.5%
Key=1,*Data*=0→1→0	16%	6.7%
Key=0,*Data*=1→0→1	3.8%	6.7%
Key=0,*Data*=0→1→0	5.1%	6.8%

Table 5.3: *PeakRatio* for strong inversion and sub-threshold in various cases.

The third point is the absolute value of the peak. The maximum peak value for strong inversion is about 2×10^{-4}, which is almost 300 times the maximum peak value for sub-threshold, 6×10^{-6}. Since the level of the power is also an important factor in side channel analysis, where low current and power levels may lead a failed attack or may force the attacker to collect more traces, sub-threshold has an important advantage in this aspect.

5.1.3 DME Signal for Fixed Data Transitions

In this section the DME calculation method is different from the previous section. Here, the *Data* transition is fixed and DME is the difference between the power trace corresponding to *Key* of 0 and the power trace corresponding to the *Key* of 1. Expression 5.3 presents the mentioned method in formula. Figures 5.15 and 5.16 show this new set of DME signals for sub-threshold and strong inversion.

$$DME0to1 = |\,|power\_key0| - |power\_key1|\,| \qquad for\ Data = 0\rightarrow1 \tag{5.3}$$

$$DME1to0 = |\,|power\_key0| - |power\_key1|\,| \qquad for\ Data = 1\rightarrow0$$

Figures 5.15 and 5.16 also propose a higher level of security for sub-threshold versus strong inversion. One can observe that the DME signal for sub-threshold is just one or two spikes, while it is a nonzero signal for most of the shown time window for strong inversion. A higher level of power consumption can also be seen in these figures.

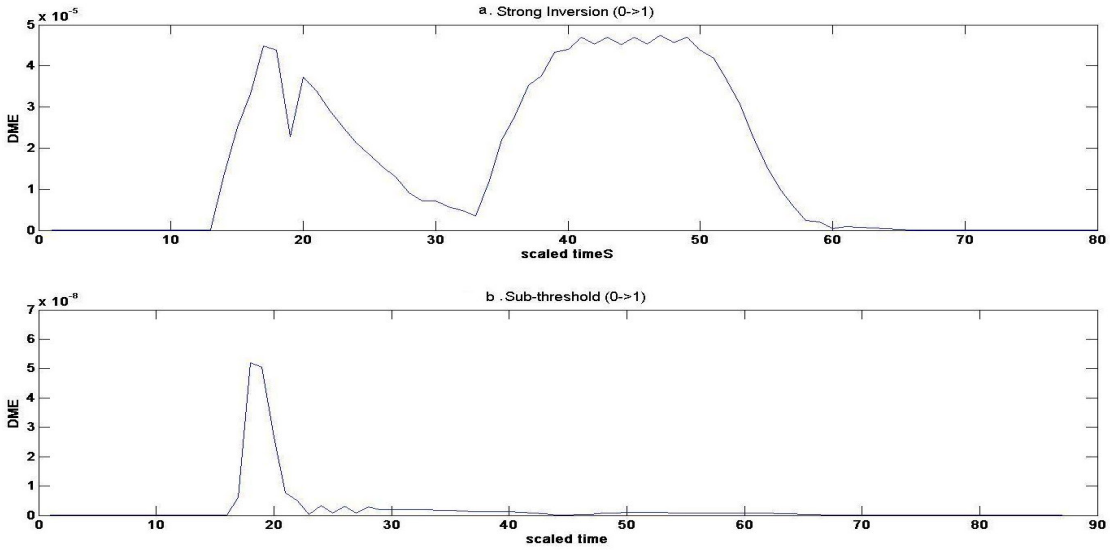


Figure 5.15: DME signals for *Data = 0→1* for a) Strong Inversion b) Sub-threshold.

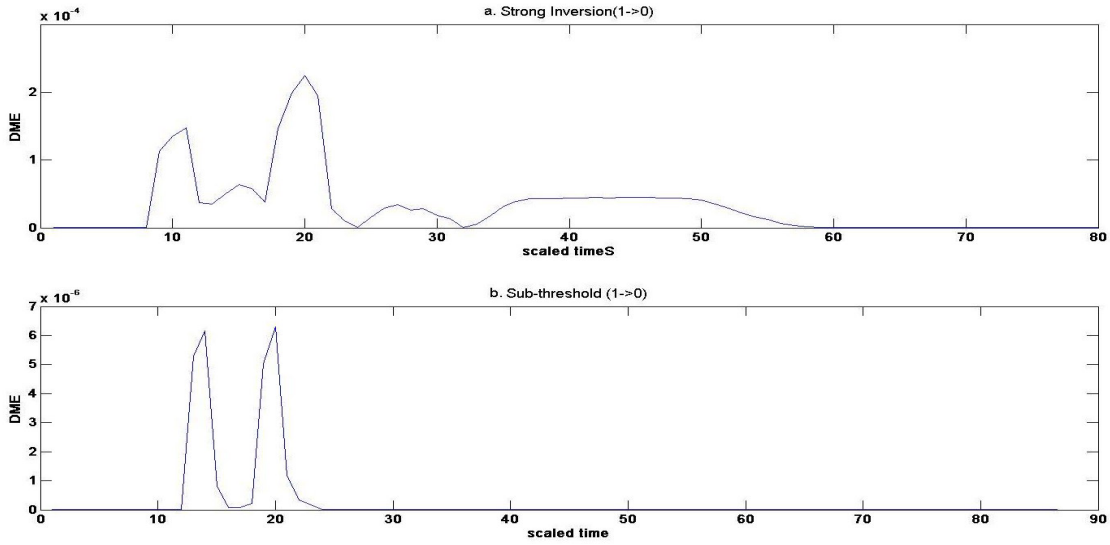


Figure 5.16: DME signals for *Data = 1→0* for a) Strong Inversion b) Sub-threshold.

5.2 Parallel XORs Architecture Analysis with DME Measure

The parallel XORs architecture previously shown in Figure 3.7 is simulated with all 256 possible inputs (b) and fixed key (k) of 5C and 5D. An important fact about the choice of 5C and 5D is that one has the lsb of 0 and the other has the lsb of 1. After simulation, 256 power traces corresponding to the applied inputs are obtained. Mean signals, M_0 and M_1, are the average of signals in set T_0 and T_1, shown in Expression 5.4, respectively. DME is calculated using Formula 4.2.

$$T_0 = \{P_i | \, b_i = 0\} \tag{5.4}$$

$$T_1 = \{P_i | \, b_i = 1\}$$

Figure 5.17 shows DME signals for strong inversion and sub-threshold. Like the previous conclusions, two important advantages can be observed in sub-threshold signals. First, the DME of sub-threshold is, on average, closer to zero. Second, the similarity between difference signals of sub-threshold for key of 0 and key of 1 is greater than the similarity between those for strong inversion. Hence, the parallel XORs architecture at sub-threshold demonstrates enhanced security performance in comparison to strong inversion.

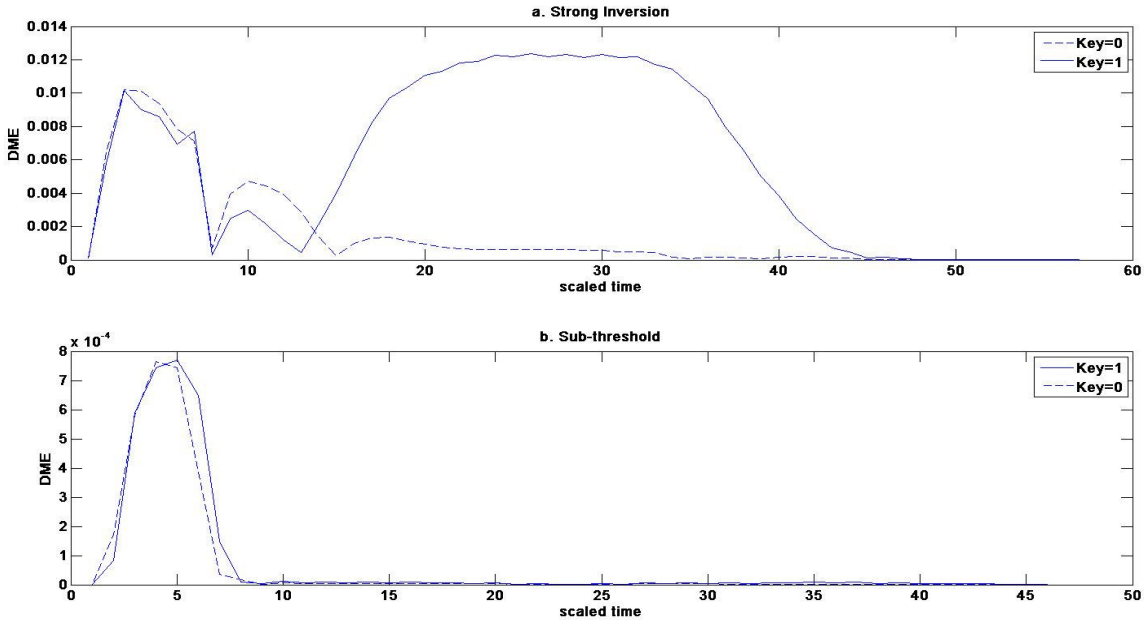


Figure 5.17: DME signals for parallel XORs architecture for a) Strong Inversion b) Sub-threshold.

5.3 NAND, NOR and XOR Gates Analysis with Frequency of Observation Measure

The frequency of observation was introduced in section 4.2. In this section, averaged power consumption per clock cycle values for designs under investigation are obtained by simulation and used to plot the frequency of observation or frequency of occurrence of each power value. The test gates for this experiment are NAND, NOR and XOR, in all logic schemes of SABL, WDDL, sub-threshold, and strong inversion. Each of the test gates, has two inputs, and each input experiences four transitions (0-0, 0-1, 1-0 and 1-1). Thus, there are 16 different possible combinations of transitions. These input transitions and the output transition for NAND, NOR and XOR gates are represented in Table 5.4 and are tested with all possible transitions and the averaged power consumption is measured for each.

in1	in2	NAND-out	NOR-out	XOR-out
0-0	0-0	1-1	1-1	1-1
0-0	0-1	1-1	1-0	1-0
0-0	1-1	1-1	0-0	1-1
0-0	1-0	1-1	0-1	1-0
0-1	0-0	1-1	1-0	0-1
0-1	0-1	1-0	1-0	0-0
0-1	1-1	1-0	0-0	1-0
0-1	1-0	1-1	0-0	1-1
1-1	0-0	1-1	0-0	1-1
1-1	0-1	1-0	0-0	1-0
1-1	1-1	0-0	0-0	0-0
1-1	1-0	0-1	0-0	0-1
1-0	0-0	1-1	0-1	1-0
1-0	0-1	1-1	0-0	1-1
1-0	1-1	0-1	0-0	0-1
1-0	1-0	0-1	0-1	0-0

Table 5.4: Testbench.

The frequency of the clock signal in SABL, WDDL and strong inversion is chosen to be 1 GHz, which is a reasonable frequency for an individual gate. The frequency for the circuits at sub-threshold is chosen to be 20 MHz. The supply voltages for SABL, WDDL and strong inversion are 1 V, and the supply voltage at the sub-threshold is 200 mV.

The results of each gate are presented individually in the following sections.

5.3.1 NAND Gate

Figure 5.18 compares SABL with strong inversion. This histogram shows that while the observed energies are spread out in a broad range for strong inversion, they remain in a narrow band for SABL. However, the average power consumption of SABL is more than five times the power consumption of SE.

Since both logic schemes are tested under the same testbench, the narrow band of the energies for SABL shows less correlation between power consumption and inputs. Thus, SABL is a more secure logic against power analysis attacks.

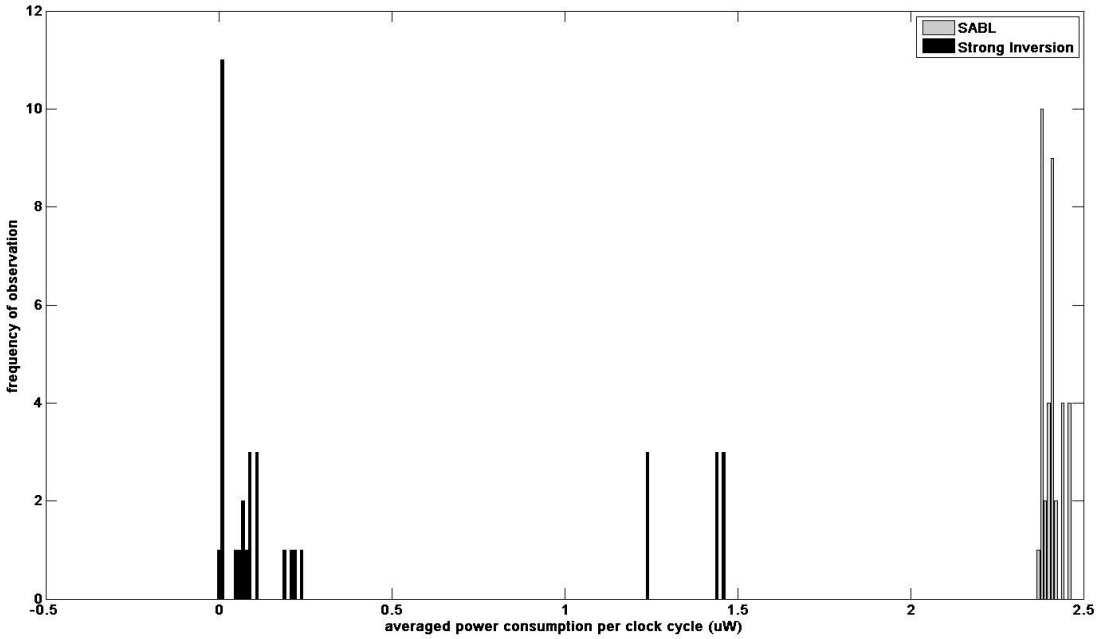


Figure 5.18: SABL vs. strong inversion for the number of observed energies per cycle (NAND).

Figure 5.19 demonstrates the same type of comparison between WDDL and strong inversion. It can be seen that the energies per cycles for WDDL remain in a narrow band while they spread out for strong inversion. As mentioned, this finding means that WDDL is more secure than strong inversion and that its power consumption is also significantly greater than strong inversion's.

Comparing SABL and WDDL can also prove useful. Figure 5.20 represents this comparison. As this histogram shows, the energies of SABL are closer than those of WDDL. Hence, SABL is a more secure logic scheme. It can also be seen that the power consumption of WDDL is less than SABL's.

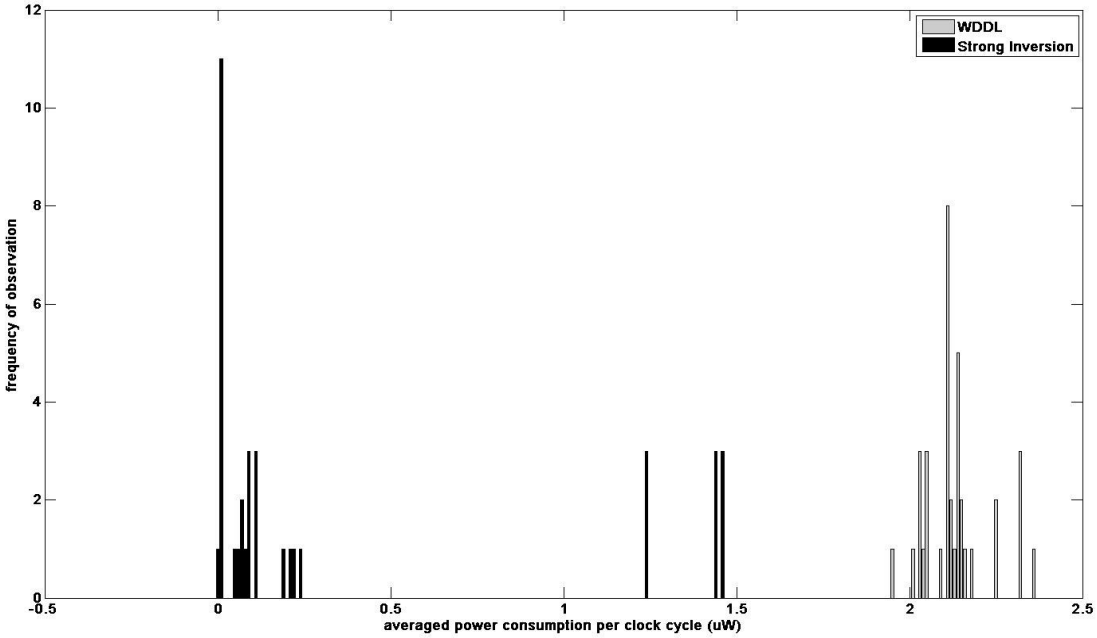


Figure 5.19: WDDL vs. strong inversion for the number of observed energy per cycle (NAND).

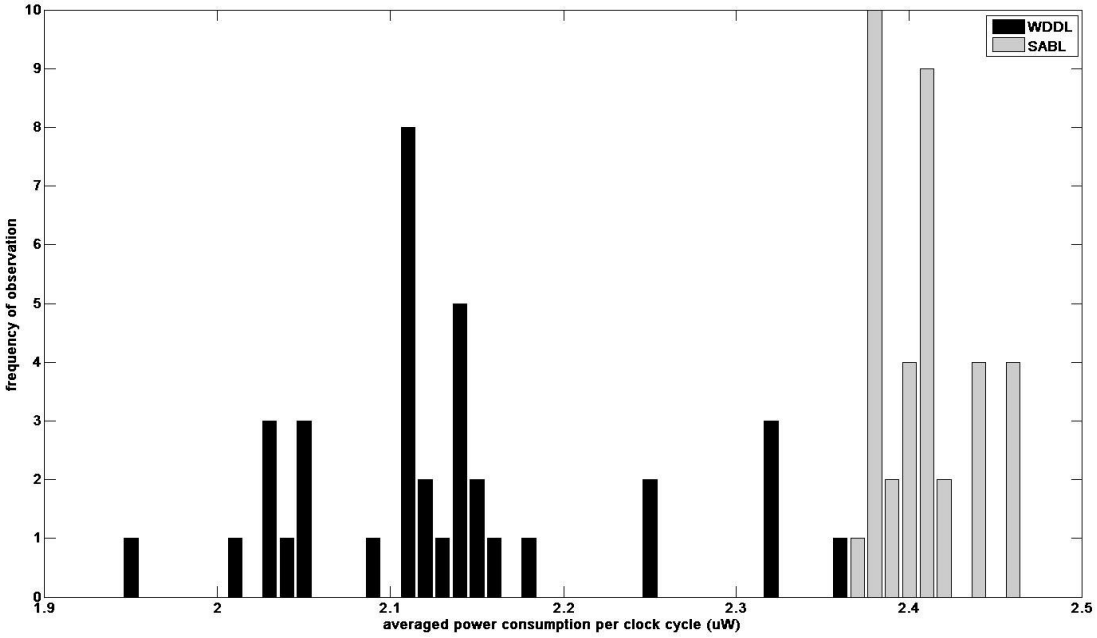


Figure 5.20: SABL vs. WDDL for the number of observed energy per cycle (NAND).

Finally, Figure 5.21 demonstrates a comparison between sub-threshold and strong inversion. The power consumption of the sub-threshold is about 1000 times less than strong inversion's, so, in order to compare the two, the power value of sub-threshold in this histogram is scaled by 1000. The histogram shows that the average power values of strong inversion are more widely spread out. Therefore, it can be concluded that sub-threshold schemes provide less correlation between power and inputs

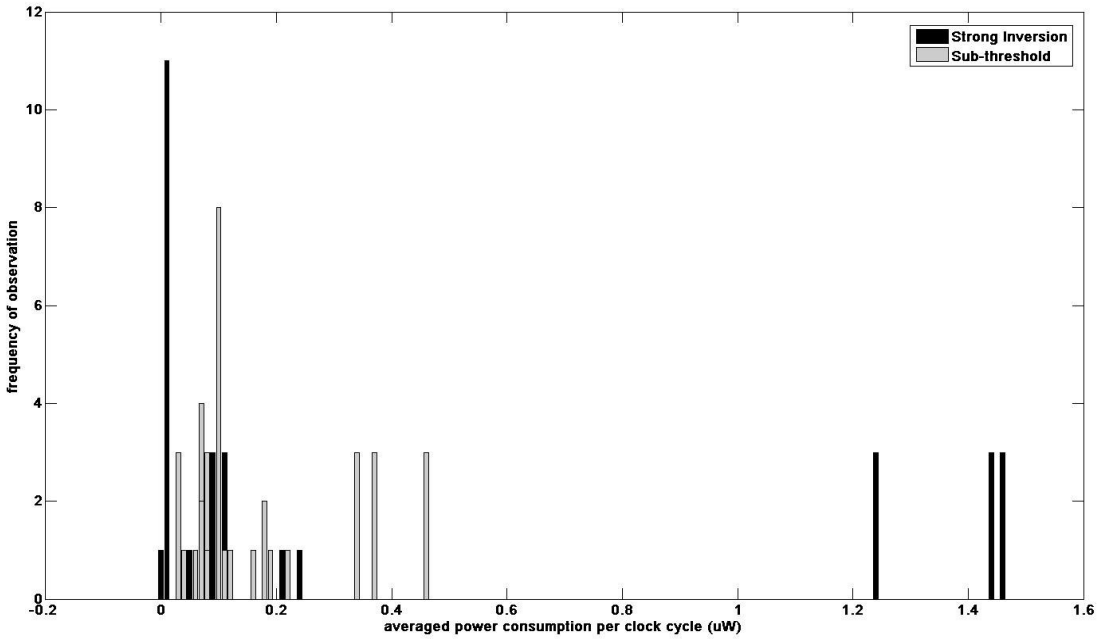


Figure 5.21: Strong inversion vs. sub-threshold for the number of observed energy per cycle (NAND).

5.3.2 NOR Gate

The same comparisons have been done for NOR gates. Figures 5.22 to 5.25 present the results. Here, the same conclusion as for the NAND gate can be made for the NOR gate. SABL's histogram has the narrowest width, followed by WDDL's. However, these two logic schemes consume substantially more power than strong inversion, which itself consumes 1,000 times more power than sub-threshold. Sub-threshold has also shown less variation in power compared to strong inversion, which means less correlation to inputs.

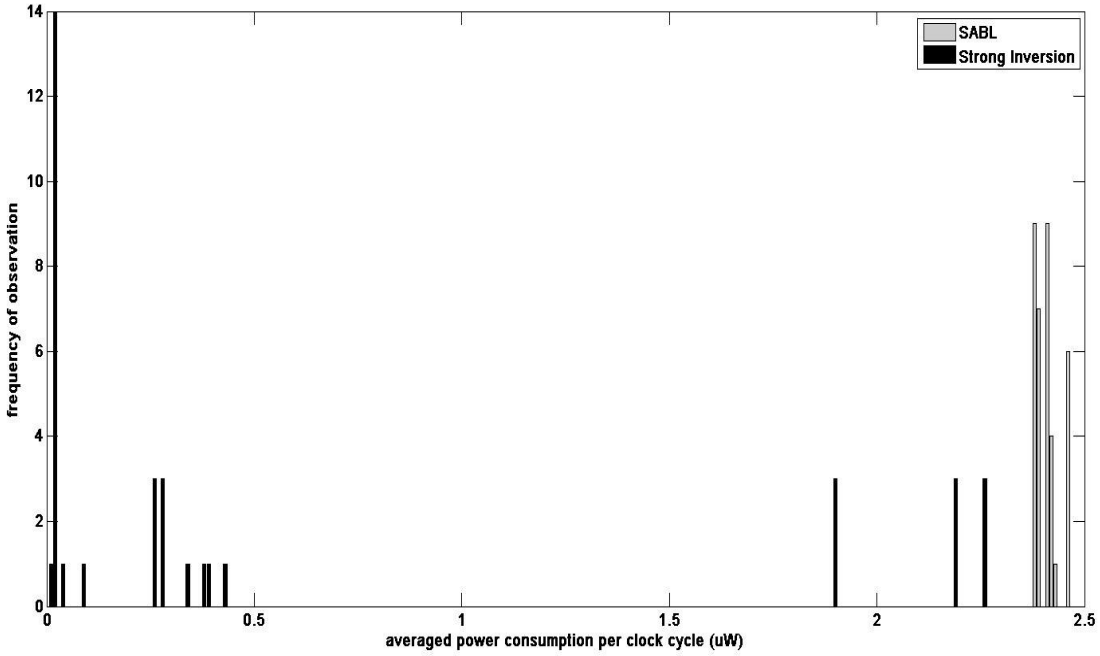


Figure 5.22: SABL vs. strong inversion for the number of observed energy per cycle (NOR).

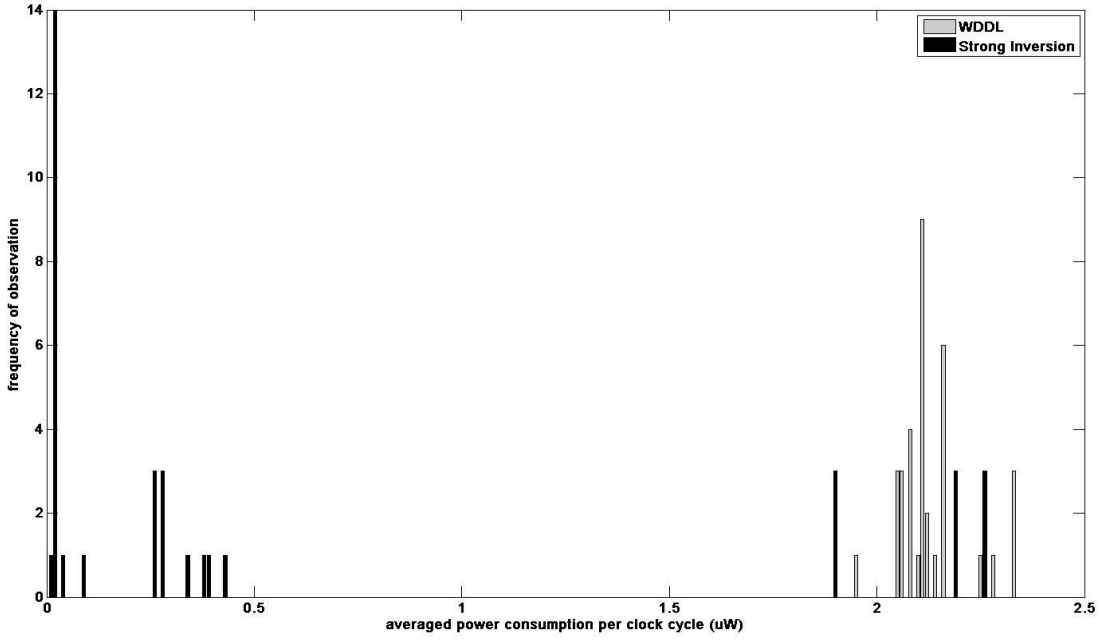


Figure 5.23: WDDL vs. strong inversion for the number of observed energy per cycle (NOR).

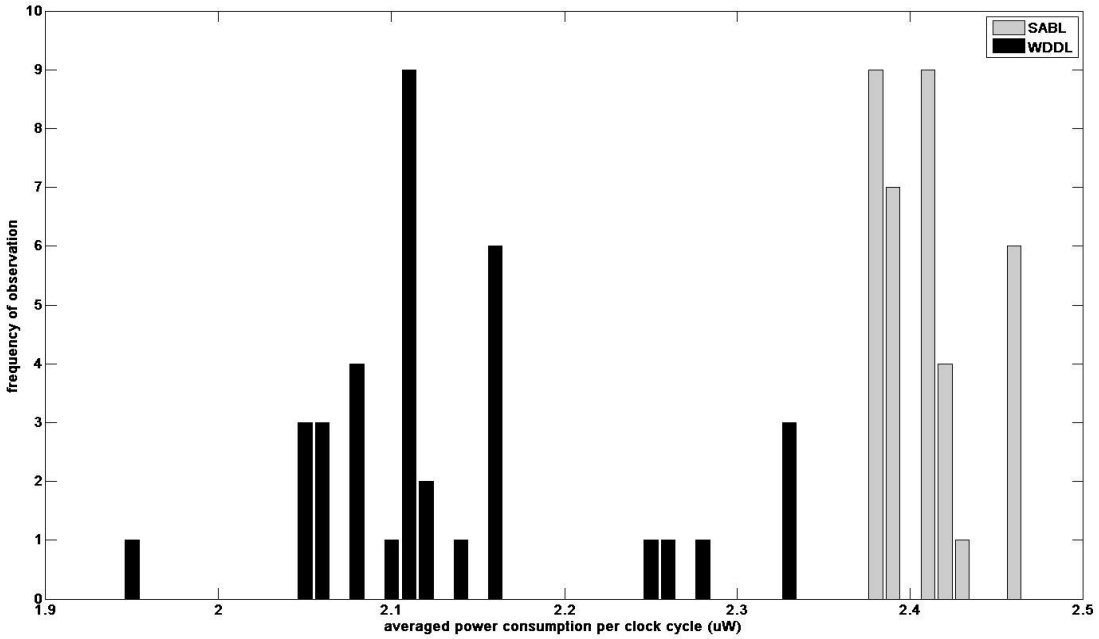


Figure 5.24: SABL vs. WDDL for the number of observed energy per cycle (NOR).

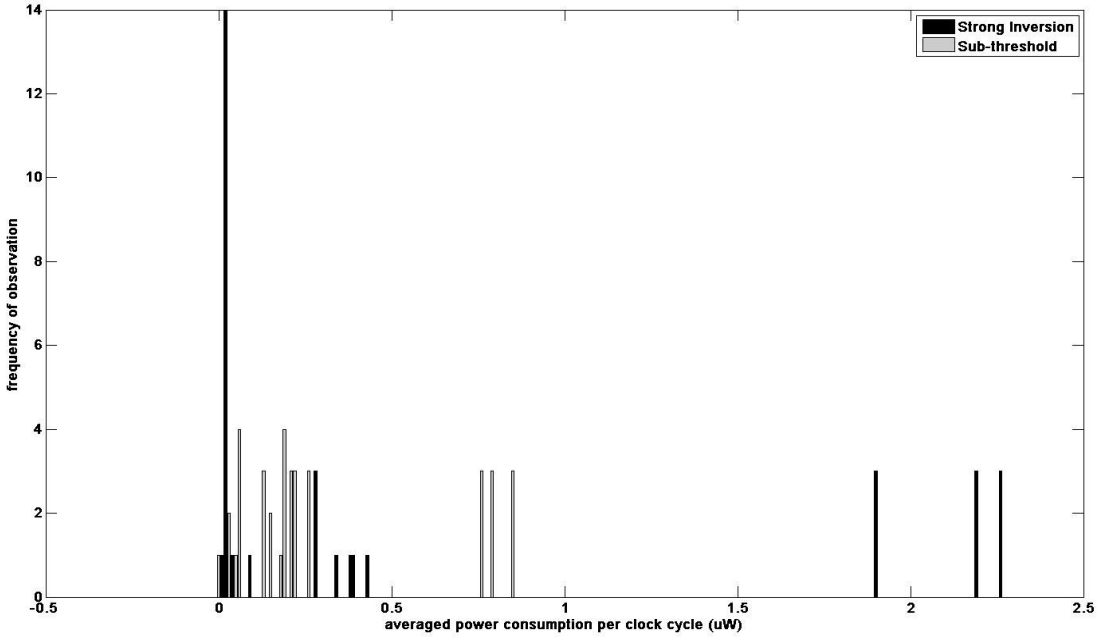


Figure 5.25: Strong inversion vs. sub-threshold for the number of observed energy per cycle (NOR).

5.3.3 XOR Gate

XOR gate has the same behavior as the previously described gates; however, in XOR the average power consumption for the transition from 0 to 1 is significantly larger than others. The reason can be found in the architecture of the XOR gate, which generates many glitches before the pull-up network is completely turned on. Figures 5.26 to 5.29 show the frequency of observation histograms for the XOR gate.

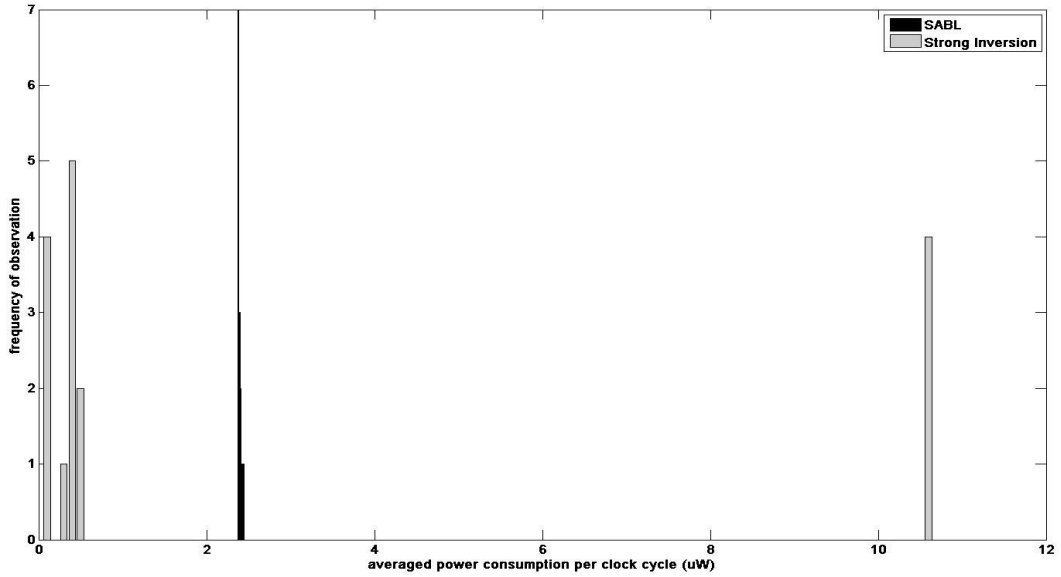


Figure 5.26: SABL vs. strong inversion for the number of observed energy per cycle (XOR).

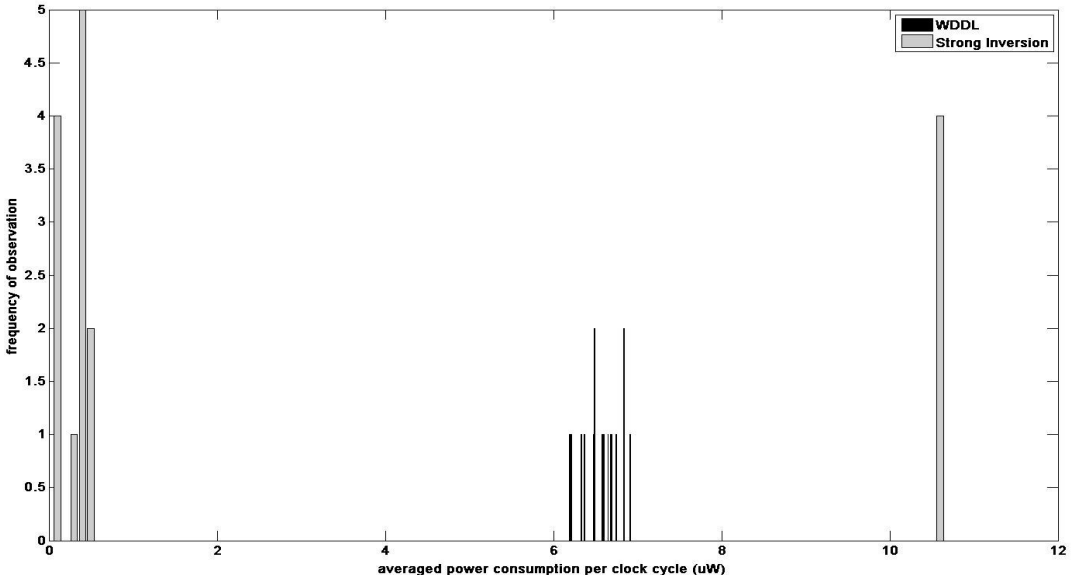


Figure 5.27: WDDL vs. strong inversion for the number of observed energy per cycle (XOR).

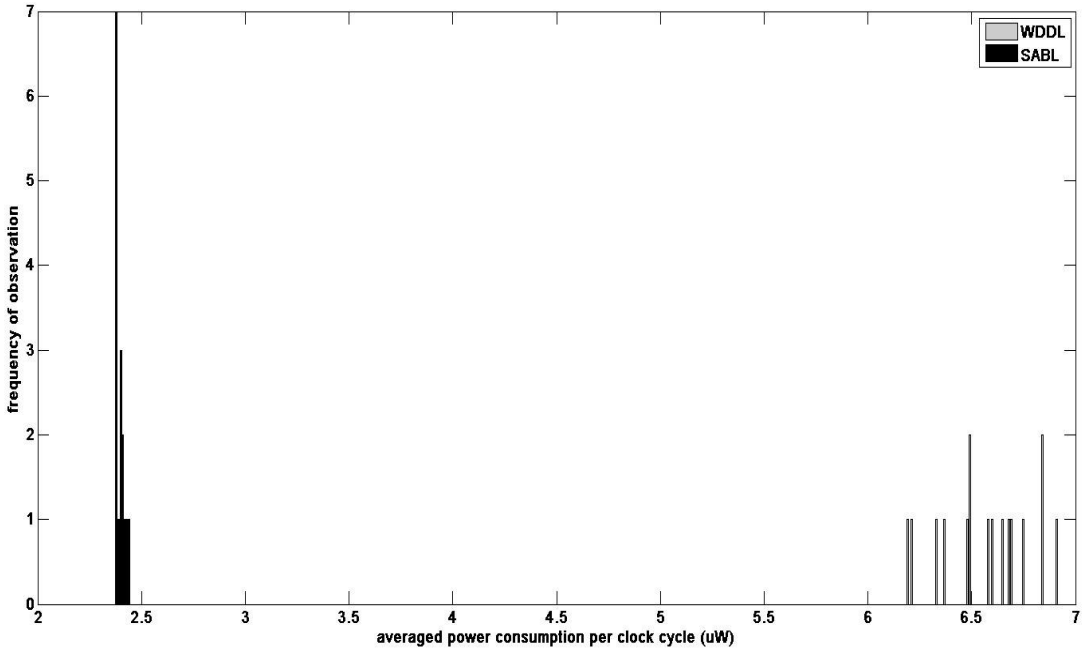


Figure 5.28: SABL vs. WDDL for the number of observed energy per cycle (XOR).

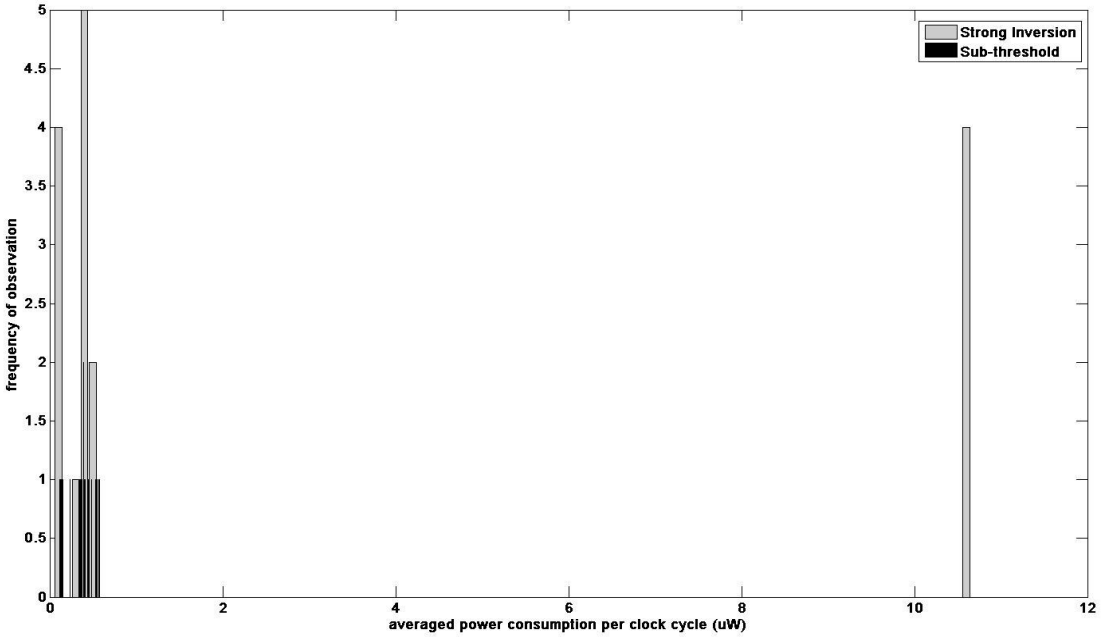


Figure 5.29: Strong inversion vs. sub-threshold for the number of observed energy per cycle (XOR).

5.4 NAND, NOR and XOR Gates Analysis with NED and NSD Measures

NED and NSD were introduced in section 4.3. Averaged power values, presented in the histograms in the previous section, are used to calculate NED and NSD based on Expressions 4.3 and 4.4. Table 5.5 demonstrates the mean, standard deviation, NED, and NSD for all three gates. The mean and STD are in μW for SABL, WDDL and strong inversion, and in nW for sub-threshold.

	NAND				NOR				XOR			
Logic / Measure	SABL	WDDL	S.I.	Subth.	SABL	WDDL	S.I.	Subth.	SABL	WDDL	S.I.	Subth.
Mean	2.41	2.13	0.34	0.17	2.41	2.13	0.63	0.31	2.4	6.57	2.87	0.377
STD	0.03	0.09	0.58	0.14	0.03	0.09	0.88	0.29	0.02	0.22	4.62	0.152
NED	0.03	0.17	1.00	0.95	0.03	0.16	1.00	1.00	0.02	0.10	1.00	0.79
NSD	0.01	0.04	1.47	0.80	0.01	0.04	1.41	0.94	0.01	0.03	1.61	0.40

Table 5.5: NED and NSD values for NAND, NOR and XOR gates.

The first noticeable point in Table 5.5 is the extremely low values of NED and NSD for SABL and WDDL logic schemes. While these two logic schemes are the most secure ones, they consume noticeably more power. Among strong inversion and sub-threshold, the later one not only consumes about 1000 times less power, but also its power values have less variation. Lower values of NED and NSD for sub-threshold in comparison to strong inversion confirms that power traces of a circuit operating at sub-threshold leak less information.

Lower levels of NED and NSD values for the XOR gate in sub-threshold compared to the same logic for NAND and NOR is most probably caused by the output load capacitance. The load capacitance for NAND and NOR gates are 1fF; however, this value for XOR is 0.1 fF. The reason is that in the XOR, the load capacitance value is a fraction of the load capacitance in strong inversion which the scaling value is the ratio of the operation frequency in strong inversion to the operation frequency at sub-threshold.

5.5 Parallel XORs Architecture Analysis with CPA Measure

As mentioned earlier, the parallel XORs architecture shown in Figure 3.7 is simulated with all 256 possible inputs (*b*) and fixed key (*k*) of 5D and 5C. An important feature of 5D and 5C is that one has the lsb of 1 and the other has the lsb of 0. After simulation, 256 power traces corresponding to applied

inputs are obtained. The CPA methodology described in section 4.4 is used to obtain the correlation trace matrix (Corr_trace). The power model used here is Hamming distance.

Various experiments are performed on the Corr_trace matrix. First, it is used to reveal the correct key. The absolute value of maximum peaks and their indexes of each of 256 traces are stored in other arrays. Let's call the array containing the maximum peak values, Max_Corr_Coeff. The index corresponding to the maximum value of Max_Corr_Coeff gives the correct key. In simpler terms, the key containing the highest correlation peak compared to all other keys is the correct key. A CPA attack executed on the correlation matrices of both sub-threshold and strong inversion was able to detect the correct key for both 5C and 5D. Figure 5.30 demonstrates the Max_Corr_Coeff traces, arrays of maximum correlation values, for both strong inversion and sub-threshold. The correlation coefficient of the correct key, 5D, is shown. Figure 5.30.c represents the difference of two histograms shown in 5.30.a and 5.30.b. The same comparison for key of 5C is shown in Figure 5.31.

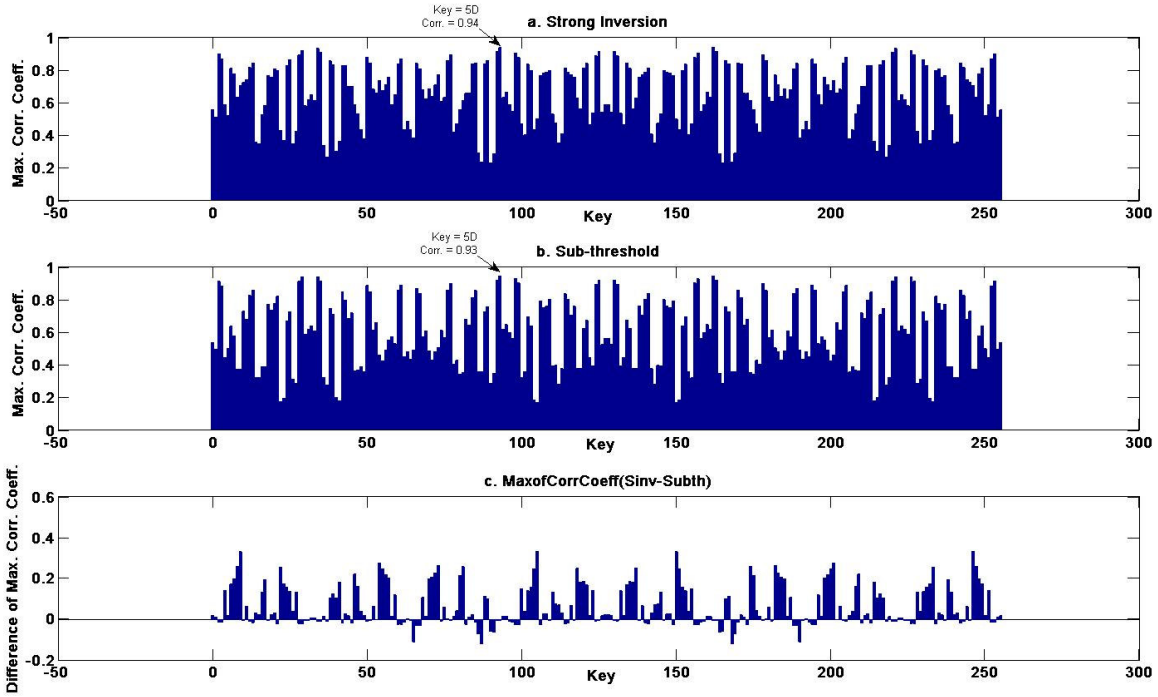


Figure 5.30: Maximum Correlation Coefficients for *correct key* = 5D, for a) Strong Inversion b) Sub-threshold c) Difference of strong inversion and sub-threshold.

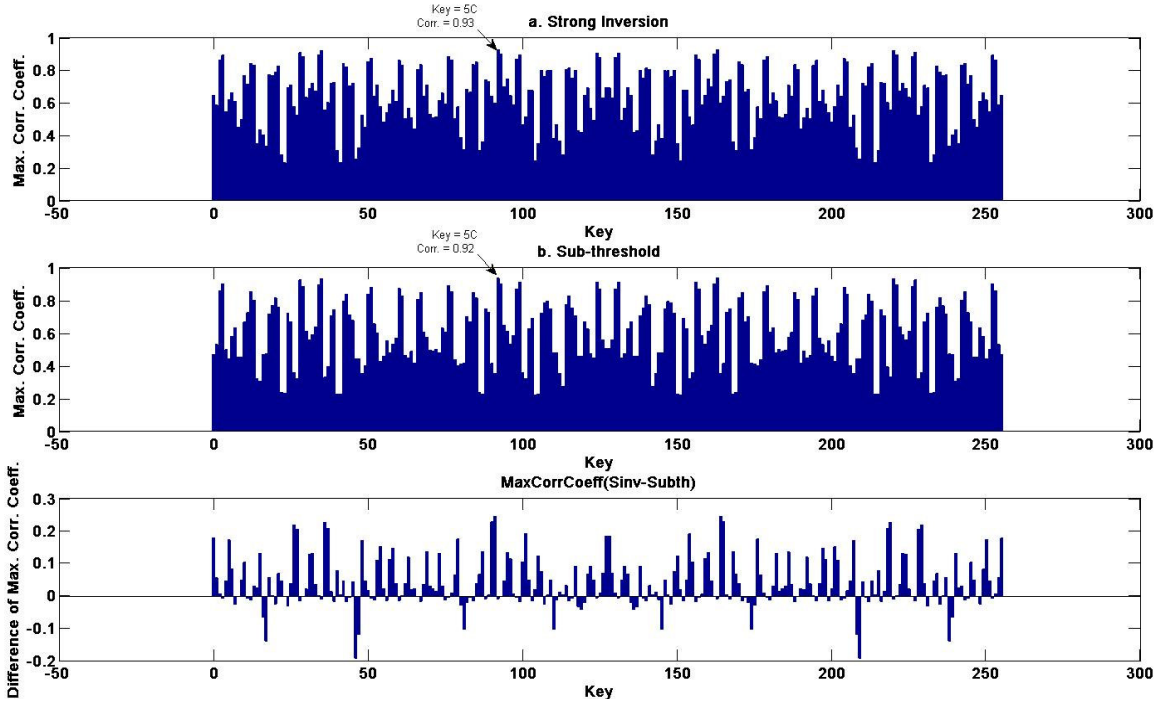


Figure 5.31: Maximum Correlation Coefficients for *correct key* = 5C, for a) Strong Inversion b) Sub-threshold c) Difference of strong inversion and sub-threshold.

The maximum correlation values of both logics are almost the same, with slightly more weight on the strong inversion side. However, parts c of Figures 5.31 and 5.30 show that the correlation coefficients of strong inversion are generally more than the sub-threshold's coefficients. Specifically, strong inversion's correlation coefficients are on average about 0.05 greater than sub-threshold's. Thus, we can deduce that the power consumption of the parallel XOR architecture at sub-threshold correlates less with its input. Nevertheless, the crucial factor that likely makes the attack harder (and is not possible to show in simulation) is the power level. Power consumption of this architecture at sub-threshold is 100 times less than strong inversion's, which makes the power analysis attack much more difficult.

Figure 5.32 presents the absolute value of the difference between the maximum correlation coefficients, Max_Corr_Coeff, of key 5C and key 5D for strong inversion and sub-threshold. The smaller this difference, the more secure the device. As can be seen in Figure 5.32, the difference is graeter for strong inversion. The maximum difference is 0.4 for strong inversion, but just 0.12 for sub-threshold. Also, the average of these differences is 0.019 for strong inversion versus 0.0058 for sub-threshold.

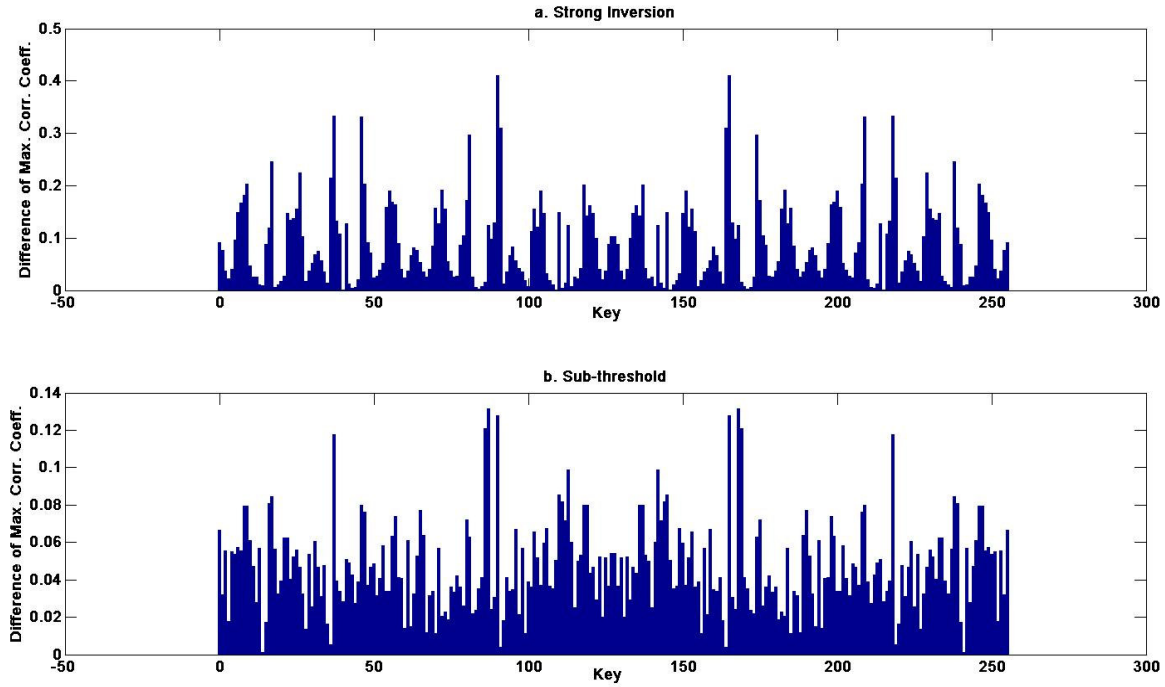


Figure 5.32: Difference of Max_Corr_Coeff traces for *Key*=5C and *Key=5D*, for a) Strong Inversion b) Sub-threshold.

The correlation trace of the correct key is shown in Figures 5.33 and 5.34 for keys of 5D and 5C, respectively.

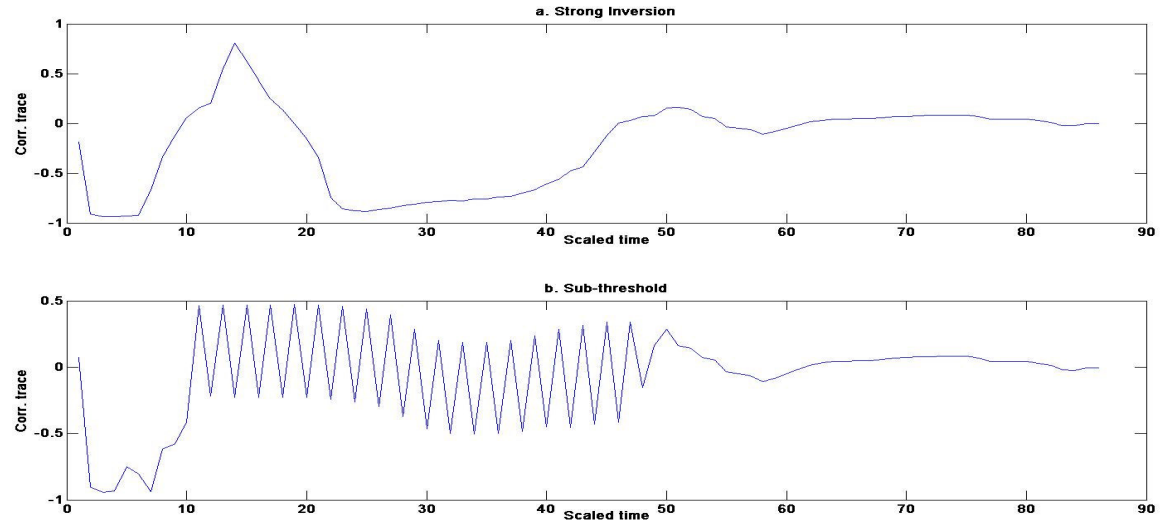


Figure 5.33: Correlation coefficient traces corresponding to the correct key, 5D, for a) Strong Inversion b) Sub-threshold.

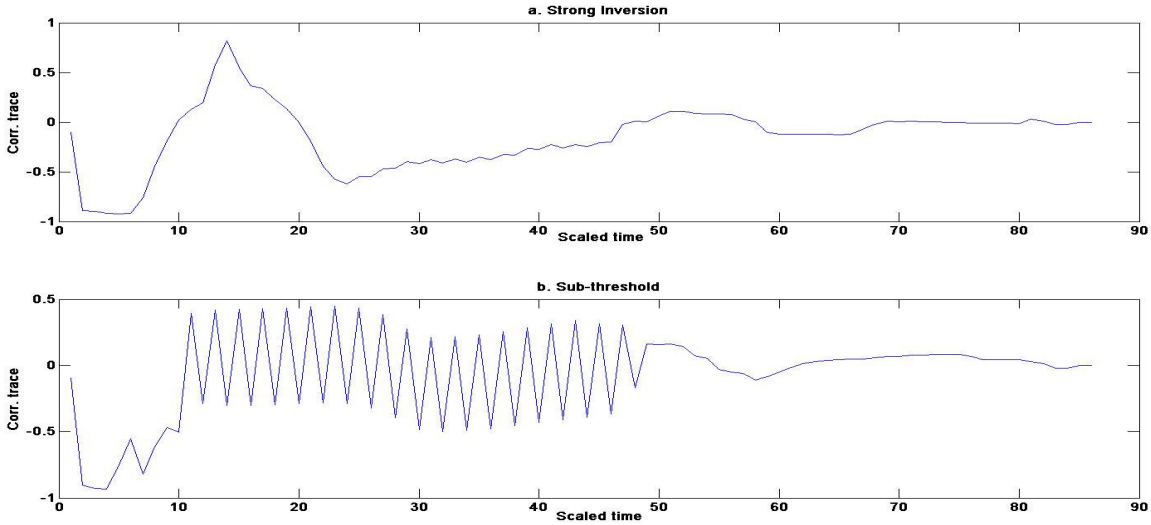


Figure 5.34: Correlation coefficient traces corresponding to the correct key, 5C, for a) Strong Inversion b) Sub-threshold.

The final point worth noting here is the time at which the power trace correlates with the acquired power model. Figures 5.35 and 5.36 illustrate the time of correlation for each key guess.

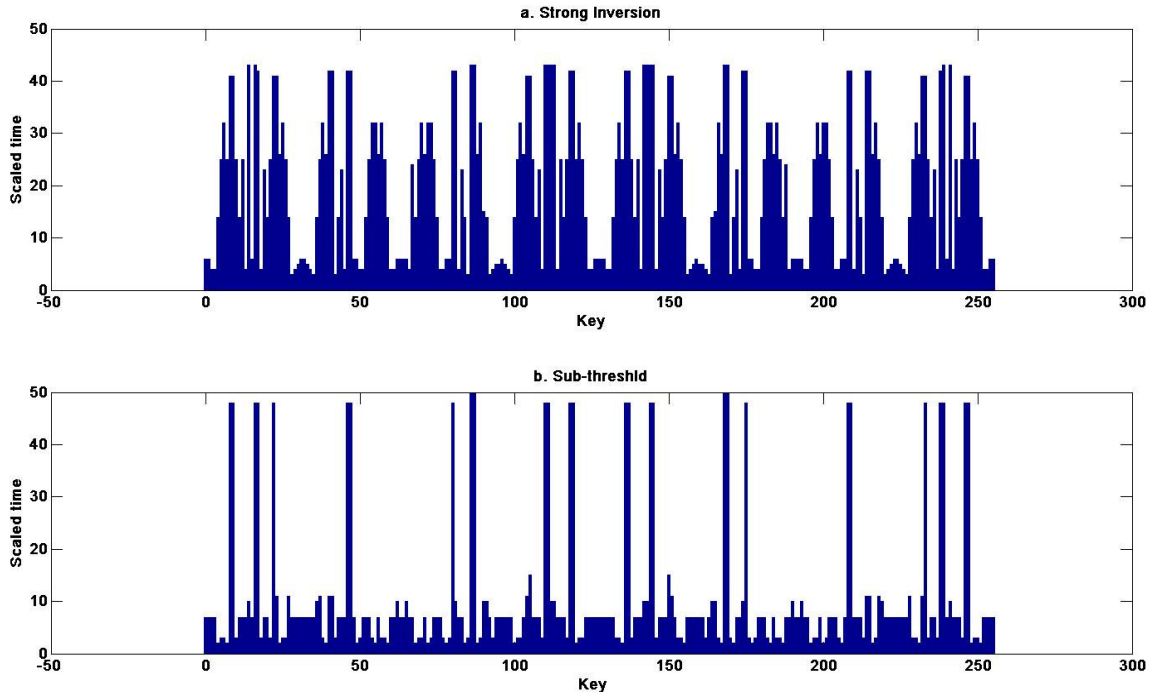


Figure 5.35: Time of occurring correlation for each key guess and correct key of 5D, for a) Strong Inversion b) Sub-threshold.

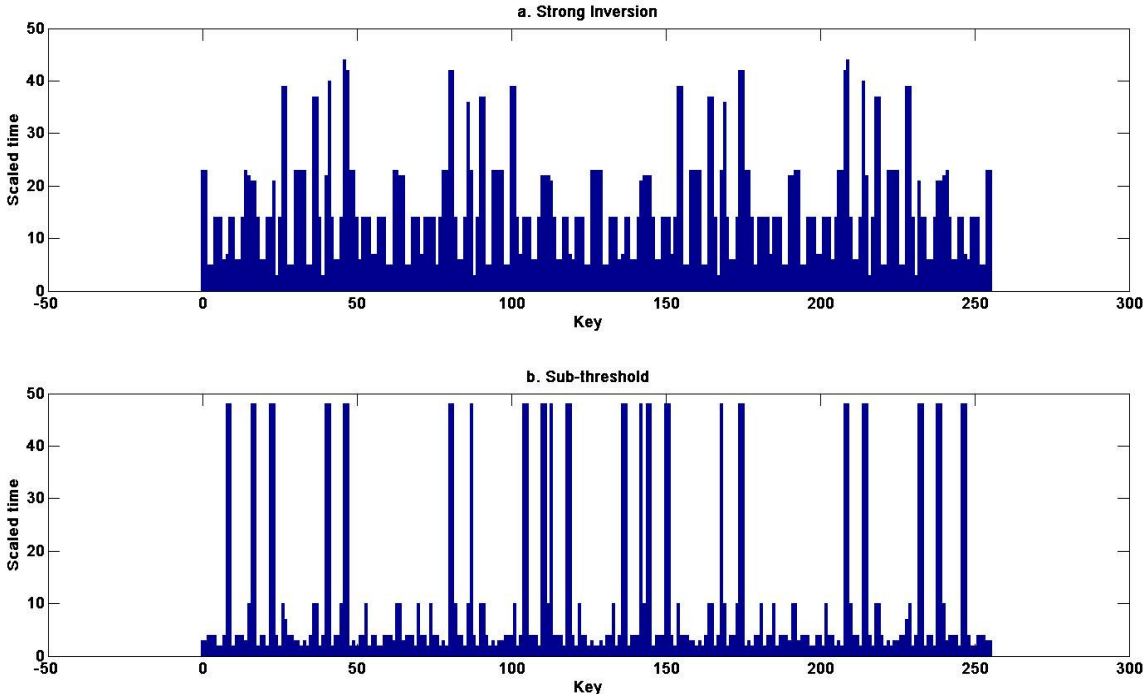


Figure 5.36: Time of occurring correlation for each key guess and correct key of 5C, for a) Strong Inversion b) Sub-threshold.

Figures 5.35 and 5.36 demonstrate that time steps for sub-threshold is more than those for strong inversion. Since the scaled time is actually the sample numbers, more time steps in strong inversion might have happened because of more glitches around the correlation time and, thus result in more recorded samples by the Cadence. When more samples are recorded, correlation time variations may increase.

5.6 Correlation Power Analysis Attack

So far, all mentioned measures were applied to the transistor level designs. Hence, the only required tool to extract power traces is Cadence Virtuoso Analog Design Environment. From here on, S-Box block and Advanced Encryption Standard core will be the focus of attacks and studied for their amount of information leakage. Since custom designing these architectures is a time consuming process, the goal of this project is to employ ASIC design methodologies, described in section 3.2, to implement these complex architectures and perform the power analysis study on them.

Designs in strong inversion are successfully implemented and power consumption traces are obtained and studied by CPA. Indeed most challenging part of this research was setting up the tools required to synthesize designs at sub-threshold. Intense and continuous efforts were made to gather information as well as to set up and run the required tools. Despite the monumental efforts, various

bugs and difficulties, such as library issues and Spectre simulation failure challenged the setting up of the Cadence Encounter Library Characterizer. Issues and errors were referred to CMC Microsystems and Cadence. After a few months of follow-up, Cadence informed us that dp_spice commands fails due to some raw directory issues. A Cadence Change Request was submitted to the R&D team for troubleshooting, but unfortunately, this process took a long time. Therefore, we could not characterize the available standard cell library for sub-threshold operation and have not been able to attain any simulation result from sub-threshold logic at this time. Simulation results of strong inversion logic will be provided here and hopefully, in the near future, we will be able to add sub-threshold results to this research to complete the work. More details about the ELC issues are provided in Appendix A.

The next section presents the strong inversion results of S-Box and AES results will follow in section 5.6.2.

5.6.1 S-Box Analysis with CPA Measure

The architecture of the S-Box block described in Section 3.3.4 is the second architecture attacked with CPA. The architecture simulated with the testbench presented in section 4.4.1 and the Hamming distance model is used to acquire the correlation trace, Corr_trace, as explained in section 4.4.3.

Like the CPA attack on Parallel XORs architecture, the key with the highest peak value compared to other keys is the correct key. The CPA attack mounted successfully on S-Box and the detected key was exactly equal to the correct key for various tested keys. Figure 5.37 demonstrates the maximum correlation trace and shows, 5C as the correct key. Figure 5.38 shows the correlation trace corresponding to the correct key.

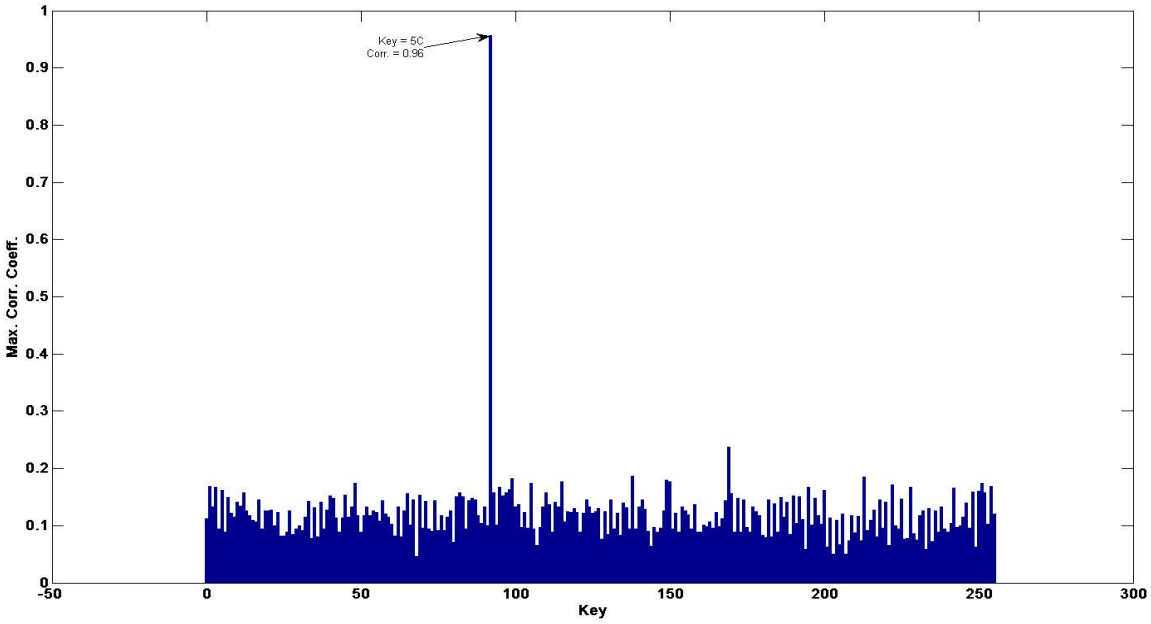


Figure 5.37: Maximum Correlation Coefficients for *correct key* = 5C.

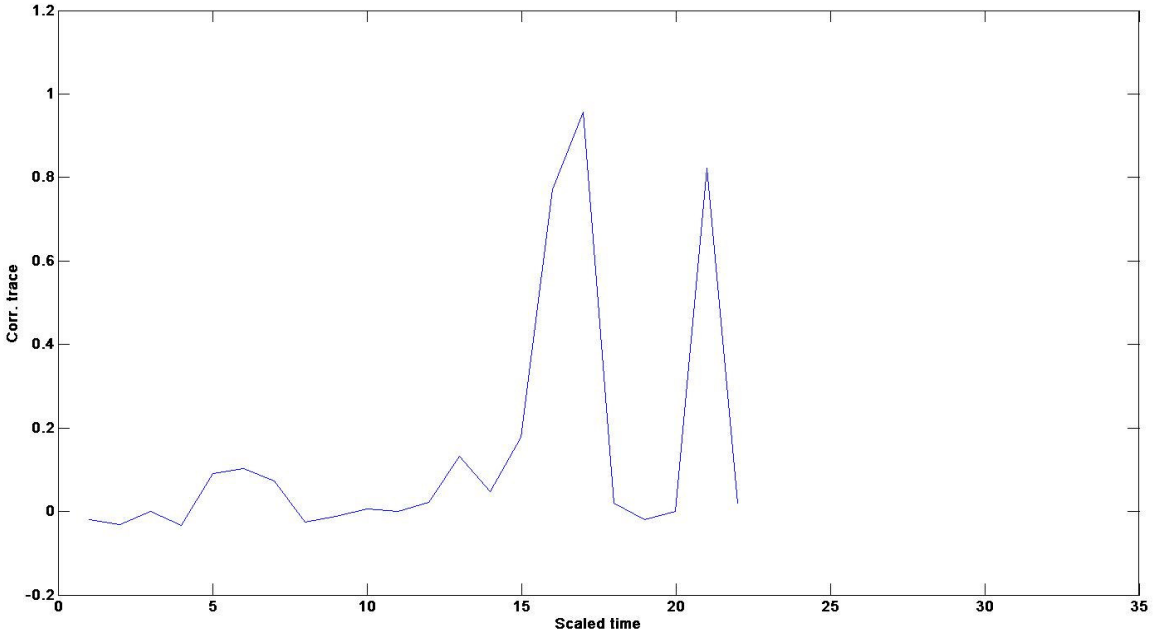


Figure 5.38: Correlation coefficient trace corresponding to the correct key, 5C.

5.6.2 AES Analysis with CPA Measure

The AES architecture described in section 3.3.3 is the last architecture attacked with CPA. The architecture simulated with the testbench presented in section 4.4.1 and Hamming weight model is used to acquire the correlation trace, Corr_trace, as explained in section 4.4.3.

The first difference between the attack on S-Box and AES is the power model. For S-Box, the model is the Hamming distance, and for AES, the model is Hamming Weight. The second difference is the size of the power traces that is significantly larger for AES, so, the attack takes longer to complete compared to S-Box. Figure 5.39 demonstrates the maximum correlation trace. The correct key, 5C, is also determined. Figure 5.40 shows the correlation trace corresponding to the correct key.

The difference in maximum correlation between S-Box and AES is also noticeable. It is 0.96 for S-Box while, it is about the half, 0.5, for AES. This shows that implementing an attack on a full AES core requires more measurements, traces and time.

The MATLAB Code written for the CPA attack on AES is provided in Appendix B. Parallel XORs and S-Box's codes are similar to AES's Code, with a slight difference in the attack model, and the size of traces. There are also some differences in importing traces into MATLAB between AES and Parallel XORs, since AES traces come from PrimeTime PX while Parallel XORs traces come from Cadence. However, changes are easy to make and there is no need to provide an individual code for each architecture.

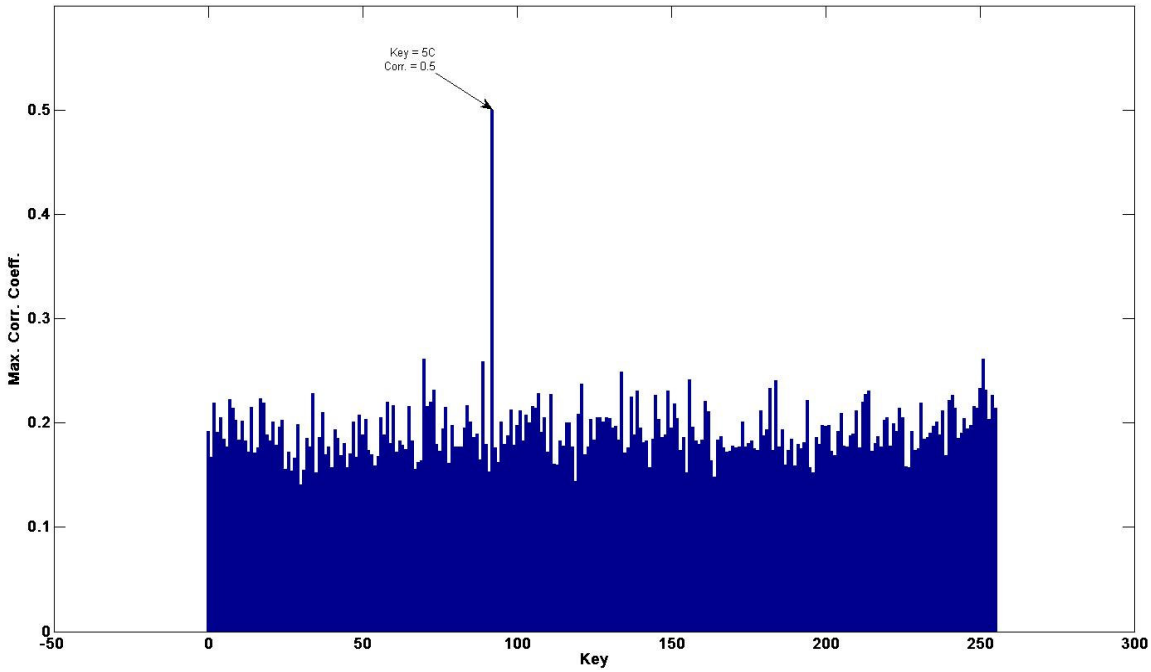


Figure 5.39: Maximum Correlation Coefficients for *correct key* = 5C.

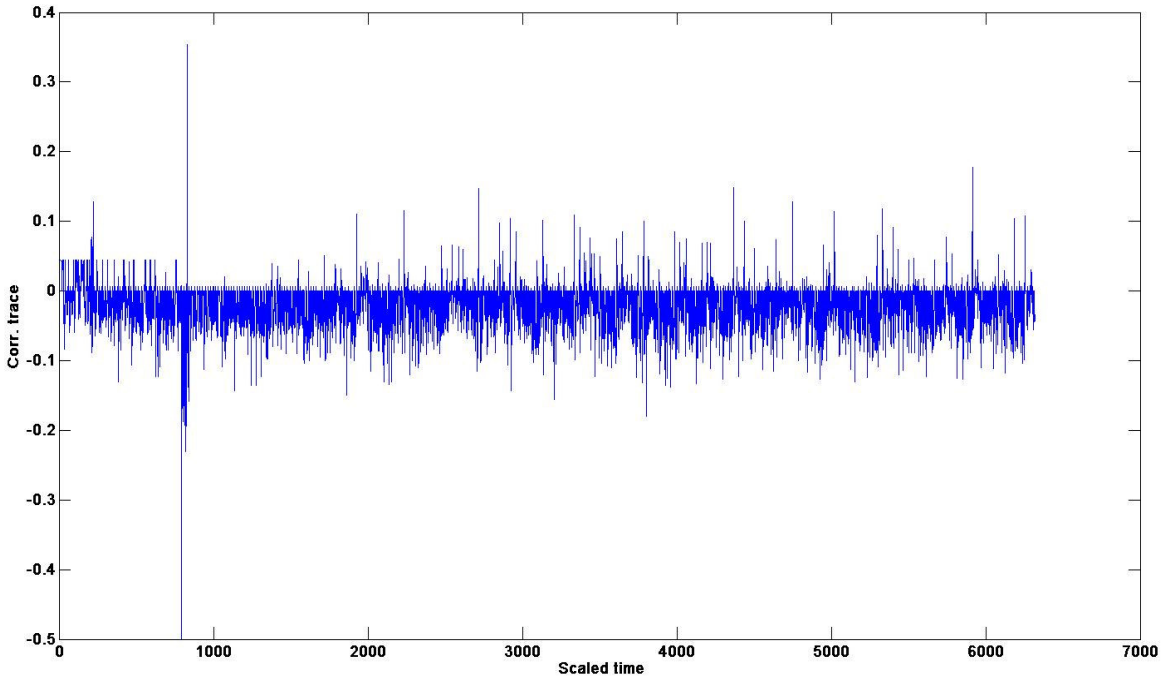


Figure 5.40: Correlation Coefficient trace corresponding to the correct key, 5C.

Chapter 6
Conclusions and Future Works

6.1 Summary and Discussion

In this thesis, side channel information leakage of the sub-threshold logic scheme against power analysis attacks was explored and compared with strong inversion logic, as the standard logic of operation, and against Sense Amplifier Based Logic (SABL) and Wave Dynamic Differential Logic, as the two of most referenced side channel resistant logics.

An XOR gate was analyzed with difference of mean energies (DME) measure to provide an overview of power consumption behavior at sub-threshold operation. Instantaneous power consumption waveforms in different transitions and with keys of 0 and 1 were obtained. Comparisons showed not only that the power level of a circuit at sub-threshold region is significantly lower than strong inversion's, but also that its power traces are more similar in different transitions and have less variation. In order to provide more detailed comparisons, DME measure was applied on power traces in experiments with fixed data transitions and fixed keys. In nearly all cases, the XOR gate showed less power variation and hence, less information leakage in sub-threshold operation.

NAND, NOR and XOR gates were also studied with frequency of observation, NED, and NSD measures. They were simulated with all possible input transitions to obtain the averaged power consumption for each transition. Despite the XOR gate measured with DME, instantaneous power trace was not used, but, the power consumption was averaged in each clock cycle to reflect the averaged power value for each transition. Frequency of observation was used as a visualized measure and showed the dispersion of averaged power values. As expected, SABL and WDDL presented small dispersion in power values in comparison to strong inversion and sub-threshold. However, they consume more energy which is their major drawback. Sub-threshold showed less dispersion in power values and hence, more security compared to strong inversion. Lower values of NED and NSD were obtained for sub-threshold compared to strong inversion, thus re-emphasizing its lower information leakage. Nevertheless, the sub-threshold cannot compete with SABL and WDDL.

Parallel XORs architecture was the next architecture tested. It was first analyzed with DME measure, after which power consumption traces were separated based on the lsb of input plaintext into two groups. Difference between the mean of these groups demonstrated two points: that the DME of sub-threshold is closer to zero on average (which shows better security) and, that DME signals for a key of bit value 1 and a key of bit value 0 are more similar compared to strong inversion. Once again this result leads to lower leakage of key information.

Subsequently, the correlation power analysis (CPA) attack was performed on parallel XORs architecture. The CPA attack was able to reveal the correct key in both sub-threshold and strong inversion; however, maximum correlation coefficients corresponding to the correct key and most of the guessed keys were lower for the circuit operating at sub-threshold. The experiment was executed with two key bit values, one with lsb of 0 and the other with lsb of 1. The difference between

maximum correlation coefficient traces for the lsb of key equals 0 and lsb of key equals 1 showed that strong inversion is about three times more than sub-threshold's. Indeed, all experiments showed that although the simulated CPA attack breaks the device for both logics, the correlation between power consumption and inputs is less for sub-threshold than strong inversion.

The final architectures, S-Box and AES, were implemented using RTL design methodologies and attacked by CPA. Both architectures had a successful attack and their secret key was successfully revealed while operating in strong inversion. In order to perform the same simulations for sub-threshold logic, we needed to characterize the available standard cell library for sub-threshold operation. However, due to library issues and Spectre simulation failure in ELC, the characterization process was aborted. Consequently, implementation of the S-Box and AES could not be completed for sub-threshold and there were no result to compare with strong inversion. The ELC issues were investigated by Cadence, who announced they would have to resolve the issues in their R&D team, which would require some time. Hence, despite several months of effort, research, and follow-up, this part of the study remains open until the near future when updates from Cadence are available.

This research tried to improve previous research on side channel analysis of sub-threshold circuits. The study of NSD in this research showed that although lowering the supply voltage down to the sub-threshold region decreases the dispersion of averaged power consumption values and as a result, lowers the value of NSD, however, the observed reduction was not on the order of 2500, as claimed in [41, 42]. The standard deviation of each region of operation was normalized with the mean value of the same region to provide a fair comparison. NSD of studied gates in this research demonstrated 30%- 75% decrease while the order of magnitude remained the same. Compared to previous research [43], we considered a CPA attack instead of a DPA attack and also employed ASIC design methodologies rather than full csutom circuit design.

6.2 Conclusions and Future Works

All experiments in this research illustrated improved power consumption behavior from the side channel information leakage aspect in lowering the operating voltage from strong inversion region down to sub-threshold. Various used measures, such as DME, NED and NSD, represented lower dependency of power consumption on handled data and executed operation. Correlation power analysis also demonstrated that power consumption traces of a device at sub-threshold are less correlated to its inputs and secret key.

Sub-threshold circuits are also expected to show a great resistivity to power analysis attacks because of their current and energy level. As a circuit operating at sub-threshold consumes from 10 to 1000 times less energy and the level of its instantaneous power consumption is significantly lower than the same circuit operating in strong inversion, extracting information from their power consumption trace will be dramatically harder. Even if an attack is possible, it will require a much larger number of traces and more run time to accomplish the attack. On the other hand, a low level of power consumption requires less noise power to hide the secret information. So, white noise which is

the same for strong inversion and sub-threshold has a stronger effect on the information leakage at sub-threshold.

In summary, this research demonstrated that sub-threshold operation has improved security against side channel analysis, but also it can decrease the amount of leaked information. As sub-threshold circuits are of recent interest for their considerably lower energy consumption, especially for RFID and biomedical applications where security is a concern, their vulnerability against power analysis attacks must be explored. This research provides a first look at this unexplored area of sub-threshold circuits.

Various topics can still be studied in power analysis of sub-threshold logics. Three possible future topics in this area are described next.

Although it has been mentioned that low current level at sub-threshold leads to a dramatically harder or even impossible attack, a measure to quantize this statement based on available measurement equipment could not be found. Thus obtaining such a measure, as the first possible extension to this research, would be beneficial for a more precise evaluation of sub-threshold circuit.

In order to further investigate the sub-threshold logic behavior against power analysis attacks, the dependency of information leakage on voltage in the sub-threshold region could be studied. It is also interesting to examine if there is any optimum voltage to achieve the least amount of information leakage.

Finally, implementing proposed side channel resistant logics for sub-threshold operation could be an interesting topic of study. A major drawback of the previously proposed logics is their high amount of energy consumption. Therefore, a sub-threshold version of those logics is expected to demonstrate excellent side channel resistivity along with reasonably low energy consumption.

Appendix A
Encounter Library Characterizer's Issues

This appendix provides more details about the issues in ELC that are reported to CMC Microsystems and a Cadence Change Request submitted to Cadence R&D team.

First issue is with the 65 nm TSMC model library. This model library contains some include instructions that includes different section of the same file within the original file and causes re-definition error. Commenting the include section was the first approach that we chose and succeed to resolve that problem; however, that is not a recommended approach and the problem has to be resolved by Cadence or TSMC. A related given error is as follows:

- `[ERROR(db_prepare)] spice syntax error: NMOS_RF : redefinition of the subckt [ file = crn65gplus_2d5_lk_v1d0.scs, #line = 3924 ]`

```
 => subckt nmos_rf d g s b inl=ne
```

Next and more important error occurs during the SPICE simulation with Spectre, after db_spice command. The error is:

- `[WARNING(db_spice)]No spice simulation to do, please check the cell/process list for any error`

The db_prepare command works fine and creates the design and all vectors but db_spice cannot recognize the simulation. Unfortunately, ELC leaves the log folders empty in most run that makes the tracking impossible; however, the error is assumed to be as follows.

There are some parameters declared in the model file but when the ELC extract the pch and nch models in the db_prepare command, it does not copy those parameters value into the pch and nch model files, so they remain undefined. This problem does not show itself until db_spice command in which the spectre cannot complete the simulation because of those undefined parameters. One of the errors is provided below and all other errors are the same.

- ```
  ERROR (SFE-1999):
  "/home/username/ELC_Test/foo.ipdb/NCH.device/simulate/model" 4:
  model
  `nch.1': parameter `wmin': Unknown parameter name `dxwn' found in
         expression.
  ```
  ```

Appendix B
MATLAB Code for Correlation Power Analysis Attack on AES

```
% AES attack. Each plain text takes 3680ns and total is 942080ns

% Workspace that provides S-Box LUT
workspace = 'WS1.mat';
method = 'CPA';
load('-mat',workspace);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Importing power traces generated by PrimeTime PX & saved in a text
file
power_input_file='AESKey5C.txt';
power=0;
data=0;

% reading in the file
data = importfile1(power_input_file);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Seperating power traces and creating power_traces matrix introduced in
%%% Chapter 4
disp('Loading data ...');

[m,n] = size(data);

%%% Some error checking to make sure that the file is generated in the
%%% format that we expect the PrimeTime PX to generate. The expected
format is:
%%% Time,1,power_atTime,2,power_atTime (for one point of power trace)
error1=0;error2=0;
for i=1:m
    if (mod(i,5) == 2)
        if (data(i,n) ~= 1)
            error1 = error1 + 1;
        end
    end

    if (mod(i,5) == 4)
        if (data(i,n) ~= 2)
            error2 = error2 + 1;
        end
    end
end

p1=0;error3=0;
for i=1:m
    if (mod(i,5)==3)
        p1=data(i,1);
    elseif (mod(i,5)==0)
        if (p1 ~= data(i,1))
            error3 = error3 + 1;
```

```
        end
        p1 = 0;
    end
end
%%% end of error checking, error1,2,3 must be 0

%%% Creating power_traces and then aligning them based on the time matrix.
%%% power_trace is the time aligned version of power
start_time = 0;
end_time = 94208000;
cycle_cnst = 368000;
cycle = 0;
plain=1;time=1;data_time=0;
power_size = zeros(256,1);

for i=1:m
    if (mod(i,5) == 1)
        data_time = data(i,1);
    end
    if (mod(i,5) == 3)
        if (start_time <= data_time)
            if ((cycle<=data_time) && (data_time<(cycle+cycle_cnst)))
                power(plain,time) = data(i,1);
                time_trace(plain,time) = (data(i-2,1)) - cycle;
                time = time + 1;
                power_size(plain) = power_size(plain) + 1;
            else
                cycle = cycle + cycle_cnst;
                if ( end_time <= cycle) break; end
                plain = plain + 1;
                time = 1;
            end
        end
    end
end

[m,n] = size(time_trace);
time_values = 6320;
Times = zeros(time_values,1);
k=2;

for i=1:m
    for j=1:n
        if (time_trace(i,j) ~= Times)
            Times(k) = time_trace(i,j);
            k = k + 1;
        end
    end
end
Times = sort(Times);
```

```
[m,n] = size(power);
k=1;
power_trace = zeros(256,time_values);

for i=1:m
    k=1;
    for j=1:time_values
        if (time_trace(i,k) == Times(j,1))
            power_trace(i,j) = power(i,k);
            k = k + 1;
        else
            power_trace(i,j) = 0;
        end
    end
end

%%% End of power_traces creation.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Power model matrix (predicted_power) matrix generation based on
Hamming
%%% Weight model and Correlating it with power_trace to create Corr_trace
%%% matrix that we know from Chapter 4.

traces = power_trace;

b=1;
inputs = [0:255];

disp('Predicting intermediate values ...');
[m,n] = size(traces);

key = [0:255];
after_sbox = zeros(m,256);

for i=1:m
    after_sbox(i,:) = SubBytes(bitxor(inputs(i),key)+1);
end



Corr_trace = zeros(256,n);

% predict the power consumption

disp('Predicting the instantaneous power consumption ...');
predicted_power = byte_Hamming_weight(after_sbox+1);

% correlate the predicted power consumption with the real power
```

```
% consumption
disp('Generating the correlation traces ...');

for i=1:n
    for j=1:m
        cmatrix=corrcoef(traces(:,i),predicted_power(:,j));
        Corr_trace(j,i)=cmatrix(1,2);
    end
end

%%% Finding the Key by finding the maximum correlation coefficient in
Corr_trace
disp('Finding the Key ...');

max = zeros(m,1);
for i=1:m
    for j=1:n
        if (max(i,1) < abs(Corr_trace(i,j)))
            max(i,1) = abs(Corr_trace(i,j));
        end
    end
end

detected_key=0;
max_value=0;
for i=1:256
    if (max_value < max(i,1))
        max_value = max(i,1);
        detected_key = i-1;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Plotting the results
x=[0:255];
fprintf('Detected Key = %x\n',detected_key);
fprintf('Max Corr = %d\n',max(max_CorrCoeff));
fprintf('Mean of Max Corr Coef = %d\n',mean(max_CorrCoeff));

% Plot maximum correlation coefficients trace and differnece of them
figure;
bar(x,max_CorrCoeff);

% Plot Time at which maximum correlation of each key is happened
figure;
bar(x,CorrTime);

% Plot correlation trace corresponding to the correct key
figure;
plot(key_trace(detected_key+1,:));
```

Bibliography

[1] B. H. Calhoun and D. Brooks, "Can Subthreshold and Near-Threshold Circuits Go Mainstream?," *IEEE Micro*, vol.30, no.4, pp.80-85, 2010.

[2] Roger Allen "Embedded systems start living in a sub-threshold world", Energy Efficiency and Technology, *April 2012 at* http://eetweb.com/Embedded-systems-start-living-sub-threshold-world/, Penton Media.

[3] C. Hocquet, D. Bol, D. Kamel, J.-D. Legat "Assessment of 65 nm subthreshold logic for smart RFID applications", in *Proc. FTFC*, 2009.

[4] J. Kwong, Y.K. Ramadass, N. Verma, A.P. Chandrakasan, "A 65 nm sub-V_t microcontroller with integrated SRAM and switched capacitor DC-DC converter," *IEEE J. of Solid-State Circuits*, vol.44, no.1, pp.115-126, 2009.

[5] Z. Bo, S. Pant, L. Nazhandali, S. Hanson, J. Olson, A. Reeves, M. Minuth, R. Helfand, T. Austin, D. Sylvester, D. Blaauw, "Energy-Efficient Subthreshold Processor Design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* , vol.17, no.8, pp.1127-1137, 2009.

[6] J. Dongsuk, S. Mingoo, C. Chakrabarti, D. Blaauw, D. Sylvester, "A Super-Pipelined Energy Efficient Subthreshold 240 MS/s FFT Core in 65 nm CMOS," *IEEE Journal of Solid-State Circuits,* vol.47, no.1, pp.23-34, Jan. 2012.

[7] P. Yu, J.P. de Gyvez, H. Corporaal, H. Yajun, "An ultra-low-energy/frame multi-standard JPEG co-processor in 65 nm CMOS with sub/near-threshold power supply," *IEEE International Solid-State Circuits Conference - Digest of Technical Papers.*, 2009, pp.146-147.

[8] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," in *Proc. of 19th International Advances in Cryptology Conference – CRYPTO '99*, 1999, pp. 388–397.

[9] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: concrete results," in *Proc. of Cryptographic Hardware and Embedded Systems*. Lecture Notes in Computer Science, vol. 2162, 2001, pp. 251–261.

[10] P. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," *in Proc. of CRYPTO '96*, vol. LNCS 1109, 1996, pp. 104-113.

[11] National Istitude of Standard and Technology (NIST), *Advanced Encryption Standard (AES)*, 2001. FIPS-197.

[12] M. B. Barron, "Low Level Currents in Insulated Gate Field Effect Transistors," *Solid-State Electronics*, vol. 15, pp. 293-302, 1972.

[13] A. Wang, B. H. Calhoun, and A. P. Chandrakasan, *Sub-threshold Design for Ultra Low-Power Systems*, Springer, 2006.

[14] R. M. Swanson and J. D. Meindl, "Ion-implanted complementary MOS transistors in low-voltage circuits," *IEEE Journal of Solid-State Circuits*, vol.7, no.2, pp. 146- 153, 1972.

[15] H. Soeleman, K. Roy, and B. C. Paul, "Robust subthreshold logic for ultra-low power operation," *IEEE Transaction on Very Large Scale Integration (VLSI) Systems*, vol.9, no.1, pp.90-99, Feb. 2001.

[16] A. Wang, A. P. Chandrakasan, and S. V. Kosonocky, "Optimal supply and threshold scaling for subthreshold CMOS circuits," in *Proc. of IEEE Computer Society Annual Symposium on VLSI*, 2002, pp.5-9.

[17] B. H. Calhoun, A. Wang, and A. P. Chandrakasan, "Device sizing for minimum energy operation in subthreshold circuits," in *Proc. of the IEEE Custom Integrated Circuits Conference*, 2004, , pp. 95- 98.

[18] B. H. Calhoun and A. P. Chandrakasan, "Characterizing and modeling minimum energy operation for subthreshold circuits," in *Proc. of the International Symposium on Low Power Electronics and Design*, 2004, pp.90-95.

[19] B. H. Calhoun, A. Wang, and A. P. Chandrakasan, "Modeling and sizing for minimum energy operation in subthreshold circuits," *IEEE Journal of Solid-State Circuits*, vol.40, no.9, pp. 1778- 1786, 2005.

[20] D. Bol, B. Kamel, D. Flandre, and J. D. Legat, "Nanometer MOSFET effects on the minimum-energy point of 45nm subthreshold logic," in *Proc. of the International Symposium on Low-Power Electronics and Design*, 2009.

[21] C. C. Enz, F. Krummenacher, and E. A. Vittoz, "An analytical MOS transistor model valid in all regions of operation and dedicated to low voltage and low-current applications," *Analog Integrated and Circuits Signal Process.*, vol. 8, pp. 83–114, 1995.

[22] T. Grotjohn and B. Hoefflinger, "A parametric short-channel MOS transistor model for subthreshold and strong inversion current," *IEEE Transactions on Electronic Devices*, vol.31, no.2, pp. 234- 246, 1984.

[23] S. Fisher, A. Teman, D. Vaysman, A. Gertsman, O. Yadid-Pecht, A. Fish, "Digital subthreshold logic design - motivation and challenges," in *Proc. of IEEE 25th Convention of Electrical and Electronics Engineers in Israel*, 2008, pp.702-706.

[24] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks, Revealing the Secret of Smart Cards*, Springer, 2007.

[25] S. Mangard, "A simple power-analysis (SPA) attack on implementations of the AES key expansion," In *Proc. of the 5th International Conference on Information security and cryptology*, vol. LNCS 2587, pp. 343-358, Springer-Verlag, 2002.

[26] R. Mayer-Sommer, "Smartly analysing the simplicity and the power of simple power analysis on smartcards," *Cryptographic Hardware and Embedded Systems*, vol. LNCS 1965, pp. 78–92, Springer-Verlag, 2000.

[27] T. H. Le, J. Cle´die`re, C. Canovas, B. Robisson, C. Servie`re, and J. L. Lacoume, "A Proposition for Correlation Power Analysis Enhancement," in *Proc. of Int'l Workshop Cryptographic Hardware and Embedded Systems*, 2006, pp. 174-186.

[28] T. S. Messerges, "Using second-order power analysis to attack DPA resistant software," in *Cryptographic Hardware and Embedded Systems*, vol. LNCS 1965, pp. 238–252, Springer-Verlag, 2000.

[29] M. L. Akkar, R. B´evan, P. Dischamp, and D. Moyart, "Power analysis, what is now possible...," in *Advances in Cryptology — ASIACRYPT*, LNCS 1976, pp. 489–502, Springer-Verlag, 2000.

[30] R. B´evan and R. Knudsen "Ways to enhance differential power analysis," in *Information Security and Cryptology*, vol. LNCS 2587, pp. 327–342, Springer-Verlag, 2002.

[31] E. Brier, C. Clavier, and F. Olivier, “Correlation power analysis with a leakage model,” in *Cryptographic Hardware and Embedded Systems*, vol. 3156 of *Lecture Notes in Computer Science*, pp. 16–29, Springer-Verlag, 2004.

[32] H. Li, K. Wu, B. Peng, Y. Zhang, X. Zheng, F. Yu, "Enhanced Correlation Power Analysis Attack on Smart Card," in *9th International Conference for Young Computer Scientists*, 2008, pp.2143-2148.

[33] J.-S. Coron, P. Kocher, and D. Naccache, “Statistics and secret leakage,” in *Financial Cryptography*, vol. LNCS 1972, pp. 157–173, Springer-Verlag, 2001.

[34] T. Popp and S. Mangard, “Masked Dual-Rail Pre-Charge Logic: DPAResistance without Routing Constraints,” in *Proc. of 7th International Workshop Cryptographic Hardware and Embedded Systems*, 2005.

[35] K. Tiri, M. Akmal, and I. Verbauwhede, "A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards," in *Proc. of the 28th European Solid-State Circuits Conference*, 2002, pp.403-406.

[36] K. Tiri, I. Verbauwhede, "A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation," in *Proc. of Design, Automation and Test in Europe Conference and Exhibition*, 2004, pp. 246- 251.

[37] M. W. Allam, M. I. Elmasry, "Dynamic current mode logic (DyCML): a new low-power high-performance logic style," *IEEE Journal of Solid-State Circuits*, vol.36, no.3, pp.550-558, 2001.

[38] I. Hassoune, F. Mac´e, D. Flandre, and J. D. Legat, “Low-swing current mode logic (LSCML): a new logic style for secure smart cards against power analysis attacks,” *Microelectronics Journal* no. 37, pp. 997–1006, 2006.

[39] T. Popp and S. Mangard, “Masked Dual-Rail Pre-Charge Logic: DPA-Resistance Without Routing Constraints,” in *Cryptographic Hardware and Embedded Systems,* vol. LNCS 3659, pp. 172–186. Springer-Verlag, 2005.

[40] F. Mac´e, F.-X. Standaert, and J.-J. Quisquater “Information theoretic evaluation of side-channel resistant logic styles” In *Cryptographic Hardware and Embedded Systems,* vol. LNCS 4727, pp. 427–442. Springer-Verlag, 2007.

[41] H. P. Alstad and S. Aunet, "Improving Circuit Security against Power Analysis Attacks with Subthreshold Operation," in *11th IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems*, 2008, pp.1-2.

[42] H. P. Alstad, S. Aunet, "Subthreshold AES S-Box with Increased Power Analysis Resistance," in *The Nordic Microelectronics event*, 2008, pp.13-16.

[43] S. I. Haider, L. Nazhandali, "Utilizing sub-threshold technology for the creation of secure circuits," in *IEEE International Symposium on Circuits and Systems*, 2008, pp.3182-3185.

[44] R. Vaddi, S. Dasgupta, and R. P. Agarwal, “Device and Circuit Design Challenges in the Digital Subthreshold Region for Ultralow-Power Applications,” *VLSI Design*, vol. 2009, pp. 1-14, 2009.

[45] Cadence, Inc., *Cadence Encounter Library Characterizer Datasheet*. http://www.cadence.com/rl/Resources/datasheets/library_characterizer_ds.pdf

[46] E. Brunvand, *Digital VLSI Chip Design with Cadence and Synopsys CAD Tools*, Addison-Wesly, 2010.

[47] Synopsys, Inc., *Synopsys DC Ultra Datasheet*. http://www.synopsys.com/Tools/Implementation/RTLSynthesis/DCUltra/Documents/DCUltra-ds.pdf

[48] Synopsys, Inc., *Synopsys PrimeTime Datasheet* http://www.synopsys.com/Tools/Implementation/SignOff/Documents/primetime_ds.pdf

[49] Mentor Graphics, Inc., *Company Web site* http://www.mentor.com/products/fv/modelsim/

[50] F. Haghighizadeh, H. Attarzadeh, and M. Sharifkhani, "A Compact 8-Bit AES Crypto-processor," in *Second International Conference on Computer and Network Technology*, 2010, pp.71-75.

[51] B. K¨opf, D. Basin, “an Information Theoretic Model for Adaptive Side-Channel Attacks,” in *The proceedings of ACM Conference on Computer and Communication Security*, 2007.