Side Channel Information Leakage: Design and Implementation of Hardware Countermeasure

by

Amirali Khatib Zadeh

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2010

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

In contrast to classical cryptanalysis attacks, which utilize the mathematical weakness of cryptographic algorithms, attacks using side channel information focus on the properties of the actual circuits and chips implementing these algorithms. Deployment of Dynamic and Differential Logics (DDL) appears to be a promising choice for providing resistance against leakage of side channel information. However, the security provided by these logics is too costly for widespread area-constrained applications. Furthermore, implementation of a secure DDL-based countermeasure involves balancing the load at the differential outputs. Thus, a complex layout methodology is required which is not available in a standard design environment provided by the commercial CAD tools.

This thesis, unlike previous logic level approaches, presents a novel exploitation of static and single-ended logic for designing the side channel resistant logic cells and registers. The proposed technique is used in the implementation of a protected crypto core consisting of the AES “AddRoundKey” and “SubByte” transformation. The test chip including the protected and unprotected crypto cores is fabricated in 180nm CMOS technology. The effectiveness of the logic-based countermeasure is assessed by mounting a side channel attack on the test chip using real power measurements. A correlation-based analysis on the unprotected core results in revealing the keys at two attack points: the output of the combinational networks (“SubByte”) and the output of the registers. The quality of the measurements is further improved by introducing an enhanced data capturing method that inserts a minimum power consuming input as a reference vector. Results obtained from analysis of the unprotected core indicate that the reference vector approach increases the correlation coefficients. In comparison, a similar analysis of the protected core shows a significant reduction in the correlation coefficients, thus no key-related information is leaked even with an order of magnitude increase in the number of averaged traces. For the first time, fabricated chip results are used to validate a new logic level side channel countermeasure that offers lower area and reduced circuit design complexity compared to the DDL-based countermeasures.

This thesis also provides insight into the side channel vulnerability of the next generation of cryptosystems. The power consumption trends in sub-90nm CMOS technology nodes are examined from a side channel perspective using simulation results. In particular, the data dependency of leakage power is analyzed. The number of traces to disclose the key is seen to decrease by 35% from 90nm to 45nm CMOS technology nodes. Thus technology scaling will have a significant impact on increasing the side channel vulnerability of nanoscale cryptosystems. Further analysis shows that the temperature dependency of the subthreshold leakage mechanism has an important role in increasing the ability to attack future nanoscale crypto cores. For the first time, the effectiveness of a circuit-based leakage reduction technique is examined for side channel security. This investigation demonstrates that high threshold voltage transistor assignment improves resistance against information leakage by increasing the number of traces for key disclosure. The analysis initiated in this thesis is crucial for rolling out the guidelines of side channel security for the next generation of Cryptosystem-on-Chip.

Acknowledgements

I would like to express my gratitude to my advisor, Professor Catherine Gebotys for her interest, support and guidance during the hardship of this research.

I would also like to thank Professor Gordon Agnew, Professor David Nairn, and Professor Urs Hengartner, who were very generous with their time in agreeing to read and comment on my thesis.

I would like to extend my special thanks to Professor Dimitrios Serpanos, my external examiner, for his time and energy so thoughtfully devoted to this endeavor.

My thanks to Brian A. White, for his many useful discussions and his assistance in parts of programming sections of this research.

My sincere thanks go to Dr. Shahab Ardalan for his technical advices and Solmaz Ghaznavi for many fruitful discussions.

I would also like to thank Professor Glenn Gulak, the director of the Advanced Digital Systems Laboratory (ADSL) and Jeetendar Narsinghani at the University of Toronto for their supports and encouragements during the testing process.

The support from Canadian Microelectronic Corporation (CMC) is appreciated.

This research has been supported in part by the Natural Science and Engineering Council of Canada (NSERC). This support is greatly appreciated.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Motivation

Today, devices such as Personal Digital Assistants (PDAs), wireless handsets, and smart cards are widely used. The demands for low-end products are increasingly growing; by the beginning of 2010, over 6 billion embedded chip smart cards will be used worldwide and approximately 3.4 billion will be sold each year [1]. The installed base of credit and debit cards issued by banks, cellular phone Subscriber Identity Module (SIM) and identity cards issued by governments and other public organizations will grow so that by 2010 each exceed 1 billion [2].

With the popularity of embedded systems processing a vast amount of confidential information, a new important dimension in design has arisen, that of security. Nearly 52% of cell phone users and 47% of PDA users feel that security is the single largest apprehension preventing the successful deployment of next generation of mobile services [2]. Increasing proliferation of security concerns in recent years has created a compelling case for attaining security envisioned in specification of different applications. The ultimate goal of all these attempts has been directed towards developing secure on/off chip communication. Hence, design and implementation of cryptosystems and in specific Cryptosystem-on-Chip have drawn increased attention.

Exponential growths of computational power and astounding advances in technology that have spurred the development of the secure systems have also ushered in seemingly parallel trends in the sophistication of security attacks. It has been seen that exploiting security vulnerabilities in hardware implementation provides exclusive opportunities for eavesdroppers to invade or weaken the functional security measures [3]. Consequently, the theoretical strength of cryptographic algorithm is no longer the only primary concern. Security in design and implementation of hardware has been given an equal weighted goal as development of enhanced encryption algorithms. Unlike the traditional model assessing security solely on the mathematical strength of the encryption algorithms, the modern security model includes a new class of cryptanalytic threat known as “implementation attacks” [4]. Implementation attacks are also known as side channel attacks. These attacks are effectual techniques for extracting critical information (e.g., key of encryption algorithm) from the physical properties such as time delay [5], power consumption [6] and electromagnetic emanation [7]. Among the known side channels, power consumption has attracted the most attention due to its ease of access and its intrinsic data dependency characteristic. The feasibility of launching power consumption-based attacks has also been examined against different types of Very Large Scale Integration (VLSI) implementation of cryptographic modules e.g., Application Specific Integrated Circuits (ASICs) [8], Field Programmable Gate Arrays (FPGAs) [9], Digital Signal Processors (DSPs) [10] and Smart Cards (SCs) [11]. The severity of the threat posed by power-based attack has already been proven [6].

To overcome the threat posed by power consumption, various software-based solutions have been proposed [12][13][14]. The software security features often overload the processing capabilities of embedded systems; therefore, significant overheads are imposed to crypto core for keeping up with computational demands [15]. The shortcomings of software countermeasures lead to an undesirable trade-off between security and performance. In the same context, hardware approaches are also recognized as they significantly increase the resistance against side channel threat such as Differential Power Analysis (DPA). It is shown that side channel security can be achieved at the hardware level if the design criteria for secure implementation are properly defined and objectively met [6]. Hardware-based techniques offer a more flexible trade-off model as they provide resistance at various design levels (e.g., system, architectural and logic levels). Therefore, although side channel security is a costly feature, hardware techniques are seen to be able to adjust the security cost within the limited trade-off budgets. The effectiveness of the hardware countermeasures and their adaptable trade-off schemes make them more appealing than the software countermeasures.

The above discussion implies that design for security is a growing demand. Serious consideration should be given to the emerging security threat such as side channel information leakage. Efficiency and effectiveness should be dealt as two important aspects in design and implementation of side channel countermeasures. Efficient side channel protection can be achieved by implementation of hardware countermeasure providing sufficient security within the limitation of resource-constrained applications.

As the technology advances many design methodologies have been influenced and often need to be modified. In order to devise new mechanisms or to assess the efficiency of the current countermeasures for security of future cryptosystems, it is essential to illustrate the impact of technology on leakage of side channel information. It is crucial to investigate whether or not side channel security gains a new definition with drastic changes on power consumption trends. Conclusive outcomes will lead to introducing more efficient and effective security mechanisms for upcoming generation of cryptosystems.

In the light of great importance to security and in response to the increasing demand for low cost side channel protection for area-constrained applications, a novel exploitation of static and single-ended (non-differential) logic is presented. Furthermore, this thesis provides insight into the side channel vulnerability of next generation of Cryptosystem-on-Chip.

1.2 Thesis Organization

This thesis is organized as follows: an overview of information leakage via power consumption is presented in Chapter 2. The common techniques for obtaining and analyzing the side channel information are also described in this chapter. Chapter 3 discusses the fundamentals of side channel protection. The most effective approaches in tackling the security threat posed by power consumption are reviewed. The latest progress in the area of side channel security is discussed in this chapter. Chapter 4 presents our proposed side channel countermeasure. Exploitation of the Current Balanced Logic (CBL) for designing logic gates and register elements is presented. A comparative analysis is included in this chapter to quantify the cost of security provided by our countermeasure. Chapter 5 describes the design and implementation of the test chip which includes protected and unprotected cores. The pre-test hardware and software preparations, instrumentation selection, and experimental setup are described. Assessment procedures including the functionality test and side channel attack are discussed. An enhanced data capturing method is also introduced. The empirical results obtained from side channel attack on the test chip are presented. A comparison between the proposed countermeasure and the previous side channel resistant logics is conducted in this chapter. Chapter 6 provides a projection of the future side channel threat with regards to CMOS technology trends. Information leakage via leakage power consumption is quantified over the technology nodes. The effectiveness of using a leakage reduction technique for side channel security is investigated. Chapter 7 presents a summary of this research, itemizes the contributions, and draws the direction of future work.

Chapter 2

Side Channel and Analysis Techniques

2.1 Introduction

A new class of cryptanalysis has been formed by utilizing the correlations that exist between the data processed by a cryptosystem and a side channel such as the power consumption of a hardware implementation. Since its discovery, the side channel threat has drawn growing attention in adversarial actions. Several analysis techniques have been developed facilitating the extraction of information contained in side channels.

This chapter serves as background for the research presented later in this thesis. The elements of power consumption in CMOS logic are described. An overview of the analysis techniques which are used in side channel attacks is also presented.

2.2 Side Channel Effects

Crypto cores are designed to map the inputs (plaintexts) to the outputs (ciphertexts) based on a predefined function using the key values. The security breaches are often reduced to the methods that expose the value of the secret keys. In the past, the focus of attackers was on exploiting the mathematical weaknesses of cryptographic algorithms for attaining the key or related information to the key. Since almost a decade ago the attention has been shifted towards extracting the critical data contained in the physical characteristics of hardware implementation, which are known as side channels. Side channels are unintentional sources that contain the signature of data transitions which occur in the crypto core. This information can lead to disclosure of secret key. The commonly used side channels are time [5], power consumption [6], electromagnetic emanations [7], and very recently acoustic [16], among which power consumption has drawn the most attention. The power consumption of a cryptosystem is seen to be highly correlated with intermediate data being processed. Moreover, the ease of access increases the popularity of the power consumption as an effective side channel. This section reviews the main sources of power consumption in digital circuits.

2.2.1 Source of Power Consumption

Complementary Metal Oxide Semiconductor (CMOS) has prevailed as the preferred choice for implementation of digital circuits. Today most digital circuits are built using CMOS technology [17]. The popularity of this technology has grown because of its low power consumption and robustness. Meanwhile, CMOS technology is also advantageous due to its low-cost and high-integration compared to bipolar process. The side channel effect in power consumption is mainly caused by the use of CMOS logic, which is the main source of information leakage. In order to explain why the power consumption of a CMOS logic gate can reveal information about the data being processed, the major components of power dissipation in CMOS circuits are first reviewed.

There are three distinct dissipation sources in CMOS logic gates, which are known as switching, short circuit and leakage power consumption [17]. The first two components are referred to as dynamic power consumption and the last one is known as static power consumption. These components are described. The expressions quantifying these elements in a typical CMOS inverter are presented.

Switching power is defined as the power consumed by the logic gate to charge the output load from ‘0’ to ‘1’. Switching power of an inverter is expressed as [17]:

$$P_{switching} = \alpha \, . \, f_{clk} \, .V_{dd} \, .V_{swing} \, .C_L \tag{2.1}$$

where α is the switching activity factor, f_{clk} is the operating frequency, V_{dd} is the supply voltage, and V_{swing} is the logic swing of the digital output. C_L is the net load capacitance, which consists of the gate capacitance of subsequent cell(s) input(s), interconnect capacitance and the diffusion capacitance of the drain of inverter transistors.

Short circuit power is a result of the transient current that flows from V_{dd} to *ground* when both NMOS and PMOS transistors are turned on during logic transitions. The non-zero rise and fall times of the input signal generate this direct path. An expression quantifying the short circuit power consumption in a CMOS inverter under assumptions of $V_{th} = V_{th\,p} = V_{th\,n}$ and $k = k_p = k_n = \mu_{p/n}\, C_{ox}\, (W/L)$ is given by [17]:

$$P_{shortcircuit} = \frac{k}{12}(V_{dd} - 2V_{th})^3 . \tau . f_{clk} \tag{2.2}$$

where V_{th} is the threshold voltage, k is gain factor, $\mu_{p/n}$ is the hole and electron mobilities, C_{ox} is the oxide capacitance per unit area, W and L are the width and the length of MOS transistor, and τ is the rise/fall time of the input signal.

Static power is consumed during the steady state when no transitions occur. A conventional CMOS gate dissipates no static power since no path exists between V_{dd} and *ground* in the steady state. However, as V_{th} is scaled down, the static power consumption caused by the leakage effects grows. Due to the increasing importance of leakage power from a side channel standpoint the major leakage mechanisms are described. The explanation is necessary as it is referred to later in Chapter 6 of this thesis where feasibility of using leakage power as an emerging side channel is investigated.

Several leakage mechanisms have been developed as the short-channel effect becomes more pronounced in submicron technology. Figure 2.1 demonstrates the three major leakage mechanisms. One of the leakage generation sources is weak inversion current, which is also known as subthreshold conduction current *(I_{sub})*. When the gate voltage of the transistor is below V_{th}, current *(I_{sub})* flowing between source and drain in a MOS transistor causes the carriers to move. Weak inversion typically dominates the modern transistor off-state leakage due to the low V_{th} that is used. I_{sub} is known as the dominant leakage source and it can be best approximated [18]:

$$I_{sub} = I_s . e^{q/(V_{gs} - V_{th_0} - \gamma V_{sb} + \eta V_{ds}).nkT} .(1 - e^{(-q.V_{ds})/kT}) \tag{2.3}$$

where I_s is the zero bias current and it is expressed as:

$$I_s = \mu_0 (\varepsilon_{ox} / t_{ox})(W_{eff} / L_{eff})(kT/q)^2 . e^{1.8} \tag{2.4}$$

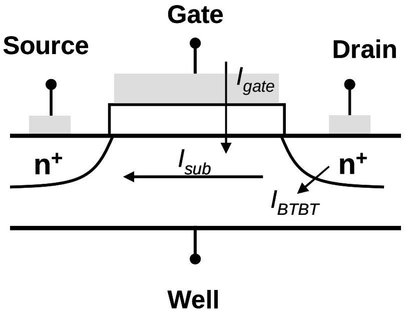


Figure 2.1 Short channel MOS leakage mechanisms [18]

μ_0 is the zero bias mobility, ε_{ox} (equals 3.97.ε_o) is the oxide permittivity, t_{ox} is the oxide thickness, W_{eff} and L_{eff} are the effective width and length of the MOS transistor, respectively. k is Boltzmann's constant, T is the operating temperature, q corresponds to the charge of an electron, $V_{th\,0}$ is the zero biased threshold voltage, γ is the linearized body-effect coefficient, η is the Drain-Induced Barrier Lowering (DIBL) coefficient, n is the subthreshold swing coefficient, and V_{gs}, V_{sb} and V_{ds} are the gate-source, substrate and drain-source voltages, respectively.

The second major leakage generation mechanism is a direct result of technology scaling. As the thickness of gate dielectric decreases below 2nm, the low oxide thickness combined with the increased electric field across the oxide results in significant electron tunneling from the substrate to the gate. Tunneling of electrons (holes) from bulk silicon *(I_{gb})*, channel *(I_{gc})*, and source/drain overlap region *(I_{gso})*/*(I_{gdo})* through the gate oxide potential barrier into the gate form the total gate leakage current *(I_g)*. I_{gso}, I_{gdo} and I_{gc} are the dominant gate leakage mechanisms in both 'on' and 'off' states of the transistor. The current density for direct gate tunneling is given [18]:

$$J_{tunnel} = (4\pi m^* q)/h^3(kT)^2.[1+(\gamma.kT/2)\sqrt{E_B}\,].e^{[(E_F/kT).\gamma\sqrt{E_B}} \qquad (2.5)$$

where m^* (equals $0.19.M_0$) is the electron transfer mass and M_0 is the electron rest mass, h is the Plank's constant, E_F is the Fermi level at the $Si\,/\,SiO_2$ interface, E_B is the height of barrier, and γ is defined:

$$\gamma = \frac{4\,\pi\,t_{ox}\,\sqrt{2m_{\;ox}}}{h} \qquad (2.6)$$

where m_{ox} equals 0.32 M_0 and denotes the effective electron mass in the oxide.

Another leakage mechanism caused by tunneling of electrons from n source/drain to p substrate is known as Band-To-Band Tunneling (BTBT) leakage current. Tunneling occurs due to the reverse biased pn junction from valance band of p region to the conduction band of the n region in existence of high field across the junction. An approximation of BTBT leakage is as follows [18]:

$$I_{BTBT} = W_{eff}\,.A\,(\frac{E_j}{\sqrt{E_g}}).\,V_{ib}.e^{[\;B(E_g)^{3/2}/E_j\,]} \qquad (2.7)$$

where E_j is the average electric field on the side and bottom of the junction which are given as [18]:

$$E_{side} = [(2q.NDEP.NSD.(V_{ib} + V_{biside}))/\varepsilon_{Si}(NDEP + NSD)]^{1/2} \qquad (2.8)$$

$$E_{bottom} = [(2q.NSUB.NSD.(V_{ib} + V_{bibot}))/\varepsilon_{Si}(NSUB + NSD)]^{1/2} \qquad (2.9)$$

NDEP, *NSUB* and *NSD* are the channel doping concentrations at depletion edge, substrate doping and source/drain diffusion doping respectively, $V_{biside / bibot}$ are the built-in potential, E_g is the band-gap, and V_{ib} is the applied potential on source/drain with respect to bulk.

The primary discussion on sources of power consumption continues with evaluating the power consumption element from side channel perspective. The following discussion will assist in understanding of how the information signature remains on the power consumption.

2.2.2 Power Consumption: An Effective Side Channel

In order to demonstrate the side channel effect in power consumption, a complete data transition in a typical CMOS inverter is reviewed. This analysis is performed in general so that it can be applied to all CMOS logic gates. The different elements of dynamic power consumption in a CMOS inverter for all possible transitions are shown in Figure 2.2. As described in Section 2.2.1 one component of dynamic power consumption associated with gate operation is known as switching power. The other element of dynamic power is short circuit power which is caused by current flowing with the non-zero rise and fall times of input signals. The current is drawn from the supply by a CMOS logic gate when a ‘0’ to ‘1’ transition occurs at the output. During a ‘1’ to ‘0’ transition, the energy previously stored in C_L is dissipated, but aside from a short circuit current no power is drawn from the supply. Unlike the short circuit power which is consumed in transitions ‘1’ to ‘0’ and ‘0’ to ‘1’ at the output of the typical complementary CMOS gates, the switching power is only consumed in ‘0’ to ‘1’ transition. The absence of switching power during ‘1’ to ‘0’ transition results in forming asymmetric behavior in power consumption in CMOS gates. Figure 2.3 simplifies this concept by showing that the power consumption values are different in transitions resulting in ‘1’ than those resulting in ‘0’.

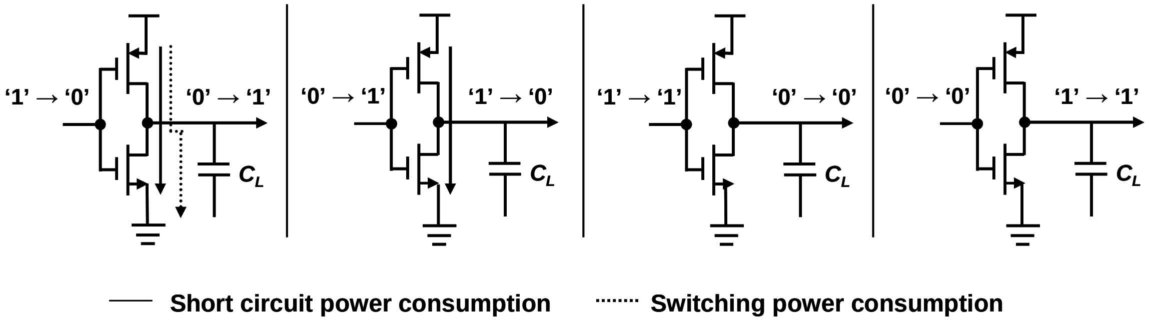


Figure 2.2 Power consumption in a typical CMOS inverter

Transition of node value	**Power Consumption**
‘0’ → ‘0’	P_{00}
‘0’ → ‘1’	$\mathbf{P_{01}}$
‘1’ → ‘0’	P_{10}
‘1’ → ‘1’	$\mathbf{P_{11}}$

$$\mathbf{P_{01}} \neq P_{10} \gg P_{00} \approx \mathbf{P_{11}} \Rightarrow P_{00} + P_{10} \neq \mathbf{P_{01} + P_{11}}$$

Figure 2.3 Transition of node value and corresponding power consumption

The variation of power consumption corresponding to different transitions highlights the data dependency of dynamic power consumption. The data dependency of switching power can also be seen in Equation 2.1, as $P_{switching}$ is shown to be a function of α (frequency of ‘0’ to ‘1’ transition). This asymmetry in switching power consumption of static CMOS logic, in fact, has formed the foundation of side channel effect in dynamic power consumption.

Static power consumption can also be included in side channel context. Leakage power as a source of static power consumption in CMOS logic gates is drastically increased in advanced technology. Equations 2.3, 2.5 and 2.7 in the previous section provide approximations on major leakage mechanisms. Parameters such as V_{gs}, V_{ds}, V_{ib}, in Equations 2.3 and 2.5 show the data dependency of subthreshold and BTBT mechanisms. Figure 2.4 shows the leakage current in an NMOS transistor with three different input patterns [18]. It is seen that the total leakage depends on the voltage values at the transistor terminal. Since voltage represents the data at the terminals, it can be concluded that the total leakage power is data dependent. This brief discussion shows that static power consumption in CMOS logic is also data dependent. An inclusive analysis on viability of using leakage power consumption as an emerging side channel will be presented in Chapter 6 where the side channel vulnerability is evaluated with regard to the technology trends.

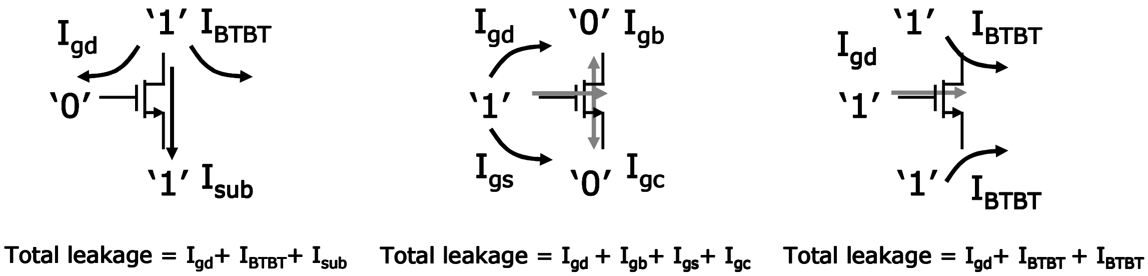

I_{gd}
'1' I_{BTBT}
'0'
'1' I_{sub}
Total leakage = I_{gd}+ I_{BTBT}+ I_{sub}
I_{gd}
'0' I_{gb}
'1'
I_{gs}
'0' I_{gc}
Total leakage = I_{gd} + I_{gb}+ I_{gs}+ I_{gc}
I_{gd}
'1' I_{BTBT}
'1'
'1' I_{BTBT}
Total leakage = I_{gd}+ I_{BTBT} + I_{BTBT}

Figure 2.4 Leakage power for different terminal voltages [18]

2.3 Side Channel Analysis Methodologies

In order to extract the data signature existing in the side channel, several methods are introduced. This section describes the principles of the techniques used for analysis of side channel information. The underlying assumptions in the analysis are that the attacker does not have access to the secret information (secret key), but the attacker can access the plaintext or ciphertext. The attacker also has access to the side channel (power consumption) measurable point on the hardware and an oscilloscope which will capture the samples of the instantaneous power (referred to as averaged traces in this thesis). In the next section the commonly used analysis methods used in side channel attacks are described.

2.3.1 Simple Analysis

Simple analysis involves direct visual examination of a cryptographic device's side channel, e.g. power consumption measurements. Typically only one trace is required for this type of analysis. The iterative operations which are executed in an underlying encryption algorithm cause a regular pattern of transistor switching [6]. This regularity is often discernible in power consumption traces of cryptographic devices. Depending on how a cipher is implemented, the information deduced from power traces may reveal the key material. If the attacker can determine where certain instructions are being executed, it becomes relatively simple to extract the useful information. An example of a simple analysis can be seen in an implementation of modular exponentiation which uses the square-and-multiply algorithm. The key value determines whether square or multiply operation should be executed. Since the conditional branches of square and multiply can be identified from the power traces, an implementation of modular exponentiation is known to be susceptible to simple analysis. The power trace acquired from the hardware implementation of the RSA algorithm can be analyzed as the multiplications require additional register loads. This increases the width of the leading spikes. As a result of this the square operation providing a narrow spike can be distinguished from the square-and-multiply operation which provides a narrow spike followed by a wider spike. In other words, if the key bit is zero the corresponding power trace contains a narrow spike, if the key bit is one the corresponding power trace consists of one wide spike after a narrow spike.

Simple analysis is useful in practice if only one or very few traces are available. This can make simple analysis attacks quite challenging in practice. Detailed knowledge about the implementation of cryptosystem is often required by the attacker. Simple analysis is not effective in revealing key related information if the Signal-to-Noise Ratio (SNR) is reduced. More enhanced analysis has been proposed in which the advanced statistical methods are used. These techniques interpreting the power traces of a cryptographic device are discussed in detail in the next section.

2.3.2 Statistical Analysis

Statistical analysis of the side channel is a more sophisticated procedure compared to simple analysis [5]. This method is used when the individual bit cannot be seen because of noise and countermeasures. The statistics used in this analysis are extremely powerful and they can extract keys even if the individual traces contain large amounts of noise. No detailed knowledge about the crypto device is necessary in statistical analysis and it is only sufficient to know the cryptographic algorithm that is executed by the device. Unlike the simple analysis in which the side channel, e.g. power consumption is analyzed along the time axis for extraction of a pattern or matching template, in statistical analysis the shape of the side channel along the time axis is not crucial. Employing this analysis, the attacker attempts to analyze how the side channel at the fixed point of time depends on the processed data. Statistical analysis-based attacks focus exclusively on the data dependency of the side channel. Therefore, a large number of power traces needs to be collected and further analyzed. The general procedure in statistical analysis-based attack is described as follows [19]:

a) Selection function: An intermediate result of the cryptographic algorithm that is executed by the attacked device needs to be a function *f (d, k)*, where *d* is a known non-constant data value and *k* is a sub-value of the secret key. Intermediate results that fulfill this condition can be used to reveal *k*. In most attack scenarios, *d* is chosen from either the plaintext or the ciphertext.

b) Side channel measurement: Side channel, e.g. power consumption, needs to be measured for *D* different data blocks. For each of the runs, the attacker needs to know the corresponding data value of *d* that is involved in calculation of the intermediate result which was initially chosen. The data value can be represented by a vector $d = (d_1, \ldots, d_D)$, where d_i denotes the data value in the i^{th} run. The side channel corresponding to data block d_i should be recorded and referred to as $(t_{i,1}, \ldots, t_{i,T})$, where *T* denotes the length of the side channel trace. Once the traces are measured for each of *D* data block, they can be recorded as matrix T of size $D \times T$. The value of each column t_j of the matrix T needs to be caused by the same operation, thus, it is imperative that the measured traces are correctly aligned. The alignment task can be done by using an accurate trigger signal that records the traces at exact same sequence of operations during each run. An enhanced alignment technique is proposed in [20].

c) Hypothetical intermediate values calculation: A Hypothetical intermediate value needs to be calculated for every possible choice *k*. The possible choices of *k* can be represented by vector k = $(k_1, \ldots, k_K)$, where *K* denotes the total number of possible choices for *k*. The elements of k are known as key hypotheses. Given the data vector d and the key hypotheses k, the attacker can calculate the hypothetical intermediate values of *f (d, k)* for all *D* runs and for all *K* key hypotheses. The result of the calculation can be represented in a matrix V of size $D \times K$.

$$v_{i,j} = f(d_i, k_j), i = 1, \ldots, D \text{ and } j = 1, \ldots, K \qquad (2.10)$$

Column *j* of V contains the intermediate results that have been calculated based on the key hypothesis k_j. Vector k contains all possible choices for *k*. Hence, the value that is used in the device is an element of k. The index of this element is referred to as *ck*. Thus, k_{ck} refers to the key of the device. The objective in statistical analysis is to find which column of V has been processed during the *D* run. Once it is known which column of V has executed in the attacked device, the k_{ck} will be revealed.

d) Side channel and intermediate values mapping: The hypothetical intermediate values V need to be mapped to a matrix H of the hypothetical side channel values. The hypothetical side channel values can be obtained by using different models in a simulation environment. The commonly used models for mapping power consumption V to H are the Hamming Distance (HD) and the Hamming Weight (HW) models. Using these models, the side channel of the device for each hypothetical intermediate value $v_{i,j}$ is simulated in order to obtain a hypothetical side channel value $h_{i,j}$. The more knowledge the attacker has about the crypto device, the better power models can be driven. The quality of the power model has a strong impact on the effectiveness of an attack.

e) Hypothetical side channel values and measured side channel values comparison: In order to map V to H, the attacker needs to compare each column h_i of the matrix H to each column t_j of the matrix T. This means that the attacker compares the hypothetical side channel values of each key hypothesis with the recorded side channels at every position.

The result of this comparison is a matrix R of size $K \times T$, where each element $r_{i,j}$ contains the results of the comparison between columns h_i and t_j. The key of the attacked device can finally be revealed based on the following observation.

The intermediate result that has been chosen in a) is part of the algorithm. The device needs to calculate the intermediate values v_{ck} during the different executions of the algorithm. Therefore, the recorded side channel traces, in fact, depend on these intermediate values at some positions referred to as *ct*, i.e. the column t_{ct} contains the trace value that corresponds to the intermediate values v_{ck}.

The hypothetical side channel values h_{ck} have been simulated by the attacker based on the values v_{ck}. Therefore, the column h_{ck} and t_{ct} are strongly related. These two columns lead to the highest value in R. The highest value of the matrix R is value $r_{ck, ct}$. All other values of R are low because the other columns of H and T are not strongly correlated. The attacker can reveal the index for the correct key *ck* and the moment of time *ct* by simply looking for the highest value in the matrix R. The indices of this value reveal the positions at which the chosen intermediate result has been processed and the key is used by the device. If all values of R are approximately the same, it means that attacker has not measured enough side channel traces to estimate the relationship between the columns of H and T. More traces provide more elements in the column of H and T which will result in more precise attack outcomes. Figure 2.5 illustrates the side channel attack procedure.

There are several well-established methods available to determine the relationship between the columns of H and T. Among them correlation and difference-of-means tests are the two powerful statistical methods which accurately express the linear relationship between two sets of data. These two methods are described in the following.

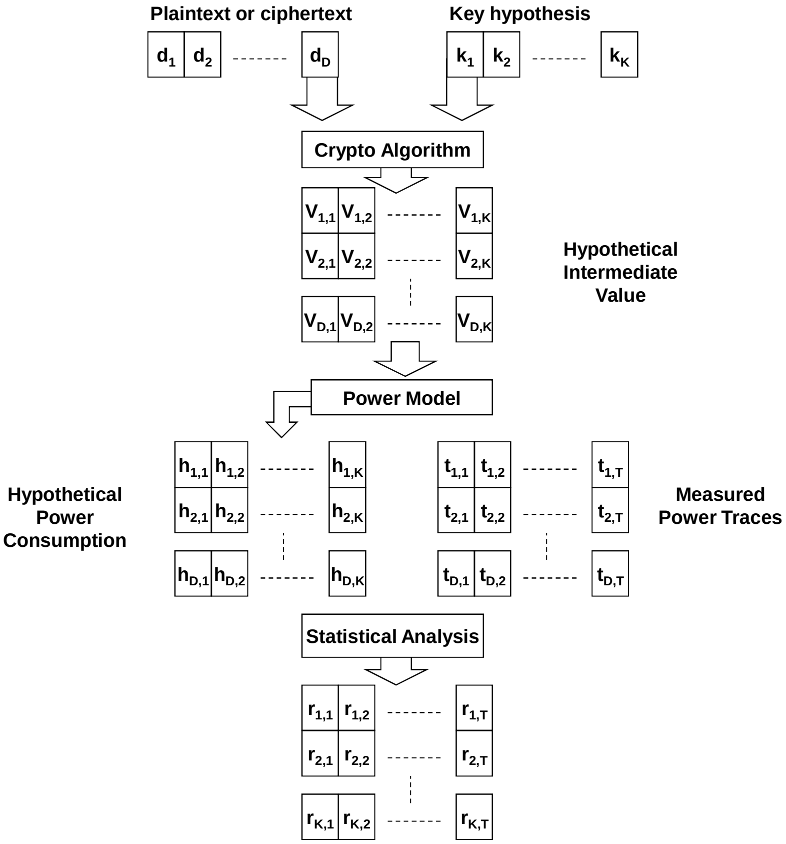

Plaintext or ciphertext
Key hypothesis
d_1 d_2 ------- d_D
k_1 k_2 ------- k_K
Crypto Algorithm
$V_{1,1}$ $V_{1,2}$ ------- $V_{1,K}$
$V_{2,1}$ $V_{2,2}$ ------- $V_{2,K}$
$V_{D,1}$ $V_{D,2}$ ------- $V_{D,K}$
Hypothetical Intermediate Value
Power Model
Hypothetical Power Consumption
$h_{1,1}$ $h_{1,2}$ ------- $h_{1,K}$
$h_{2,1}$ $h_{2,2}$ ------- $h_{2,K}$
$h_{D,1}$ $h_{D,2}$ ------- $h_{D,K}$
$t_{1,1}$ $t_{1,2}$ ------- $t_{1,T}$
$t_{2,1}$ $t_{2,2}$ ------- $t_{2,T}$
$t_{D,1}$ $t_{D,2}$ ------- $t_{D,T}$
Measured Power Traces
Statistical Analysis
$r_{1,1}$ $r_{1,2}$ ------- $r_{1,T}$
$r_{2,1}$ $r_{2,2}$ ------- $r_{2,T}$
$r_{K,1}$ $r_{K,2}$ ------- $r_{K,T}$

Figure 2.5 The steps in launching a side channel attack

Correlation coefficient test: The correlation coefficient test is a common approach to determine the linear relationship between sets of data. The correlation test is also known suitable for modeling the statistical properties in side channel attacks. In statistics the linear relationship can be expressed based on the covariance or the correlation. The definition of the covariance is given by [21]:

$$Cov\ (X, Y) = E((X - E(X)).(Y - E(Y))) = E(XY) - E(X).\ E(Y) \qquad (2.11)$$

where *X* and *Y* are sets of data, *E(X)* and *E(Y)* are expected values of *X* and *Y*, respectively. The covariance quantifies the deviation from the mean. Equation 2.11 shows that the covariance is related to the concept of statistical dependence. If *X* and *Y* are statistically independent then *E (XY) = E(X). E(Y);* therefore, *Cov (X,Y) = 0*. The covariance is typically not known and needs to be estimated. The estimator *c* of the covariance is given as [21]:

$$c = \frac{1}{n-1}.\sum_{i=1}^{n} (x_i - \bar{x}).(\ y_i - \bar{y}) \qquad (2.12)$$

where *n* is the number of data point in the set and $\bar{x}$, $\bar{y}$ are the mean values. A more commonly used method to measure a linear relationship between two values is the correlation coefficient ρ *(X, Y)*. The correlation coefficient is defined as a function of covariance [21]:

$$\rho\,(X,Y) = \frac{Cov\,(X,Y)}{\sqrt{Var(X).Var(Y)}} \qquad (2.13)$$

where *Var(X)* and *Var(Y)* are the variances of data set *X* and *Y*. The correlation coefficient is a dimensionless quantity and it can only take values between plus and minus one. ρ is also not known and needs to be estimated. The estimator *r* is defined by Equation 2.14 as:

$$r = \frac{\sum_{i=1}^{n} (x_i - \bar{x}).(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 . \sum_{i=1}^{n} (y_i - \bar{y})^2}} \tag{2.14}$$

In a statistical-based attack, the correlation coefficient is used to determine the linear relationship between the columns h_i and t_j for $i = 1, \ldots, K$ and $j = 1, \ldots, T$.

This results in a matrix R of estimated correlation coefficient. Each value of $r_{i,j}$ based on the D elements of columns h_i and t_j are estimated. Equation 2.14 is can now be rewritten as:

$$r_{i,j} = \frac{\sum_{d=1}^{D} (h_{d,i} - \bar{h}_i) .(t_{d,j} - \bar{t}_j)}{\sqrt{\sum_{d=1}^{D} (h_{d,i} - \bar{h}_i)^2 . \sum_{d=1}^{D} (t_{d,j} - \bar{t}_j)^2}} \tag{2.15}$$

The results of a successful attack on a large set of smart card chips show the validity and the advantages of the correlation-based power attack [19].

Difference-of-Means Test: An alternative technique to determine the relationship between the columns of H and T is the difference-of-means. The statistical analysis using this method follows similar steps as described earlier in Section 2.3.2. The only difference occurs when matrix V is mapped to H. In case of correlation coefficient, there are no special constraints for this mapping, while in case of using the difference-of-means only binary side channel models are possible. This means that the side channel model needs to be chosen in such a way that $\mathrm{h}_{i,j} \in \{ 0,1 \}, \forall\, i, j$. Therefore, direct use of the HW model is not feasible. This model should be modified before it is used, e.g. setting $h_{i,j} = 1$ if $HW (v_{i,j}) > 4$ and $h_{i,j} =0$ if $HW (v_{i,j}) < 4$ if $h_{i,j} = 0$. However, it is clear that such a binary model describing the power consumption of the attacked device is not as accurate as a non-binary model. This causes the attack based on difference-of-means to be less effective than the correlation-based attack. Besides the different requirements for the side channel modeling, the comparison procedure between the hypothetical values and real side channel values is also different in difference-of-means. The matrix R has the exact size as in the case of correlation coefficient; however, its elements are calculated using other statistical method. The following explanation is given to elaborate the method that is used in difference-of-means approach.

Performing the attack procedure explained in Section 2.3.2, the attacker creates a binary matrix H and further makes the assumption that the side channel for certain intermediate value is different from the side channel corresponding to other values. The sequence of zeros and ones in each column of H is a function of input data *(d)* and a key hypothesis *(k_i)*. In order to verify whether k_i is correct or not, the attacker splits the matrix T into sets of rows, i.e. two sets of side channel traces, according to h_i. The first set contains those rows of T whose indices correspond to the indices of the zeros in the vector h_i. The second set contains all remaining rows of T. Subsequently, the mean of the rows are calculated. The vector m'_{0i} denotes the mean of the rows in the first set and m'_{1i} denotes the mean of the rows in the second set. For the number of power traces then:

$$m_{1\,i,j} = \frac{1}{n_{1_i}} . \sum_{l=1}^{n} h_{l,i} \, . t_{l,_j} \tag{2.16}$$

$$m_{0\,i,j} = \frac{1}{n_{0i}} . \sum_{l=1}^{n} h_{l,i} \, . t_{l,j} \tag{2.17}$$

where n denotes the number of rows of H, i.e. the number of power traces that are used in the attack, $n_{1\,i}$ and $n_{0\,i}$ are expressed by Equations 2.18 and 2.19, respectively:

$$n_{1_i} = \sum_{l=1}^{n} h_{l,i} \tag{2.18}$$

$$n_{0_i} = \sum_{l=1}^{n} (1 - h_{l,i}) \tag{2.19}$$

The key hypothesis (*k_i)* is correct, if there is a significant difference between $\mathrm{m}_{1i,j}$ and $\mathrm{m}_{0i,j}$ at some point. The difference indicates that there is a correlation between h_{ck} and some columns of T. Similar to the correlation coefficient-based analysis, the difference occurs at the exact moments of time when the intermediate values that correspond to h_{ck} are processed. At the other time the difference between the vectors is essentially zero. The result of the attack using difference-of-means method is a matrix R, where each row of R corresponds to the difference between the mean vectors $m_{1_{i,j}}$ and $m_{0_{i,j}}$ of one key hypothesis. The original side channel attack on implementation of DES reported in [6] is mounted using the difference-of-means test.

2.4 Discussion on Analysis Methods and Evaluation Strategy

Power-based attacks using statistical analysis are generally known as Differential Power Analysis (DPA). The concept of DPA is independent of the statistical tests or the type of attacked data [15]. Distinguishing simple analysis from DPA is straightforward. The differences can be itemized as; first, the attacker's knowledge and capabilities, e.g. whether the attacker has access to only few power traces or many power traces, and whether the attacker is able to characterize the device, second, the type of leakage that is exploited by the attackers, e.g. if the information leakage is easily observable by the visual inspection of the trace. If information leakage via side channel is not discernable directly in a trace, then DPA will be the preferred approach for side channel attack. In general, DPA attacks are more powerful. The popularity of the DPA attack is due the fact that no detailed knowledge about the attacked device is required. DPA attacks have become common tools for evaluating the resistance against side channel information leakage [6].

DPA-based attack may not always lead to the same results. The effectiveness of DPA is first influenced by the model which is enumerated in the attacks. The models used by attackers are chosen based on their knowledge. Second consideration which should be taken into account is the type of the crypto cores. The attack on a core that has fewer components is more efficient than on a core which executes several instructions at a time. It is important that effectiveness of the attacks is compared on the basis of common hardware implementations. It is shown in [21] that using the same model and identical hardware, the correlation-based DPA provides successful results with less attempts compared to the difference-of-means hypothetical test. In other words, correlation-based DPA is a stronger evaluation technique for assessing the leakage of side channel information.

The brief discussion in this section clarifies that the leakage of side channel information is best evaluated by employing a powerful attack based on an appropriate model examined by an effective analysis technique.

The following are the conditions that form our strategy for evaluating the leakage of side channel information. The assessment will be conducted by:

- considering an access to plaintext and power consumption measurable point
- mounting DPA attack
- analyzing the data using correlation coefficient test
- mapping the hypothetical and measurements values utilizing the HD model
- providing a realistic reference model for evaluating side channel information leakage by attacking on an unprotected crypto core
- strengthening the attack by developing an enhanced data capturing method which improves the result of correlation coefficients test (the concept of reference vector insertion is discussed later in Chapter 5)
- evaluating the side channel information leakage using real power measurement from test chip
- delivering the final results by number of averaged traces for key revelation.

2.5 Summary and Conclusion

This chapter provided the background on information leakage via power consumption. The elements of power consumption in CMOS circuits were reviewed from a side channel perspective. The data dependency of different elements of power consumption in CMOS circuits was described. The detailed analysis was also presented on leakage power mechanisms which will be referred to later in Chapter 6 (where the trend of side channel threat versus technology advancement is evaluated). Attacks for extracting the data carried by the power consumption traces were described. Simple and statistical analysis techniques were reviewed. DPA using correlation coefficient and difference-of-means were explained. A conclusive review on attack methodologies and analysis techniques were presented. Furthermore, an assessment strategy used for measuring the side channel security in this thesis was established.

Chapter 3

Side Channel Countermeasures

3.1 Introduction

Side channels such as power consumption, electromagnetic (EM) emission of the crypto devices, and the time of execution of crypto operations are recognized as driving elements of modern cryptanalysis. Exploitation of the information leaked by side channels has become more popular due to increasing demand for hardware implementation of cryptosystems [4]. Thus, special attention has been drawn towards the ongoing issue related to side channel information leakage. Significant efforts have been made to define new design constraints for reducing the data dependency of power consumption. Therefore, major improvements have been achieved in implementation of secure crypto cores.

This chapter describes two main streams for counteracting leakage of information via power consumption: data concealing and data masking. These two approaches are explained and their applications in reducing data signature in amplitude and time dimensions of power consumption are described. The countermeasures that have been presented so far in the literature are reviewed. The operations of these countermeasures are briefly explained. The discussion in this chapter illustrates the image of the recent progresses which has been made in the area of side channel security.

3.2 Side Channel Resistance Methods

The implementation of side channel countermeasures can be distinguished between software and hardware approaches. Software protection methods proposed earlier for side channel protection have demonstrated either a limited degree or inefficient protection levels [15]. The shortcomings of software protection techniques shifted the attention towards design and implementation of side channel countermeasures at the hardware level. Methods that are proposed to secure the critical information (e.g., key related) contained in side channels typically fall into two main categories: data concealing and data masking. The concept of data concealing is, first explained.

3.2.1 Data Concealing

Data concealing-based countermeasures have been proposed to make the power consumption of crypto devices independent of the intermediate values and also independent of the operations executed by crypto devices. Since the power consumption is characterized by its time and amplitude, the approaches proposed for data concealing are also categorized by those that affect the time and those that affect the amplitude dimension of the power consumption.

Data concealing on time dimension: Revisiting the side channel analysis methodologies explained in Section 2.3.2, one can realize that a successful side channel attack requires the recorded side channel traces to be correctly aligned. This means that the power consumption of each operation should be located at the same position in each power trace. If this condition is not fulfilled, analysis of side channel information will be more challenging. This observation motivates the designer to randomize the execution of the crypto operation in security sensitive applications so that the device performs the operation of the algorithm at different time points during each execution. The power consumption appears to be noticeably random. The attack procedure becomes more complicated if the randomization increases.

Data concealing on amplitude dimension: Changing the amplitude of power consumption representing the performed operations or data values is also considered as a side channel countermeasure. In order to hide the data in the amplitude dimension both equalization and randomization are proposed. Both of these techniques are shown to be effective as they lower the SNR associated with the performed operations or processed data. SNR can be expressed as follows:

$$\text{SNR} = P_{exp} / (P_{noise}) \qquad (3.1)$$

where the signal corresponds to P_{exp} contains relevant information for analysis, and noise component is given by P_{noise} which is sum of the electrical noise and switching noise.

Attaining ideal SNR (SNR = 0) is not practical; however, low SNR can be achieved by reducing $Var\ (\ P_{exp}\ )$ to zero or by increasing $Var\ (P_{noise})$ to infinity. These techniques are referred to as equalization or randomization, respectively (*Var* denotes the variance). Reducing $Var\ (P_{exp}\ )$ to zero means that the power consumption should be identical for all operations and data values. Increasing $Var\ (P_{noise}\ )$ to infinity means that the amplitude of noise needs to be infinitely increased. It is shown that lower SNR provides significant side channel resistance [6].

3.2.2 Data Masking

Masking data is another alternative for developing side channel resistant architectures. Masking provides resistance by randomizing the intermediate values that are processed by the crypto device. An advantage of this approach is that masking allows the power consumption of the intermediate values to be independent, even if the device has data dependent power consumption. In a masked implementation, each intermediate value v is concealed by a random value m that is called mask and it results in: $v_m = v * m$. The mask m is generated internally and varies in each execution round of algorithm. Hence, it is not known by the attacker. The operation * is typically Boolean exclusive-or function, the modular addition, or the modular multiplication. The masks are directly applied to the plaintext or the key. The result of the encryption is also masked. The mask needs to be removed at the end of computation. A typical masking scheme specifies how all intermediate values are masked and how to apply, remove, and change the masks throughout the algorithm. Masking schemes such as Boolean and arithmetic masking are described next. The concepts of secret sharing and blinding in masking are also explained.

a) Boolean and arithmetic masking: Masking is divided into Boolean and arithmetic masking. In Boolean masking, the intermediate value is concealed by exclusive-oring with the mask, e.g. $v_m = v \otimes m$. In arithmetic masking; however, masking is performed by arithmetic operation such as addition or multiplication [22]. Modular addition and modular multiplication are also common masking schemes, e.g. $v_m = v + m$ and $v_m = v \times m$ (mod n). Masking provides side channel resistance if each masked intermediate value v_m is pairwise independent of v and m. Hence, every masked intermediate value should induce a distribution which is independent of the unmasked intermediate value. Some cryptographic algorithms which are based on Boolean and arithmetic operations require both types of masking. Such combined masking is often problematic from implementation perspective since switching between masking schemes involves significant amount of additional operations [13][23]. Efficient algorithms for switching between Boolean and arithmetic masking are presented in [24]. The intermediate value v can be computed given v_m and m. In other words, intermediate value v is represented by two shares: v_m and m. Only two given shares will allow to determine v. Consequently, masking corresponds to a secret-sharing scheme that uses two shares. Similar concept can be extended to secret-sharing with several masks [25]. Masking technique for Data Encryption Standard (DES) [26] and Advanced Encryption Standard (AES) [27] are discussed in [28] and [12], respectively. Applying several masks to one intermediate increases the cost of implementation. In practice secret sharing based on two shares is described as an efficient masking scheme. The application of arithmetic masking in asymmetric cryptographic schemes is called blinding. In the blinding approach, multiplicative masking is applied to input message (*v*) in decryption in [29]. This type of masking is known as message blinding. Another slightly different masking technique can be applied to the exponent [30]. This technique is used for particularly Elliptic Curve Cryptography (ECC) [31].

3.3 Hardware Implementation of Side Channel Countermeasures

An overview on the hardware implementation of data concealing (time and amplitude dimensions) and data masking countermeasures are presented. The discussion distinguishes the implementations between two design abstraction levels: architecture and logic.

3.3.1 Data Concealing at the Architectural Level

As discussed in the previous section in order to obtain data/operation-independent power consumption, randomizing the sequence of operation and randomizing the value of the consumed power can be used.

In practice, several implementations of data concealing countermeasures at the architectural level are proposed. Most of these implementations operate based on the data concealing either in time or amplitude dimensions. Implementation of data concealing in both time and amplitude dimensions is also reported.

Data concealing in time dimension (architectural level): It is crucial that countermeasures affecting the time dimension of power traces remain unidentifiable. This means that attackers must not be able to detect the countermeasures. The goal of these countermeasures is to randomly insert dummy operation for shuffling the performed operations or randomly changing the clock signal to make the alignment of the power traces more difficult. These methods are briefly discussed.

a) Inserting random cycle or dummy operations: The random insertion of dummy operations shuffles the sequence of executed crypto algorithms. Similarly dummy clock cycle can be randomly inserted; however, the random operation may take multiple clock cycles. To randomly insert dummy cycles, the registers of a protected cryptographic device are usually duplicated. The original registers are used to store the intermediate values while other registers are used to store the random values. During the execution, a random number generated in each clock cycle is used to determine whether the clock cycle is a dummy cycle or not. If the dummy cycle enters, the device will perform a computation by using random data stored in the duplicated registers. Otherwise, the device continues with execution of the algorithm. An example of this approach is the non-deterministic processor which randomly changes the sequence of the program during each execution [32]. The countermeasure introduced in [32] effectively counteracts power-based attack. The same concept is improved and presented in [33] in which additional instructions are randomly inserted.

b) Randomly skipping clock or changing frequency: This approach adds a filter into the clock signal path. The filter randomly skips the clock signal pulses. Random numbers are used to determine which clock pulses are skipped. An alternative to skipping of clock pulse is to generate a clock signal with a randomly changing frequency, e.g. controlling the frequency of an internal oscillator. Multiple clock domains can also be considered as several clock signals are randomly applied to the core. A side channel resistant architecture in [34] uses a Dynamic Voltage and Frequency Switching for adding randomization to clock frequency and supply voltage (Figure 3.1).

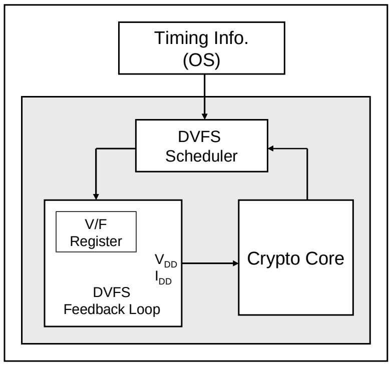


Figure 3.1 Dynamic Voltage and Frequency Switching (DVFS) [34]

Data concealing in amplitude dimension (architectural level): The power consumption of cryptographic devices can be made nearly equal for all operations and all data values by using filtering. To counteract the power-based attack noise can also be added to the power supply; however, it is important to realize that SNR not only depends on the cryptographic device, but also on the measurement setup that is used for the attack. A countermeasure which reduces SNR of the measurement setup may not necessarily reduce SNR for all setups. Therefore, the measurement techniques should be carefully considered by one who designs the countermeasures. The following explanations are given to describe the filtering and noise injection.

a) Filtering: In order to remove exploitable components of the power consumption, a filter is inserted between the power supply pin of the cryptographic device and the circuit that computes the crypto algorithm. Power consumption can be filtered by using switched capacitors or constant current sources. The effect of an Resistance-Inductance-Capacitor (RLC) filter inserted into the power supply line of the cryptographic device is analyzed in [35]. Decoupling the power supply of crypto core is also presented in [36]. The basic idea is that a capacitor is charged by the power supply while the other capacitor supplies power for the core. The capacitors are periodically switched as shown in Figure 3.2 (a). A similar idea is proposed in [37] by using a three-phase charge pump to deliver power to the device. Using active circuits is also considered in [38][39][40][41] which flatten the power consumption to reduce the leakage of data from crypto device. These approaches are shown in Figure 3.2 (b) and (c).

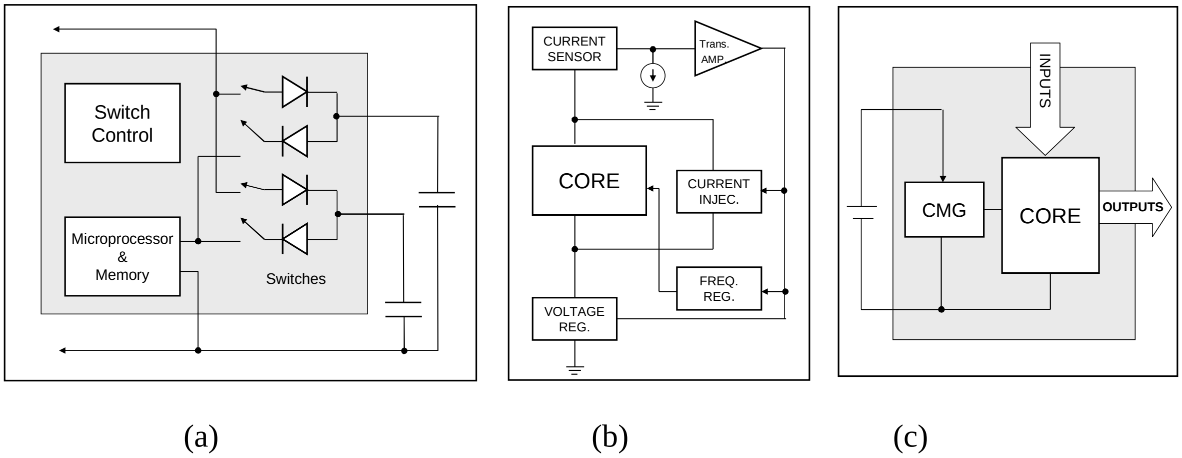


Figure 3.2 a) Decoupling the supply [36] b) Flattening the power consumption [40] c) Current Masking Generation (CMG) [41]

b) Noise insertion: An alternative method to filtering is to generate noise in parallel to the computation of the crypto algorithm. Noise engines are typically built by using random number generators. To provide sufficient entropy on the power consumption, the random generators need to be connected to a network of large capacitors. The random charging and discharging the capacitor network leads to noise in the power consumption. Other approaches introduced in [42][43] integrate different components with same functionality into the crypto core. Random numbers are then used to decide which component performs the operations in the executed cryptographic algorithm. Supplying noise that is induced by this countermeasure depends on the degree of randomness that is applied.

3.3.2 Data Masking at the Architectural Level

A significant amount of research has been devoted in applying masking schemes. Recent research on masking mostly focuses on the AES algorithm [27]. Implementation of masking schemes in hardware requires similar consideration as implementation in software. Boolean masking schemes are known as effective side channel protection methods for block ciphers.

Nonetheless, significant overhead is imposed on performance and area as a result of using masking. The implementation of masking schemes is briefly discussed next.

a) Masking Multiplier (MM): Adders and multipliers are the basic building blocks in cryptographic algorithms. The Substitution Box (SBOX) in AES can be decomposed into a sequence of addition and multiplication. In order to realize a Masked Multiplier (MM), a circuit has to compute the product of two masked inputs $v_m = v \oplus m_v$, $w_m = w \oplus m_w$, and some masks *(m_v, m_w, m)* such that:

$$MM(v_m, w_m, m_v, m_w, m) = (v \times w) \oplus m \qquad (3.2)$$

This observation can be used to build a masked multiplier (Figure 3.3).

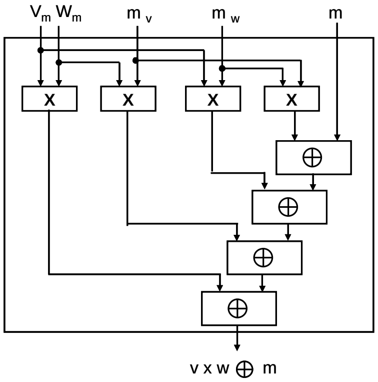


Figure 3.3 A masked multiplier (four standard multipliers and standard adders) [22]

b) Random precharging: Randomly precharging all the combinational and sequential cells of the core is another approach. The typical implementation of random precharging requires duplicating the sequential cells [47]. The duplicates of the registers are inserted between the original registers and the combinational cells. Random precharging operates such that in the first clock cycle, the duplicate of the registers, containing random values, are connected to the combinational cells, so that the outputs of all combinational cells are randomly precharged. In the second clock cycle, the result of the combinational cell is stored in the original registers that contain the intermediate values of the executed algorithm. At the same time, the intermediate values are moved from these registers to the duplicates. This switches the role of the registers. Thus, in the second clock cycle the combinational cells are connected to registers that contain the intermediate values of the algorithm. The switching continues in the third cycle. Random precharging is implemented in a very similar way as the random insertion of dummy cycles. A method to implement random precharging is discovered in [48] by randomizing the register usage. Hence, the intermediate values of the algorithm are stored in different registers with potentially different data during each execution.

c) Masking buses: Buses are particularly vulnerable to power-based attack due to the large capacitance associated with them. Data encryption is often used to prevent attacking on the buses. A pseudo-random key is generated and used in a simple scrambling algorithm. Hence, the simplest version of bus encryption, where a random value is exclusive-ored to the value on the bus, corresponds to masking the bus. Bus encryption is presented in [49][50][51]. In general, masked implementations involve redundancies since the mask values are frequently added at some points in the algorithm and removed or changed at other points. The overhead may exceed the overall design limitations.

3.3.3 Data Concealing at the Logic Level

The concept of data concealing at the logic level is mainly realized by employing logic style whose power consumption is independent of the processed data and performed operations. This is achieved by making the power consumption of the logic constant for all the processed logic values. A consequence of such behavior is that the logic consumes the maximum amount of power at all the time. Logic styles with constant power consumption are typically implemented in Dynamic and Differential Logic (DDL) scheme. In addition to power consumption overhead in DDL, the total area cost of this logic family is considerably greater than single-ended logics. Because of significantly increased area and power consumption, often not all components of cryptographic devices are implemented using these logics. A precise side channel leakage evaluation is necessary to ensure no sensitive blocks leave the secured component of crypto core. Furthermore, “logic style conversion”, “high-level design capture” and “logic synthesis” steps must be integrated into the standard semi-custom design flow. These steps provide support for logic synthesizers which are typically use single-ended logic style. The logic synthesis of the high-level design can be performed by using an single-ended cell library. These cell netlist is then converted to a DDL cell netlist. Moreover, side channel aware floorplanning, placement and routing are still required for ensuring that the capacitance and resistance of complementary wires are pairwaise the same.

The side channel resistant logic styles can be categorized into two main groups: those which require generating new logic cells and those whose logic cells are based on standard library. Several DDL styles such as Sense Amplifier-Based Logic (SABL) [52], Wave Dynamic Differential Logic (WDDL) [53], Dual-Spacer Dual-Rail (DSDR) [54], Three-Phase (TP) Dual-rail precharge logic [55] and 3-state Dynamic Logic (3sDL) [56] are proposed. The most known DDL logic styles for side channel security are reviewed in the following.

a) Sense Amplifier-Based Logic (SABL): The concept of balancing the power consumption at the logic level is first introduced in [52]. Sense Amplifier-Based Logic (SABL) consumes a fixed amount of charge for every transition including the degenerated events in which a gate does not change state. The combinational SABL cells are designed such that the cells only evaluate after all input signals have been set to complementary values. All the SABL cells are connected to the clock signal which can be precharged simultaneously. In balanced SABL circuits, the propagation delays of the complementary wire should be pairwise identical so that the combinational SABL cells can always evaluate at a certain time in each clock cycle. SABL implementation requires several times more area compared to corresponding CMOS circuits. The maximum clock rates are typically halved. The power consumption in SABL is significantly high. Actual increase of the power consumption depends on several aspects: the size of the circuits, ratio between the number of combinational and sequential cells which defines the increase of the clock tree, and the circuit architecture. A power efficient version of SABL named as Charge Recycling Sense Amplifier-Based Logic (CRSABL) [57] is also presented for power-constrained applications. The side channel resistance of SABL circuit is typically very high if all the complementary wires are sufficiently balanced. A general SABL configuration is shown in Figure 3.4. A methodology to route multiple differential wires between secure dynamic differential standard cells is proposed in [58]. The library of SABL cells need to be generated since the standard cells cannot be used.

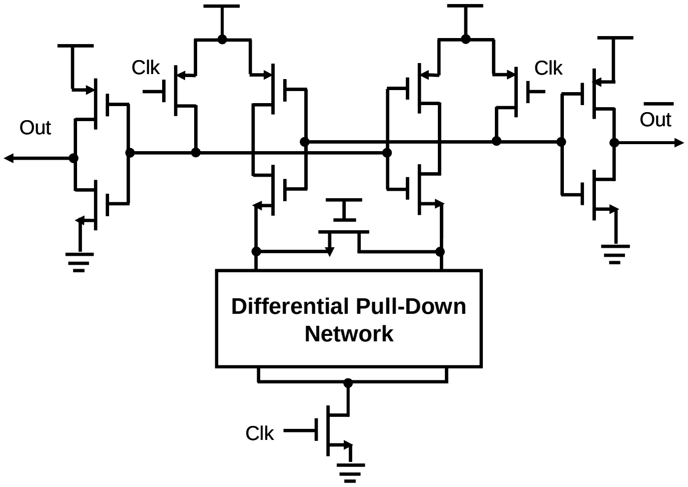


Figure 3.4 Generic SABL gate [52]

b) Wave Dynamic Differential Logic (WDDL): Wave Dynamic Differential Logic (WDDL) gates introduced in [53] are synthesized based on standard cells that are available in existing libraries. WDDL has less complexity in its structure and shows less side channel resistance compared to DDL logic family. In WDDL the sequential cells are connected to the clock signal, thus, only these cells precharge and evaluate at the same time. Combinational WDDL cells precharge when their inputs are set to precharge value. Figure 3.5 shows the schematic of a WDDL NAND cells. A combinational WDDL cell consists of two circuits that realize the Boolean functions: F_1 and F_2. The functions must be defined such that: if the input signals $in_1, \overline{in_1},..., in_z, \overline{in_z}$ are set to complementary values, complementary output values are calculated according to the intended logic function of the cell. Thus, for the complementary input values, F_1 and F_2 must satisfy $F_1(in_1,...,in_z) = \overline{F_1(\overline{in_1},...,\overline{in_z})}$.

The area overhead of WDDL circuits is significantly higher than CMOS circuits with equivalent functionality. The power consumption overhead of WDDL is considerably large. The maximum clock rates are approximately same between WDDL and corresponding CMOS circuits. However, since a WDDL delay flip-flop consists of two stages, in order to obtain the same throughput as in CMOS, the clock frequency in WDDL should be increased by factor of two. WDDL is suitable for reconfigurable (FPGA) platform implementation. The ASIC implementation of WDDL requires a semi-custom design flow which includes generating a library of the logic gates and special procedure for differential routing and placement.

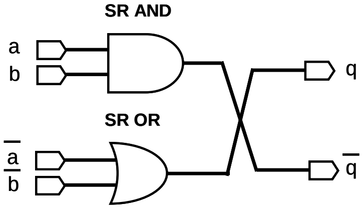


Figure 3.5 Cell schematic of WDDL NAND cell [53]

c) Current Mode Logic (CML): Current Mode Logic (CML) is suggested to tackle the threat of side channel attacks. CML utilizes the current mode scheme in which the output value of a logic cell is defined by current that is passing through the cell. The sum of these currents is rather constant and likely independent of the actual output values. Such characteristic is advantageous for side channel resistant architecture. MOS Current Mode Logic (MCML) is introduced for side channel security [59]. Dynamic Current Mode Logic (DyCML) shown in Figure 3.6 is proposed which cancels the static power dissipation inherited from its predecessor [60]. Using DDL style, DyCML achieves steady power consumption. The dynamic power consumption of DyCML gates is also small compared to other dynamic differential logics because of its low swing output. This logic shows same level of side channel resistance as SABL with an improved power-delay product [57]. The effect of unbalanced capacitive load at the output is still an ongoing issue in design of DyCML for side channel applications [61]. An extra area is imposed by implementation of virtual ground element. Deployment of DyCML also involves generating the cell library. DyCML's output is not rail-to-rail; therefore, level shifter is required at the output terminals. The impact of adding level shifter on side channel information leakage of DyCML is not investigated in [61].

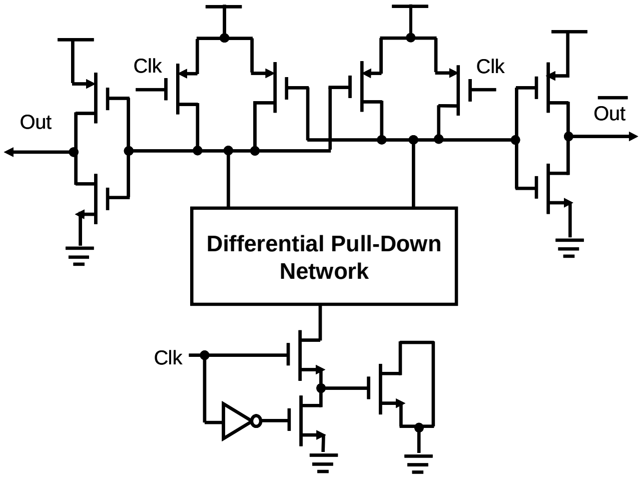


Figure 3.6 Generic DyCML gate [59]

d) Adiabatic Differential Switch Logic (ADSL): Adiabatic Differential Switch Logic (ADSL) is explored for side channel security in [62]. ADSL is an efficient energy recovery logic style pertaining to ultra-low energy low voltage digital circuits (Figure 3.7). The notion of adiabatic energy theorem employed in design of ADSL gates requires a time dependent power supply which is known as the power clock. The power consumption in adiabatic logic is known to be proportional to the inverse of rise time square of the power clock. ADSL's slow power clock reduces the peak of the supply current. The characteristic of power traces in ADSL provides less variation in power consumption which consequently reduces the side channel information leakage. ADSL saves up to 50% power consumption for a typical inverter. ADSL requires a four-phase power clock with a very high rise time. In providing such a complex clocking network the effect of clock skew cannot be ignored. The clock tree of ADSL also imposes a significant area overhead. Application of ADSL is limited to very low speed digital systems.

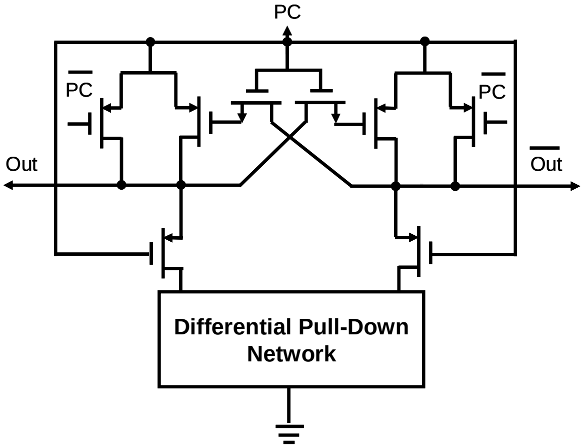


Figure 3.7 Generic ADSL gate [62]

e) Asynchronous circuits: Asynchronous circuits are known as effective side channel countermeasures [63][64]. Asynchronous circuits that counteract power-based attack are implemented in DDL scheme. Thus, side channel resistance of asynchronous circuits still relies on balancing the complementary wires. Further consideration for balancing the power consumption in asynchronous circuits is presented in [65]. Design flow for side channel resistant asynchronous implementation is described in [66]. Design of an asynchronous architecture is complex and time consuming due to lack of EDA tools that support the design of such circuits. The general configuration of asynchronous logic is shown in Figure 3.8.

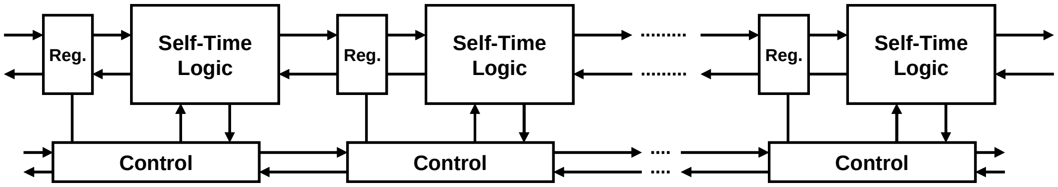


Figure 3.8 Asynchronous architecture

3.3.4 Data Masking at the Logic Level

Side channel security based on masking notion is often implemented at the architectural level. Nonetheless, several logic styles using the masking concept are also introduced which are referred to as masked logic styles. Masked logic styles operate on masked values and the corresponding masks. Since the masked values are independent of unmasked values, the power consumption of the masked cells should also be independent of the unmasked values. Consequently, the total power consumption of the cryptographic device becomes independent of the processed data and the performed operations. Boolean masking is commonly used. Figure 3.9 shows a two-input unmasked cell and a corresponding two-input masked cell. The input and output signals of the unmasked cells are carried on signal wires. In a masked cell; however, the input and output signals are split into masked values and the corresponding masks. Masking is characterized by the number of different masks in a circuit or the frequency of changing the mask values. Masking number schemes can be applied by using one distinct mask for each signal. All the masked values are pairwise independent of each other. The functionality of the logic cell becomes very complex by using this masking method. To reduce the number of necessary masks, the circuits can be clustered so the

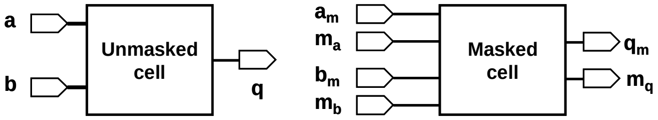


Figure 3.9 A 2-input unmasked cell and a corresponding 2-input masked cell [72]

same mask can be used for each cluster. For every masked signal that passes over from one signal cluster C_1 to another cluster C_2 an additional interface is required to change the mask from mc_1 to mc_2. Defining the number of clusters with different masks is a non-trivial task. Changing the single mask value can typically be detected via power consumption of the mask distributing net. The mask net can be implemented in DDL logic to overcome such a weakness. The masking frequency scheme determines how often the mask values are changed. If the mask is changed in each clock cycle, the rate at which new masks must be generated is very high. This would be more complicated when several masks are used. In order to reduce the rate of mask generation, the mask values can be used in several clock cycles. The approach of masked logic cells are developed by several researchers. Masked AND gate is proposed in [67][68]. Secret sharing at the cell level is also discussed in [69]. A complete review on implementation of masked logic style is presented in [70] [71]. One of the main logic styles proposed for masking the side channel data is introduced as Masked Dual-rail Pre-Charge Logic (MDPL) [72]. The operation of MDPL is reviewed.

a) Masked Dual-rail Pre-Charge Logic (MDPL): Masked Dual-rail Pre-charge Logic (MDPL) uses the same mask m for all signals in the circuits [72]. Masked signal d_m corresponds to an unmasked value $d = d_m \otimes m$. MDPL is implemented by using DDL circuits. MDPL cells are built based on single-ended cell that are available in existing standard-cell libraries. The general architecture of an MDPL is shown in Figure 3.10. As seen only the sequential cells are connected to the clock signal, thus, only these cells precharge and evaluate at the same time. Combinational MDPL cells precharge when inputs have been set to the precharge values and evaluate when their inputs are set to complementary values. MDPL flip-flops perform three operations. In the precharge phase, they start the precharge wave. In the evaluation phase flip-flops provide the stored complementary values that are masked with mask $m(t)$ of the current clock cycle. MDPL flip-flops complete the mask changing from $m(t)$ to $m(t+1)$, where $m(t+1)$ is mask value of the next clock cycle. The mask nets in MDPL circuits are complementary wires that are precharged. This allows balancing the power consumption of the mask nets to a degree that prevents simple analysis (determining the mask value is not feasible by simply monitoring the power traces of the crypto core). The structure of combinational MDPL cells is similar to combinational WDDL [53] cells. The main difference is that the inputs to the Boolean functions, F_1 and F_2, are masked values and corresponding masks. The side channel resistance of MDPL is limited as not all internal nodes of the single-ended cells on which MDPL cells are based perfectly masked. Eliminating this limitation increases the size and the complexity of MDPL cells. Implementation of MDPL requires larger area than that of unmasked CMOS. The maximum clock frequency is typically halved. The power consumption of MDPL is significantly increased due to the fact that MDPL are DDL and the mask nets must be switched frequently.

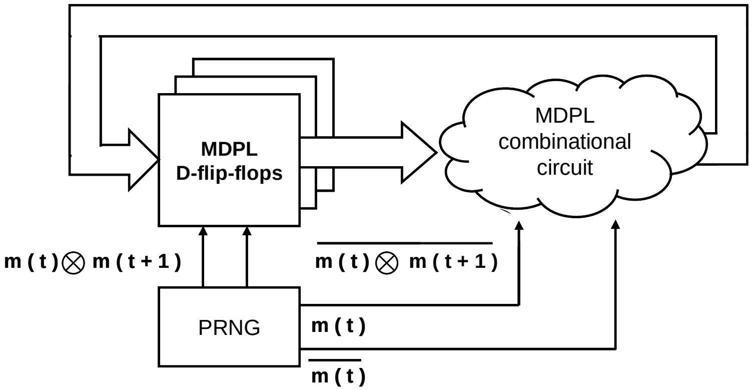


Figure 3.10 Architecture of an MDPL circuit [72]

3.4 Summary and Conclusion

This chapter presented an overview of methods which have been used for tackling the threat of side channel information leakage. The hardware protection methods in two main streams of data concealing and data masking were studied. Applications of these two concepts in amplitude and time dimensions of power traces were discussed. The hardware countermeasures based on concealing and masking were discussed in both architecture and logic levels. The most effective side channel countermeasures were seen to be those implemented in hardware and particularly at the logic level. The review of the previously known countermeasures not only demonstrated the expenditure involved with security but also addressed the complexity that designers face in providing side channel security.

Chapter 4

Side Channel Countermeasure: The Proposed Approach

4.1 Introduction

The previous chapter discussed the root causes of information leakage via power consumption. The recent progress in tackling the advanced power-based attacks was reviewed. The issues in delivering side channel secure hardware were also highlighted. This chapter presents a review on design and evaluation of the proposed side channel countermeasure. The objectives and strategy of this thesis for developing resistance against side channel information leakage are discussed. The design of logic cells and storage elements with novel exploitation of constant power consuming logic is presented. Side channel information leakage is evaluated at the simulation level. A comparative analysis with standard CMOS logic is also included for quantifying the cost of side channel resistance.

4.2 The proposed Approach

Recent research has shown that concealing the data signature in amplitude or time dimensions of the power traces forms an effective protection mechanism against leakage of information. Furthermore, it is seen that tackling the side channel threat at the logic level is the most effective approach [52]. Several DDL-based countermeasures were examined for realization of data concealing [52][57][59][60]. As discussed in the previous chapter, the cost associated with design and implementation of these countermeasures is significantly high. DDL styles in earlier works demonstrate considerable side channel security with significant amount of area overhead in addition to the performance degradation and power consumption. The extra cost of clocking at the gate level further increases the total trade-off rate. Moreover, the side channel security achieved by DDL requires balancing the load at the differential outputs. In practice satisfying such condition appears to be an ongoing challenge [58]. To reduce the expenditure of the side channel security several other approaches have been proposed. Applying data concealing in the time dimension of power traces is shown to reduce the overhead cost. These techniques are based on introducing misalignment to power traces by applying a random frequency [47] or employing delay variant datapath [48]. Nevertheless, recent developments show that some of these protection techniques can be defeated by enhanced attack methodologies: advanced energy-based partitioning [76] or reshaping-based techniques [77].

Design objectives: In response to the increased demand for side channel security mechanisms the primary objectives of:

- providing a cost effective trade-off scheme for area-constrained cryptosystems
- proposing a less complex logic-based approach

are considered in this thesis for the design and implementation of a countermeasure against information leakage via power consumption.

Design strategy: The strategy for achieving the design objectives is adapted to suppress the data signature in the amplitude of the power consumption of logic cells and storage elements. A novel use of Current Balanced Logic (CBL) is proposed for designing elements of combinational and sequential logics. Design objectives are expected to be met as:

- dynamic and differential properties of the logic style are eliminated (resulting in significant area efficiency)
- single-ended logic is used leading to less design complexity (removing the balancing load requirements).

The concept of data hiding by CBL is described. Design procedure of CBL gates for combinational networks is reviewed. To provide side channel resistant for sequential networks the design of Current Balanced Edge Triggered Registers (CBETRs) is also discussed.

4.3 Data Concealing: Amplitude Dimension

In order to remove the data signature from the amplitude of power consumption, the logic must consume power which is independent of the data processed by the circuit. This condition can be satisfied by employing either equalization or randomization of the power consumption in each transition. Several proposals are discussed in Chapter 3 applying the concepts of equalization and randomization. The characteristic of CBL is investigated as a potential equalization-based approach for data hiding in amplitude dimension of power consumption. CBL is introduced in [78][79] and recommended for low-noise applications on mixed mode (analog/digital) circuits. The operation and design criteria of CBL logic gates are described.

4.3.1 Current Balanced Logic (CBL) for Combinational Network

CBL originated from pseudo-NMOS logic with an extra NMOS transistor which balances the current drawn from the supply in each transition. CBL is a static logic which does not require a clock signal at the cell structure. CBL operates when the data is available at the input, thus, unlike dynamic logic [17], CBL's power consumption is static. This means that CBL cells dissipate power continuously and regardless of presences or absence of the data at the input terminal. Figure 4.1 shows the general configuration of the CBL. The CBL gate operates by PMOS load (M_1) which functions as a constant current source. M_1 and M_2 are assumed to be sized to operate as an identical transistor, so $k_p = k_n = k$ and $V_{th\ p} = V_{th\ n} = V_{th}$, where k_p and k_n are called gain factors and they equal $\mu_p\ C_{ox}\ W_1\ /L_1$ and $\mu_n\ C_{ox}\ W_2\ /L_2$, (μ_p and μ_n are hole and electron motilities, W_1, W_2, L_1 *and* L_2 are the widths and lengths of M_1 and M_2), $V_{th\ p}$ and $V_{th\ n}$ are threshold voltages of M_1 and M_2 transistors, respectively. The voltage values at the gate-source *($V_{gs1,2}$)* and drain-source *($V_{ds1,2}$)* of M$_1$ and M$_2$ can be extracted from Figure 4.1 as follows:

$$V_{gs1} = V_{dd},\ V_{ds1} = V_{dd} - V_{out},\ V_{gs2} = V_{out},\ V_{ds2} = V_{dd}$$

where V_{dd} and V_{out} are supply and output voltage, respectively.
The current of M$_1$ *(I_{M1})* and M$_2$ *(I_{M2})* in triode and saturation modes are expressed as:

$$I_{M1}(triode) = k(V_{dd} - V_{th} - \frac{V_{dd} - V_{out}}{2})(V_{dd} - V_{out}) \quad (4.1)$$

$$I_{M1}(sat.) = k\frac{(V_{dd} - V_{th})^2}{2} \quad (4.2)$$

$$I_{M2}(triode) = k(V_{out} - V_{th} - \frac{V_{dd}}{2})(V_{dd}) \quad (4.3)$$

$$I_{M2}(sat.) = k\frac{(V_{out} - V_{th})^2}{2} \quad (4.4)$$

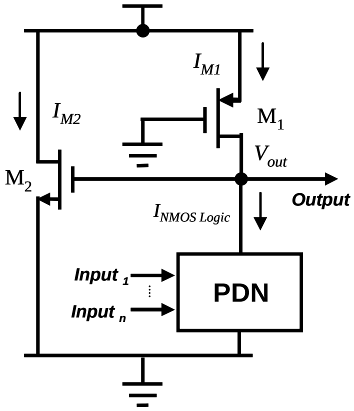


Figure 4.1 CBL structure

First order analysis: The first order analysis of power consumption of the CBL gate is performed with regard to the output voltage *(V_{out})* [75]. For simplicity, the Pull-Down Network (PDN) is modeled by a single NMOS transistor. For different logic functions the PDN can be easily defined and sized. If $V_{out} = 0$ then M$_1$ operates in saturation and M$_2$ operates in cut-off mode. The total supply current *(I_{Vdd})* which equals sum of the currents of M$_1$ *(I_{M1})* and M$_2$ *(I_{M2})* is expressed by Equation 4.5:

$$I_{Vdd} = k\frac{(V_{dd} - V_{th})^2}{2} \quad (4.5)$$

If $V_{out} = V_{dd}$, then M$_1$ and M$_2$ operate in triode and saturation modes, respectively, thus, I_{Vdd} is:

$$I_{Vdd} = k\,(V_{dd} - V_{th} - \frac{V_{dd} - V_{out}}{2})(V_{dd} - V_{out}) + k\,\frac{(V_{out} - V_{th})^2}{2} \tag{4.6}$$

By simplifying Equation (4.6) for $V_{out} = V_{dd}$, I_{Vdd} equals:

$$I_{Vdd} = k\,\frac{(V_{dd} - V_{th})^2}{2} \tag{4.7}$$

Equation 4.7 gives the same value for I_{Vdd} as it is calculated by Equation 4.5. The total I_{Vdd} remains the same for both $V_{out} = 0$ and $V_{out} = V_{dd}$, hence, I_{Vdd} is constant and independent of the logic value at V_{out} . In practice and particularly for short channel, CBL gates do not deliver perfect constant I_{Vdd}. For more realistic analysis we remove the conditions of $V_{th\,p} = V_{th\,n} = V_{th}$ and provide a second order analysis with consideration of the short channel effects. It is still assumed that $k_p = k_n = k$ (The proper sizing should satisfy the condition of $k_p = k_n = k$; however, the impact of process variation does not allow the perfect matching. An evaluation of the data hiding characteristic of CBL with consideration of process variation is presented later in this chapter).

Second order analysis: The transition at the output node of CBL gate can be divided into three regions.

a) $0 < V_{out} < min(\left|V_{th\,p}\right|, V_{th\,n})$

b) $min(\left|V_{th\,p}\right|, V_{th\,n}) < V_{out} < V_{dd} - min(\left|V_{dsat\,p}\right|, V_{dsat\,n})$

c) $V_{dd} - min(\left|V_{dsat\,p}\right|, V_{dsat\,n}) < V_{out} < V_{dd}$

where $V_{dsat\ p,n}$ are the velocity saturation voltages of M_1 and M_2, respectively. The velocity saturation currents for M_1 and M_2 are shown by Equations 4.8 and 4.9 [17]:

$$I_{M_1}\ (velocity\, sat.) = k\,(V_{dd} - V_{th\,p} - \frac{V_{dsat\,p}}{2})\ V_{dsat\,p} \tag{4.8}$$

$$I_{M_2}\ (velocity\ sat.) = k\,(V_{out} - V_{th\,n} - \frac{V_{dsat\,n}}{2})\ V_{dsat\,n} \tag{4.9}$$

The total I_{Vdd} equals I_{M1} + I_{M2} and it is recalculated for the distinct regions: (a) M_1 is in velocity saturation and M_2 is in cut-off (b) both M_1 and M_2 are in velocity saturation (c) M_1 is in triode and M_2 is in velocity saturation.

The total supply current (I_{Vdd}) can be expressed by the following equations:

a) $$I_{Vdd} = k\,(V_{dd} - V_{th\ p} - \frac{V_{dsat\ p}}{2})V_{dsat\ p} \tag{4.10}$$

b) $$I_{Vdd} = k\ [(V_{dd} - V_{th\,p} - \frac{V_{dsat\ p}}{2})\ V_{dsat\ p}] + k\ [(V_{out} - V_{th_n} - \frac{V_{dsat\,n}}{2})\ V_{dsat\,n}] \tag{4.11}$$

c) $$I_{Vdd} = k\ [(V_{dd} - V_{th\,p} - \frac{(V_{dd} - V_{out})}{2})(V_{dd} - V_{out})] + k\ [(V_{out} - V_{th\,n} - \frac{V_{dsat\ n}}{2})V_{dsat\ n}] \tag{4.12}$$

CBL gate (XOR) is designed in 180nm CMOS technology. The operation of the gate is simulated using HSPICE. Figure 4.2 shows the transient waveform of supply current (I_{Vdd}) of CBL gate once the V_{out} changes from logic ‘0’ to ‘1’. Considering the short channel transistor model, one can observed that the constant current objective is not perfectly satisfied.

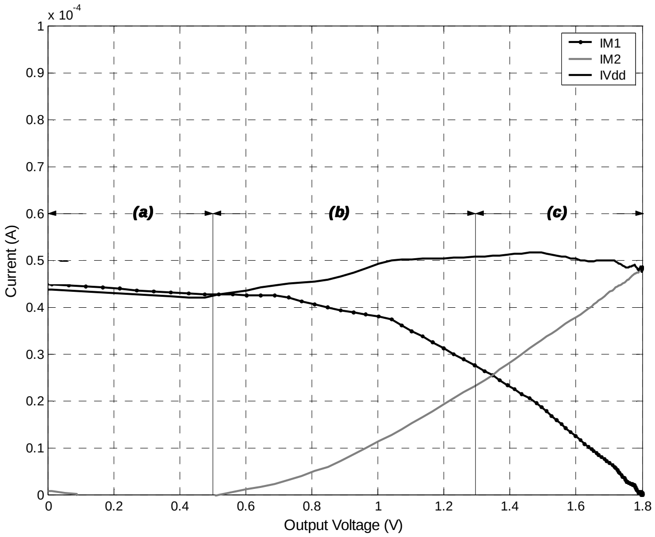


Figure 4.2 Current waveform versus output voltage for CBL XOR gate

The deviations of I_{vdd} (ΔI_{vdd}) in regions (b) and (c) with regard to the I_{vdd} value in region (a) are extracted from Equations 4.11 and 4.12 and presented as follows:

$$\text{b) } \Delta I_{vdd} = I_{vdd}\,(b) - I_{vdd}\,(a)$$

$$= k\;[(V_{out} - V_{th\,n} - \frac{V_{dsat\,n}}{2})(\,V_{dsat\,n})] \qquad (4.13)$$

$$\text{c) } \Delta I_{Vdd} = I_{Vdd}\,(c) - I_{Vdd}\,(a)$$

$$= k\;[(V_{dd} - V_{th\,p} - \frac{(V_{dd} - V_{out})}{2})\,(V_{dd} - V_{out}) - \qquad (4.14)$$

$$(V_{dd} - V_{th\,p} - \frac{(V_{dsat\,p})}{2})(V_{dsat\,p})] + I_{M2}\ (velocity\ sat.)$$

In region (b) the deviation is shown to be equal to velocity saturation current of M_2 (Equation 4.13). Equation 4.14 can determine ΔI_{vdd} for given value of V_{out} which is $V_{dd} - V_{dsat\ p} < V_{out} < V_{dd}$; therefore, ΔI_{vdd} for region (c) is:

$$I_{M2}\ (velocity\ sat.) - I_{M1}\ (velocity\ sat.) < \Delta I_{vdd}\,(c) < I_{M2}\ (velocity\ sat.) \qquad (4.15)$$

Equations 4.13 depicts the role of M_2 in deviation of I_{Vdd} in regions (b). Equation 4.15 also shows that the maximum deviation in region (c) is determined by M_2. It can be concluded that the supply current deviation in operation regions (b) and (c) can be reduced if I_{M2} is decreased. In order to identify the influential parameter in reducing I_{Vdd} Equation 4.9 is revisited. It is seen that by increasing $V_{th\ n}$, saturation current of M_2 is reduced and as a result the deviations in region (b) and (c) are reduced. If M_2 is replaced by a high threshold voltage (V_{th}) NMOS transistor then ΔI_{vdd} is reduced.

Figure 4.3 shows the current deviation in regions (b) and (c) for different values of the $V_{th\ n}$.

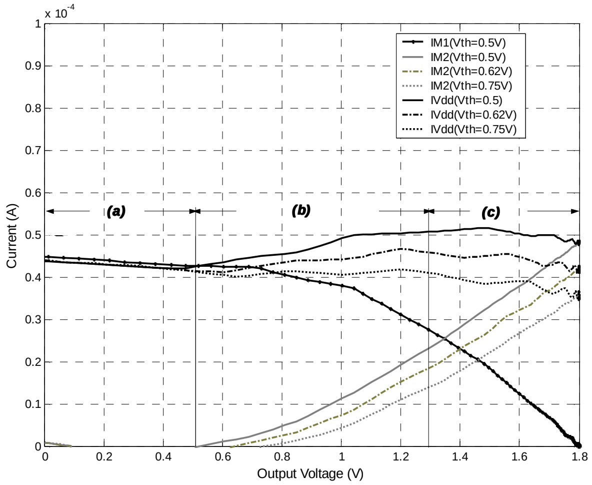


Figure 4.3 Current waveform versus output voltage for CBL gate (M_2 with different V_{th})

Figure 4.4 shows I_{Vdd} in a typical CBL and CMOS XOR gates[1]. As expected the I_{Vdd} deviation in CBL is significantly lower than that of CMOS. Further reduction is achieved as a result of using high threshold balancing transistor (M_2). The deviation of I_{Vdd} in CBL XOR results in current spike approximately 3μA while the current spike in CMOS XOR reaches more than 78μA. Such characteristic can be utilized to reduce the leakage of side channel information. The design procedure of CBL gates is reviewed in the next section.

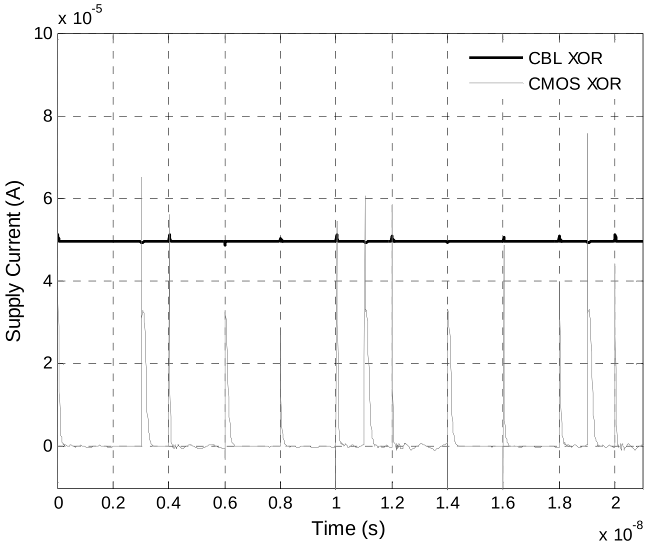


Figure 4.4 Transient waveforms of supply current of CMOS and CBL XOR gates

CBL design procedure: In order to obtain the appropriate values for width *(W)* and length *(L)* of M_1 and M_2, the procedure can be followed based on the given static or dynamic design criteria.

a) Static-based design: If the static criterion, e.g. lower-band of output voltage *(V_{out-L})* is given, thus in region (a) $V_{out} = V_{out-L}$ can be used to determine I_{Vdd} from Equation 4.16 [17]:

[1] XOR configuration for CMOS implementation is shown in Appendix A.

$$V_{out\text{-}L} = (V_{dd}\text{-}V_{th})(1\text{-}\sqrt{1\text{-}\frac{I_{Vdd}}{I_{sat}}}) \tag{4.16}$$

where I_{sat} is the saturation current. Since M$_1$ is in saturation, then, W_1 / L_1 is determined by the I_{Vdd} value. The sizing of Pull-Down Network (PDN) block is also found since M_2 is cut-off and $I_{Vdd} = I_{M1} = I_{PDN}$. Considering the matching criteria, W_2 / L_2 can be set accordingly.

b) Dynamic-based design: If the propagation delays, rise-time *(t_{PLH})* and fall-time *(t_{PHL})*, for a given load capacitance *(C_L)* are specified, the design procedure is similar to sizing of conventional pseudo-NMOS gates. Since M$_1$ operates as a Pull-Up Network (PUN) W_1 / L_1 can be found by approximation of the rise-time as follows [17]:

$$t_{PLH} = (1.7C_L)/k_pV_{dd} \tag{4.17}$$

PDN is modeled by a NMOS transistor, thus, it can be sized by approximation for fall-time as given [17]:

$$t_{PHL} = (1.7C_L)/k_n(1\text{-}\frac{0.46}{k_p/k_n})V_{dd} \tag{4.18}$$

The design procedures described above is helpful for obtaining the initial sizes, the W / L values of M$_1$ and M$_2$ transistors can be further adjusted in a CAD environment. The dynamic-based sizing is used in this research for designing the logic gates. The circuit schematic of inverter, NAND and XOR gates are shown in Figure 4.5. All the gates are designed to operate comparably with their standard CMOS[2] counterparts in 180nm CMOS process. The ratio of the NMOS and PMOS transistors in the logic gates are available in Appendix A.

[2] Standard CMOS inverter and NAND gates are used for comparison.

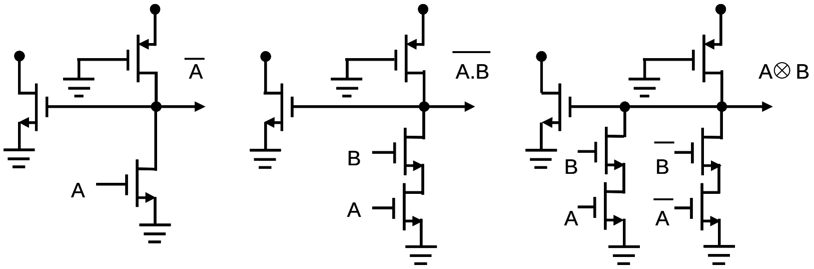


Figure 4.5 CBL a) Inverter b) NAND c) XOR gates

Evaluation of side channel leakage: As discussed in Chapter 2 leakage of information from logic gates is the result of data dependency in the power consumption. The power consumption can be represented by the current in the supply node. The amount of current variation depends on the present state and the next state the gate will switch to. The current variation in the supply branch of a logic gate can be used as an indicator of data leakage. In order to quantify the current variation a test bench shown in Figure 4.6 is used. The fanout signal degradation is considered in both the previous and the succeeding stage (C_l =10fF). Instantaneous current is simulated at the transistor level using HSPICE. In order to capture the current variations, measurements are conducted for 100-cycle-long pseudorandom data sequences. Normalized Current Deviation (NCD) [52] is used as a measure of the variation on the current consumption.

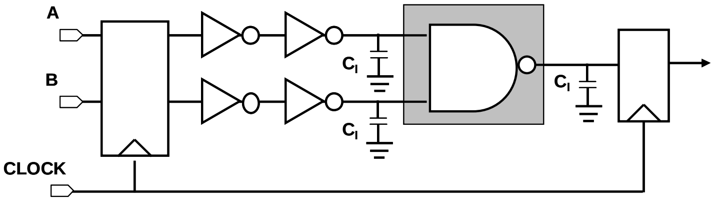


Figure 4.6 The simulation test bench

NCD is defined as:

$$\text{NCD} = \frac{\text{Max (current /cycle) - Min (current /cycle)}}{\text{Max (power /cycle)}} \tag{4.19}$$

The value of NCD ranges from 0 to 1. Smaller NCD indicates less leakage, thus, more effort needs to extract the side channel information. In general NCD can be used to illustrate the degree of susceptibility to side channel attack [52]. The measurement is only related to the gate in the gray box. The results are presented in Table 4.1. The area consumptions of the gates are also reported (normalized based on CMOS gate). Significant decrease is seen in NCD of the CBL gates. However, area consumption depends on the gate type. Inverter and XOR are the most and least area consuming gates in CBL design, respectively.

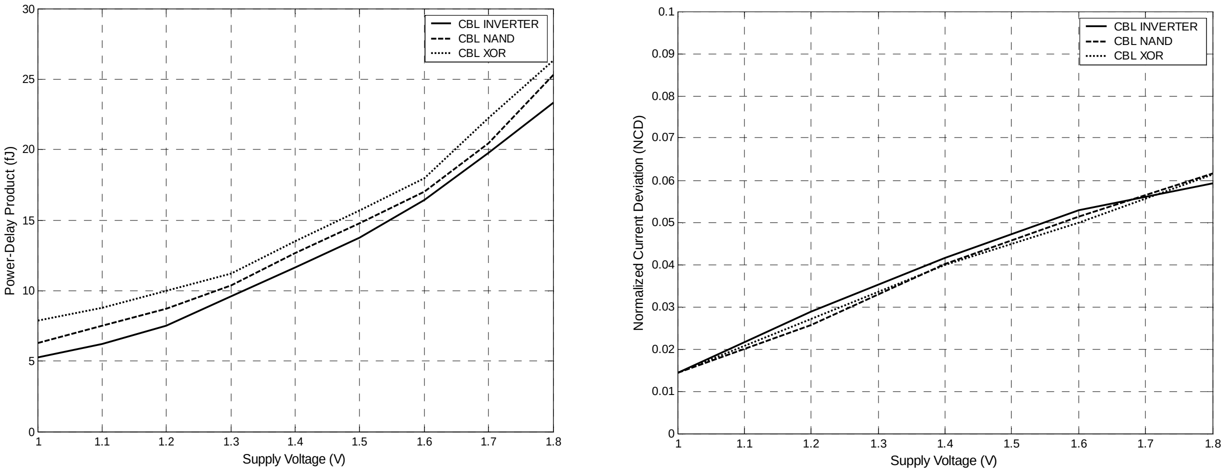
Table 4.1 Comparison of NCD and area between CMOS and CBL gates

Logic	**NCD**			**Area**		
	Inverter	NAND	XOR	Inverter	NAND	XOR
CMOS	1.00	1.00	1.00	1.00	1.00	1.00
CBL	0.058	0.064	0.063	1.79	0.82	0.8

Further observation: CBL operates based on a constant current source. The operation principle already suggests that ideally the power is consumed continuously in CBL, regardless of the switching activity. Thus, the power consumption is essentially independent of the switching frequency. On one hand, this could be advantageous for very high-speed operation since the total power consumption may become smaller than the power consumption of CMOS gates [80]. On the other hand, significant static power consumed by CBL gates is a limiting factor for deployment of this logic in power-constrained applications. The potential approach for reducing the power consumption in CBL gate is investigated.

Unlike other Current Mode Logic (CML) styles such as Current Steering Logic (CSL) [81] or MOS Current Mode Logic (MCML) [59] which operate by using V_{bias}-controlled current source, CBL is structured with a current equalizer transistor. Decreasing the supply voltage *(*V_{dd} *)* for power consumption reduction is shown to affect the correct operation of CSL and MCML [75]. The current source of CBL is composed of a PMOS saturated transistor which tolerates greater variation of the supply current. Lowering V_{dd} can, thus, be used as a potential power saving option. Nonetheless, Equations 4.17 and 4.18 show that performance of the CBL gate depends on supply voltage. Therefore, a realistic comparison requires considering the effect of supply voltage reduction on both power and performance. The product of power and delay known as power-delay is used for comparison. Power-delay products of CBL gates for different V_{dd} values are plotted in Figure 4.7 (a). The variation of NCD with supply is illustrated in Figure 4.7 (b). It is evident that lowering V_{dd} reduces the amplitude of variation in I_{vdd}, subsequently develops more resistance against side channel information leakage. Therefore, lowering the supply offers an alternative trade-off scheme which gives an option for reduced power consumption and less side channel information leakage at the cost of performance degradation[3].

[3] Note that for smart card applications speed performance is not a prior concern as most smart cards have an internal clock frequency ranging from 10 to 20 MHz. Current state-of-art smart cards operate at maximum 50 MHz [SLE88CX720P Short Product Information, http://www.infenion.com].



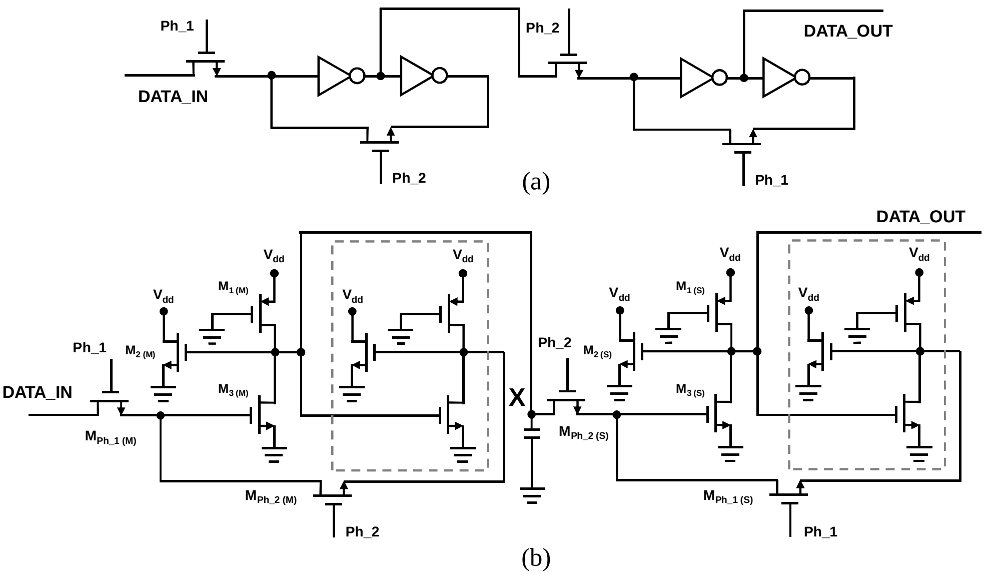
(a) (b)

Figure 4.7 a) Power-delay b) NCD variation with supply voltage for CBL gates

4.3.2 Current Balanced Logic (CBL) for Sequential Network

Data-dependency of the power consumed by logic cells causes information leakage in combinational networks. Similar to logic gates, the power consumption of storage elements is also data dependent. Thus, storage elements are susceptible to information leakage. A strong side channel aware design strategy requires both logic cells and storage elements to be protected against the threat imposed by side channel attack. Architecture of a side channel resistant storage element is described.

a) Current Balanced Edge Triggered Register (CBETR) I: Registers or so called memories can be either designed in a dynamic or static style. Dynamic memories store data for a short period of time. They are based on the principle of temporary charge storage on parasitic capacitors associated with MOS transistors. Storage in the static scheme relies on the concept of producing a bistable element. Static memories preserve the state as long as the power is turned on. They are built using positive feedback, where the circuit topology consists of intentional connection between the output and input of a combinational circuit. A leakage resistant register is designed by integrating CBL gates into a similar structure. The current behavior of CBL is utilized to suppress the data signature in power consumption of the register. In order to operate as a positive edge triggered register, the register is designed based on master-slave concept. Two CBL inverters are connected in two-phase non-overlapping clock scheme. Figure 4.8 (a) and (b) show the gate and transistor levels of the Current Balanced Edge Triggered Register (CBETR). CBETR consists of two identical blocks which serve as master and slave stages. Transistors in master and slave stages are distinguished by indices of M and S in Figure 4.8 (b).



(a)

(b)

Figure 4.8 Current Balanced Edge Triggered Register a) gate level b) transistor level

Transistors configuring the CBL inverter are $M_{1(M)}$, $M_{2\,(M)}$ and $M_{3(M)}$ in master stage and $M_{1(S)}$, $M_{2(S)}$ and $M_{3(S)}$ in slave stage. One extra inverter is included in each stage for closing the feedback loop to preserve the correct value during the holding period. Transistors controlling the clock in the master stage are $M_{Ph\_1\ (M)}$, and $M_{Ph\_2\ (M)}$. Transistors $M_{Ph\_1\ (S)}$, and $M_{Ph\_2\ (S)}$ form the clocking network in the slave stage. The clock scheme of CBETR is composed of a two-phase non-overlapping clock signal which controls sampling and holding the data in master and slave stages. Employing the non-overlapping clock instead of using complementary clock scheme removes the condition of generating ideal CLK and $\overline{CLK}$ (with zero delay) in design of the register. Non-overlapping clock also avoids occurrence of undefined state at the output of register. Figure 4.9shows an implementation of the clock circuitry in CBETR for generating a two phase non-overlapping clock.

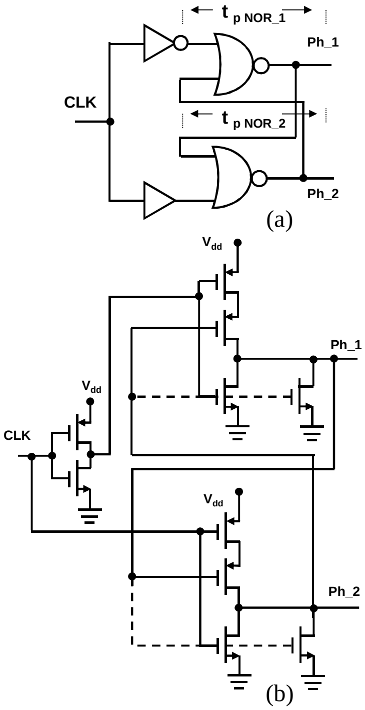


Figure 4.9 Clock generating a) gate level b) transistor level

The operation of CBETR in one clock cycle is explained as follows. When the Ph_1 signal is ‘1’, transistor $M_{Ph\_1\ (M)}$ is turned on; DATA_IN is sampled by the master stage into node

‘X’. During this period, $M_{Ph\_1\ (S)}$ is on and forms the feedback loop at the slave stage. Ph_1 and Ph_2 are non-overlapped, thus, Ph_2 is ‘0’ so $M_{Ph\_2\ (S)}$ is off and the output is decoupled from input. Consequently, the slave stage is in hold mode and DATA_OUT retains its previous value.

When the Ph_1 signal is ‘0’, $M_{Ph\_1\ (M)}$ is turned off, hence, the master stage stops sampling the input. $M_{Ph\_1\ (S)}$ is also turned off and the feedback loop at the slave stage becomes open. Due to non-overlapping, Ph_2 is ‘1’and it turns on $M_{Ph\_2\ (S)}$. The inverse value of DATA_IN stored in ‘X’ is inversed and copied to the output. Ph_2 also turns on $M_{Ph\_2\ (M)}$. The feedback loop at the master stage is formed to hold the state of node ‘X’. For comparison purposes, the inverters are replaced by standard CMOS gates. The test bench in Figure 4.10 is used and a similar procedure as the one in Section 4.3.1 is followed for evaluating the leakage of side channel information.

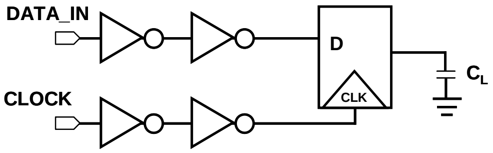


Figure 4.10 Test bench

Figure 4.11 shows the transient waveforms of data-in/out, and supply current of the CBETR compared to the one designed by using CMOS gates. NCD is measured and results are shown in Table 4.2. As expected utilizing the CBL gates provides significant reduction in NCD values in CBETR indicating the resistance that CBL style provides against leakage of data via power consumption. The area consumed by CBETR follows same trend as seen in designed gates in Section 4.3.1.

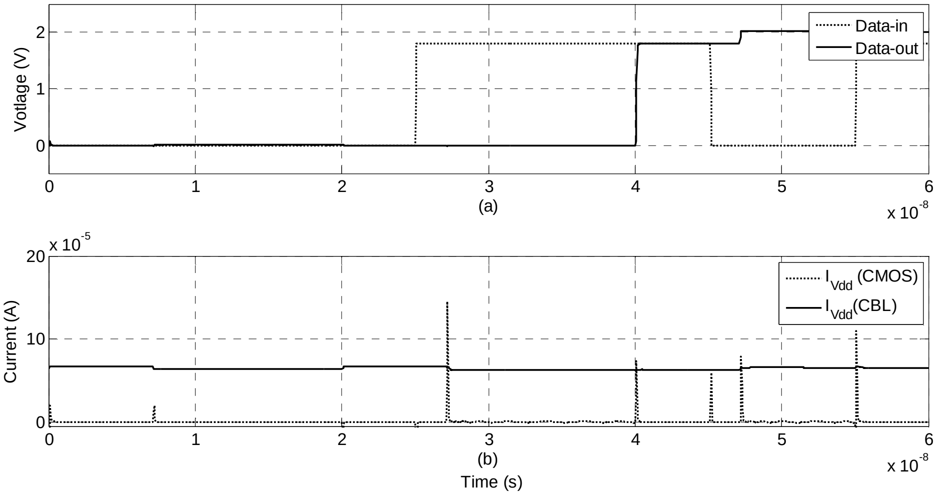


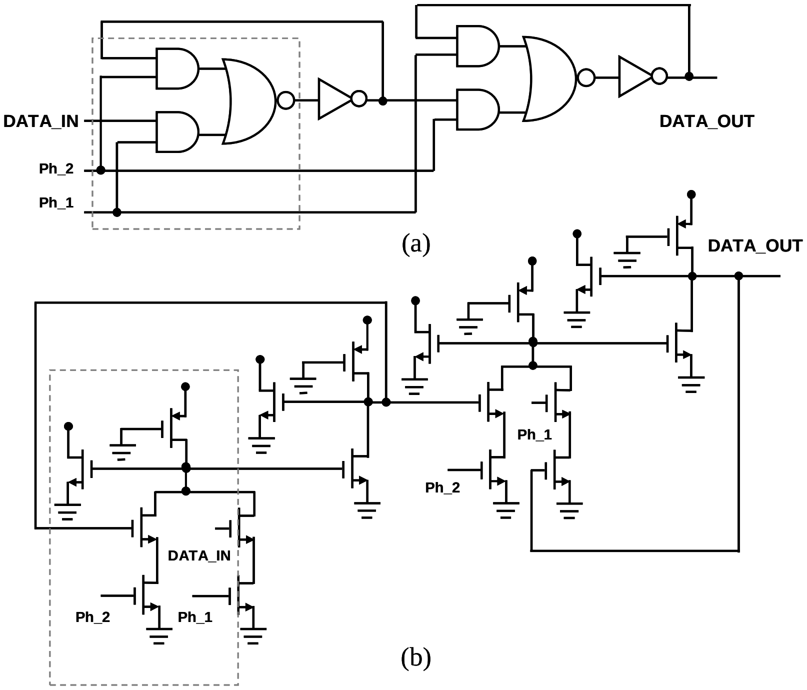
Figure 4.11 Transient waveforms of a) input/output b) supply current of CMOS and CBL registers

Table 4.2 Comparison results

Logic	**NCD**	**Area**
CMOS-ETR	1	1
CBETR (I)	0.06	1.61

Further Observation: As shown in Section 4.3.1.2, the power consumption of CBL gates can be decreased by lowering the supply. A similar approach is investigated in CBETR. The Ph_1 and Ph_2 signals controlling the sampling and hold sequences are applied by an NMOS-only switch. NMOS switch passes a degraded high voltage *(V_{dd} - $V_{th\ n}$)* to the input of the sampling inverters. Consequently, CBETR operates under the condition of input logic equals V_{dd} - $V_{th\ n}$. Lowering the supply as an option for reducing the power consumption is restricted as it affects the input logic value. In order to effectively reduce the power consumption and ensure that the CBETR still reliably operates, the NMOS switches have to be removed. A modified version of CBETR is presented in which the NMOS switches are replaced by gates, so that lowering the supply can be applied with no concerns about degrading the input logic values.

b) Current Balanced Edge Triggered Register (CBETR) II: Architecture of the modified CBETR is shown in Figure 4.12. The NMOS-only switches are replaced by combination of AND and OR gates. The sampling and hold operations are similar to the first version of CBETR. The DATA_IN and Ph_1 signals command the sampling (master stage) through the AND gate instead of NMOS-only switch. The feedback loop is also modified by adding an AND gate operating between Ph_2 and sampled DATA_IN. The operation of the second version of CBETR is reviewed.



(a)

(b)

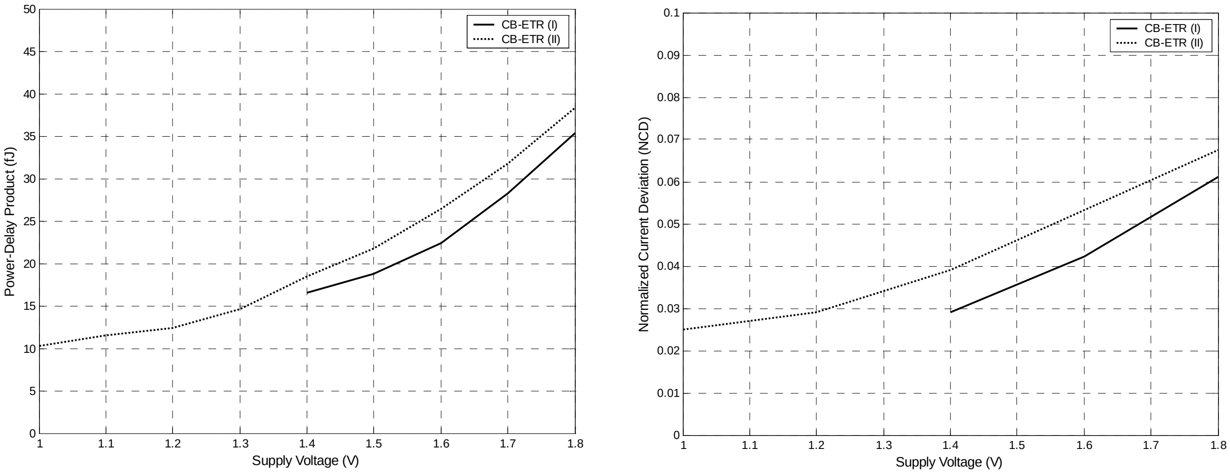
Figure 4.12 Modified CBETR a) gate level b) transistor level

When Ph_1 is ‘1’, the AND operation between Ph_1 and DATA_IN results in completion of sampling at the master stage. Ph_1 and Ph_2 are non-overlapped, thus, Ph_2 is ‘0’ which allows the feedback loop to be formed at the slave stage and hold operation is executed. By changing the status of the Ph_1 and Ph_2 signals to ‘0’ and ‘1’, respectively, the master stage performs hold and the slave stage begins sampling.

It is evident that area is increased in second version of CBETR; however, since the voltage drop caused by NMOS-only switch is no longer an issue it is expected that further lowering V_{dd} can be considered for reducing the power consumption. The comparison is presented in the following to quantify the trade-off between the area and power reduction in two versions of CBETR.

Trade-off scheme: Trade-off scheme can be drawn by comparing two versions of CBETR. An evaluation is conducted by using same simulation setup (Figure 4.10). The comparison results are given as follows.

a) Performance of registers is often measured by the setup and hold times. The setup time and hold time constraints are defined to ensure that the data signal is properly propagated by the register and the result remains valid at the output during the sampling period. The setup time is the time before the rising edge of the clock that the input data must be valid. For CBETR I, the DATA_IN signal has to be propagated through an NMOS and the CBL inverter of the master stage. This ensures that input data (DATA_IN) is sampled at the node ‘X’. The Ph_1 signal which is the actual clock into the master stage requires a setup time that is equal to the propagation delay of an NMOS-only switch ($t_{p\ NMOS}$) and two CBL inverters ($t_{p\ CBL\text{-}inverter}$). Thus, the setup time with regard to the original clock equals: $t_{p\ NMOS} + 2 \times t_{p\ CBL\text{-}inverter} - t_{p\ NOR\_2}$. For CBETR II, the time required by the Ph_1 signal equals delay of the CBL combined AND/NOR gate ($t_{p\ CBL\text{-}AND/NOR}$) plus $t_{p\ CBL\text{-}inverter}$. Hence, the setup time in CBETR II equals: $t_{p\ CBL\text{-}AND/NOR} + t_{CBL\text{-}inverter} - t_{p\ NOR\_2}$. The hold time represents the time that input must be held stable after the rising edge of the clock. Hold time for CBETR I and II equals: $t_{p\ NOR\_1} + t_{p\ NOR\_2}$. The propagation delay is also defined as the time it takes for delivering the DATA_IN from the slave stage to the final output. For CBETR I the propagation delay equals: $t_{p\ NMOS} + t_{p\ CBL\text{-}inverter} + t_{p\ NOR\_1} + t_{p\ NOR\_2}$.



(a) (b)

Figure 4.13 a) Power-delay b) NCD variation with supply voltage for CBETR I and II

For CBETR II, the propagation delay is $t_{p\ \text{CBL- AND/NOR}} + t_{p\ \text{CBL-inverter}} + t_{\text{P NOR\_1}} + t_{\text{P NOR\_2}}$.

b) Power consumption in CBETR II is slightly greater than in CBETR I; however, CBETR II can operate with supply as low as 1V whereas CBETR I may not operate reliably with supply lower than 1.5V due to voltage drop on NMOS switch. This means that reducing the power consumption in CBETR II can be more effectively attained by lowering the supply voltage. Figure 4.13 (a) presents the power-delay products versus the supply voltage in CBETR I and II. NCD variation with supply voltage is shown in Figure 4.13 (b).

c) Area consumed by CBETR I and II is compared. CBETR II is designed by replacing the NMOS-only switch and the first CBL inverter in master stage of CBETR I with the CBL AND/NOR gates. As shown in Figure 4.12 (dashed boxes) AND/NOR gate can be implemented in a combined architecture resulting in less area consumption comparing to implementation of individual gates. CBETR II consumes 6.96μm^2 compared with CBETR I which consumes 6.27μm^2. An increase of 11% in area is, thus, seen.

d) Comparison Overview: The above review of performance, power and area assists a designer to select CBETR I or II for a particular application. For example, the setup time in CBETR I is slightly longer than CBETR II (approx. $t_{p\ NMOS}$), whereas the propagation delay of CBETR II is larger for about $t_{p\ CBL\text{-}AND/NOR}$ - $t_{p\ NMOS}$.

Power consumption in CBETR designs is slightly different; however, the characteristic of CBETR II to operate with V_{dd} as low as 1V provides more power savings at the cost of performance. NCD follows a similar trend as it is seen in CBL gates. The reduction of NCD with lowering V_{dd} can be advantageous for CBETR II. CBETR I is more area efficient than CBETR II.

4.4 The Proposed Countermeasure: Architectural Level

A subset of an encryption algorithm is chosen as a test structure for analyzing the data-dependency of the power consumption of CBL gates at the architectural level. The results are compared with an identical architecture that is designed using standard CMOS logic gates. The test structure is briefly introduced and detail of the analysis process is discussed.

4.4.1 Information leakage at the Architectural Level

The test structure is composed of "SubByte" in AES algorithm [27]. "SubByte" consists of an affine function and inverse multiplication in GF(2^8). "SubByte" is executed by a block known as Substitution Box (SBOX). Figure 4.14sows the architecture of the SBOX.

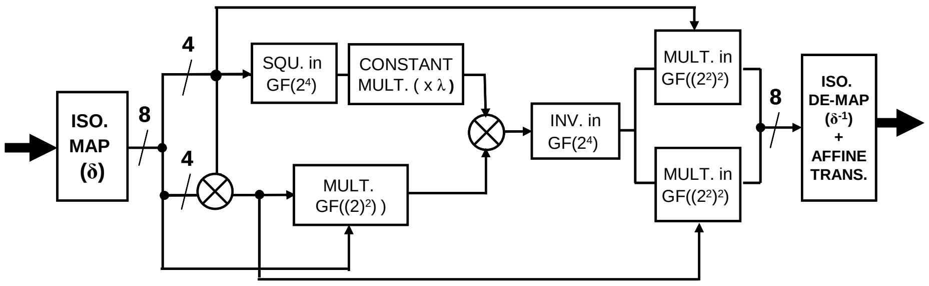
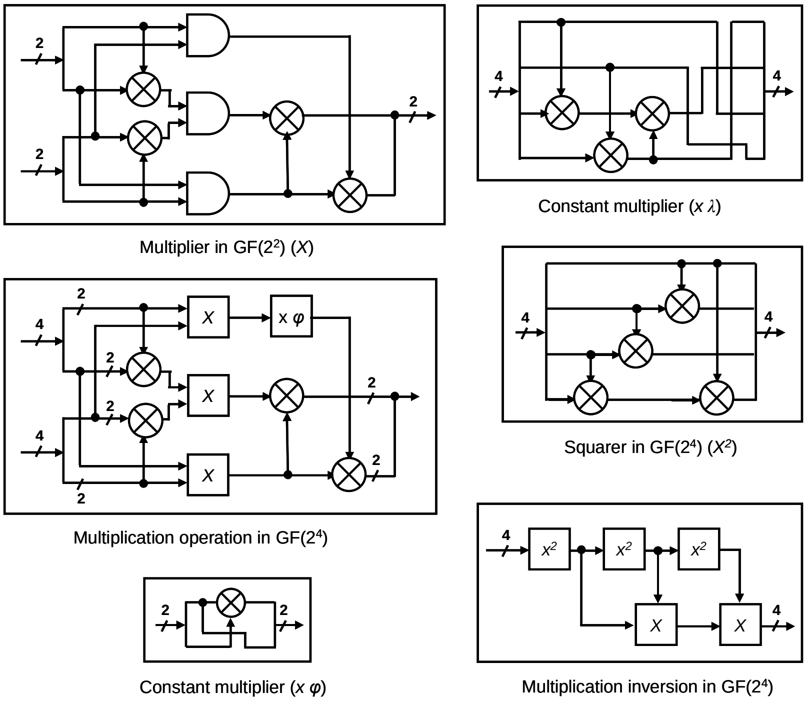


Figure 4.14 Block diagram of "SubByte" operation (SBOX) [27]



Multiplier in GF(2^2) (X)

Constant multiplier (x λ)

Multiplication operation in GF(2^4)

Squarer in GF(2^4) (X^2)

Constant multiplier (x φ)

Multiplication inversion in GF(2^4)

Figure 4.15 SBOX building blocks [27]

The gate level schematics of the SBOX blocks are shown in Figure 4.15. Further details on the architecture of SBOX are given in Appendix B. The SBOX consists of 45 two-input NAND gates, and 140 two-input XOR gates [27]. The test structure is designed in 180nm CMOS technology. For a comparative analysis, the test structure is designed using CBL and standard CMOS logic cells. Variation of the power consumption is measured for a random input sequence of 500 clock cycles. Figure 4.16 shows a superposition of the power supply current of the transient response. The instantaneous current of the CMOS implementation is highly irregular. However, the instantaneous current of CBL SBOX is subjected to minor variations since nearly same amount of power is consumed in each cycle. The instantaneous current of the CMOS design at its peak reaches 40mA, while the current fluctuation of CBL remains confined to a narrow margin of approximately 2mA, with constant value of 34.6mA.

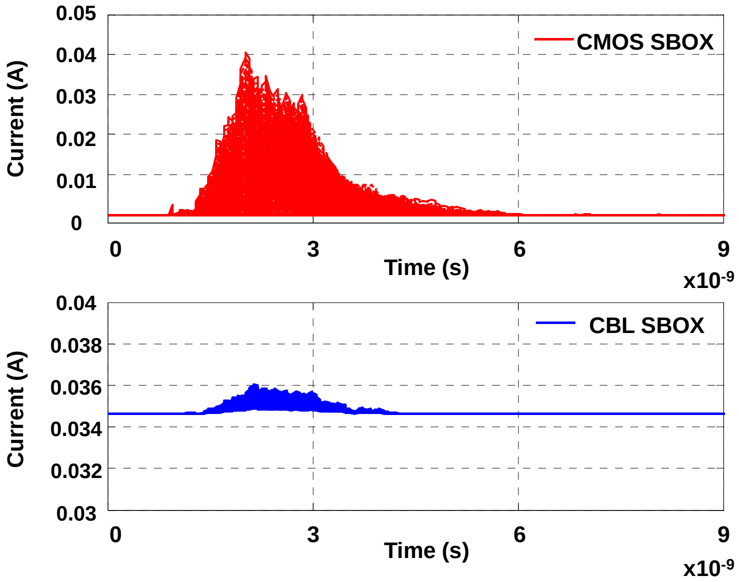


Figure 4.16 Superposition of the simulated instantaneous power supply current for 500 clock cycles a) CMOS b) CBL SBOX

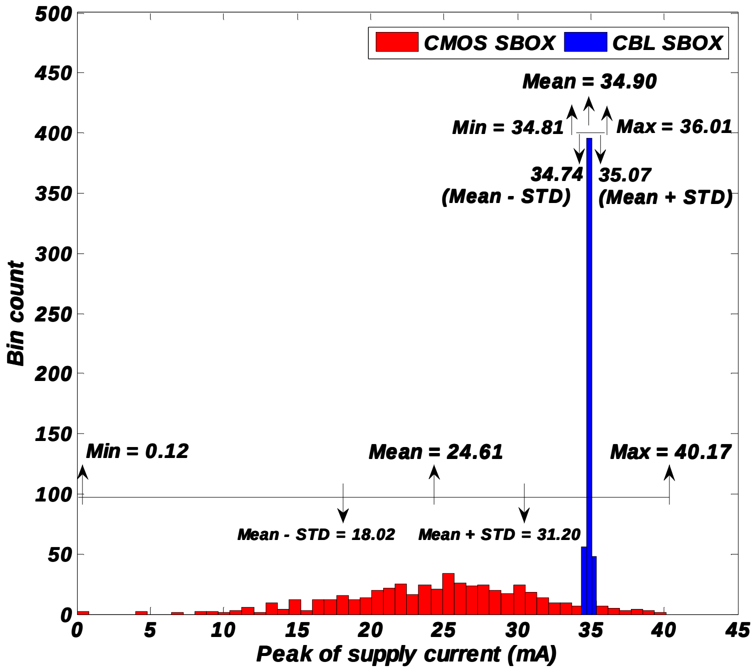


Figure 4.17 Histogram of the simulated peak of the current per cycle for 500 cycles

Table 4.3 Comparison results

Design Style	**NCD**	**Area**
CMOS	1	1
CBL	0.06	0.8027

Figure 4.17 shows the histogram of the peak of supply current. The results show that the peak of supply current in CMOS implementation experiences a large variation, whereas the deviation of the peak of supply current of CBL remains in a narrow band.

Table 4.3 presents the NCD for CMOS and CBL SBOX. NCD values also show significant reduction in CBL compared to CMOS. This indicates that the power behavior of CBL gates is also effective at the architectural level by significantly reducing the variation of power consumption. In other words, the data signature in power consumption of test structureis drastically reduced. The reduction is obtained by increasing the power consumption. The discussions and simulation results demonstrate that the proposed approach effectively reduces the current variations caused by the data values. This consequently shows the suitability of using CBL for applications that requires the data-independent power.

4.4.2 Information Leakage: Technology Impact

The most governing technology variations are seen in sizing of the transistors and the assigning the threshold voltages. As discussed in Section 4.3.1, the effect of the process variation should be considered in evaluation of the data hiding characteristic of CBL. Analysis of power behavior of CBL-based architecture under process variation is presented.

Monte Carlo is a commonly used method for analyzing the impact of process variation on design parameters. The deviation from the nominal value can be computed for design parameters by running the simulation for *n* iteration rounds. The Monte Carlo simulation is an available feature in Cadence design environment which computes the impact of both mismatch and process variation on a defined parameter. We use NCD as a parameter which shows the variation on the current consumption. Monte Carlo simulation is run for 2000 iteration rounds. Thus, in total 2000 values are obtained for NCD. The distribution of these values in Figure 4.18 shows that the variation of NCD is between 0.053 and 0.088. In worst case NCD reaches to 0.088. The NCD value of the CMOS architecture in Section 4.4.1 is 1. It is seen in worst case, NCD of the CBL architecture is only 8% compared to that of the CMOS architecture. Therefore, NCD is still significantly smaller even with consideration of the process variation. This implies that CBL can be considered as an option for further investigation. The next chapter presents an implementation of a crypto core on a fabricated chip and investigates the side channel resistance of CBL in a real attack scenario.

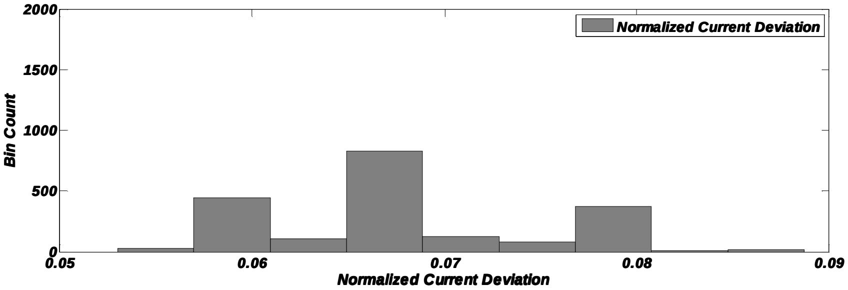


Figure 4.18 Histogram of NCD distribution of the CBL SBOX in 2000 iterations

4.5 Summary and Conclusion

This chapter defined our objectives and strategy for designing a side channel countermeasure at the logic level. Area efficiency and reduced complexity were given priority in our proposed design. A novel exploitation of Current Balance Logic (CBL) was proposed for designing side channel resistant logic cells. Current Balanced Edge Triggered Register (CBETR) was also introduced to provide side channel resistance for sequential network. The major trade-off element for obtaining side channel resistance was the power consumption. It was seen that power savings can be achieved as a result of reduced supply voltage. The result of this supply reduction also improved the side channel resistance. A subset of the AES algorithm was used as a test bench for assessing the proposed countermeasure at the architectural level. The supply current variation in CBL was seen as low as 6%. The results showed that side channel resistance of CBL was achieved by 20% area saving compared to CMOS. Furthermore, the impact of process variation was investigated on power consumption behavior of CBL-based architecture. Measure of Normalized Current Deviation (NCD) was computed over 2000 iteration rounds in Monte Carlo simulation. The worst case deviation in NCD being 8% was significantly lower than that of the CMOS architecture. The simulation-based results in this chapter showed that CBL has potential warranting further investigation.

Chapter 5
The Empirical Results

5.1 Introduction

In order to validate the results from the earlier studies on the proposed countermeasure, a realistic security assessment should be performed. A test chip is designed and fabricated. A side channel attack is mounted on the test chip using real power measurements. This chapter first reviews the implementation of the test chip which includes two separate cores: the proposed side channel resistant logic core (protected core) and standard CMOS logic core with no countermeasure (unprotected core). The functional test and attack procedure including data capturing and data analyzing are explained. The empirical results of the attack on the test chip are presented. Furthermore, a comparative analysis is included in this chapter to highlight the characteristics of the proposed countermeasure.

5.2 Test Chip: Design and Implementation

The proposed countermeasure in Chapter 4 employs non-standard cells at the logic level; therefore, a full custom approach is required for design synthesis, floorplanning, placement, routing, verification and finally tape-out. A brief overview of the design procedure is given.

5.2.1 Implementation of the Test Chip

Basic logic functions including inverter, NAND and XOR are designed using Current Balanced Logic (CBL). The cell library including the CBETR I is implemented in 180nm CMOS technology. The design procedure is performed in Cadence [82], layout and post layout simulations are executed by HSPICE. Layout to netlist is done in Analog Artist of Cadence. Layout is created with Layout Plus of Cadence [82].

The test structure is composed of the “AddRoundKey” and “SubByte” transformation of AES algorithm [27]. “AddRoundKey” is an XOR operation between the plaintext and the key. “SubByte” operation is executed by SBOX. The SBOX blocks (See Section 4.4.1) are recognized by their Boolean equations which are presented in Appendix B. Figure 5.1 depicts the architecture of the test chip.

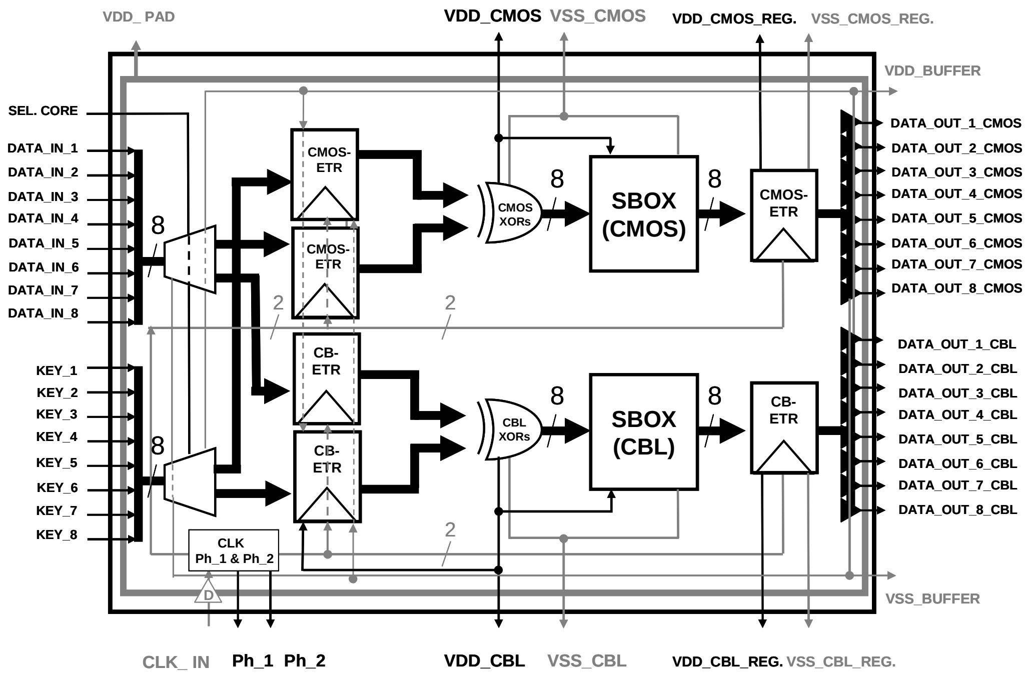
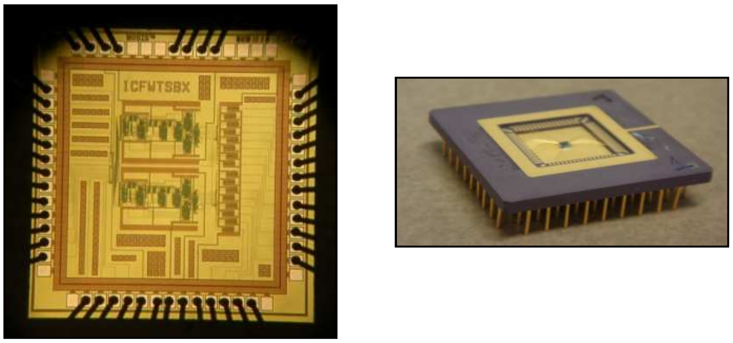


Figure 5.1 The architecture of the test chip

To allow for comparison the test chip includes protected and unprotected cores which are designed using CBL and standard CMOS logic, respectively. The test chip is fabricated in 180nm CMOS TSMC 6M technology. The total number of pins in the test chip is 47. In total five test chips are fabricated. Three test chips are tested for functionality and one is used in the side channel attack. The die photo of the chip is shown in Figure 5.2 (a). The photograph of the final packaged (PGA 68) test ship is presented in Figure 5.2 (b).



(a) (b)

Figure 5.2 a) The photograph of the die b) The packaged fabricated-chip (PGA 68)

5.3 Testing and Verification

Pre-test hardware and software setup are performed. The hardware setup involves designing a platform which enables sending and receiving the data between the tester and the chip. The software setup involves defining the test flow, characterizing the Device-Under-Test (DUT) and setting up the parameters for the testing procedure. Functional test is explained after describing the pre-test setup. Detailed explanation of functional test is useful as some of the tester features are introduced which are also utilized in side channel attack.[4]

[4] Testing took place at the Advanced Digital Systems Laboratory (ADSL) at the University of Toronto (Figure 5.3). The access to the ADSLab was provided by Canadian Microelectronic Corporation (CMC).

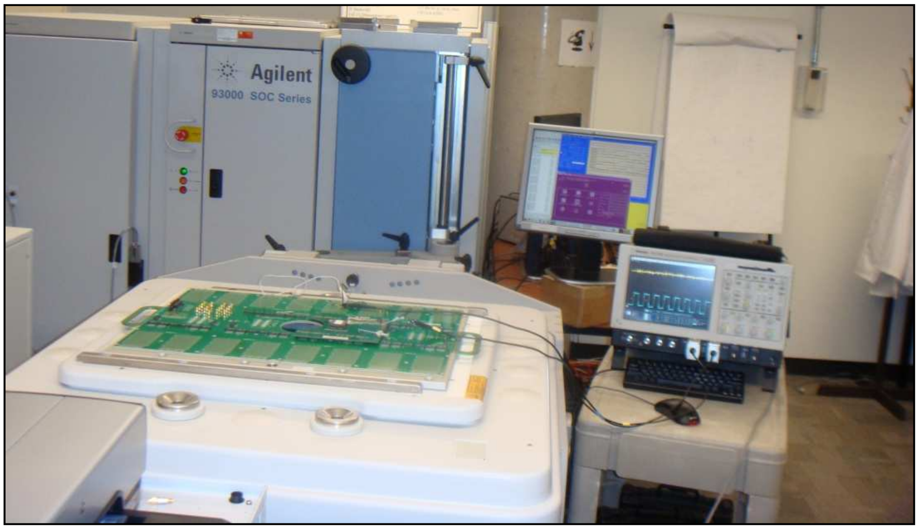


Figure 5.3 Experimental setup in the Advanced Digital Systems Laboratory (ADSL)

5.3.1 Pre-test Setup

In order to evaluate the performance of the test chip, the testing equipment at the Advanced Digital Systems Laboratory (ADSL) including the Agilent 93000 SOC tester is used. The specifications of the tester are available in Appendix C. Some of the pre-test hardware and software setup are performed remotely via the Virtual Network Computing (VNC) connection. A summary of the preparation is discussed.

a) Pre-test hardware setup: As mentioned earlier for comparative analysis the identical core is designed using standard CMOS cells and it is included in the test chip. Separate V_{dd} and *GND* pins are assigned for each core's logic and output registers (CBL and CMOS). Buffers and pads are supplied with separate V_{dd} pins. The power consumption of the additional circuits is, thus, excluded from the measurements.

Design of a Printed Circuit Board (PCB) is required so that the chip and the tester can communicate through the PCB and test fixture. Figure 5.4 (a) shows the schematic of the PCB top layer. Figure 5.4 (b) depicts the fabricated PCB, test chip and the mounted probe.

The tester provides reliable channels for data flow to the test chip and also reduces the impact of environmental noise. The tester is controlled with a PC terminal using Linux-based smart test software provided by Verigy [83]. This feature facilitates the functional test and provides an automated platform for mounting a side channel attack. A brief explanation of pre-test software setup is given.

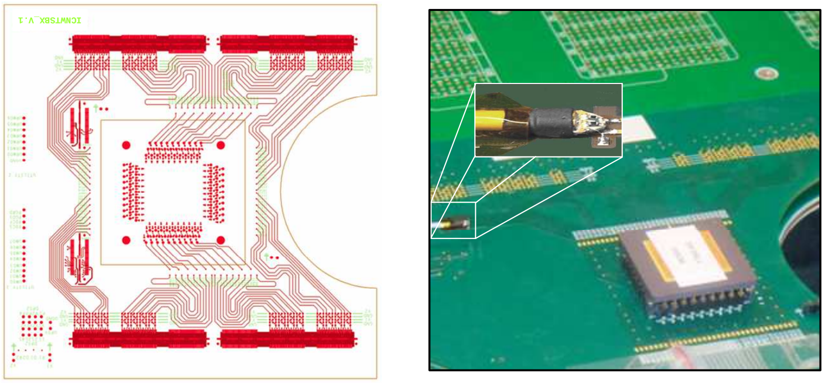


Figure 5.4 a) Schematic of the top layer of PCB b) Fabricated PCB, test chip and mounted probe for measuring the voltage variation in supply branch

b) Pre-test software setup:

The pre-test software setup involves generating several files to characterize the test chip and tune the tester parameters. The following files should be created.

Pin configuration needs to be defined. This step determines the paths between the chip and the tester for receiving the DATA_IN (plaintext), KEY (key), CLK (CLK_IN), V_{dd}, *GND* and sending DATA_OUT (SBOX output) signals. Defining the pins involves assigning the tester channels to the pins, characterizing the power supply operating range and setting the pin groups. Pin numbers (extracted from the bonding diagram and the packaging data sheet) are assigned to the tester channels. The pin identification numbers are also set. Defining the levels for the pins is included in this step. The pins are characterized as I/O or supply with the assigned communication voltages and low and high termination currents. Appendix D provides the detail of the pin configuration.

Timing setup is the next step of the pre-test preparation. Timing setup defines the duration of the input signals, clocking event and output sampling.

Vector file is the last pre-test procedure which provides the testing pattern. The test pattern is first generated for verifying the functionality of the test chip. The patters are modified later for using in the side channel attack. The general format for the vector patterns in the Agilent 93000 SOC tester is shown in Figure 5.5. In the first column, clock signal (CLK_IN) is introduced to the test chip. The DATA_IN, KEY and corresponding DATA_OUT vectors are represented in 8-bit binary format in each clock cycle. The DATA_OUT vectors are "SubByte" results of DATA_IN which is XORed with the KEY prior to entering to the SBOX. Note that the output data corresponding to the input and key values arrives with one clock cycle delay; therefore, the output values at the first line of the test vector are considered as "don't care" (xxxxxxxxxxx). In the first row in Figure 5.5 the "0xff" is XORed with "0x88" and final result (after "SubByte" operation) appears at the DATA_OUT column of the second row. The SBOX table and complete test vector for a given key is available in Appendix E. The vectors are created in an ASCII file and converted to the format (binL) that is readable by the tester. In the remainder of the thesis DATA_IN, KEY and DATA_OUT signals are referred to as plaintext, key and SBOX output. Pre-test setup is completed once all pin configuration, level, timing and vector files are appropriately created.

```
CLK_IN          DATA_IN             KEY           DATA_OUT

  1         11111111(0xff)     10001000(0x88)    xxxxxxxx(0x00)
  1         11111110(0xfe)     10001000(0x88)    11110101(0xf5)
  1         11111101(0xfd)     10001000(0x88)    00111000(0x38)
  .               .                  .                 .
  .               .                  .                 .
  1         00000001(0x01)     10001000(0x88)    10001011(0x83)
```

Figure 5.5 Test vectors

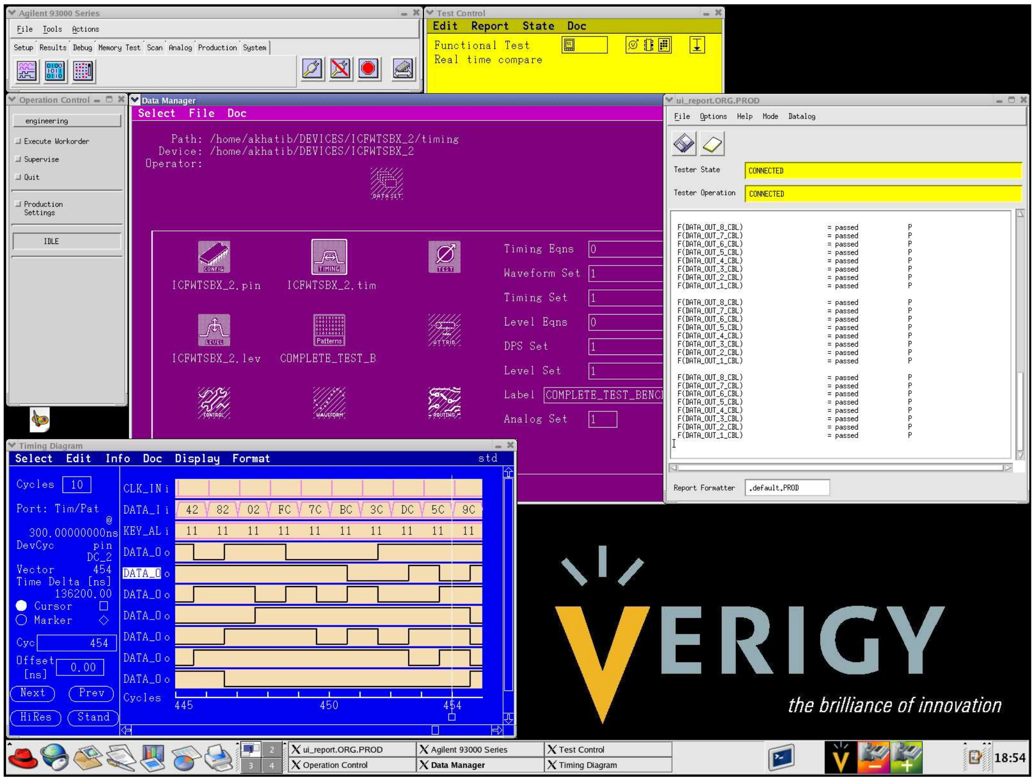
5.3.2 Test Execution

In order to reduce the possible debug time, it is highly recommended to run a continuity test. Continuity test is a DC test function that checks the connectivity of the pins to the tester channels. It is likely that pins of the DUT are not properly contacted by the sockets. This causes failure in the testing. The continuity test verifies all the contacts from the channel to the pads (internal connection of the test chip). Failure in the continuity test is reported by the pin number. By physically locating the pin and inspecting the device, one can make sure that pins are properly connected and secured within the socket holes.

Functional test: Once all the above steps are completed, the functional test can be executed by applying the vector patterns to the test chip. The Agilent 93000 SOC tester provides several options for functional test. For general functionality purpose the DC characterization test is used. Three options are available in the test control window. The vector patterns can be applied “only one time”, “endlessly” or “run until fails”. Depending on the sequence and purpose of testing, one of these options can be used. The “only one time” is used at the beginning of the testing procedure. If the test result fails, “only one time” test identifies the error via error map function. This shows whether or not the failure is caused by a particular vector or the failure occurs as a result of the incorrect timing setup (e.g. clock event or sampling time). For evaluating the performance and reliability, the “endless” and “run until fails” options can also be used.

The test results are reported in several format, e.g. per pin and per vector. There are several debugging tools available in Verigy showing the error map and the timing diagram. These options report the failures with the details and possible root causes. Further information on test options can be found in [80]. Figure 5.6 and Figure 5.7 are the snapshots of the test results windows of the CMOS and CBL cores, respectively. The red window is the data manager consul, where the pre-test preparations steps are defined. The blue window represents the timing diagram showing the DATA_IN, KEY and corresponding output DATA_OUT. The yellow window is the test control, where the testing option is selected.

The result of the functional test is reported per pin and appears in the gray window. Test result on each output pin is identified by ‘P’ and ‘F’ indicating “Passed” or “Failed”, respectively. When the functionality of the cores is verified it is time to evaluate the leakage of side channel information. The procedure of security assessment is explained in the next section.



Functional Test
Real time compare

```
Path: /home/akhatib/DEVICES/ICFWTSBX_2/timing
Device: /home/akhatib/DEVICES/ICFWTSBX_2
Operator:
```

ICFWTSBX_2.pin ICFWTSBX_2.tim ICFWTSBX_2.lev COMPLETE_TEST_B

Timing Eqns	0
Waveform Set	1
Timing Set	1
Level Eqns	0
DPS Set	1
Level Set	1
Label	COMPLETE_TEST_BENC
Analog Set	1

Tester State CONNECTED

Tester Operation CONNECTED

```
F(DATA_OUT_8_CBL)                    = passed          P
F(DATA_OUT_7_CBL)                    = passed          P
F(DATA_OUT_6_CBL)                    = passed          P
F(DATA_OUT_5_CBL)                    = passed          P
F(DATA_OUT_4_CBL)                    = passed          P
F(DATA_OUT_3_CBL)                    = passed          P
F(DATA_OUT_2_CBL)                    = passed          P
F(DATA_OUT_1_CBL)                    = passed          P

F(DATA_OUT_8_CBL)                    = passed          P
F(DATA_OUT_7_CBL)                    = passed          P
F(DATA_OUT_6_CBL)                    = passed          P
F(DATA_OUT_5_CBL)                    = passed          P
F(DATA_OUT_4_CBL)                    = passed          P
F(DATA_OUT_3_CBL)                    = passed          P
F(DATA_OUT_2_CBL)                    = passed          P
F(DATA_OUT_1_CBL)                    = passed          P

F(DATA_OUT_8_CBL)                    = passed          P
F(DATA_OUT_7_CBL)                    = passed          P
F(DATA_OUT_6_CBL)                    = passed          P
F(DATA_OUT_5_CBL)                    = passed          P
F(DATA_OUT_4_CBL)                    = passed          P
F(DATA_OUT_3_CBL)                    = passed          P
F(DATA_OUT_2_CBL)                    = passed          P
F(DATA_OUT_1_CBL)                    = passed          P
```

Report Formatter .default.PROD

Cycles 10

Port: Tim/Pat @ 300.00000000ns

DevCyc pin DC_2

Vector 454

Time Delta [ns] 136200.00

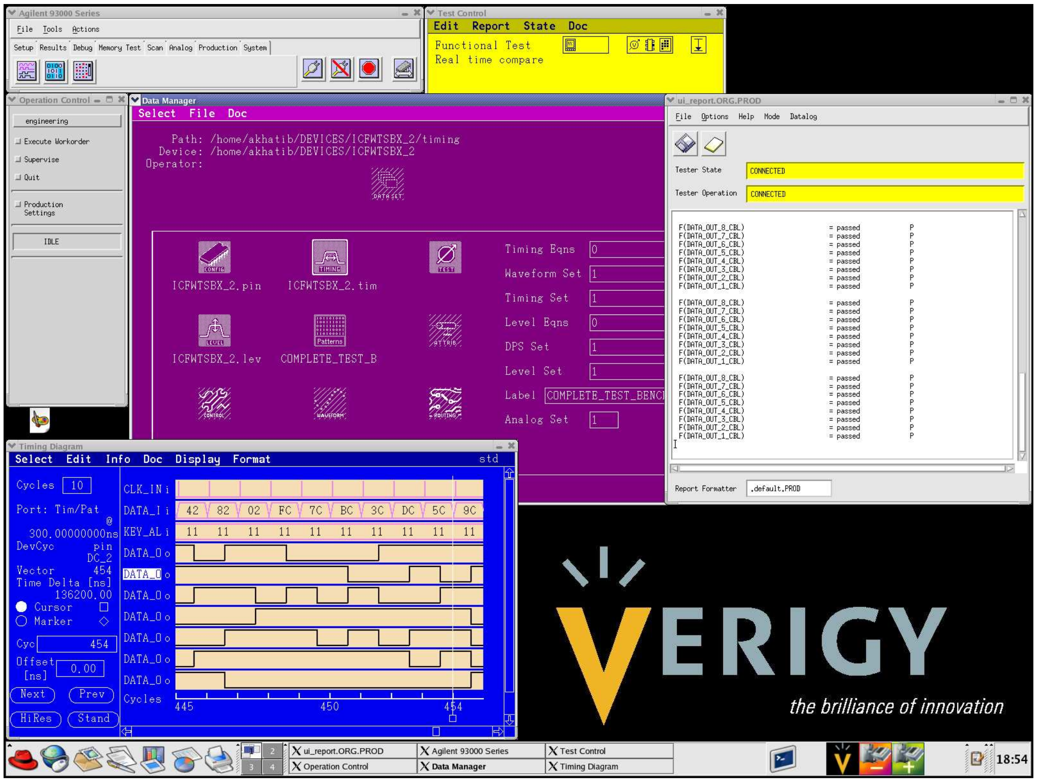
● Cursor □

○ Marker ◇

Cyc 454

Offset [ns] 0.00

Figure 5.6 Test report of the CMOS core



Functional Test
Real time compare

```
   Path: /home/akhatib/DEVICES/ICFWTSBX_2/timing
 Device: /home/akhatib/DEVICES/ICFWTSBX_2
Operator:
```

ICFWTSBX_2.pin ICFWTSBX_2.tim
ICFWTSBX_2.lev COMPLETE_TEST_B

Timing Eqns	0
Waveform Set	1
Timing Set	1
Level Eqns	0
DPS Set	1
Level Set	1
Label	COMPLETE_TEST_BENC
Analog Set	1

Tester State CONNECTED
Tester Operation CONNECTED

```
F(DATA_OUT_8_CBL)                = passed        P
F(DATA_OUT_7_CBL)                = passed        P
F(DATA_OUT_6_CBL)                = passed        P
F(DATA_OUT_5_CBL)                = passed        P
F(DATA_OUT_4_CBL)                = passed        P
F(DATA_OUT_3_CBL)                = passed        P
F(DATA_OUT_2_CBL)                = passed        P
F(DATA_OUT_1_CBL)                = passed        P

F(DATA_OUT_8_CBL)                = passed        P
F(DATA_OUT_7_CBL)                = passed        P
F(DATA_OUT_6_CBL)                = passed        P
F(DATA_OUT_5_CBL)                = passed        P
F(DATA_OUT_4_CBL)                = passed        P
F(DATA_OUT_3_CBL)                = passed        P
F(DATA_OUT_2_CBL)                = passed        P
F(DATA_OUT_1_CBL)                = passed        P

F(DATA_OUT_8_CBL)                = passed        P
F(DATA_OUT_7_CBL)                = passed        P
F(DATA_OUT_6_CBL)                = passed        P
F(DATA_OUT_5_CBL)                = passed        P
F(DATA_OUT_4_CBL)                = passed        P
F(DATA_OUT_3_CBL)                = passed        P
F(DATA_OUT_2_CBL)                = passed        P
F(DATA_OUT_1_CBL)                = passed        P
```

Report Formatter .default.PROD

Cycles 10
Port: Tim/Pat @ 300.00000000ns
DevCyc pin DC_2
Vector 454
Time Delta [ns] 136200.00
● Cursor □
○ Marker ◇
Cyc 454
Offset [ns] 0.00

```
CLK_IN i
DATA_I i   42  82  02  FC  7C  BC  3C  DC  5C  9C
KEY_AL i   11  11  11  11  11  11  11  11  11  11
DATA_O o
DATA_O o
DATA_O o
DATA_O o
DATA_O o
DATA_O o
DATA_O o
Cycles     445            450            454
```

Figure 5.7 Test report of the CBL core

5.4 Evaluation of Side Channel Information Leakage

In order to evaluate the resistance provided by the proposed countermeasure, a side channel attack using real power consumption is launched on the test chip. This requires that first the traces corresponding to the power consumption are extracted. The collected traces are then analyzed using the method previously described in Chapter 2. These two steps are referred to as data capturing and data analysis. Data capturing is described.

5.4.1 Data Capturing

Performing a side channel analysis, one should apply a series of input data to the test chip. The input includes 8-bit plaintext and key and the output is 8-bit ciphertext. The plaintext and corresponding ciphertext are changed every clock cycle whereas the key remains constant in an entire experiment. In each clock cycle one complete operation shown in Figure 5.8 is executed. The power traces are recorded in every clock cycle. The data capturing procedure has a direct influence on the accuracy and efficiency of the attack, thus, it is imperative to select the proper tools and employ appropriate measurement techniques.

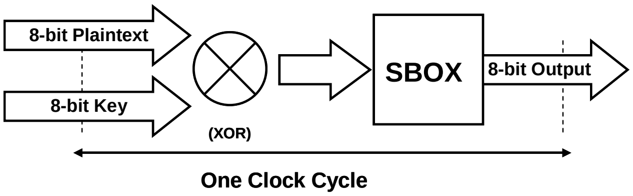


Figure 5.8 Operation of the test chip in one clock cycle

Data capturing tools: The measurement tools and experimental setup are shown in Figure 5.9. The current variation of supply in each core is measured in the form of voltage variation across a resistor which is between the Vdd pins of the test chip (VDD_CMOS/CBL and VDD_CMOS/CBL_REG in Figure 5.1) and the supply to the PCB. The measurements are performed separately on each core. The amount of supply voltage to the test chip is 1.8V. The measurement setup includes a 20Ω series resistor, high-frequency probes, and a digital sampling oscilloscope.

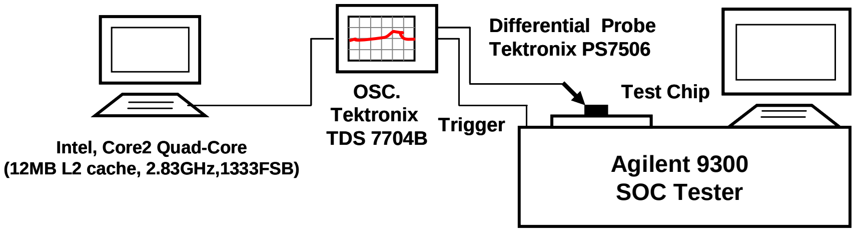


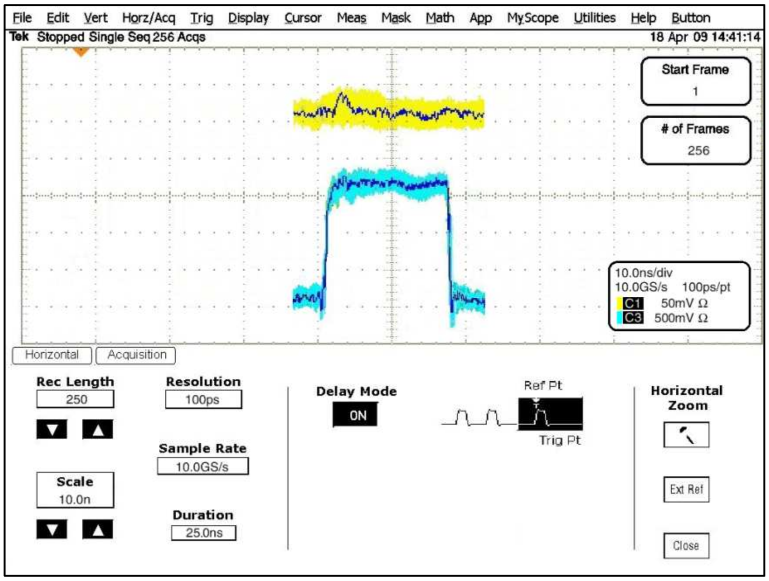
Figure 5.9 Experimental setup for launching a side channel attack

a) High-frequency Differential Probe: Voltage variation across the measurement resistor is sensed and captured by a PS7506 differential probe from Tektronix. The high performance solder tip with a long reach is used for the measurements. The specification of PS7506 differential probe is available in Appendix F.

b) Digital Oscilloscope: The traces obtained in this experiment are captured by the Tektronix TDS7704B oscilloscope [84] (Appendix G). The “FastFrame” feature is used for capturing the traces. “FastFrame” operates as follow.

Every time the clock triggers, a new plaintext enters into the chip and one frame is captured. Thus, for *n* clock cycles, the oscilloscope captures *n* frames. Each frame contains a waveform corresponding to the power variation of the core for one clock cycle. “Scale” and “Resolution” buttons of the scope are used for adjusting the time duration of the frame. Resolution is sacrificed if a long frame is captured. Therefore, in order to increase the accuracy only a portion of the waveform which carries the useful information is selected and captured. The number of data points in each frame is specified by the record length. Once the acquisition completes the captured frames are aligned and displayed in superposition format. Figure 5.10 shows the “FastFrame” screen. The light shadow (the top signal) shows the frames corresponding to the operation of 256 input plaintexts. The Ph_2 signal (the bottom signal) corresponding to 256 clock cycles is also captured. The last trace (256th) is shown in dark color superimposed on top of all the frames at the top and bottom of the scope plots. The acquired frames are recorded by the oscilloscope in the form of one data file (.dat). The data file is accompanied by a header file (hdr.) which can be used for splitting the frames to *n* traces during the data analysis.

Further details on setting up the “FastFrame” data acquisition can be found in [81]. The duration of each frame is 25ns. There are 250 sample points in each frame (10GS/s sampling rate). The oscilloscope specification and the setup of the data acquisition are available in Appendix G.


File Edit Vert Horz/Acq Trig Display Cursor Meas Mask Math App MyScope Utilities Help Button
Horizontal
Acquisition
Ref Pt
Trig Pt
Ext Ref
Close

Tek Stopped Single Seq 256 Acqs 18 Apr 09 14:41:14

Start Frame 1

# of Frames 256

10.0ns/div
10.0GS/s 100ps/pt
C1 50mV Ω
C3 500mV Ω

Rec Length 250

Resolution 100ps

Delay Mode ON

Horizontal Zoom

Sample Rate 10.0GS/s

Scale 10.0n

Duration 25.0ns

Figure 5.10 “FastFrame” feature for signal acquisition

Data capturing setup: In addition to employing measurement tools, some important issues with regards to setup for data capturing are explained.

a) Trigger setup: a trigger signal is needed for notifying the oscilloscope to begin sampling. In our experiment the clock signal to the chip (channel 11513) is used as a trigger to the oscilloscope. Dummy elements are added to delay the Ph_1 and Ph_2 signals (See Section 4.3.2) by approximately 2 ± 0.5ns. This intentional delay ensures that the sampling starts sufficiently earlier than the actual operation; hence, no data is lost during the sampling. Details on designing the delay element are given in Appendix H.

b) Signal acquisition: Noise is an influential factor causing error in data analysis. Thus, noise reduction should be considered during data capturing. Assigning separate supply pins to the cores, registers, output buffers and pads reduces the impact of noise. Furthermore, averaging is used as a practical technique for reducing the noise. A brief explanation is given on how averaging is performed.

A string of data consisting of *p* different plaintexts is formed. The string is applied *m* times resulting *p*m* frames (Figure 5.11). The frames are averaged so that *p* averaged traces correspond to the *p* plaintexts are acquired. In our experimental setup 50 times is seen to be sufficient for averaging (*m* = 50).

```
                  CLK_IN       DATA_IN             KEY            DATA_OUT

String              1      11111111(0xff)     10001000(0x88)     xxxxxxxxxxxxx
containing          1      11111110(0xfe)     10001000(0x88)     11110101(0xf5)
p plaintext         1      11111101(0xfd)     10001000(0x88)     00111000(0x38)
cycle 1             .            .                  .                  .
                    .            .                  .                  .
                    1      00000010(0x02)     10001000(0x88)     01111100(0x7c)
                    1      00000001(0x01)     10001000(0x88)     10001011(0x83)
                    1      00000000(0x00)     10001000(0x88)     10100111(0xa7)

                    .            .                  .                  .
                    .            .                  .                  .
                    .            .                  .                  .
                    .            .                  .                  .

String              1      11111111(0xff)     10001000(0x88)     xxxxxxxxxxxxx
containing          1      11111110(0xfe)     10001000(0x88)     11110101(0xf5)
p plaintext         1      11111101(0xfd)     10001000(0x88)     00111000(0x38)
cycle m             .            .                  .                  .
                    .            .                  .                  .
                    1      00000010(0x02)     10001000(0x88)     01111100(0x7c)
                    1      00000001(0x01)     10001000(0x88)     10001011(0x83)
                    1      00000000(0x00)     10001000(0x88)     10100111(0xa7)
```

Figure 5.11 Test vector for averaging *m* times

5.4.2 Data Analysis

In order to analyze the data contained in the captured frames attacker needs to know where in the power traces the useful information is hidden. It is also important that attacker relate the power consumption to some characteristics of the useful information. The attack points and the proposed technique for enhancing the quality of data capturing are reviewed.

a) Attack points: As discussed in Section 4.3.2 data dependency of power consumption in both logic gates and registers can be exploited for extracting the key related information. If attackers distinguish the power consumption of the SBOX and the registers, they will be able to launch two separate side channel attacks. Each attack requires analysis and extraction of the power consumed by the combinational networks (SBOX) and the power consumed at the moment of time the SBOX output is sampled by the registers. The structure of the test chip allows analyzing the side channel information leaked by the SBOX as well as the registers. During one clock cycle the operation of the SBOX and registers can be tracked by the Ph_2 and Ph_1 signals which are generated by the external clock. We use these two signals as references for setting up the capturing windows in “FastFrame” data acquisition.

Attack to the SBOX (combinational logic): When Ph_2 is ‘1’; the plaintext sampled by input registers (slave stage) enters into the chip. “AddRoundKey” and “SubByte” operations are executed. The power variation during this time depends on the intermediate values processed by the logic gates of the SBOX. Thus, the extracted traces from the V_{dd} pins (VDD_CMOS/CBL in Figure 5.1) during this time (Ph_2 is ‘1’) can be exploited for attacking the SBOX.

Attack to the registers (sequential logic): When Ph_1 is ‘1’ the results of the core operation are sampled by output registers (master stage). This means that the power trace extracted from the V_{dd} pins (VDD_CMOS/CBL_REG in Figure 5.1) for duration of this time (Ph_1 is ‘1’) contain useful information that can be used for attacking the registers. Extra pins are assigned to monitor the Ph_1 and Ph_2 signals. Figure 5.12 shows the sampling duration for attacking the core and the registers.

Figure 5.13 (a) is a snapshot from the oscilloscope screen which shows the operation of the CMOS SBOX for two clock cycles. The top yellow waveform (Channel 1) is a power trace of the CMOS SBOX. The light blue waveform is the Ph_2 signal (Channel 2). The dark blue waveform is the original clock signal. The power trace exhibits two significant peaks.

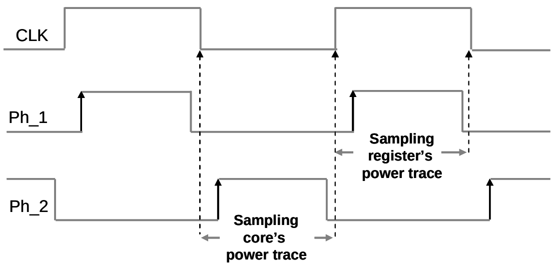
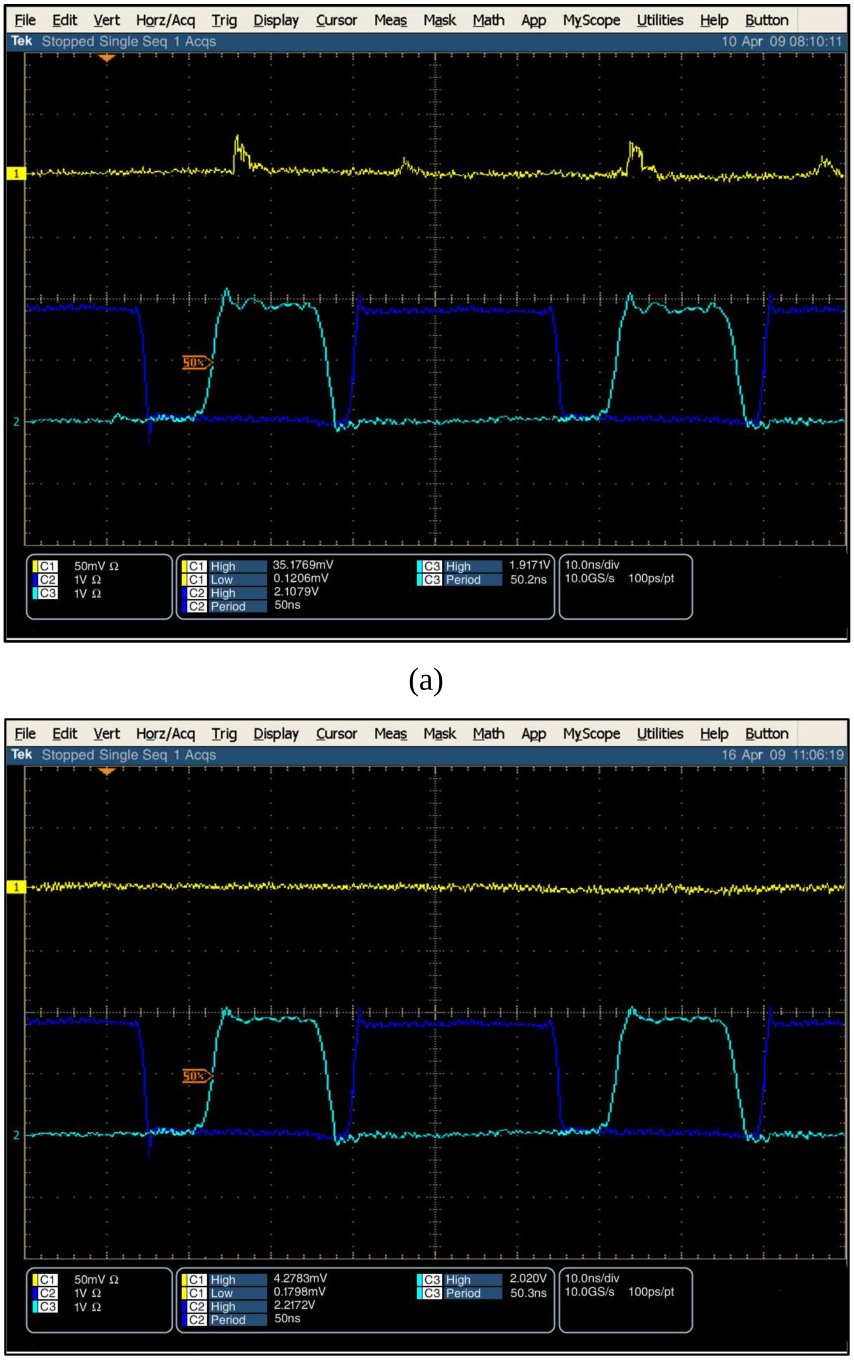


Figure 5.12 Sampling duration for attacking the SBOX and the registers

These peaks occur when the input registers (slave stage) apply a plaintext into the CMOS SBOX. As mentioned earlier the trace is captured by the "FastFrame" in each clock cycle only for duration of the time that the Ph_2 signal is '1' (the light blue waveform). The two minor peaks occur as a result of switching the Ph_1 signal (Ph_1 is absent in the snapshot). This portion of waveform is not sampled since the attack is being mounted on the SBOX. Figure 5.13 (b) shows the trace, the corresponding Ph_2 and the clock signals in the CBL SBOX. Similar setup is used for capturing the waveforms for attacking the CBL SBOX.

Figure 5.14 (a) is a snapshot showing the trace corresponding to operation of the output registers in the CMOS core. The power trace is the yellow waveform at the top. The Ph_1 signal is shown by light blue waveform. Two significant peaks occurring during Ph_1 is '1' should be captured as they correspond to sampling of the SBOX output. "FastFrame" window is setup to capture the waveform for this duration. The other two peaks correspond to switching of Ph_2. These peaks are out of the sampling duration. Figure 5.14 (b) shows the same procedure for attacking CBL register (CBETR).


File Edit Vert Horz/Acq Trig Display Cursor Meas Mask Math App MyScope Utilities Help Button
1
2
50%

Tek Stopped Single Seq 1 Acqs 10 Apr 09 08:10:11

C1 50mV Ω
C2 1V Ω
C3 1V Ω

C1 High 35.1769mV
C1 Low 0.1206mV
C2 High 2.1079V
C2 Period 50ns

C3 High 1.9171V
C3 Period 50.2ns

10.0ns/div
10.0GS/s 100ps/pt

(a)

Tek Stopped Single Seq 1 Acqs 16 Apr 09 11:06:19

C1 50mV Ω
C2 1V Ω
C3 1V Ω

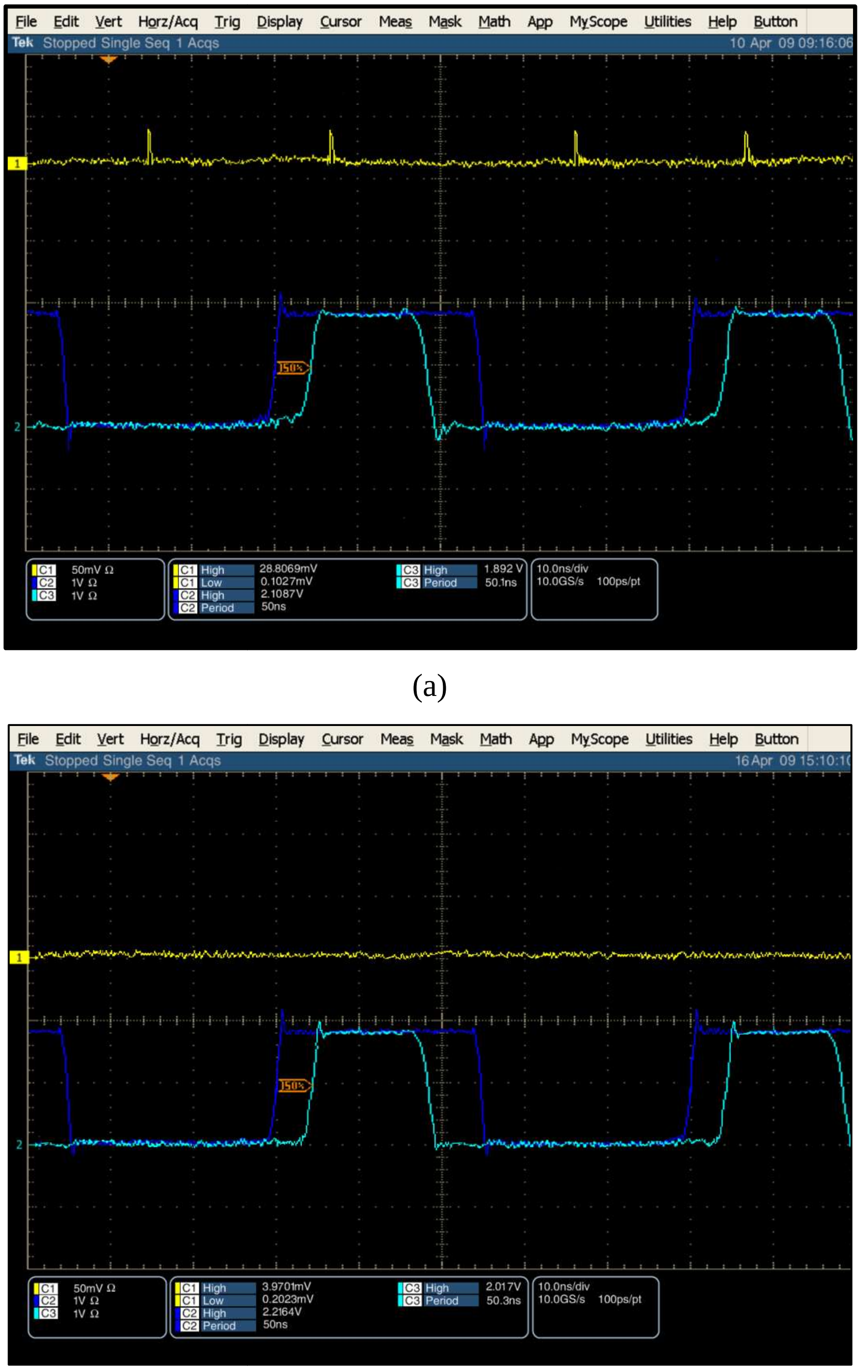
C1 High 4.2783mV
C1 Low 0.1798mV
C2 High 2.2172V
C2 Period 50ns

C3 High 2.020V
C3 Period 50.3ns

10.0ns/div
10.0GS/s 100ps/pt

(b)

Figure 5.13 Traces of the SBOX a) CMOS b) CBL



```
Tek Stopped Single Seq 1 Acqs                                   10 Apr 09 09:16:06

C1  50mV Ω     C1 High    28.8069mV     C3 High    1.892 V    10.0ns/div
C2  1V Ω       C1 Low     0.1027mV      C3 Period  50.1ns     10.0GS/s   100ps/pt
C3  1V Ω       C2 High    2.1087V
               C2 Period  50ns
```

(a)

```
Tek Stopped Single Seq 1 Acqs                                   16 Apr 09 15:10:1

C1  50mV Ω     C1 High    3.9701mV      C3 High    2.017V     10.0ns/div
C2  1V Ω       C1 Low     0.2023mV      C3 Period  50.3ns     10.0GS/s   100ps/pt
C3  1V Ω       C2 High    2.2164V
               C2 Period  50ns
```

(b)

Figure 5.14 Traces of the registers a) CMOS b) CBL

b) Enhanced data capturing method: According to the evaluation strategy discussed earlier in Section 2.4 the Hamming Distance (HD) model is chosen in this thesis for launching an attack. Using the HD model, the attacker should be able to compute the hamming weight differences between two consecutive SBOX outputs. This is a reasonable assumption in known and chosen plaintext attacks [21]. In order to obtain more precise values of power variation corresponding to the HD model an enhanced data capturing method using reference vector is introduced. The concept of a reference vector is explained.

If a fixed plaintext is inserted between every of successive plaintexts, in every other clock cycle the power consumption will become almost equal. This provides a more precise reference point for power consumption. The following discussion explains how a reference vector can be chosen.

A fixed plaintext can be chosen as a reference vector based on the SBOX property. In composite field implementation, SBOX will consume significantly less power for input “0x00” than all other input values [27]. This is due to the fact that input “0x00” causes multiplications by zero. Since the multipliers are the major sources of power consumption in SBOX, the total power consumption essentially is reduced. In order to provide “0x00” at the SBOX input, a plaintext equals the key should be applied to the core. The XOR result of plaintext and key applies “0x00” at the input, therefore, SBOX consumes minimum power. Since the output of SBOX corresponding to “0x00” is “0x63”, the hamming distance of the switching between successive traces is equal to the hamming weight of the XOR of “0x63” and upcoming SBOX output.

Figure 5.15 shows the modified patterns which hold the minimum power consumption reference vector “0x88”. The reference vector is underlined. In a real scenario attackers may not have sufficient knowledge to apply the reference vectors. However, playing the designer role we are able to assign the reference vector by knowing the key.

The results in Section 5.5 show that the CMOS core is successfully attacked without applying the reference vectors. However, significant increases in correlation coefficients are seen as a result of applying minimum power consumption input signal as a reference vector.

```
              CLK_IN         DATA_IN              KEY          DATA_OUT

               1        10001000(0x88)     10001000(0x88)    xxxxxxxxxxxx
               1        11111111(0xff)     10001000(0x88)    01100011(0x63)
 n cycles      1        10001000(0x88)     10001000(0x88)    11110101(0xf5)
               1        11111110(0xfe)     10001000(0x88)    01100011(0x63)
               1        10001000(0x88)     10001000(0x88)    00111000(0x38)
               .              .                  .                  .
               .              .                  .                  .
               .              .                  .                  .
```

Figure 5.15 Vector pattern composed of plaintext with reference “0x88”

c) Correlation-based attack: The attack procedure is now reviewed. Test structure carries out the operation of “AddRoundKey” and “SubByte”. The intermediate result which is a function of the plaintext (*d*) and key (*K*) is defined as:

$$f(d, k) = (SBOX\ (d_i\ XOR\ K)) \qquad (5.1)$$

The maximum value of the averaged traces is used for representing the power variation of SBOX and registers corresponding to each plaintext [22]. These values form a *N x 1* matrix which is referred to as the measurement matrix, where *N* is the number of averaged traces. The values representing the power consumption of the reference vectors are excluded from the measurements matrix since they will be removed from the traces.

In order to calculate the hypothetical model of the intermediate values, the HD of two successive values of the SBOX output is exploited. The HD model maps the transitions that occur at the output as a result of operation $f\ (d_i, k)$. Matrix V of size *N* x *256* is generated which is referred to as estimation matrix. Matrix V represents HD of the SBOX outputs for *N* plaintexts. This model is generated for all the possible keys (2^8). The final step is to compare the estimation (V) and the measurements (T) matrixes. The correlation coefficient shown by Equation 2.14 is calculated between the elements of the V and T matrixes. The correct key guess is the one that results in the highest correlation coefficient between the vector of the model and the vector of measurements. Figure 5.16 illustrates the execution of the attacks including the data capturing, processing and analyzing.

```
Inputs
Plaintext ($PT_i$), i = (1, …, N)
               $PT_i = (pt_{i8}, pt_{i7}, … pt_{i1})$
KEY ($K_j$), j $\in$(0, …, 255)
               $K_j = (k_{j8}, k_{j7}, … k_{j1})$
Reference Vector (RV), for given $K_j$;
               $RV_j = (rv_8, rv_7, …, rv_1)$
DATA_IN String ($D_n$)for given $K_j$,
               $D_n = (RV_j, PT_1, RV_j, ..., RV_j, PT_n)$; n = (1, …, N)
Outputs
Power of SBOX corresponding to $K_j$ and $PT_i$; $P_{SBOXij}$
Power of Registers corresponding to $K_j$ and $PT_i$; $P_{Reg.ij}$

Apply $D_n$ for a given $K_j$ using $RV_j$;

Acquire traces corresponding for $PT_i$ & given $K_j$; $P_{SBOXij}$ & $P_{Reg.ij}$
    Run for M times, (M times of averaging)
    Average pointwise
    Remove traces of of $RV_j$, $PT_{RVj}$
    Extract traces corresponding to all $PT_i$ & given $K_j$; $P_{SBOXij}$ & $P_{Reg.ij}$

Find $P_{SBOXij}$ = Max.of $P_{SBOXij}$ for duration of [Ph_2];
Find $P_{REG.ij}$ = Max. of $P_{Reg.ij}$ for duration of [Ph_1];

Create Power model $P_{Mi[0, 255]}$ = HamDis (SBOX($PT_i$ XOR $K_{[0, 255]}$)),
        for all i = (1, …, n)
Compute Correlation ($P_{Mi[0, 255]}$, $P_{SBOXij}$) for all i; $Func._{cost\ SBOXi}$
Compute Correlation ($P_{Mi[0, 255]}$, $P_{REG.ij}$) for all i; $Func._{cost\ REG.i}$
    If $Func._{Cost\ SBOXi}$ is max. for given $K_j$,
        then $K_j$ to be verified as correct key,
        if “not” increase N and go to Acquire
    If $Func._{Cost\ REG.i}$ is max. for given $K_j$;
        then $K_j$ to be verified as correct key,
        if “not” increase N and go to Acquire
```

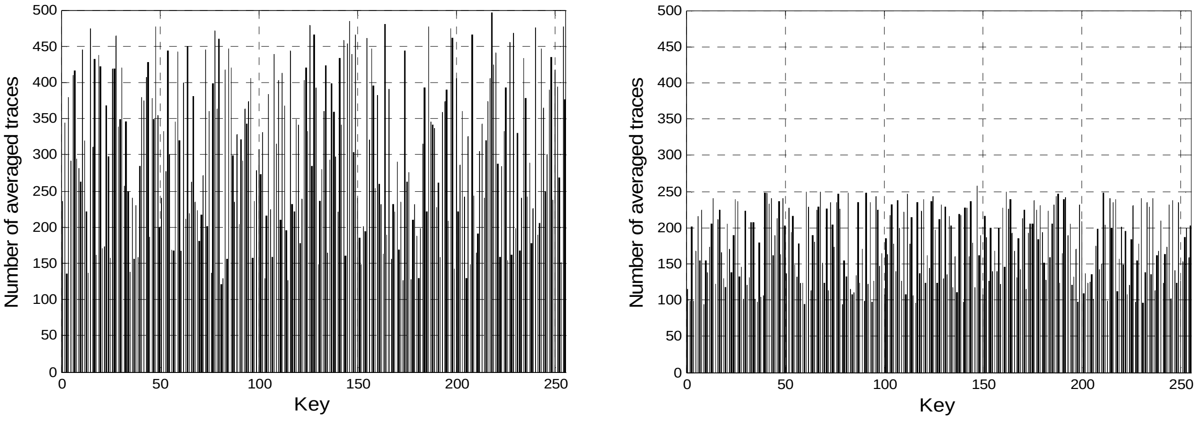
Figure 5.16 The procedure of correlation-based attack

5.5 Side Channel Leakage Assessment: Result Review

The resistance against side channel information leakage is quantified with the number of averaged traces (corresponding to number of the plaintext) to disclose the key. This measure is defined as “the cross-over point between the correlation coefficient of the correct key and the maximum correlation coefficient of all the wrong keys guesses [52]. For further clarification the term “averaged trace” is defined as: the minimum number of plaintexts that reveals the correct key, e.g. for a key which requires 100 plaintexts to be revealed, 100 averaged traces are used with 50 times averaging over a fixed plaintext, thus, in total 5000 frames need to be captured. The results of the attack on the CMOS core will be presented next.

5.5.1 Result Review: CMOS Core

The correlation-based side channel attack on CMOS core was successful at two points: the output of the SBOX and the output of the registers. Figure 5.17 (a) and (b) depict the number of the averaged traces required for revealing 256 key from the SBOX and registers, respectively.

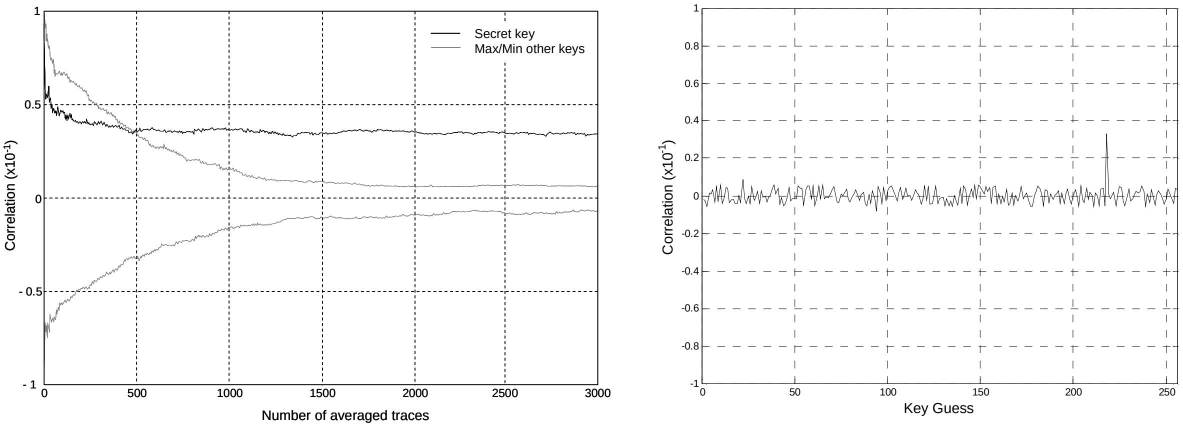


(a)

(b)

Figure 5.17 Number of averaged traces for all the keys extracted from the CMOS a) SBOX b) registers

In our experimental setup maximum 497 averaged traces are required to reveal key “0xda” from the CMOS SBOX. In attacking the registers, maximum 258 averaged traces are needed for extracting key “0x93”. The result of the attack on CMOS block is shown in Figure 5.18 (a). The point where the black line crosses the gray line depicts the number of averaged traces required for revealing the key “0xda”. The correlation values for all keys are also shown in Figure 5.18 (b).



(a)

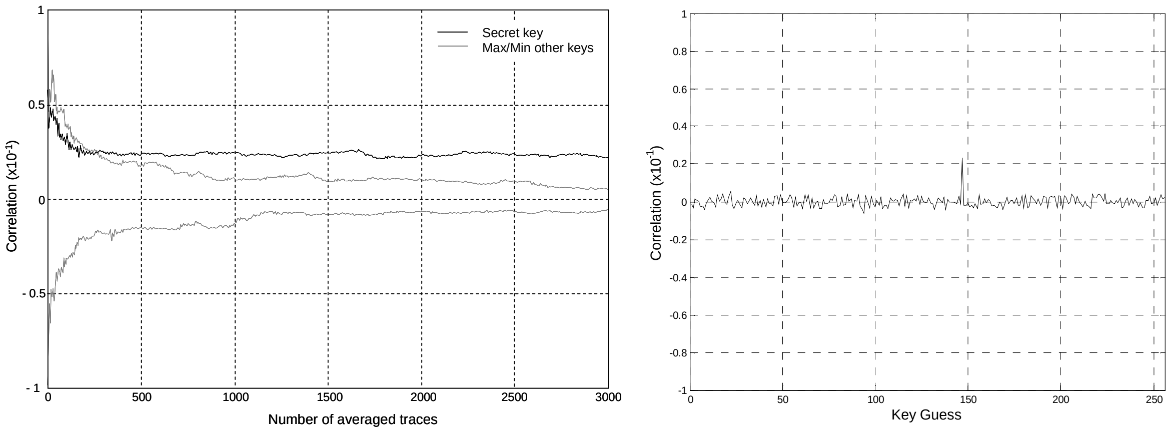
(b)

Figure 5.18 Correlation a) vs number of averaged traces b) for all key in the CMOS SBOX (key “0xda”)

Similar results are presented in Figure 5.19 (a) and (b) for the CMOS registers. The result shows that key “0x93” is revealed by averaged 258 traces.

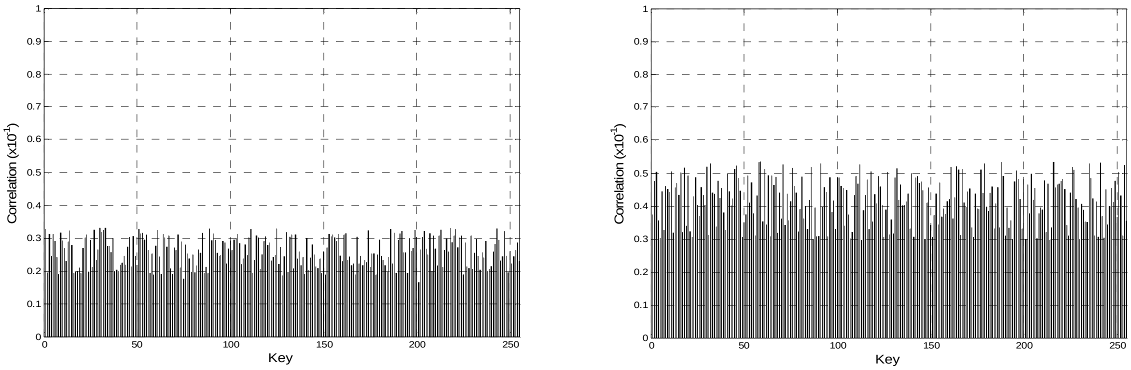
The correlation coefficients corresponding to the correct keys are ranged from 0.0175 to 0.0336. This range changes from 0.0299 to 0.0535 when the reference vector is applied. Although the increasing scaling is not same for all the keys, correlation coefficients are seen to increase at least by 1.5 times. This shows the effectiveness of the proposed enhanced data capturing method. The values of correlation coefficient are shown in Figure 5.20 (a) and (b).

Nearly similar trend is observed in correlation corresponding to the correct key from the CMOS registers.



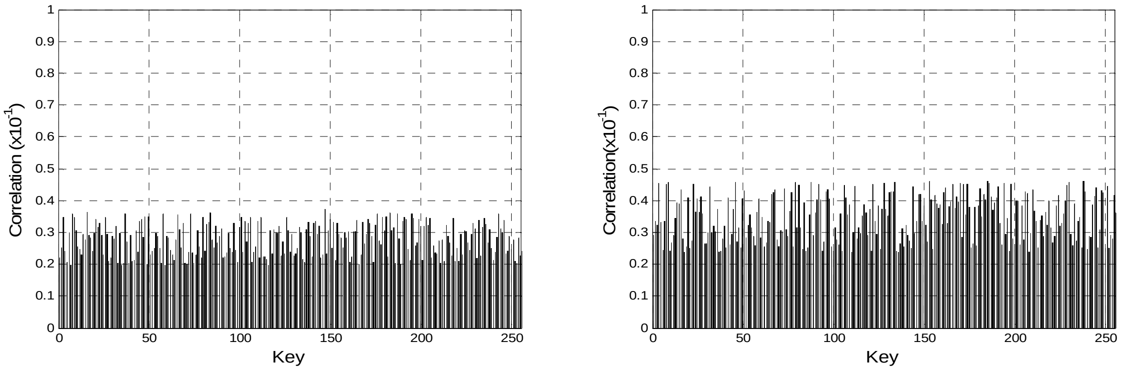
(a) (b)

Figure 5.19 Correlation a) vs number of averaged traces b) for all key in the CMOS registers (key "0x93")



(a) (b)

Figure 5.20 Correlations for all the keys extracted from the CMOS SBOX a) without b) with applying reference vector



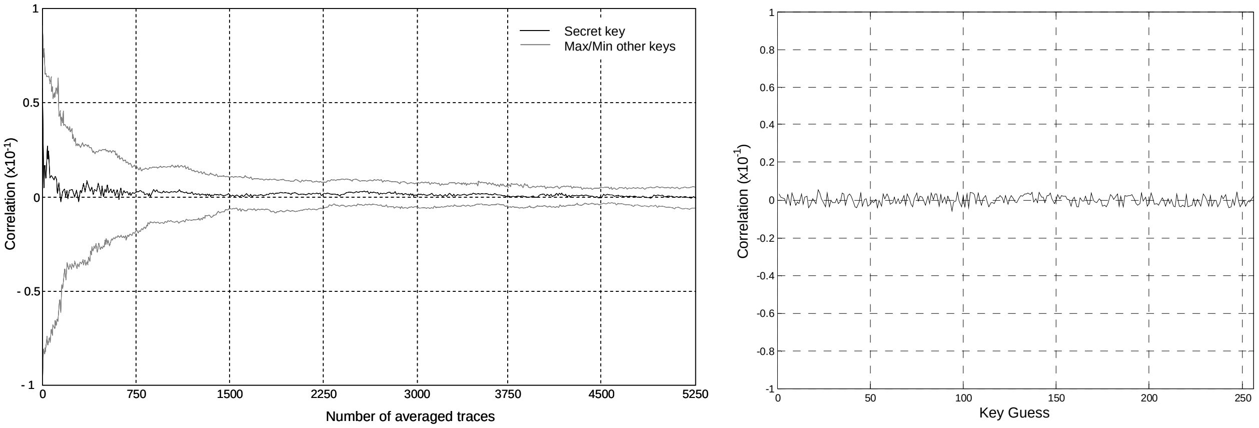
(a) (b)

Figure 5.21 Correlations for all the keys extracted from the CMOS registers a) without b) with applying reference vector

Figure 5.21 (a) and (b) show the correlation for key extracted from the CMOS registers. The increase in correlation coefficient in registers is about 1.35 times. The large resolution in correlation values caused by the reference vector facilitates distinguishing the correct key with more certainty. Unlike the improved correlation coefficients, only slight changes are observed in the number of averaged traces for key disclosure.

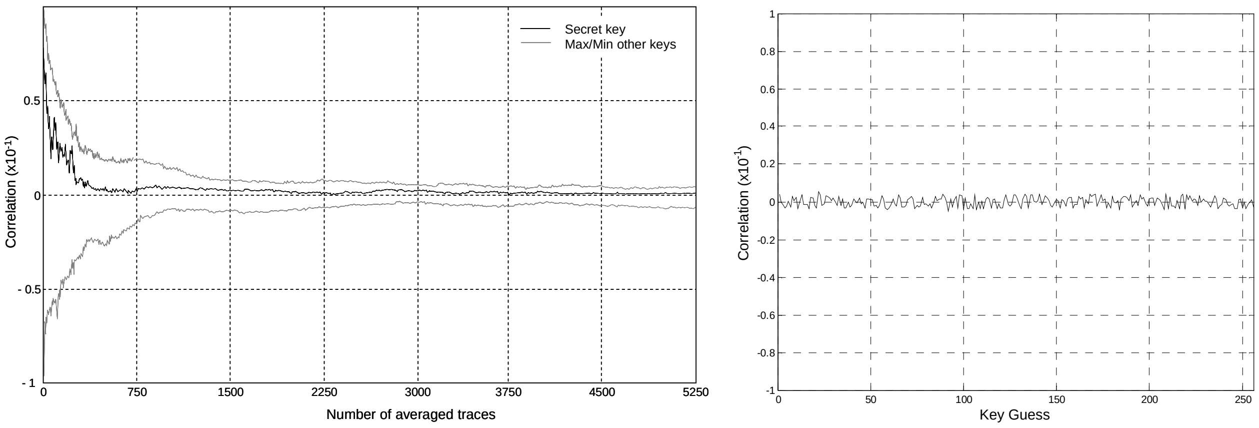
5.5.2 Result Review: CBL Core

The attack is launched on CBL core using the reference vector. Correlation coefficients are reduced drastically and no significant correlation is observed for any of the correct key in the CBL SBOX and the registers. The number of averaged traces is increased from 500 to 5000. There is still no significant correlation seen leading to the correct key. Figure 5.22 (a) plots the correlation for the correct key in the CBL SBOX. The correlation values for all the possible keys are also shown in Figure 5.22 (b). No significant result is obtained from the attack on CBL register (CBETR). The results of attack on CBL registers are shown in Figure 5.23 (a) and (b).



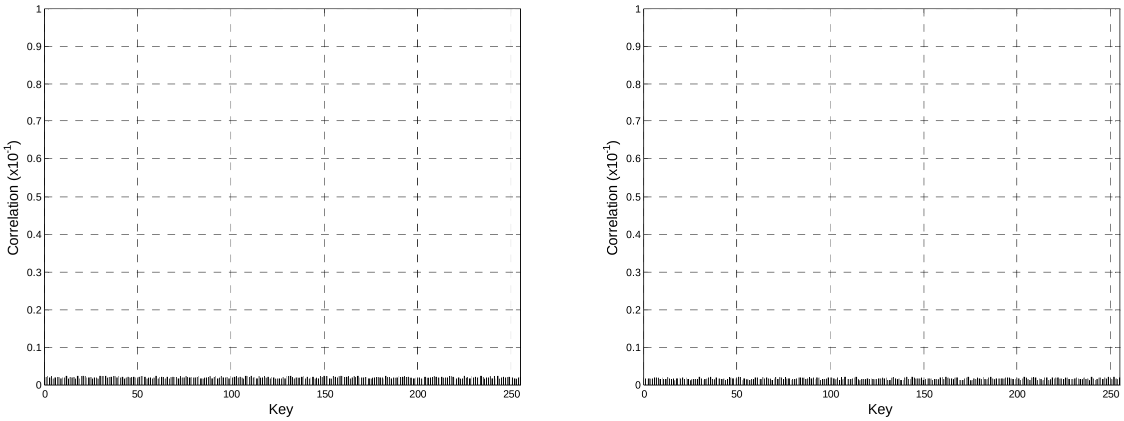
(a) (b)

Figure 5.22 Correlation a) vs number of averaged traces b) for all key in the CBL SBOX (key "0xda")



(a) (b)

Figure 5.23 Correlation a) vs number of averaged traces b) for all key in the CBL registers (key "0x93")



(a)

(b)

Figure 5.24 Correlations for all the keys in CBL a) SBOX b) register

The correlation coefficients for all the keys are acquired for the CBL SBOX and the registers. The results in Figure 5.24 (a) and (b) illustrate significant reduction in amplitude of correlation coefficients. Nonetheless, none of the coefficients led to significant differences between the correct and incorrect keys. The result of the attack on the CBL core demonstrates that the proposed method develops as a resistance against power attacks. Table 5.1 summarizes the implementation results.

Table 5.1 Comparison results between CMOS and CBL core

Characteristic	**Standard CMOS**		**Proposed design**	
Area (mm^2)	0.0987		0.0861	
Maximum Frequency[*] (MHz)	150		120	
Power Consumption (mW)	0.156[§]		72.56[†]	
No. of averaged traces for keys extraction ‡	SBOX	Register	SBOX	Register
Min	120	90	-	-
Mean	310	170	-	-
Max	500	258	-	-

[**] Duty factor of clock = 50% (1.8V)

[§] Dynamic power consumption (at 1.8V, 20MHz)

[†] Static power consumption

‡ The number of averaged traces to disclose the key varies for each correct key. This is also observed that same key may be obtained with slightly different number of traces if the experiment is repeated.

Figure 5.25 shows the area estimation of the CMOS and CBL cores on the actual die. It is seen that area consumption of the protected core is 15% less than that of the unprotected core. A comparative analysis with the previous side channel resistant logics will be presented.

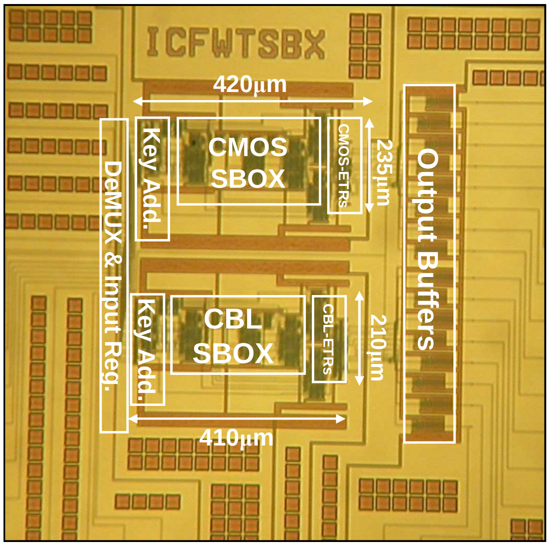


Figure 5.25 The photograph of die shows the CBL and CMOS core

5.5.3 Comparison: The Proposed Countermeasure and its Counterparts

The proposed approach in this thesis is compared with other logic level countermeasures. Table 5.2 shows a list of side channel resistant logic styles: SABL [52], WDDL [53], MCML [59], DyCML [60], and RSL [86]. The power and area are reported based on different hardware architectures, simulation conditions and in two cases in different implementation technology. An assessment based on these figures may not project a proper quantitative scheme for highlighting the advantages and drawbacks of the different logics. Therefore, in addition to considering the data in Table 5.2 a qualitative analysis is also delivered.

Table 5.2 Characteristics of the side channel resistant logics

Logic*	**Design****	**Tech** (nm)	**Power** (mW)***	**Area** (mm^2)	**Scheme**‡
SABL [52]	DES	180	2.81	-	DDL
WDDL [53]	AES†	180	200	2.45	COM.
MCML [59]	Kasumi	180	20.7	-	DL
DyCML [60]	Khazad	130	0.027	-	DDL
RSL [86]	AES	130	-	30K§	RSL
Proposed	AES	180	72.56	0.0861	SSL

* Results reported for SABL, MCML and DyCML are simulation-based whereas the result of WDDL, RSL and the proposed countermeasure are based on the fabrication results.
** Subset of different encryption algorithms are implemented, [52]: 92 XOR + 86 NAND, [59] MCML: 77 XOR + 105 NAND, and [60] (DyCML 754 transistors).
*** Power in dynamic architectures is reported at different frequencies.
‡ DDL, COM., DL and SSL are referred to as Dynamic Differential Logic, Combinational, Dynamic Logic and Static Single-ended Logic, respectively.
† An entire AES with no detail of an individual SBOX is reported in [53].
§ Area in [86] is reported by gate counts. The results in [86] show that RSL requires two times more gates than standard CMOS.

Performance: The side channel resistant logics listed in Table 5.2 are dynamic (except for MCML and RSL) whereas the proposed countermeasure is static; as a result, the performance comparison mainly highlights the characteristics between the dynamic and static logics. The operation of the dynamic logic gates is executed in precharge and evaluation cycles which are controlled by the clock signal. The maximum clock rates in dynamic-based structures are typically halved due to their two-phase operation. This is a common property among the dynamic logics particularly those are proposed for side channel security e.g. SABL, WDDL, and DyCML.

MCML, RSL and CBL are static logics which operate on one-phase cycle. The maximum operation frequency in static-based architecture is determined by delay of the critical path. Thus, the circuit structure of the gates should be compared for performance evaluation. MCML is low swing logic with average 20% less delay compared to CBL [59]. RSL gates experience approximately 50% performance degradation comparing to the standard CMOS gates [86]. Analysis in Section 4.3.2 shows the performance is also degraded by 10%-15% in CBL gates compared to standard CMOS. It can been seen that performance penalty is expected between 20% and 50%, once standard CMOS logic is replaced by static-based side channel resistant logics. The performance overhead can be reduced by using MCML; however, noise margin is decreased as a result of low swing outputs.

Power Consumption: Dynamic logics such as SABL, DyCML, and RSL do not consume static power. The major source of the power consumption in these logics is switching activity which is determined by the clock frequency. In order to lower the power consumption in dynamic logic, a common approach is to reduce the swing at the output of the logic gates [17]. DyCML [60] is an example of side channel resistant logic with low swing output. DyCML consumes significantly low power; however, in practice a level-shifter should be added to the output to preserve the logic level in an acceptable swing for the cascaded blocks. The impact of adding a buffer at the output of low swing logic such as DyCML needs to be investigated from both the power consumption and the side channel perspectives. This issue has not been addressed in [60].

The proposed countermeasure operates with no clock signal at the gate level. Despite elimination of the clocking networks, the total power consumption of CBL is significantly high. The characteristic featuring the suitability of CBL for side channel security also limits the use of this logic family for power-constrained applications. A potential approach for reducing the power consumption in CBL was introduced in Section 4.3.2. An alternative trade-off scheme was presented for less power consumption. The functionality of CBL under low supply voltage is examined. The CBL SBOX operates with supply voltage as low as 1.2V at the cost of 40% performance degradation.

Area: Since different hardware architectures are compared in Table 5.2, the area comparison based on those figures may not be fair. For more realistic area evaluation SABL, DyCML, MCML and RSL gates are used in design of a same SBOX architecture. Table 5.3 presents the total transistor counts and total area consumption by the transistors. The results are normalized based on the area consumption of CBL gates. The area of the clocking networks and routing are excluded in this comparison and minimum transistor size is used in the design of the gates. The results in Table 5.3 show the superiority of CBL in both transistor counts and area consumption. MCML also consumes significantly less area compared to DyCML and SABL. Between dynamic logic DyCML uses less area than SABL. RSL also involves with several times area consuming.

Table 5.3 Area comparisons of side channel resistant logics in 180nm CMOS technology (normalized based on CBL)

	SABL		**DyCML**		**MCML**		**RSL**	
	No. of Trans.*	Area	No. of Trans.	Area	No. of Trans.	Area	No. of Trans.	Area
SBOX	1.9	6.74	1.8	6.30	1.2	4.26	4.3	15.2

* Transistors

Information Leakage: We use the number of averaged traces for quantifying the side channel resistant (Sections 5.5.1 and 5.5.2). The comparison presented in those sections is believed to be fair since the empirical results are attained from the same architecture using similar data capturing and analysis methods. However, not all the logics in Table 5.2 use the same measure for assessing side channel resistance. The effectiveness of WDDL is examined in complete AES implementation and results are presented by one million measurements. The side channel resistant of RSL is also quantified by number of measurements for key disclosure. The results in [86] show that significant resistance is achieved by RSL. Over one million measurements are performed in [86] for attacking 16-bit key. In comparison, we attack on an 8-bit key on a similar architecture which is fabricated in 180nm CMOS technology. In total approximately 260,000 (maximum number of traces x number of averaging) measurements are performed. The results of the attacks can be interpreted for comparison between CBL and RSL. The attack in [86] is launched on a longer key in 130nm CMOS process. This attack without doubt requires more effort due to the reduced signal-to-noise ratio of the power traces which is caused by the technology scaling.

Complexity-driven Constraint: Although, the design complexity is not always simple to measure, a brief discussion is presented to qualitatively compare the side channel resistant logics. As described in Section 3.4.2 if differential logic style is used for protection against side channel information leakage, balancing the loads at the differential outputs is a condition that must be met. A complex layout methodology is required in order to satisfy this condition. Balancing the load adds more complexity to design and implementation of SABL and DyCML. The proposed countermeasure is a single-ended logic, thus, it is free from such constraint. In same context, MCML and CBL can also be compared. Unlike MCML requiring V_{bias} for operation, CBL does not require any extra circuitry for supplying the current at the cell level, thus, CBL can be designed and implemented with less complexity. Unlike the CBL which is custom-designed logic, RSL can be implemented using standard CMOS cells. From a design effort perspective RSL is the most efficient approach.

5.6 Summary and Conclusion

Implementation of the proposed countermeasure is reviewed in this chapter. The test chip architecture, including protected and unprotected cores, was detailed in addition to discussion of pre-test hardware and software preparations. The experimental setup and measurement tools were also introduced. The testing procedures were described in two parts: functionality and side channel assessment. Utilizing the tester features, both cores were verified to be 100% functional. Power consumptions of the CMOS SBOX and registers were exploited for mounting correlation-based attacks on the unprotected core. The vulnerability of the CMOS core was quantified by the number of averaged traces for key disclosure. It was seen that CMOS SBOX and registers revealed the keys within a maximum of 497 and 258 averaged traces, respectively. Increasing the number of averaged traces to over 5000 still did not result in key revelation from the protected core. Comparative analysis was presented to rank the side channel resistant logics based on: performance, power, area, information leakage and complexity. The superiority of the proposed countermeasure in transistor count and total area was evident. Therefore, a cost effective approach for side channel security of area-constrained crypto core was provided. The condition of balancing the load at the differential outputs was not required by the proposed countermeasure, thus, side channel resistance is achieved with less complexity compared to previous approaches.

Chapter 6

Side Channel Information and Technology

6.1 Introduction

In order to fulfill standards such as FIPS 140-3 [87] the security issues associated with deployment of advanced technology must be investigated. Side channel attack is known as one of the most severe breaches threatening the security of cryptosystems. Thus, it becomes crucial to explore the impact of technology on side channel vulnerability of cryptosystems.

This chapter reviews the trends of power consumption in CMOS technology from a side channel perspective. The simulation-based results are used to quantify the leakage of information via leakage power consumption. The trend of side channel threat posed by leakage power consumption is drawn over several technology nodes. The role of different leakage generating mechanisms is considered in this investigation. The effectiveness of leakage control technique is examined for side channel security.

6.2 Power Consumption: Technology Trends

Over the past 25 years, the transistor minimum features size has scaled down from 6μm to the present sub-90nm. As indicated in Figure 6.1 the number of transistors per chip has been quadrupling every three to four years, while the speed of microprocessors has been more than doubling, increasing 2MHz for the Intel 8080 in the mid-1970's to well over 10GHz for present leading-edge chips [88]. Meanwhile the supply voltage *(V_{dd})* must also continue to scale down at the historic rate of 30% per technology generation in order to minimize power dissipation and power delivery costs in future high-performance microprocessor designs. To maintain the speed enhancement per technology generation, the transistor threshold voltage *(V_{th})* and the gate oxide thickness *(t_{ox})* of the transistor must be scaled with the supply voltage. However, reducing V_{th} causes transistor leakage current ($I_{leakage}$) to increase exponentially (Equation 2.3). The leakage mechanisms were already discussed in detail in Section 2.2.1. A brief review of the root causes of the dominant leakage mechanisms is given.

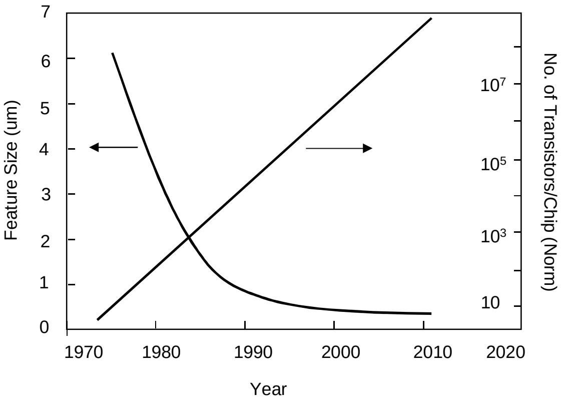


Figure 6.1 Historical trend of LSI's [89]

6.2.1 Leakage Power Elements

Contribution of the major leakage mechanisms in NMOS transistor with the technology advances is shown in Figure 6.2. Subthreshold, gate and Band-to-Band Tunneling (BTBT) are identified as the dominant leakage power mechanisms and they increase significantly as feature size decreases.

As discussed V_{th} reduction (scaling) results in exponential increase in subthreshold current due to the Short Channel effects (SCEs) such as Drain-Induced Barrier Lowering (DIBL). To control SCE and to increase the transistor drive strength, oxide thickness must also become thinner in each technology. Aggressive scaling of t_{ox} results in a high direct-tunneling current through the transistor's gate insulator.

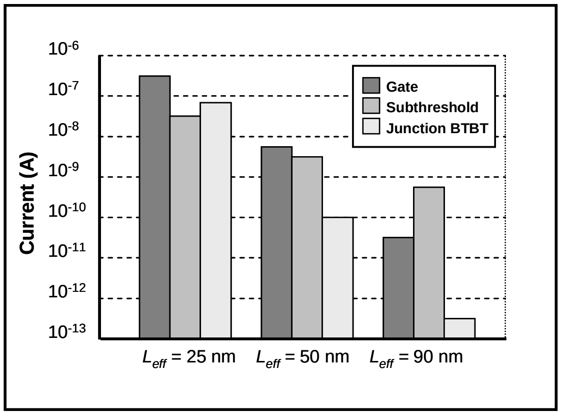


Figure 6.2 Major leakage mechanisms at different technology nodes in an NMOS transistor [18]

The scaled transistors require higher substrate doping densities to reduce the width of the depletion region for the source- and drain-substrate junctions. A narrower depletion region width helps to control the short-channel effect. The high doping density near the source- and drain-substrate junctions causes a significantly large BTBT current through these junctions under high reveres bias. It is evident that increasing the total leakage power consumption is due to the increase of all these three leakage mechanisms [18].

6.3 Leakage Power: An Emerging Side Channel

As the technology scales down the leakage current becomes a more effective element of static power consumption [89]. The data dependency characteristic of leakage power is also increased [90]. This increasing trend has drawn attention to leakage power which will likely offer a new power related side channel threat. By far only a handful of researchers investigated the side channel role of leakage power consumption [91][92].

In order to provide insight into security of next generation of cryptosystems we present a series of simulation-based analysis to:

- quantify the vulnerability of nanoscale cryptosystems to leakage of information via leakage power consumption

- highlight the role of leakage generation mechanisms in leakage of information
- examine the effectiveness of the circuit-based leakage power reduction technique for side channel security.

The detail of the simulation setup and review of the results are presented next.

6.3.1 Side Channel Trends of Leakage Power Consumption

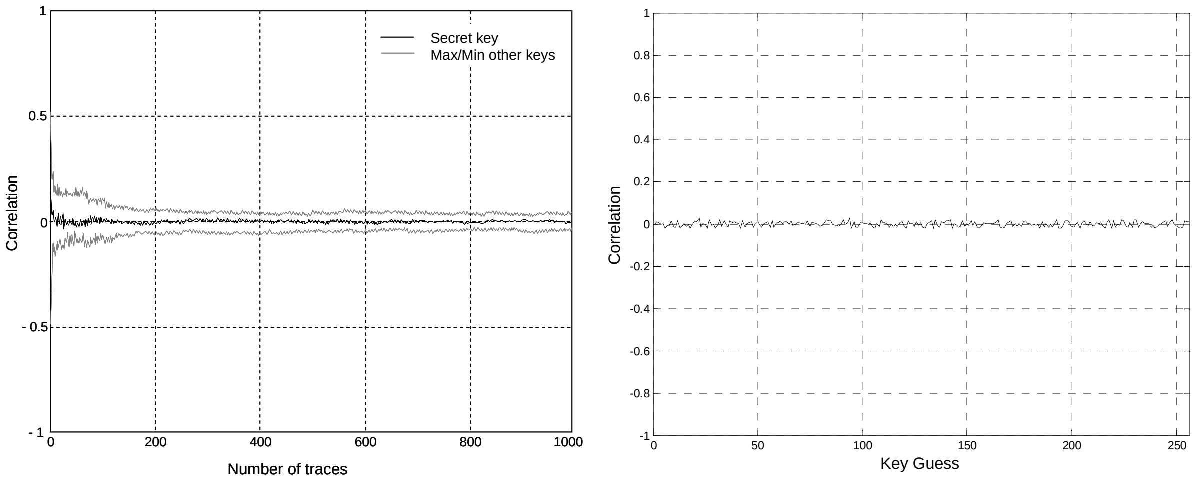
The test bench previously introduced in Section 5.2.1 is used for evaluating the side channel information leakage via leakage power. The test bench is designed using gates from standard CMOS libraries in 180, 90 and 45nm technology nodes[5]. The simulation-based side channel attack exploiting leakage power consumption is launched. The number of traces to reveal the key from the test bench is used for comparison. In order to extract the leakage power corresponding to data processed by the core, a plaintext is applied to the test bench and the supply current is sampled after an intentional delay. The intentional delay is slightly longer than the propagation delay of the core. This means that switching is complete and data at the output is stable. By maintaining the clock signal "on" and holding same data at the input no more switching occurs; however, the power is still consumed by the core. The current drawn from the supply after switching represents the leakage power corresponding to the present input vector. DC value of the supply current after switching represents the leakage power consumption. The simulation environment in HSPICE is setup to record the value of the leakage current for every plaintext. The procedure described in Section 2.3.3 is followed for mounting a side channel attack on the SBOX output. The correlation analysis is performed. By changing the plaintext (*N* times) and maintaining the key a matrix of $N \times 1$ is generated containing the values of the leakage power for N plaintexts and the key. This matrix plays a role of the measurement in a real side channel attack, thus, it is denoted measurement matrix. A hypothetical model required for the attack is formed based on the HD of the SBOX

[5] CMOS Technologies are used in this thesis including 180nm which is provided by TSMC, 90nm and 45nm are made available by STMicroelectronics. In design of the test structure standard cells with minimum transistor size are used.

outputs. The hypothetical model is obtained for all the possible keys (2^8). The result is an *N x 256* matrix which is referred to as the estimation matrix. The final step is to compare the one-column measurement matrix with the *256*-column estimation matrix for *N* numbers of inputs. Applying the correlation coefficient test, one expects to see the highest correlation only for the correct key. As the number of inputs increases the correlation coefficient which is calculated between the hypothetical model and the measurement reduces. Only for the correct key the correlation coefficient should remain high. Side channel information leakage is quantified using the number of traces (corresponding to the number of plaintexts) for disclosing the key. The results of simulation-based attack using leakage power consumption will be discussed next.

6.3.2 Result Review

The results of the attack on test bench designed in 180nm CMOS process are shown. Figure 6.3 (a) and (b) illustrate the correlation versus the number of traces and correlation for all the key guesses, respectively.



(a) (b)

Figure 6.3 Result of the simulated attack on the test bench in 180nm CMOS process

a) correlation vs number of traces b) correlations corresponding to all keys

No significant correlation is seen for the test bench in 180nm CMOS technology. The results appear the same for all the key values and no keys can be extracted from the leakage power. Hence, exploiting leakage power consumption did not lead to leakage of useful information. This is due to the fact that in 180nm CMOS process the leakage power consumption is significantly low. A similar attack is launched on the test bench which is designed in 90nm CMOS process. Unlike the previous attack, exploiting leakage power consumption provides sufficient information for extracting the keys. Some keys are revealed with less number of traces; however, some keys require more traces. Figure 6.4 shows the number of traces for extracting different keys. Keys are extracted by minimum 290 and maximum 400 traces. In particular, key "0x1c" requires 365 traces to be revealed. This key is used as a benchmark since its number of traces is almost in the mid range of total number of trace.

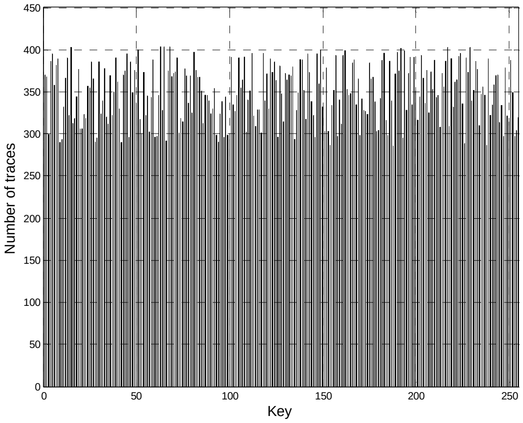
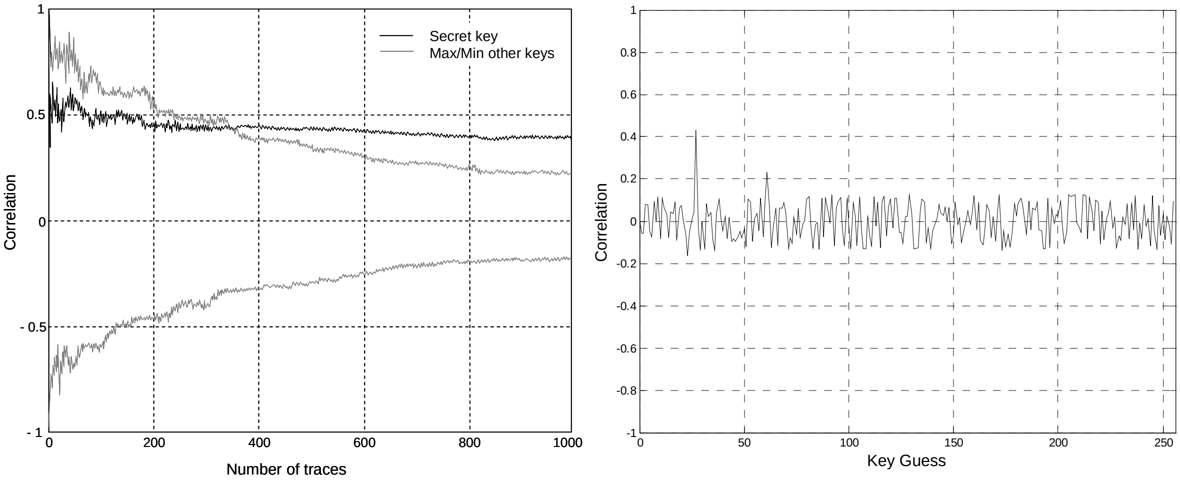


Figure 6.4 Number of traces for extracting different keys in 90nm CMOS process



(a) (b)

Figure 6.5 Result of the simulated attack on the test bench in 90nm CMOS process a) number of traces for extracting the correct key “0x1c” b) correlations corresponding to all keys

Figure 6.5 (a) shows the results of correlation analysis for key equals “0x1c”. Correlation coefficients for all keys are also shown in Figure 6.5 (b). Attack on the test bench designed in 45nm CMOS process is also successful. All keys are extracted. The number of traces for extracting different keys varies between 200 and 250 (Figure 6.6).

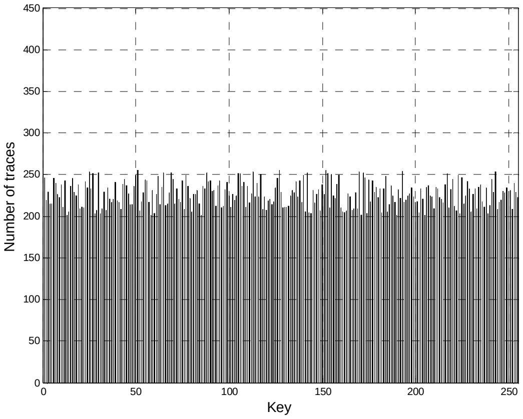
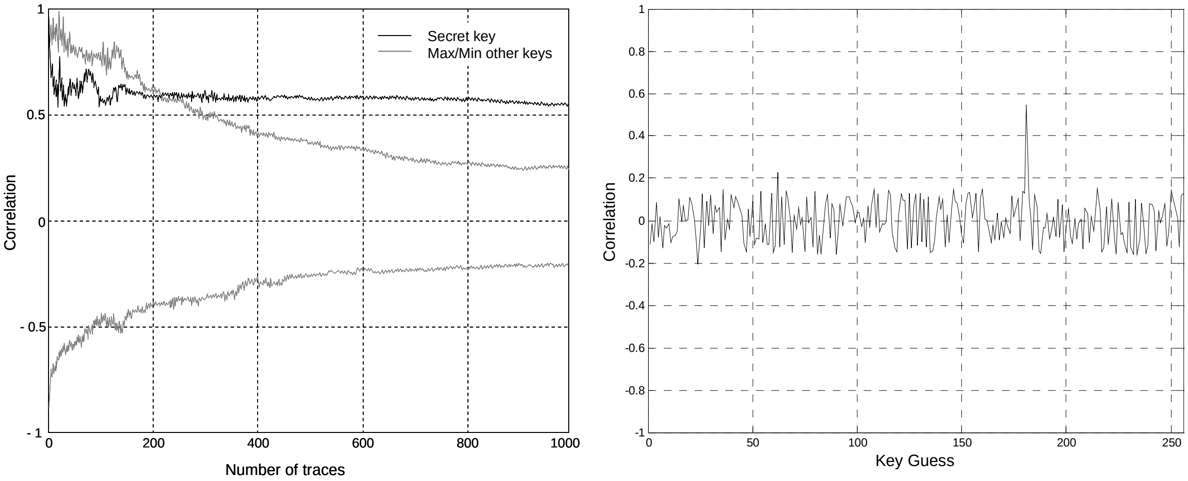


Figure 6.6 Number of traces for extracting different keys in 45nm CMOS process

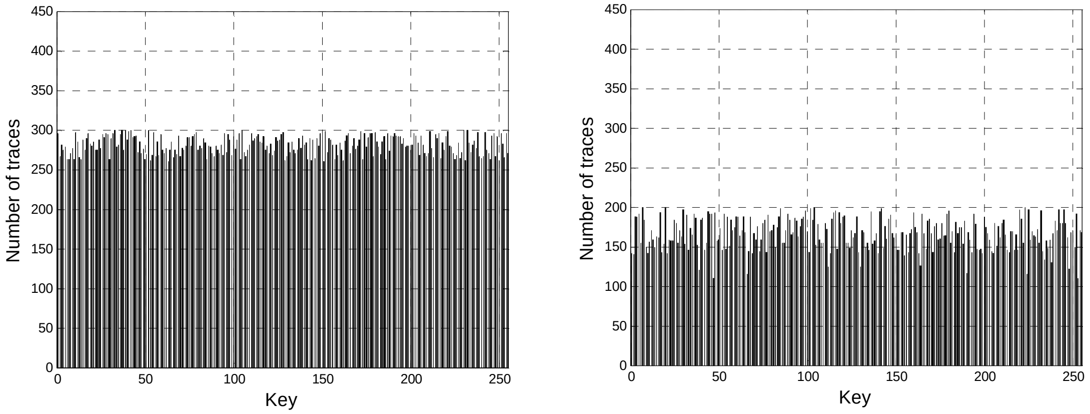


(a) (b)

Figure 6.7 Result of the simulated attack on the test bench in 45nm CMOS process a) number of traces for extracting the correct key “0xb5” b) correlations corresponding to all keys

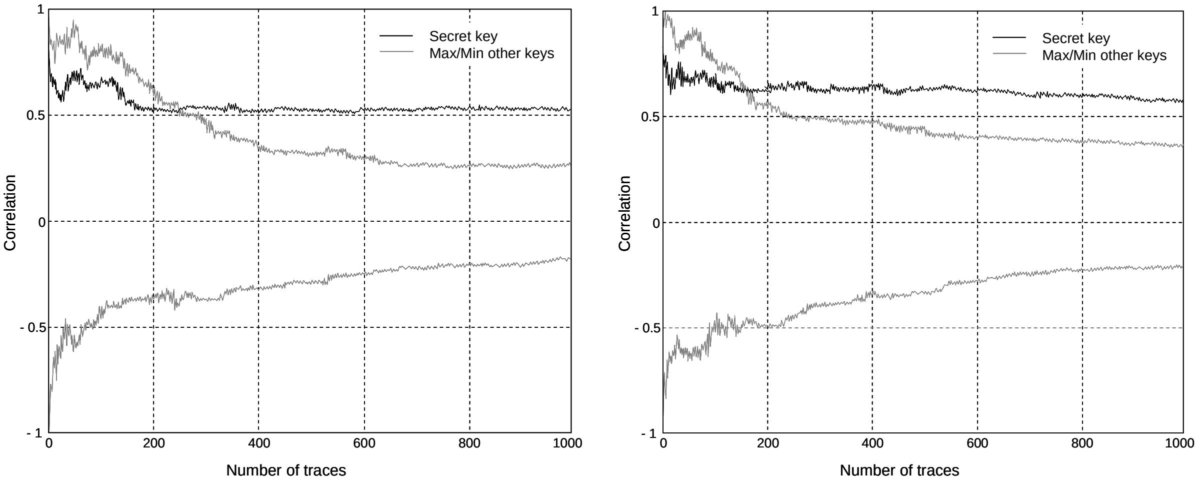
The results of the correlation analysis in Figure 6.7 (a) show that 225 traces are sufficient for revealing the correct key “0xb5”. The correlation coefficients of all the keys are shown in Figure 6.7 (b). Although the number of traces is not scaled with same rate, it was seen that the overall number of traces for key revelation reduces approximately 35% when the technology node changes from 90 to 45nm CMOS process. According to the number of traces, one can conclude that the resistance of crypto core to side channel information leakage reduces as the transistor feature size scales down. Hence, the increasing trend of leakage power is also highly correlated with security vulnerability of cryptosystems.

Further Observation: The results from the previous simulations are obtained at the temperature of 25^C (Figure 6.4 and Figure 6.6). In reality, the attack is performed when the temperature is expected higher than 25^C (during the core operation). For further analysis we repeat our previous simulation at 125^C. The test benches are attacked using the same set of plaintexts and the results are reviewed.



(a) (b)

Figure 6.8 Number of traces for all the keys in a) 90nm b) 45nm CMOS process at 125^C



(a) (b)

Figure 6.9 Result of the simulated attack on the test bench at 125^C for extracting a) “0x1c” in 90nm and b) “0xb5” in 45nm CMOS process

No significant results are obtained from the attack on the test bench in 180nm CMOS process. The attacks on test benches in 90 and 45nm are successful. The results illustrated in Figure 6.8 (a) and (b) show that the number of traces for revealing the keys is reduced. The keys are revealed by 260-310 traces in 90nm CMOS process. The number of traces for extracting the key from the test bench designed in 45nm CMOS process is changed to 115-220. Figure 6.9 (a) shows that key “0x1c” can be extracted by 280 traces. Figure 6.9 (b) depicts that in 45nm CMOS process key “0xb5” is revealed by 155 traces. Approximately 20% reduction is seen in the number of traces in 90nm CMOS process as a result of 100^{C} increase in temperature. The reduction is seen 5% more in 45nm CMOS process. It is evident that temperature rising causes more reduction in number of traces. Therefore, the second simulation-based attack led to the two following results:

- the core becomes more vulnerable at higher temperature
- impact of temperature on side channel vulnerability is more pronounced for advanced technology.

The first outcome can be explained by revisiting the mechanisms involved in generating leakage power. It is shown in [18] that the different leakage components have different temperature dependence. Subthreshold current is governed by the carrier diffusion that increases with temperature. Tunneling probability of an electron through potential barrier does not directly depend on temperature; however, increasing temperature reduces silicon’s band gap, which is the barrier height for tunneling in BTBT leakage current. In general, the gate and the junction BTBT leakage are less sensitive to temperature variations. Figure 6.10 shows the effect of temperature variation on leakage component of 25nm NMOS transistor [18].

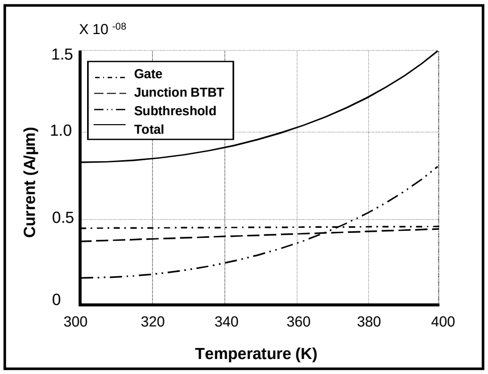


Figure 6.10 Leakage mechanisms and temperature dependency in an NMOS transistor [18]

It is seen that subthreshold leakage increases exponentially with temperature, the junction BTBT increases slowly and the gate leakage are almost independent of temperature variation. From the brief review of leakage mechanisms and the simulation-based results of the attack at higher temperature, one can conclude that the increased vulnerability to leakage of side channel information is due to temperature dependency of the subthreshold leakage mechanism. This is an important observation as it highlights the role of subthreshold leakage in a real attack.

The second outcome can also be explained by reviewing the trend of leakage power versus the technology nodes (Figure 6.2). The total leakage power consumption increases with technology scaling e.g. total leakage power in 45nm is greater than 90nm CMOS process. The upward trend also applies directly to the portion of the leakage that is generated by subthreshold mechanism [18]. This means that subthreshold leakage in 45nm becomes greater than in 90nm CMOS process. By taking the temperature dependency of the subthreshold leakage into consideration, it can be understood why the core becomes more vulnerable to information leakage at a higher temperature when the implementation technology is scaled down.

The above discussion addresses the increasing role of leakage power from a side channel perspective. Further analysis also highlights the importance of subthreshold leakage in side channel information leakage. In order to explore the potential methods for increasing the resistance against information leakage via leakage power consumption, the effectiveness of leakage reduction technique will be examined next.

6.4 Side Channel Aware Leakage Control

The feasibility of exploiting leakage power consumption in a side channel attack was discussed in the previous section. The reduced number of traces for key revelation shows that the threat posed by leakage power consumption increases in the advanced technologies. Previous research proposals have presented several techniques for controlling the mechanisms generating the leakage power [18]. The issue raised now is whether or not these techniques are suitable for developing side channel security. In response to this question, we analyze the side channel resistance of the test bench in the presence of a popular leakage power reduction technique. The quantifiable results are then compared to the simulation results in Section 6.3.2, where no leakage reduction techniques are employed.

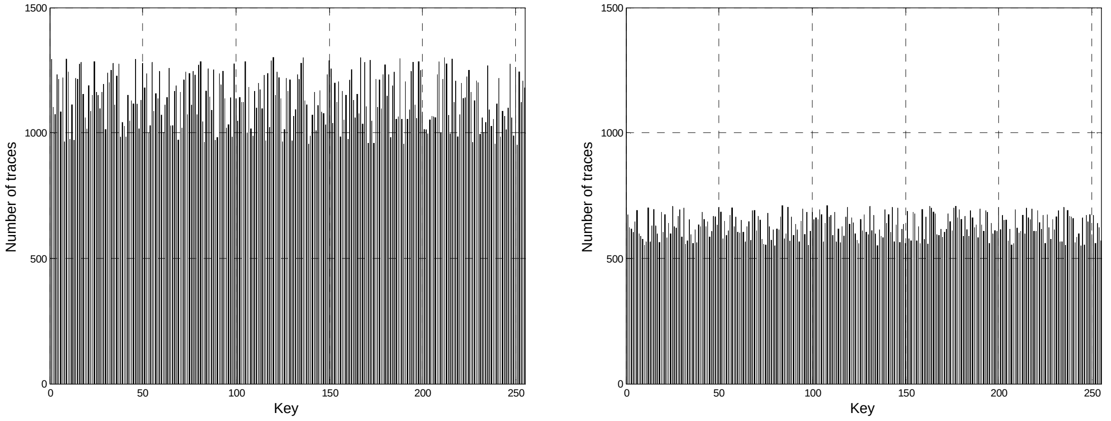
6.4.1 Leakage Power Control and Side Channel Effect

The importance of subthreshold leakage in real attack scenario is shown in Section 6.3.2. Therefore, we focus on subthreshold leakage reduction technique. The main approach for reducing the subthreshold leakage is based on increasing the transistor threshold voltage *(V_{th})* (Equation 2.3). In practice, the threshold voltage is determined during the fabrication process. Transistors can be designed to operate with dual or multiple threshold voltages. The threshold voltage can also be increased at the circuit level by using forced stack effect or reverse body bias. Studies in [18] showed that increasing V_{th} during the fabrication of transistor is the most efficient technique for lowering leakage power. Thus, instead of using extra circuitry for increasing V_{th}, e.g. stacking effect and reverse body bias we assign V_{th} by using the low V_{th} and high V_{th} transistor models available. Transistors are then divided to operate either at low or high V_{th}. The effectiveness of leakage power control technique on developing resistances against side channel attack is evaluated as follows.

The test structure in Section 6.3.1 was designed using low V_{th} transistors. All low V_{th} NMOS and PMOS transistors in the test bench are now replaced with high V_{th} transistors[6] and the test bench is redesigned in 90nm and 45nm CMOS processes. All the other characteristics of the test structure remain the same. A similar procedure was followed as described in Section 6.3.1 for obtaining the number of trace to disclose the key. The results are presented as follows.

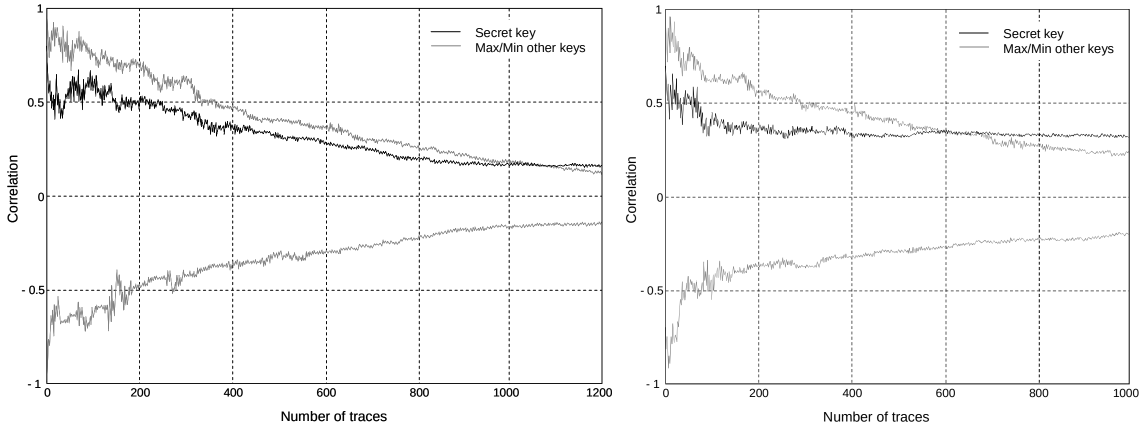
[6] The HSPICE models are available for high threshold and low threshold voltage. The high and low threshold voltages are -0.55, -0.48, -0.46, -0.42, and 0.55, 0.48, 0.45, 0.42V are recorded for PMOS and NMOS transistors at 90 and 45nm CMOS processes.

Figure 6.11 (a) shows the results of simulation-based attack on test bench in 90nm CMOS process. It is seen that retrieving the keys requires between 950 and 1300 traces.



(a) (b)

Figure 6.11 Number of traces for all the keys in a) 90nm b) 45nm CMOS process

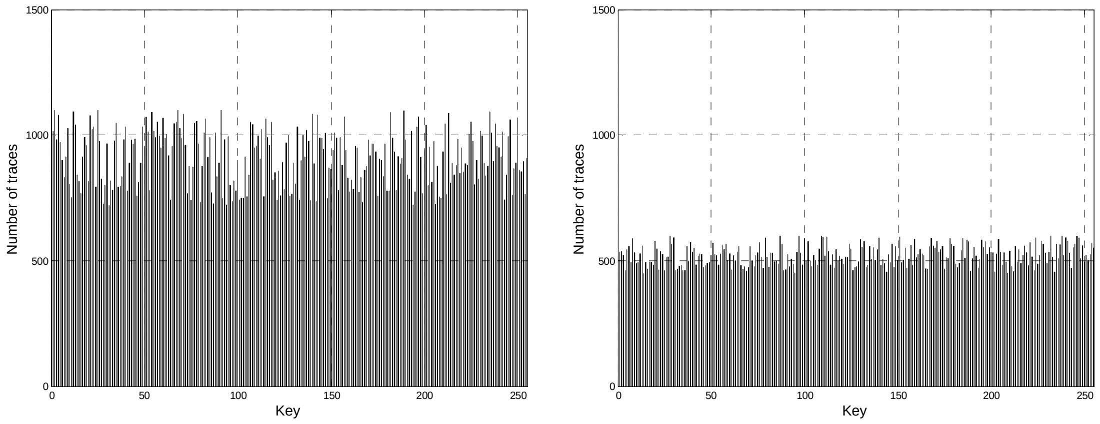


(a) (b)

Figure 6.12 Result of the simulated attack on the test bench for extracting a) “0x1c” in 90nm and b) “0xb5” in 45nm CMOS process

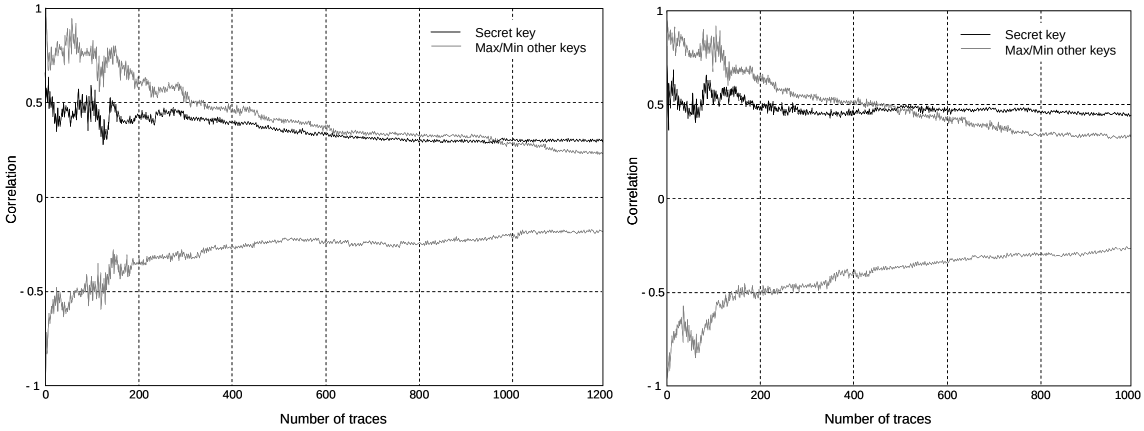
Figure 6.11 (b) shows that in 45nm design between 550 and 710 traces can reveal the keys. The number of traces versus correlation coefficient of the particular keys ("0x1c" and "0xb5" in 90nm and 45nm CMOS processes) are illustrated in Figure 6.12 (a) and (b). Approximately, 1150 and 655 traces can reveal the same key from the high V_{th} test bench designed in 90nm and 45nm CMOS processes. The results are compared with those in Figure 6.5 (a) and Figure 6.7 (a) of the previous test benches which were designed using only low V_{th} transistors. Increases of 3.2 and 2.9 times are observed in number of traces in 90nm and 45nm CMOS processes, respectively. Although the keys are still revealed, the increased number of traces shows that deployment of high V_{th} transistors provides resistance against side channel information leakage in both technology nodes.

Further Observation: To investigate the impact of temperature variation, the above simulation-based attack is repeated at 125^C. The number of traces for extracting the key is determined and shown in Figure 6.13 (a) and (b). Revealing the keys requires 710-1100 and 450-600 traces in 90nm and 45nm CMOS processes, respectively. Keys "0x1c" and "0xb5" are retrieved using 980 and 510 traces (Figure 6.14). A comparison between the results of the simulations in 25^C and 125^C shows that 20% and 17% reduction can be seen in the number of traces in test bench in 90nm and 45nm CMOS processes. This equals approximately to the reduction caused by the temperature increase in our attack in Section 6.3.2, where low V_{th} transistors are used.



(a) (b)

Figure 6.13 Number of traces for all the keys in a) 90nm b) 45nm CMOS process at 125^{C}



(a) (b)

Figure 6.14 Result of the simulated attack on the test bench at 125^{C} for extracting a) “0x1c” in 90nm and b) “0xb5” in 45nm CMOS process

6.5 Summary and Conclusion

This chapter investigated the evolution of the side channel threat with technology. The number of traces for key extraction is seen to reduce by 35% from 90nm to 45nm CMOS technology nodes. It was shown that leakage power plays an increasingly important role from side channel perspective as we venture into the nanoscale technology era. It was also observed an increase of 100^{C} in temperature reduces the number of traces (for key extraction) by 20% and 25% in 90nm and 45nm CMOS processes, respectively. This result highlighted the importance of subthreshold leakage mechanism in side channel information leakage. For the first time, the effectiveness of a circuit-based leakage reduction technique was examined for the side channel application. Security analysis of a crypto core showed that assigning high V_{th} transistor increased the resistance against information leakage. Although not completely removing the side channel threat, high V_{th} transistor assignment can be employed as a part of a hybrid solution in design of a side channel resistant crypto core. This analysis provided insight into design and implementation of side channel aware cryptosystem in nanoscale technologies where the role of leakage power is expected to be significantly important.

Chapter 7
Discussions and Conclusions

7.1 Summary of Work

This thesis studies the leakage of side channel information and introduces a new logic level countermeasure. The novel use of a constant power consuming single-ended logic style is proposed for designing side channel resistant logic gates. A similar concept is employed in the design and implementation of an edge triggered register. The proposed approach provides resistance against leakage of side channel information for both combinational and sequential logic networks. A test chip is fabricated in 180nm CMOS technology for validation of the results. The test chip includes a protected crypto core which is designed using the proposed countermeasure and an unprotected core (CMOS) with no countermeasure. An attack mounted on the CMOS core leads to revelation of the key. The attack of the core protected by the proposed countermeasure did not reveal the keys even with the enhanced analysis technique. The results show that the proposed countermeasure provides a cost effective security mechanism for area-constrained applications. Single-ended (non-differential) output logic cells and registers also reduce the overall design complexity.

Furthermore, this thesis analyzes the impact of technology scaling on the side channel vulnerability of cryptosystems. The role of leakage power consumption from a side channel perspective is investigated. The leakage of information via leakage power consumption is quantified by the number of traces for key extraction. It is shown that the upward trend of the leakage power correlates with leakage of information. The impact of temperature is included in this investigation identifying the important role of subthreshold leakage. In response to growing concerns about the potential security threat posed by leakage power, the effectiveness of a circuit-based leakage reduction technique is examined. To the best of our knowledge, this investigation for the first time provides an analysis of information leakage in the presence of high V_{th} transistor. Although, the attack is successful, some resistance provided by the high threshold voltage *(V_{th})* is evident. These results can be used for developing a side channel aware leakage strategy for future resistant Cryptosystem-on-Chip.

7.2 Comparison to Previous Research

The major differences between the countermeasure proposed in this thesis and the previous research are discussed in this section.

- The countermeasures presented in [52][53][54][55][56][57][60] are based on dynamic logic. Thus, a significant area overhead associated with the clocking network at the gate level is seen. The static logics for side channel security are introduced in [59][86]. The logic discussed in [59] needs differential outputs. The overhead of having double the number of transistors still exists in the logic introduced in [59]. The technique discussed in [86] uses a random logic scheme. A considerable amount of area is also required for implementation of the random switching circuitry. The proposed countermeasure is the lowest area consuming logic among both dynamic and static side channel resistant logics (See Table 5.3).
- The threat of side channel leakage at the logic level in [52][53][54][55][56][57][59][60] is tackled by using a differential logic style. Unlike differential logic, the proposed approach suggests employing a single-ended logic style, thus, no specific layout methodology is required (See Section 5.5.3).
- Research in [59][86] provides side channel resistance only for combinational networks using a static logic scheme; however, the proposed approach presents a side

channel resistance register element in addition to resistant logic cells. Thus, side channel resistance is provided for both elements of combinational and sequential networks (See Section 4.3.2).

- Previous research in [59] uses a current mode logic which requires a bias voltage *(V_{bias})* at the gate level. This thesis proposes a countermeasure which is also current mode logic, but it uses a PMOS load as a current source. Therefore, it does not require an extra circuitry for V_{bias}. This reduces the design complexity as well as the area consumption (See Section 5.5.3).
- The impact of process variation on side channel analysis is lacking in previous research. Unlike [52][53][54][55][56][57][60][86], this research includes the effect of process variation on analysis of the proposed countermeasure (See Section 4.4.2).
- Evaluation of the side channel resistance in [52][59][60] is based on simulation results. Unlike those, this research provides analysis based on empirical results. Using the real power traces validates the concept of the reference vector insertion which is also introduced in this research (See Section 5.5).
- This research, unlike [91][92], considers the impact of temperature on information leakage via leakage power. Thus, for the first time, the role of subthreshold leakage mechanism in leakage of side channel information is highlighted (See Section 6.3).
- To the best of our knowledge, the proposed research is the first to explore the effectiveness of high threshold voltage transistors for developing side channel resistance (See Section 6.4).

7.3 Summary of Contributions

This section outlines the contributions of this thesis to the field of side channel security.

a) Contribution to developing resistance against information leakage via power consumption (silicon-based result)

- Achieving resistance against power-based side channel attacks for combinational networks by the novel use of Current Balanced Logic (CBL) in the design and implementation of basic logic cells.
- Providing side channel resistance for sequential networks by designing Current Balanced Edge Triggered Registers (CBETRs).
- Proposing a cost effective approach which provides side channel resistance for area-constrained applications.
- Introducing a logic level side channel countermeasure with reduced complexity.
- Investigating the impact of process variation on side channel information leakage of the proposed countermeasure.
- Validating the effectiveness of the proposed countermeasure using empirical results.

b) Contribution to evaluating information leakage (designer perspective)

- Introducing an enhanced evaluation technique, a known-value reference vector, for analyzing the side channel information.
- Verifying the effectiveness of using the minimum power consuming vector as a reference (using real power measurement).

c) Contribution to future technology trends in side channel analysis (simulation-based results)

- Examining the impact of technology scaling on information leakage via leakage power consumption and quantifying the vulnerability of the next generation of cryptosystems to side channel information leakage.
- Emulating the real attack scenario by considering the impact of temperature variation and addressing the increasing role of subthreshold leakage mechanism.
- Investigating the effectiveness of the high threshold transistor assignment in providing side channel resistance.

7.4 Future Direction and Solution Extensions

Important future research directions emerging from this thesis can be divided into two areas: the analysis of side channel information and the synthesis of side channel countermeasures.

Analysis of side channel information and potential strategies

- **Enhanced model for side channel information leakage:** Leakage of side channel information has been modeled based on signal-to-noise ratio. The effects of parasitics and crosstalk noise associated with pads and packaging can also be added to the current model. The enhanced model will not only provide a realistic evaluation of the side channel threat but also assist to characterize the leakage of information via a more complex side channel such as electromagnetic emission.
- **Advanced Electrical Design Automation (EDA) tools for side channel evaluation:** New metrics can be defined and added to the current simulation tool. The predefined metrics will be used to evaluate the side channel vulnerability at the design time. Design optimization for side channel resistance will also be feasible if such metrics are available. Thus, the side channel evaluation will be integrated into advanced EDA tools.
- **Empirical analysis on side channel vulnerability of nanoscale cryptosystems:** The viability of exploiting leakage power consumption in a side channel attack needs to

be empirically proven. Providing a detailed analysis of side channel leakage will also address the practical issues involved in data capturing and data analysis. This will provide a realistic image of the future side channel threat.

Synthesis of side channel countermeasure and potential strategies

- **Power/area efficient side channel countermeasure:** More effort is still needed for developing power efficient side channel countermeasures for area-constrained application. Providing a side channel resistant methodology for sensitive applications with a limited source of power consumption such as hand held devices is still an open issue. The challenge increases more when extra constrains such as performance and area are added to the trade-off model.
- **Effective countermeasure for cryptosystems in nanoscale:** Design and implementation of countermeasures for thwarting the threat of leakage-based side channel attack are necessary. Techniques can range from circuit level countermeasures to architectural level countermeasures.

References

[1] ePaynews – Mobile Commerce Statistics.
http://www.epaynews.com/statistics/mcommstats.html

[2] J. Markoff, “Secure Digital Transaction Just Got a Little Less Secure,” New York Times, 11 December 1995.

[3] J. Kelsey, B. Schneier, D. Wagner, and C. Hall, “Side Channel Cryptanalysis of Product Cipher,” in Proc. of 5th European Symposium of Research in Computer Security (ESORICS), Lectures Notes in Computer Science, vol. 1485, Springer, pp. 97-110, September 1998.

[4] http://csrc.nist.gov/publications/PubsFIPS.html

[5] P. Kocher, “Timing Attack on Implementation of Diffie-Hellman, RSA, DSS and other Systems,” in Proc. of 16th Advances in Cryptology (CRYPTO), Lectures Notes in Computer Science, vol. 1109, pp. 104-113, 1996.

[6] P. Kocher, J. Jaffe and B. Jun, “Differential Power Analysis,” in Proc. of 19th Advances in Cryptology (CRYPTO), Lectures Notes in Computer Science , vol. 1666, pp. 388-397, 1999.

[7] J. Quisquater and D. Samyde, “Electromagnetic Analysis (EMA): Measures and Counter-measures for Smart-cards,” in Proc. of Smart Card Programming and Security (E-smart), Lectures Notes in Computer Science, vol. 2140, Springer, pp. 200-210, 2001.

[8] S. B. Ors, F. Gurkaynak, E. Oswald and B. Preneel, “Power- Analysis Attack on an ASIC AES Implementations,” in Proc. of International Conference on Information Technology (ITCC), 2004.

[9] B. Örs, E. Oswald and B. Preneel, “Power-Analysis Attacks on an FPGA--First Experimental Results,” in Proc. of 4th Workshop on Cryptographic Hardware and Embedded Systems (CHES), Lecture Notes in Computer Science, vol. 2779, pp. 35-50, Sept. 2003.

[10] C.Gebotys, "Design of Secure Cryptography against the Threat of Power-Attacks in DSP Embedded Processors," ACM Transactions on Embedded Computer Systems, vol. 3, no. 1, 2004.

[11] T. S. Messergers and E. A. Dabbish, "Power Analysis of Modular Exponentiation in Smart Cards," in Proc. of 1st Cryptography Hardware and Embedded Systems (CHES), Lectures Notes in Computer Science, vol. 1717, Springer, pp. 144-157, 1999.

[12] T. S. Messerges, "Securing the AES Finalists against Power Analysis Attacks," in Proc. of 7th International Workshop on Fast Software Encryption, Lecture Notes in Computer Science, vol. 1978, Springer, pp. 150-164, 2000.

[13] J.S. Coron and L. Goubin, "On Boolean and Arithmetic Masking against Differential Power Analysis," in Proc. of 2nd Cryptography Hardware and Embedded Systems (CHES), Lecture Notes in Computer Science, vol. 1965, Springer, pp. 231-237, 2000.

[14] A. Akkar and B. Giraud, "An Implementation of DES and AES, Secure against Some Attacks," in Proc. of 3rd Cryptography Hardware and Embedded Systems (CHES), Lecture Notes in Computer Science, vol. 2162, Springer, pp. 309-318, 2001.

[15] C. Karlof and D. Wagner, "Hidden Markov Model Cryptanalysis," in Proc. of 5th Cryptography Hardware and Embedded Systems (CHES), Lecture Notes in Computer Science, vol. 2779, Springer, pp. 17-30, 2003.

[16] A. Shamir and E. Tromer, "Acoustic Cryptanalysis,"
http://www.wisdom.weizmann.ac.il/~tromer/acoustic/, 2004.

[17] J. Rabaey, A. Chandrakasan, and B. Nikolic, Digital Integrated Circuits, 2nd edition, Prentice Hall, 2003.

[18] K Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage Current Mechanisms & Leakage Reduction Techniques in deep-submicron CMOS Circuits," in IEEE, vol. 91. issue 2, pp 305-327, Feb. 2003.

[19] E. Brier, C. Clavier, and F. Olivier, "Correlation Power Analysis with Leakage Model," in Proc. of 6th Cryptography Hardware and Embedded Systems (CHES), Lecture Notes in Computer Science, vol. 3156, Springer, pp. 16-29, 2004.

[20] T. Le, J. Clediere, C. Serviere and J. Lacoume, "Efficient Solution for Misalignment of Signal in Side Channel Analysis," in Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2007.

[21] D. Moore and G. MaCabe, Introduction to the Practice of Statistic, Freeman, 1993.

[22] N. Pramstaller, E. Oswald, S. Mangard, F. Gurkaynak, and S. Haene, "A Masked AES ASIC Implementation," in Proc. of Austrochip, Springer, pp. 77-82, 2004.

[23] L. Goubin, “A Sound Method for Switching between Boolean and Arithmetic Masking,” in Proc. of 3rd Cryptography Hardware and Embedded Systems (CHES), Lecture Notes in Computer Science, vol. 2162, Springer, pp. 3-15, 2001.

[24] J. S. Coron, and A. Tchulkine, “A New Algorithm for Switching from Arithmetic to Boolean Masking,” in Proc. of 2nd Cryptography Hardware and Embedded Systems (CHES), Lecture Notes in Computer Science, vol. 2779, Springer, pp. 89-97, 2003.

[25] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi, “Towards Sound Approaches to Counteract Power Analysis Attacks,” in Proc. of 19th Annual International Cryptography Conference, Lecture Notes in Computer Science, vol. 1666, Springer, pp. 398-412, 1999.

[26] http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf

[27] http://www.csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[28] M. L. Akhtar, and L. Goubin, “A Generi Protection against High Order Differential Power Analysis,” in Proc. of 10th International Fast Software Cryptography (FSE), Lecture Notes in Computer Science, vol. 2887, Springer, pp. 192-205, 2003.

[29] Rivest, R.; A. Shamir; L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems". Communications of the *ACM* 21 (2), pp.120–126, 1978.

[30] J. S. Coron, “Resistance against Differential Power Analysis for Elliptic Cure Cryptosystems,” in Proc. of 1st Cryptography Hardware and Embedded Systems (CHES), Lecture Notes in Computer Science, vol. 1717, Springer, pp. 292-302, 2003.

[31] http://csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTReCur.pdf

[32] D. May, H. L. Muller, and N. P. Smart, “Non-deterministic Processors,” in Proc. of 6th Australasian Conference (ACISP), Lecture Notes in Computer Science, vol. 2119, Springer, pp. 115-129, 2001.

[33] J. Irwin, D. Page, and N. P. Smart, “Instruction Stream Mutation for Non-Deterministic Processors,” in Proc. of IEEE International Conference on Application-Specific Systems, Architecture and Processors, pp. 286-295, 2002.

[34] S. Yang, W. Wolf, N. Vijaykrishnan and D. Serpanos, “Power Attack Resistant Cryptosystem Design: A Dynamic Voltage and Frequency Switching Approach,” in Proc. of Design, Automation and Test in Europe Conference (DATE), pp. 64-69, 2005.

[35] J. Coron, P. Koucher and D. Naccache, "Statistics and Secret Leakage," in Proc. of 4th International Financial Cryptography (FC), Lecture Notes in Computer Science, vol. 1962, Springer, pp. 157-173, 2001.

[36] A. Shamir, "Protecting Smart Cards from Passive Power Analysis with Detached Power Supply," in Proc. of 2nd Cryptography Hardware and Embedded Systems (CHES), Lectures Notes in Computer Science, vol. 1965, Springer, pp. 71-77, 2000.

[37] P. Corsonello, S. Perri, and M. Margala, "A New Charge-pump Based Countermeasure against Differential Power Analysis," in Proc. of 6th IEEE International Conference on ASIC (ASICON), Lecture Notes in Computer Science, vol.1, pp. 66-69, 2005.

[38] P. Rakers, L. Connell, T. Collins, and D. Russell, "Secure Contactless Smartcard ASIC with DPA Protection," in IEEE Journal of Solid-State Circuits (JSSC), vol. 36, no. 3, pp. 559-565, 2001.

[39] G. B. Rantanpal, R. D. Williams, and T. N. Blaock, "An On-chip Signal Suppression Countermeasure to Power Analysis Attacks," in IEEE Transactions on Dependable and Secure computing, vol. 1, no. 3, pp. 179-189, 2004.

[40] R. Muresan, H. Vahedi, Y. Zhanrong, and S. Gregori, "Power Smart System-on-Chip Architecture for Embedded Cryptosystems," in Proc. of 3rd IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES + ISSS), ACM, pp. 184-189, 2005.

[41] D. Mesquita, J. D. techer, L. Torres, G. Sassatelli, G. Cambon, M. Robert, and F. Moraes, "Current Mask Generation; A Transistor Level Security against DPA Attacks," in Proc. of 18th Annual Symposium on Integrated Circuits and System Design (SBCCI), ACM, pp. 115-120, 2005.

[42] L. Benini, A. Macii, E. Macii, E. Omerbegovic, F. Pro, and M. Poncino, "Energy-Aware Design Techniques for Differential Power analysis Protection," in Proc. of 40th Design Automation Conference (DAC), ACM, 2003.

[43] L. Benini, A. Macii, E. Macii, E. Omerbegovic, M. Poncino, and F. Pro, "A Novel Architecture for Power Maskable Arithmetic Units," in Proc. of 13th ACM Great Lake Symposium on VLSI (GLVLSI), ACM, pp. 136-140, 2003.

[44] M. L. Akhtar, C. Giraud, "An Implementation of DES and AES Secure against Some Attacks," in Proc. of 3rd Cryptography Hardware and Embedded Systems (CHES), Lecture Notes in Computer Science, vol. 2162, Springer, pp. 309-318, 2001.

[45] K. Tiri, and I. Verbauwhede, "Securing Encryption Algorithm against DPA at the Logic Level: Next Generation Smart Card Technology," in Proc. of 5th Cryptography Hardware and Embedded Systems (CHES), Lecture Notes in Computer Science, vol. 2779, Springer, pp. 137-151, 2003.

[46] J. D. Golic, and C. Tymen, "Multiplicative Masking and Power Analysis of AES," in Proc. of 4th Cryptography Hardware and Embedded Systems (CHES), Lecture Notes in Computer Science, vol. 2779, Springer, pp. 137-151, 2002.

[47] M. Bucci, M. Guglielmo, R. Luzzi, and A. Trifiletti, "A Power Consumption Randomization Countermeasures for DPA-resistant Cryptographic Processor," in Proc. of 14th International Workshop on Integrated Circuit and System Design, Power, and timing Modeling, Optimization and Simulation (PATMOS), Lecture Notes in Computer Science, vol. 3254, Springer, pp. 481-490, 2004.

[48] D. May, H. L. Muller, and N. P. Smart, "Random Register Renaming to Foil DPA ," in Proc. of 3rd Cryptography Hardware and Embedded Systems (CHES), Lecture Notes in Computer Science, vol. 2162, Springer, pp. 28-38, 2001.

[49] L. Benini, A. Galati, A. Macii, E. Macii, and M. Poncino, "Energy Efficient Data Scrambling on Memory-Processor Interfaces," in Proc. of International Symposium on Low Power Electronics and Design (ISLPED), ACM, pp. 26-29, 2003.

[50] J. Golic, "DeKaRT: A New Paradigm for Key-dependent Reversible Circuits," in Proc. of 5th Cryptography Hardware and Embedded Systems (CHES), Lecture Notes in Computer Science, vol. 2779, Springer, pp. 98-112, 2003.

[51] R. Elbaz, L. Torres, G. Sassatelli, P. Guillemin, C. Anguille, M. Bardouillet, C. Buatois, and J. B. Riguad, "Hardware Engines for Bus Encryption: A Survey of Existing Techniques," in Proc. of Design Automation and Test in Europe Conference and Exposition (DATE), pp. 40-45, 2005.

[52] K. Tiri M. Akmal and I. Verbauwhede, "A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Power Analysis on Smart Cards," in Proc. of 28th European Solid-State Circuits Conference (ESSCIRC), pp. 403-406, 2002.

[53] K. Tiri, and I. Verbauwhede, "A Logic Level Design Methodology for Secure DPA Resistant ASIC or FPGA Implementation," in Proc. of Design Automation and Test in Europe Conference and Exposition (DATE), pp. 246-251, 2004.

[54] D. Sokolov, J. Murphy, A. Bystrov, and A. Yakovlev, “Design and Analysis of Dual-Rail Circuits for Security Applications,” in IEEE Transactions on Computer, vol. 54, no. 4, pp. 449-460, 2005.

[55] M. Bucci, L. Giancane, R. Luzzi, and A. Trifiletti, “Three-phase Dual-rail Pre-charge Logic,” in Proc. of 8th Cryptography Hardware and Embedded Systems (CHES), Lecture Notes in Computer Science, vol. 4249, Springer, pp. 232-241, 2006.

[56] M. Aigner, S. Mangard, R. Menicocci, M. Olivieri, G. Scotti, and A. Trifiletti, “A Novel CMOS Logic Style with Data Independent Power Consumption,” in Proc. of International Symposium on Circuits & Systems (ISCAS), pp. 1066-1069, 2005.

[57] K. Tiri and I. Verbauwhede, “Charge Recycling Sense Amplifier Based Logic: Securing Low Power Security IC’s vs. DPA,” in Proc. of 30th European Solid-State Circuits Conference (ESSCIRC), pp. 179-182, 2004.

[58] Kris Tiri, and Ingrid Verbauwhede, “Place and Route for Secure Standard Cell Design”, in Proc. of 6th International Conference on Smart Card Research and Advanced Applications (CARDIS), pp. 143-158, 2004.

[59] Z. Toprak, and Y. Leblebici “Low-Power Current Mode Logic for Improved DPA-Resistance in Embedded Systems,” in Proc. of International Symposium on Circuits & Systems (ISCAS), pp. 1059-1062, 2005.

[60] F. Macé, F. Standaert, J. Quisquater and J. Legat, “A Dynamic Current Mode Logic to Counteract Power Analysis Attacks,” in Proc. of 19th Conference on Design of Circuits and Integrated Systems (DCIS), pp. 186-191, 2004.

[61] M. Allam and M. Elmasry, “Dynamic Current Mode Logic (DyCML): A New Low-Power High Performance Logic Style,” in IEEE Journal of Solid-State Circuits (JSSC), vol. 36, no. 3, pp. 550-559, 2001.

[62] Y. Zhang. A Novel Adiabatic Differential Switch Logic Technique for Ultra-Low Energy VLSI Design and Security Analysis. Thesis, University of Waterloo, Waterloo, Canada, 2003.

[63] J. Fournier, S. Moore, H. Li, R. Mullins and G. Taylor, “Security Evaluation of Asynchronous Circuits,” in Proc. of 2nd Cryptography Hardware and Embedded Systems (CHES), Lectures Notes in Computer Science, vol. 2779, Springer, pp. 137-151, 2003.

[64] Z. C. Yu, S. B. Furber, and L. A. Plana, “An Investigation into Security of Self-timed Circuits,” in Proc. of 9th International Symposium on Advanced research in Asynchronous Circuits and Systems (ASYNC), pp. 206-215, 2003.

[65] K. J. Kulikowski, M. Su, A. B. Smirnov, A. Taubin, M. G. Karpovsky, and D. MacDonald, “Delay Insensitive Encoding and Power Analysis: A Balancing Act,” in Proc. of 11th International Symposium on Advanced research in Asynchronous Circuits and Systems (ASYNC), pp. 116-125, 2005.

[66] K. J. Kulikowski, A. B. Smirnov, and A. Taubin, “Automated Design of Cryptographic Devices Resistant to Multiple Side Channel Attacks,” in Proc. of 8th Cryptography Hardware and Embedded Systems (CHES), Lectures Notes in Computer Science, vol. 4249, Springer, pp. 399-413, 2006.

[67] E. trichina, T. Korkishko, and K. Lee, “Small Size, Low Power, Side Channel-Immune AES Coprocessor: Design and Synthesis Results,” in Proc. of 4th International Conference, AES, Lectures Notes in Computer Science, vol. 3373, Springer, pp. 113-127, 2005.

[68] j. D. Golic, and R. Menicocci, “Universal Masking on Logic Gate Level,” in IEE Electronic Letters, vol. 40, no. 9, pp. 526-527, 2004.

[69] Y. Ishai, A. Sahai, and D. Wagner, “Private Circuits: Securing Hardware against Probing Attacks”, in Proc. of 23rd International Cryptology Conference, AES, Lectures Notes in Computer Science, vol. 2729, Springer, pp. 463-481, 2003.

[70] W. Fischer, and B. M. Gammel, “Masking at Gate Level in the Presence of Glitches,” in Proc. of 7th Cryptography Hardware and Embedded Systems (CHES), Lectures Notes in Computer Science, vol. 3659, Springer, pp. 187-200, 2005.

[71] Z. Chen, and Y. Zhou, “Dual-Rail Random Switching Logic: A Countermeasure to Reduce Side Channel Leakage”, in Proc. of 8th Cryptography Hardware and Embedded Systems (CHES), Lectures Notes in Computer Science, vol. 4249, Springer, pp. 242-254, 2006.

[72] T. Popp and S. Mangard, “Implementation Aspects of the DPA-Resistant Logic Style MDPL,” in Proc. of IEEE International Symposium on Circuits & Systems (ISCAS), pp. 2913-2916, 2006.

[73] K. Tiri, and I. Verbauwhede, “Design Method for Constant Power Consumption of Differential Logic Circuits,” in Proc. of Design Automation and Test in Europe Conference and Exposition (DATE), pp. 628-633, 2005.

[74] S. Guilley, P. Hoogvorst, Y Mathieu, and R. Pacalet, "The Backend Duplication Method," in Proc. of 7th Cryptography Hardware and Embedded Systems (CHES), Lectures Notes in Computer Science, vol. 3659, Springer, pp. 383-397, 2005.

[75] D. Real, C. Canovas, J. Clediere, and M. Drissi, "Defeating Classical Hardware Countermeasures: A New Processing for Side Channel Analysis," in Proc. of Design Automation and Test in Europe Conference and Exposition (DATE), pp. 978-981, 2008.

[76] T. H. Le, J. Clediere, C. Serviere, and J. Lacoume, "Efficient Solution for Misalignment of Signal in Side Channel Analysis" in Proc. of IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP), pp. 257-260, 2007.

[77] D. Real, C. Canovas, J. Clediere, M. Drissi, and F. Valette, "Defeating Classical Hardware Countermeasure: A New Processing for Side Channel Analysis," in Proc. of Design Automation and Test in Europe Conference and Exposition (DATE), pp. 1274-1279, 2008.

[78] E. M. Albuquerque, and M. M. Silva "A Comparison by Simulation and by Measurement of the Substrate Noise Generated by CMOS, CSL, and CBL Digital Circuits," in IEEE Transactions on Circuits and Systems-I, vol. 52, no. 4, pp. 734-740, 2005.

[79] L. Yang, and J. S. Yuan, "Design of Enhancement Current-Balanced Logic for Mixed Signal ICs," in Proc. of IEEE International Symposium on Circuits & Systems (ISCAS), pp. 761-764, 2003.

[80] J. Musicer et al., "MOS Current Mode Logic for Low-power, Low-noise CORDIC Computation in Mixed-signal Environment," in Proc. of International Symposium on Low Power Electronics and Design (ISLPED), pp. 102-107, 2000.

[81] H. T. Ng, and D. J Allstot, "CMOS Current Steering Logic for Low-Voltage Mixed-Signal Integrated Circuits ," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 5, no. 3, pp. 301-308, 1997.

[82] http://www.cadence.com/products/custon_ic/index.aspx?lid=custom_ic_design

[83] http://www.imse.cnm.es/recursos/laboratorio/93000_Specifications.pdf

[84] http://www2.tek.com/cmswpt/madetails.lotr?ct=MA&cs=mur&ci=9573&lc=EN

[85] http://eprint.iacr.org/2008/188.pdf

[86] M. Saeki, D. Suzuki, K. Shimizu, and A. Satoh, “A Design Methodology for a DPA-Resistant Cryptographic LSI with RSL Techniques”, in Proc. of 10th Cryptography Hardware and Embedded Systems (CHES), Lectures Notes in Computer Science, vol. 5747, Springer, pp. 189-205, 2010.

[87] http://csrc.nist.gov/publications/PubsFIPS.html

[88] G. Moore, “Progress in Digital Integrated Circuits,” in Proc. of the International Electronic Meeting, pp. 11-13, 1975

[89] (2005) International Technology Roadmap for Semiconductors. International SEMATECH, Austin, TX. http://public .itrs.net/

[90] G. Merrett et al., “Leakage Power Analysis and Comparison of Deep Submicron Logic Gates,” in Proc. *Power and Timing Modeling Optimization and Simulation Conference* (PATMOS), pp. 198-207, 2007.

[91] J. Giorgetti, et al., “Analysis of Data dependence of Leakage Current in CMOS Cryptographic Hardware,” in Proc. Great Lake Symposium of VLSI (GLVLSI), pp. 78-83, 2007.

[92] L. Lin, and W. Burleson “Leakage-based Differential Power Analysis on Sub-90nm CMOS Cryptosystems,” in Proc. of IEEE International Symposium on Circuits & Systems (ISCAS), pp. 78-83, 2008.

Appendix A

Basic Logic Cells

The library of basic logics is shown in Figure A.1 (CMOS) and Figure A.2 (CBL). Table A.1 shows the transistor dimensions and total area. The high threshold transistor (M_2) is used in design of CBL gates (V_{th} = 0.6).

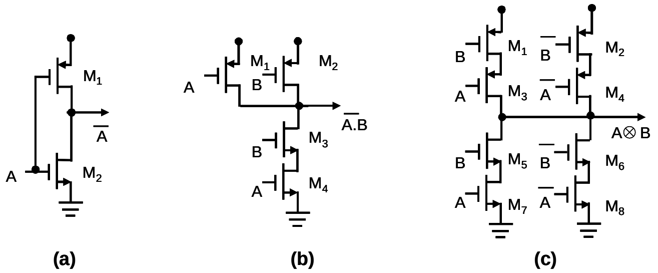


Figure A.1 CMOS) inverter b) NAND c) XOR gate

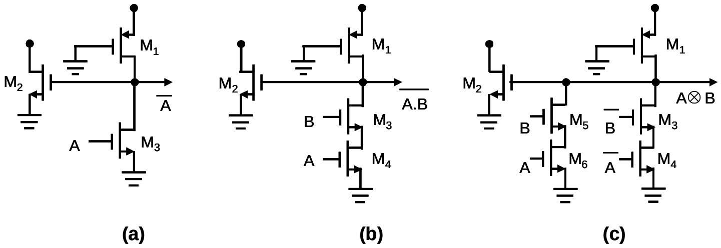


Figure A.2 CBL) inverter b) NAND c) XOR gate

Table A.1 Transistor dimensions and total area of logic gates

	Inverter		NAND		XOR	
	CMOS	CBL	CMOS	CBL	CMOS	CBL
M_1 *(W/L)*	8	3	8	3.5	8	3.5
M_2 *(W/L)*	3	0.5	8	0.7	8	0.7
M_3 *(W/L)*	-	2.5	6	5	8	2.5
M_4 *(W/L)*	-	-	6	5	8	2.5
M_5 *(W/L)*	-	-	-	-	6	2.5
M_6 *(W/L)*	-	-	-	-	6	2.5
M_7 *(W/L)*	-	-	-	-	6	-
M_8 *(W/L)*	-	-	-	-	6	-
Total Area *(μm^2)*	14.28	25.56	53.2	43.62	96.56	77.24

Appendix B

Substitution Box (SBOX)

The SubByte transformation is computed by taking the multiplicative inverse in $GF(2^8)$ followed by an affine transformation. The Affine Transformation can be represented in matrix form and it is shown below [27].

$AT(b)$ is an Affine Transformation while the vector b is the multiplicative inverse of the input byte from the state array (Figure B.1). It is observed that “SubByte” involves a multiplicative inversion operation. Multiplicative inverse module can be implemented by using look-up table or computational-based circuits. A common side channel aware strategy is to design and implement computational-based circuits by using secure logic style. The multiplicative inverse computation is described and then affine transformation will follow to complete the review of the test structure.

$$AT(b) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} b_7 \\ b_6 \\ b_5 \\ b_4 \\ b_3 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Figure B.1 Affine Function

The individual bits in a byte representing a $GF(2^8)$ element can be viewed as coefficient to each power term in the $GF(2^8)$ polynomial, e.g. $\{10001011\}_2$ is representing the polynomial $p^7 + p^3 + p + 1$ in $GF(2^8)$. Any arbitrary polynomial can be represented as $ax + b$, given an irreducible polynomial of $x^2 + Ax + B$. Thus, elements in $GF(2^8)$ may be represented as $ax + b$ where a is the most significant nibble while b is the least significant nibble. Therefore, the multiplicative inverse can be computed using the equation below.

$$(ax+b)^{-1} = a(a^2B+abA+b^2)^{-1}x+(b+aA)(a^2B+abA+b^2)^{-1} \quad \text{(B.1)}$$

The irreducible polynomial that is selected is $x^2+x+\lambda$. Since $A=1$ and $B=\lambda$, then the Equation 4.23 is simplified to the form as shown below:

$$(ax+b)^{-1} = a(a^2\lambda+b(a+b))^{-1}x+(b+a)(b^2\lambda+c(b+c))^{-1} \quad \text{(B.2)}$$

Equation (A.2) indicates that there are multiply, addition, squaring and multiplication inversion in $GF(2^4)$ operations in Galois Field. Each of these operators can be transformed into individual blocks when constructing the circuit for computing the multiplicative inverse.

a) Isomorphic Mapping

The multiplicative inverse computation can be performed by decomposing the more complex $GF(2^8)$ to lower order fields of $GF(2^1)$, $GF(2^2)$ and $GF((2^2)^2)$. In order to accomplish the above, the following irreducible polynomials are used:

$GF(2^2) \rightarrow GF(2): x^2+x+1$

$GF((2^2)^2) \rightarrow GF(2^2): x^2+x+\varphi$

$GF(((2^2)^2)^2) \rightarrow GF((2^2)^2): x^2+x+\lambda$

where $\varphi=\{10\}_2$ and $\lambda=\{1100\}_2$.

Computation of the multiplicative inverse in composite fields cannot be directly applied to an element which is based on $GF(2^8)$. The element has to be mapped to its composite field representation via an isomorphic function, δ. Likewise, after performing the multiplicative inversion, the result will also have to be mapped back from its composite field representation to its equivalent in $GF(2^8)$ via inverse isomorphic function δ^{-1}. Both δ and δ^{-1} can be represented as an 8×8 matrix. Let $a$be the elements in $GF(2^8)$, then the isomorphic mappings and its inverse can be written as $\delta\times a$ and $\delta^{-1}\times a$, which is a case of matrix multiplication as shown in Figure A. 2 where a_7 is the most significant bit and a_0 is the least significant bit.

$$\delta \times \mathrm{a} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} \mathrm{a}_7 \\ \mathrm{a}_6 \\ \mathrm{a}_5 \\ \mathrm{a}_4 \\ \mathrm{a}_3 \\ \mathrm{a}_2 \\ \mathrm{a}_1 \\ \mathrm{a}_0 \end{bmatrix} \qquad \delta^{-1} \times \mathrm{a} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \mathrm{a}_7 \\ \mathrm{a}_6 \\ \mathrm{a}_5 \\ \mathrm{a}_4 \\ \mathrm{a}_3 \\ \mathrm{a}_2 \\ \mathrm{a}_1 \\ \mathrm{a}_0 \end{bmatrix}$$

Figure B.2 Isomorphic functions

The matrix multiplication can be translated to logical XOR operation.

$$\delta \times \mathrm{a} = \left| \begin{array}{c} a_7 \oplus a_5 \\ a_7 \oplus a_6 \oplus a_4 \oplus a_3 \oplus a_2 \oplus a_1 \\ a_7 \oplus a5 \oplus a_3 \oplus a_2 \\ \mathrm{a}_7 \oplus a_5 \oplus a_3 \oplus a_2 \oplus a_1 \\ a_7 \oplus a_6 \oplus a_2 \oplus a_1 \\ a_7 \oplus a_4 \oplus a_3 \oplus a_2 \oplus a_1 \\ a_6 \oplus a_4 \oplus a_1 \\ a_6 \oplus a_1 \oplus a_0 \end{array} \right| \qquad \delta^{-1} \times \mathrm{a} = \left| \begin{array}{c} a_7 \oplus a_6 \oplus a_5 \oplus a_1 \\ a_6 \oplus a_2 \\ a_6 \oplus a_5 \oplus a_1 \\ a_6 \oplus a5 \oplus a_4 \oplus a_2 \oplus a_1 \\ a_5 \oplus a_4 \oplus a_3 \oplus a_2 \oplus a_1 \\ a_7 \oplus a_4 \oplus a_3 \oplus a_2 \oplus a_1 \\ a_5 \oplus a_4 \\ a_6 \oplus a_5 \oplus a_4 \oplus a_2 \oplus a_0 \end{array} \right|$$

Figure B. 3 Xor representation of isomorphic functions

b) Composite Field Arithmetic Operations

As previously mentioned any arbitrary polynomial can be represented by $ax + b$ where a is upper half term and b is the lower half term. Therefore, a binary number in Galois Field a can be split to $a_H x + \mathrm{a}_L$, e.g. if $a = \{1010\}_2$, it can be represented as $\{10\}_2 x + \{10\}_2$, where a_H is $\{10\}_2$ and a_L is $\{10\}_2$. a_H and a_L can be further decomposed to $\{1\}_2 x + \{0\}_2$ and $\{1\}_2 x + \{0\}_2$, respectively. The decomposing is executed by using irreducible polynomial introduced earlier. The logical equations for addition, squaring, multiplication and inversion can be derived.

Addition in $GF(2^4)$ can be translated to simple XOR operation.

Squaring in $GF(2^4)$ for a, an element in $GF(2^4)$ represented by binary number $\{a_3, a_2, a_1, a_0\}$, is $k = a^2$ with representation of $\{k_3, k_2, k_1, k_0\}$

$k_3 = a_3,\ k_2 = a_3 \oplus a_2,\ k_1 = a_2 \oplus a_1,\ k_0 = a_3 \oplus a_1 \oplus a_0$

Multiplication with constant $\lambda = \{1100\}_2$ in $GF(2^4)$ for a, an element in $GF(2^4)$ represented by binary number $\{a_3, a_2, a_1, a_0\}$, is $k = a\lambda$ with representation of $\{k_3, k_2, k_1, k_0\}$

$k_3 = a_2 \oplus a_0,\ k_2 = a_3 \oplus a_2 \oplus a_1 \oplus a_0,\ k_1 = a_3,\ k_0 = a_2$

Multiplication in $GF(2^4)$ for a and b, two elements in $GF(2^4)$ represented by binary number $\{\underbrace{a_3, a_2}_{a_H}, \underbrace{a_1, a_0}_{a_L}\}$ and $\{\underbrace{b_3, b_2}_{b_H}, \underbrace{b_1, b_0}_{b_L}\}$, is $k = ab$ with representation of $\{\underbrace{k_3, k_2}_{k_H}, \underbrace{k_1, k_0}_{k_L}\}$

$k_3 = k_H x + k_L = (a_H b_H + a_H b_L + a_L b_H)x + a_H b_H \lambda + a_L b_L$

Multiplication in $GF(2^2)$ for a and b, two elements in $GF(2^2)$ represented by binary number $\{a_1, a_0\}$and $\{b_1, b_0\}$, is $k = ab$ with representation of $\{k_1, k_0\}$

$k_1 = a_1 b_1 \oplus a_0 b_1 \oplus a_1 b_0,\ k_0 = a_1 b_1 \oplus a_0 b_0$

Multiplication with constant $\varphi = \{10\}_2$ in $GF(2^2)$ for a, an element in $GF(2^2)$ represented by binary number $\{a_1, a_0\}$, is $k = a\varphi$ with representation of $\{k_1, k_0\}$

$k_1 = a_1 \oplus a_0,\ k_0 = a_1$

Multiplication inverse in $GF(2^4)$ for a, an element in $GF(2^4)$ represented by binary number $\{a_3, a_2, a_1, a_0\}$, is a^{-1} with representation of $\{a_3^{-1}, a_2^{-1}, a_1^{-1}, a_0^{-1}\}$

Appendix C

Agilent SOC 93000 Tester

Digital Specification	
No. of Digital Channel	480
Max. Serial Data Rate	500 Mbits/s @ $3V_{pp}$
64 Channel – Data rate	1GHz clock@ 3 V_{pp}
416 Channel – Data rate	330 Mbits/s@ 3 V_{pp}
AC Performance	
Min. Pulse width	1ns@ $3V_{pp}$ 0.8ns@ $3V_{pp}$
DC Performance	
Level Range	-2 V to 6.5 V
Level Resolution	2.5mV
Level Accuracy	+-10mV
Impedance	
Source Impedance	50 Ohm +- 2.5 Ohm
Programmable Load	
Current (I_{oh}, I_{ol})	0 to 35 mA
Current Resolution	12.5uA
Current Accuracy	75 uA 1% of max (I_{oh}, I_{ol})
Max. Vector Memory	16 Million vectors/pin
Edge Placement Accuracy	+-100 ps
Device Power Supplies	
2DPS board with 4 channel each	
Max. Current per Channel	

Appendix D

Pin Configuration (ICFWSBX)

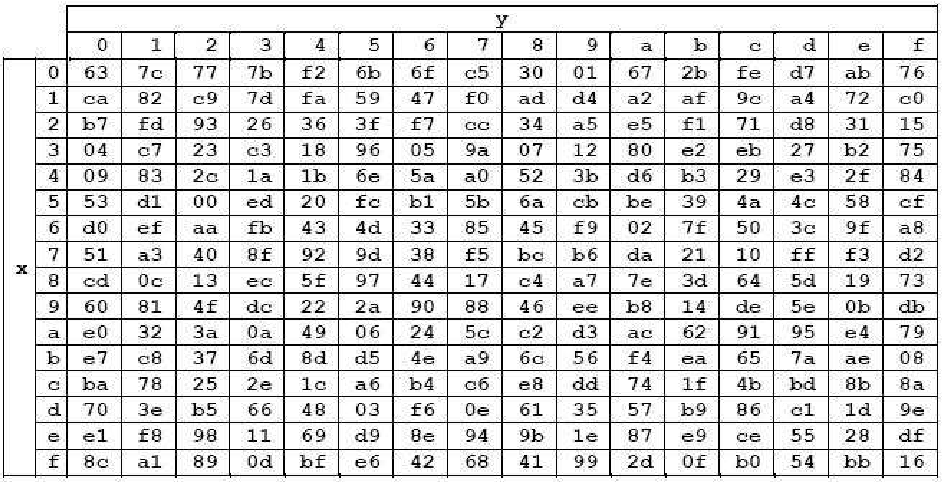
Bonding Diagram	Pin # (PGA69A)	Tester Channel	Pin ID
1-4	C1	10310	DATA_OUT_4_CMOS
2-5	D2	10311	DATA_OUT_3_CMOS
3-6	D1	10312	DATA_OUT_2_CMOS
4-7	E2	10313	DATA_OUT_1_CMOS
5-8	E1	10314	DATA_OUT_8_CBL
6-9	F2	10315	DATA_OUT_7_CBL
7-10	F1	10702	DATA_OUT_6_CBL
8-11	G2	10704	DATA_OUT_5_CBL
9-12	G1	10703	DATA_OUT_4_CBL
10-13	H2	10706	DATA_OUT_3_CBL
11-14	H1	10705	DATA_OUT_2_CBL
12-15	J2	10708	DATA_OUT_1_CBL
13-16	J1	10707	VDD_CMOS_REG_OUT
14-22	K4	10713	VSS_CBL
15-31	L8	11103	VDD_CBL
16-32	K9	11107	VDD_BUFFER
17-33	L9	11105	VDD_PAD
18-34	L10	11106	VSS_CMOS_REG_OUT
19-35	K10	10909	KEY_1
20-36	K11	11108	KEY_2
21-37	J10	11109	KEY_3
22-38	J11	11110	KEY_4
23-39	H10	11111	KEY_5
24-40	H11	11112	KEY_6
25-41	G10	11113	KEY_7
26-42	G11	11114	KEY_8
27-43	F10	11502	DATA_IN_1
28-44	F11	11115	DATA_IN_2
29-45	E10	11504	DATA_IN_3
30-46	E11	11503	DATA_IN_4
31-47	D10	11506	DATA_IN_5

32-53	A10	11510	DATA_IN_6
33-54	B9	11511	DATA_IN_7
34-55	A9	11512	DATA_IN_8
35-56	B8	11513	CLK_IN
36-57	A8	11514	VSS_BUFFER
37-59	A7	11516	VSS_CMOS
38-60	B6	11316	VDD_CMOS
39-61	A6	10101	VDD_CBL_REG
40-62	B5	10302	DATA_OUT_8_CMOS
41-63	A5	10301	DATA_OUT_7_CMOS
42-64	B4	10304	DATA_OUT_6_CMOS
43-65	A4	10303	DATA_OUT_5_CMOS
44-66	C3	11517	SEL CORE
45-67	D4	11518	VSS_CBL_REG

Appendix E

SubByte Transformation

The SBOX used in the SubByte transformation in AES. Figure A. 4 shows the Substitution values for the byte xy (in hexadecimal format).



		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Figure E. 1 AES SBOX Table

Functional Test Pattern

Clock	**Data_In**	**Key**	**Data_Out**
1	11111111	10001000	00000000
1	11111110	10001000	00010100
1	11111101	10001000	11111001
1	11111100	10001000	00100111
1	11111011	10001000	10001100
1	11111010	10001000	11010001
1	11111001	10001000	11110100
1	11111000	10001000	10011000
1	11110111	10001000	11010101
1	11110110	10001000	11011101
1	11110101	10001000	11001111
1	11110100	10001000	01110101
1	11110011	10001000	01001101
1	11110010	10001000	10111000
1	11110001	10001000	00011010
1	11110000	10001000	11010000
1	11101111	10001000	01001110

1	11101110	10001000	01110001
1	11101101	10001000	11001100
1	11101100	10001000	00100100
1	11101011	10001000	11101111
1	11101010	10001000	00101101
1	11101001	10001000	01011010
1	11101000	10001000	00100010
1	11100111	10001000	11110110
1	11100110	10001000	01000010
1	11100101	10001000	00011100
1	11100100	10001000	01110010
1	11100011	10001000	10100000
1	11100010	10001000	01101111
1	11100001	10001000	10001101
1	11100000	10001000	00001001
1	11011111	10001000	11100010
1	11011110	10001000	11100001
1	11011101	10001000	01000000
1	11011100	10001000	00110101
1	11011011	10001000	10100111
1	11011010	10001000	00101110
1	11011001	10001000	01101011
1	11011000	10001000	01111110
1	11010111	10001000	11100110
1	11010110	10001000	10110100
1	11010101	10001000	01011011
1	11010100	10001000	00101111
1	11010011	10001000	00000001
1	11010010	10001000	11101010
1	11010001	10001000	01111101
1	11010000	10001000	00011101
1	11001111	10001000	01000101
1	11001110	10001000	00011001
1	11001101	10001000	01010101
1	11001100	10001000	01011100
1	11001011	10001000	11001001
1	11001010	10001000	10100100
1	11001001	10001000	00110100
1	11001000	10001000	11001000
1	11000111	10001000	11101110
1	11000110	10001000	10010001
1	11000101	10001000	00000010
1	11000100	10001000	11101100
1	11000011	10001000	11000100
1	11000010	10001000	10101101
1	11000001	10001000	00000000
1	11000000	10001000	11110010
1	10111111	10001000	10010011
1	10111110	10001000	01110011
1	10111101	10001000	00001010
1	10111100	10001000	10001001
1	10111011	10001000	10001110
1	10111010	10001000	11010010
1	10111001	10001000	10010100
1	10111000	10001000	00100110
1	10110111	10001000	01111111

1	10110110	10001000	00001101
1	10110101	10001000	00001000
1	10110100	10001000	10100110
1	10110011	10001000	11010111
1	10110010	10001000	01100001
1	10110001	10001000	01010010
1	10110000	10001000	01111011
1	10101111	10001000	00111001
1	10101110	10001000	10010110
1	10101101	10001000	11000010
1	10101100	10001000	10010010
1	10101011	10001000	01101100
1	10101010	10001000	00111000
1	10101001	10001000	11011000
1	10101000	10001000	11111010
1	10100111	10001000	01001111
1	10100110	10001000	11111101
1	10100101	10001000	01001001
1	10100100	10001000	10110001
1	10100011	10001000	00011000
1	10100010	10001000	00010010
1	10100001	10001000	00000100
1	10100000	10001000	01000100
1	10011111	10001000	01011111
1	10011110	10001000	11011001
1	10011101	10001000	10100010
1	10011100	10001000	01000011
1	10011011	10001000	00101100
1	10011010	10001000	00010111
1	10011001	10001000	01001010
1	10011000	10001000	00100011
1	10010111	10001000	00001100
1	10010110	10001000	10000010
1	10010101	10001000	00111101
1	10010100	10001000	00110110
1	10010011	10001000	11100000
1	10010010	10001000	10000110
1	10010001	10001000	01010110
1	10010000	10001000	01100010
1	10001111	10001000	10110101
1	10001110	10001000	10000111
1	10001101	10001000	00001011
1	10001100	10001000	00000111
1	10001011	10001000	11101101
1	10001010	10001000	01011101
1	10001001	10001000	10010000
1	10001000	10001000	10110011
1	10000111	10001000	11000110
1	10000110	10001000	00110001
1	10000101	10001000	10001010
1	10000100	10001000	11100111
1	10000011	10001000	00100000
1	10000010	10001000	00001110
1	10000001	10001000	11001010
1	10000000	10001000	00000110
1	01111111	10001000	01010011

1	01111110	10001000	11111011
1	01111101	10001000	00010101
1	01111100	10001000	10011110
1	01111011	10001000	10101000
1	01111010	10001000	01010001
1	01111001	10001000	00100001
1	01111000	10001000	11001110
1	01110111	10001000	01101110
1	01110110	10001000	01101000
1	01110101	10001000	01001011
1	01110100	10001000	00010000
1	01110011	10001000	10101110
1	01110010	10001000	01111001
1	01110001	10001000	11110011
1	01110000	10001000	11011011
1	01101111	10001000	00000011
1	01101110	10001000	00101001
1	01101101	10001000	10100001
1	01101100	10001000	00111010
1	01101011	10001000	00110011
1	01101010	10001000	01100011
1	01101001	10001000	00000101
1	01101000	10001000	11101000
1	01100111	10001000	10100011
1	01100110	10001000	00010110
1	01100101	10001000	10101111
1	01100100	10001000	10010101
1	01100011	10001000	01011001
1	01100010	10001000	01110000
1	01100001	10001000	11011010
1	01100000	10001000	00010001
1	01011111	10001000	00001111
1	01011110	10001000	10010111
1	01011101	10001000	11111110
1	01011100	10001000	01000110
1	01011011	10001000	10001111
1	01011010	10001000	11111000
1	01011001	10001000	11001101
1	01011000	10001000	10111100
1	01010111	10001000	11010100
1	01010110	10001000	11110000
1	01010101	10001000	10000100
1	01010100	10001000	01010111
1	01010011	10001000	01000111
1	01010010	10001000	10011101
1	01010001	10001000	10011100
1	01010000	10001000	00101000
1	01001111	10001000	11110101
1	01001110	10001000	10001000
1	01001101	10001000	11011111
1	01001100	10001000	01010000
1	01001011	10001000	01100100
1	01001010	10001000	01110100
1	01001001	10001000	01011000
1	01001000	10001000	00110111
1	01000111	10001000	11011110

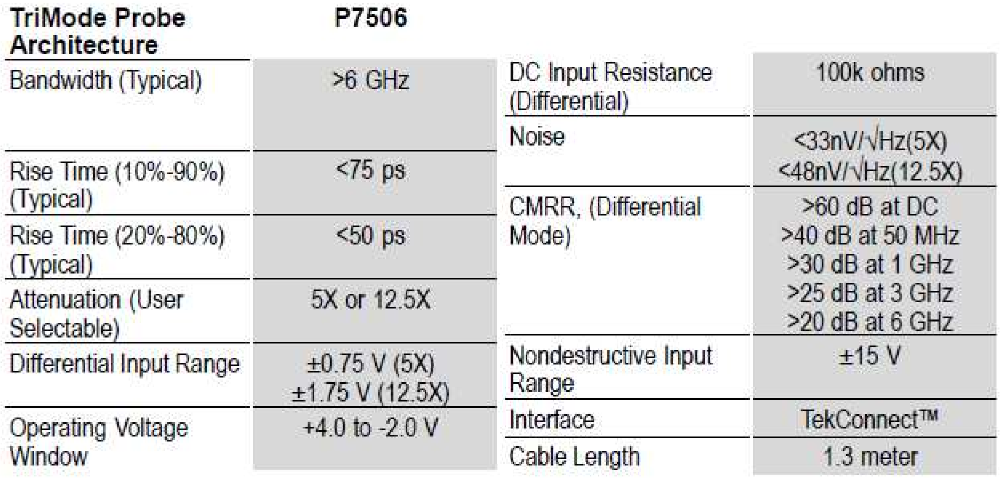
1	01000110	10001000	10110000
1	01000101	10001000	11110001
1	01000100	10001000	10110110
1	01000011	10001000	11000011
1	01000010	10001000	01100110
1	01000001	10001000	10110111
1	01000000	10001000	00111011
1	00111111	10001000	10111110
1	00111110	10001000	10101010
1	00111101	10001000	00111100
1	00111100	10001000	10101001
1	00111011	10001000	00011011
1	00111010	10001000	10111101
1	00111001	10001000	11000111
1	00111000	10001000	10111010
1	00110111	10001000	11101011
1	00110110	10001000	00101010
1	00110101	10001000	11111111
1	00110100	10001000	01011110
1	00110011	10001000	11100100
1	00110010	10001000	10000011
1	00110001	10001000	00110010
1	00110000	10001000	01111010
1	00101111	10001000	00100101
1	00101110	10001000	10011011
1	00101101	10001000	10110010
1	00101100	10001000	01100000
1	00101011	10001000	11111100
1	00101010	10001000	01100101
1	00101001	10001000	01110110
1	00101000	10001000	11101001
1	00100111	10001000	11010110
1	00100110	10001000	01100111
1	00100101	10001000	10111001
1	00100100	10001000	10101011
1	00100011	10001000	01101001
1	00100010	10001000	11000000
1	00100001	10001000	00111111
1	00100000	10001000	01010100
1	00011111	10001000	10011010
1	00011110	10001000	01111000
1	00011101	10001000	10011111
1	00011100	10001000	11001011
1	00011011	10001000	10100101
1	00011010	10001000	10111011
1	00011001	10001000	11011100
1	00011000	10001000	11100101
1	00010111	10001000	10000000
1	00010110	10001000	10011001
1	00010101	10001000	01101101
1	00010100	10001000	01101010
1	00010011	10001000	01001000
1	00010010	10001000	10101100
1	00010001	10001000	11010011
1	00010000	10001000	01110111
1	00001111	10001000	00101011

1	00001110	10001000	00011111
1	00001101	10001000	11110111
1	00001100	10001000	01001100
1	00001011	10001000	10111111
1	00001010	10001000	00011110
1	00001001	10001000	11000001
1	00001000	10001000	00110000
1	00000111	10001000	00111110
1	00000110	10001000	10000101
1	00000101	10001000	11000101
1	00000100	10001000	00010011
1	00000011	10001000	11100011
1	00000010	10001000	01111100
1	00000001	10001000	10001011
1	00000000	10001000	10000001
1	00000000	10001000	01000001

Appendix F

High Frequency Differential Probe

Tektronix P7506



TriMode Probe Architecture	P7506
Bandwidth (Typical)	>6 GHz
Rise Time (10%-90%) (Typical)	<75 ps
Rise Time (20%-80%) (Typical)	<50 ps
Attenuation (User Selectable)	5X or 12.5X
Differential Input Range	±0.75 V (5X) ±1.75 V (12.5X)
Operating Voltage Window	+4.0 to -2.0 V
DC Input Resistance (Differential)	100k ohms
Noise	<33nV/√Hz(5X) <48nV/√Hz(12.5X)
CMRR, (Differential Mode)	>60 dB at DC >40 dB at 50 MHz >30 dB at 1 GHz >25 dB at 3 GHz >20 dB at 6 GHz
Nondestructive Input Range	±15 V
Interface	TekConnect™
Cable Length	1.3 meter

Appendix G

Digital Oscilloscope

Tektronix

TDS 7704B Oscilloscope

7 GHz with 20 GS/s

Oscilloscope Setup	
Acquisition Mode	HiRes/FastFrame
Trigger Mode	Positive Edge
Record Length	250
Duration of One Frame	25ns
Frequency Span	5GHz
Sampling Rate	10GSample/s

Appendix H

Design and Implementation of Dummy Delay Chain

A delay element is a circuit that produces an output waveform similar to its input waveform which is only delayed by a certain amount of time. Constant-delay element can be designed by using transmission gate or cascaded m series-connected PMOS and NMOS (when m = 1 the delay element is a typical CMOS inverter). Each delay element is suitable for different range of delays. Transmission gate-based is highly recommended for its area efficiency; however, the cascaded-based delay element is recognized for its high yield. Yield is defined as the percentage of total delay elements whose propagation delay falls within a certain delay cut-off. In the presence of parameter variations, delays are distributed over certain range. The distribution of delay can be evaluated by normalized variability, *3σ/μ*, where $\sigma$and μ are the standard deviation and mean of measured delay respectively. The cut-off delay is defined between $\pm 5\,\%$ and $\pm 10\,\%$ of the mean delay.

In order to design an area efficient and robustness delay chain, the delay chains are designed and implemented by cascaded-based delay element. The design process of cascaded inverter-based delay elements is reviewed in the following.

Cascaded inverter-based delay element: Design of inverter-based delay element depends on the time taken to (dis)charge the load capacitance. A reasonable delay approximation can be derived by using a typical inverter model. If average value of the charging current equals to the saturation current of a PMOS (Equation H.1):

$$I_{av}(\,PMOS\,) = k_p \frac{(V_{gs} - V_{th\,p}\,)^2}{2} \tag{H.1}$$

where $k_p = \mu_p C_{ox} W / L$ is gain factor (μ_p, C_{ox}, W and L are mobility, capacitance per unit and width and length of the PMOS device),V_{gs}, $V_{th\,p}$ are gate-source voltage and threshold voltage, respectively. Since $V_{dd} > \left| V_{thp} \right|, V_{thn}$ then $I_{av}(\,PMOS\,)$can be approximated as:

$$I_{av}(PMOS) \approx k_p \frac{(V_{dd})^2}{2} \tag{H.2}$$

Using the $I_{av}(PMOS)$, the propagation delay is expressed as [17]:

$$t_p = \frac{(t_{pLH} + t_{pHL})}{2} = \frac{C_L}{2V_{dd}}(\frac{1}{k_p} + \frac{1}{k_p}) \tag{H.3}$$

where t_{pLH} is propagation delays for low to high, t_{pHL} is propagation delay for high to low output transition, and C_L is the load capacitance. The above expression can be modified when the effect of a nonzero input rise $t_r > t_{pHL}$ on propagation delay is considered:

$$t_{pHL_{(actual)}} = (t^2{}_{pHL(step)} + (t_r / 2)^2)^{1/2} \tag{H.4}$$

Equation H.4 shows that the delay is proportional to gain factor $(k_{p,n})$ and subsequently to $(W/L)_p$ and $(W/L)_n$. Therefore, by sizing the PMOS and NMOS devices approximate delay can be obtained. In order to obtain more delay per unit area and subsequently higher delay values each inverter can be replaced by multiple series-connected PMOS and NMOS devices in pull-down and pull-up networks, respectively. Figure H.1 shows the multiple-transistor cascaded inverter-based delay elements. Similar steps taken for sizing single inverter-based delay element can be followed for sizing the devices in multiple-transistor cascade inverters.

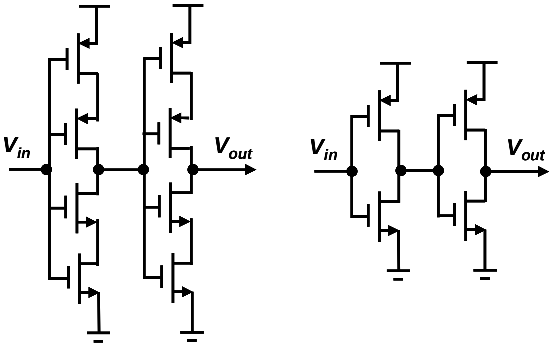


Figure H.1 Delay element