8-2007

Transmission gate based dual rail logic for differential power analysis resistant circuits

Srinidhi Narasimha char
Clemson University, cnarasi@clemson.edu

TRANSMISSION GATE BASED LOGIC FOR DIFFERENTIAL POWER ANALYSIS RESISTANT CIRCCUITS

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Computer Engineering

---

by
Srinidhi Narasimha Char
August 2007

Accepted by:
Dr. Samuel T Sander, Committee Chair
Dr. Richard R Brooks
Dr. Rajendra Singh

ABSTRACT

Cryptographic devices with hardware implementation of the algorithms are increasingly being used in various applications. As a consequence, there is an increased need for security against the attacks on the cryptographic system. Among various attack techniques, side channel attacks pose a significant threat to the hardware implementation. Power analysis attacks are a type of side channel attack where the power leakage from the underlying hardware is used to eavesdrop on the hardware operation. Wave pipelined differential and dynamic logic (WDDL) has been found to be an effective countermeasure to power analysis. This thesis studies the use of transmission gate based WDDL implementation for the differential and dynamic logic.

Although WDDL is an effective defense against power analysis, the number of gates needed for the design of a secure implementation is double the number of gates used for non-secure operations. In this thesis we propose transmission gate based structures for implementation of wave pipelined dynamic and differential logic to minimize the overhead of this defense against power analysis attacks. A transmission gate WDDL design methodology is presented, and the design and analysis of a secure multiplier is given. The adder structures are compared in terms of security effectiveness and silicon area overhead for three cases: unsecured logic implementation, standard gate WDDL, and transmission gate WDDL. In simulation, the transmission gate WDDL design is seen to have similar power consumption results compared to the standard gate WDDL; however, the transmission gate based circuit uses 10-50% fewer gates compared to the static WDDL.

DEDICATION

This work is dedicated to my parents for all their support and encouragement.

ACKNOWLEDGMENTS

I would to thank God for my current achievements and for being a moral support throughout my life. I would like to thank my parents and my family for their constant guidance and support. I would like to thank Dr. Sander, my advisor, for providing valuable advice, support and encouragement throughout the course of my study. I would like to thank Dr. Singh and Dr. Brooks for their valuable input, encouragement and for serving on my thesis committee. I would like to thank my friends for shaping my non academic personality.

TABLE OF CONTENTS

Table of Contents (Continued)

Page

LIST OF TABLES

LIST OF FIGURES

List of Figures (Continued)

CHAPTER ONE
INTRODUCTION

Cryptographic secure systems form a unique application of information theory and also relate to other engineering branches involving communication systems, mathematics, software design and circuit design. A central problem for such cryptographic systems is to know that they have not been tampered with or replaced by a malicious third party. These attacks, also called cryptanalysis attacks, capitalize on the implementation defects inherent in any system.

Technological advances in application specific integrated circuit (ASIC) design have had a significant impact on the field of cryptography. The needs for increased speed, reduced area and reduced cost have resulted in customized implementations of cryptographic devices. However, even if the algorithms used are cryptographically secure, their implementation may be susceptible to hardware attacks. Furthermore, even if the hardware implementation is secure, it may still be susceptible to a number of side-channel attacks, an issue of much research. Of these side channel attacks, one specific concern is the power analysis attack, which uses power fluctuation information measured during operation to extract information out of a cryptographic device.

One way of securing the cryptographic systems against these attacks is through software countermeasures. However, it has been found that these software countermeasures are not completely effective since power leakage is largely dependent on the underlying logic architecture and cmos device characteristics [26]. As a result, there is a need to implement efficient hardware countermeasures against such attacks at logic architecture level.

To address this need, this thesis implements an efficient dual rail logic methodology for power analysis prevention at the logic architecture level. Chapter 2, discusses the basics of cryptography and power analysis attacks. While Chapter 3 explains the related work done in dual-rail logic, Chapter 4 details the implementation of dual rail logic on FPGA. Chapter 5 proposes the transmission gate-based logic structure. Chapter 6 summarizes results and conclusions.

CHAPTER TWO
BASICS OF CRYPTOGRAPHIC SYSTEMS

2.1 Background

Cryptographic algorithms and protocols form the basis for secure computing and communication systems. These algorithms are typically implemented as abstractions, resulting in a complete system that targets specific objectives. In such a system, each block is assumed to be independently secure, and so the final implementation is also assumed to be secure.

The algorithms and protocols are implemented either as software running on a processor or on custom hardware. It is assumed that the hardware used has little or no impact on the cryptographic security of the implemented system. In addition, the hardware is assumed to be a black box [3], since that the attacker can only look at signals as they enter or leave the system. Since internal computations are not revealed to the outside world, the security of the system is primarily dependent on the specific algorithm.

2.1.1 Algorithmic security parameters

Secrecy algorithms can be described as a transformation of one space (i.e. a set of possible messages) to another space (a set of possible cryptograms) [1]. Ideally these transformations are reversible such that they can be deciphered only with the correct decryption key.

The security of the algorithm is tested by analyzing the one used and its key manipulations. The following parameters are typically used to evaluate the strength of the secrecy system [1].

1. Key length

A cryptographic key is used to encrypt the plain text. Generally, a longer key length increases the strength of the implementation; however, it also increases the processing overhead. Thus the key length must be small enough to be practical while at the same time long enough to ensure security.

2. Secrecy level

The secrecy level of a security system is defined by the amount of partial information needed to compromise the entire system. For some secrecy algorithms, partial information is enough to decrypt the cryptogram. However, other systems based on only a partial compromise of information will not provide a unique solution for the cryptogram. In addition, any increase in overhead needed to break the system will ensure a higher the level of security.

Thus, it is needed for the key length to be small enough to be practical while at the same time long enough for the system to be secure. The key length and secrecy level for a typical system can be determined by the theoretical analysis.

2.2 Theoretical analysis

Theoretical analysis of secrecy systems arises from the basic postulates covered in detail by Shannon. His communication theory of secrecy systems and the underlying equations form basis of secrecy systems and cryptography [1]. Understanding of security requires the basic understanding of the mathematical aspects of secrecy systems.

The mathematical model of a cryptography system can be designed similar to the noisy communication system [1]. The message or the transmitted signal is modified through

statistical or mathematical properties by a statistically chosen key, which is accomplished through encryption. The result is the cryptogram which is analogous to the noisy signal. This cryptogram is then used to extract the original signal which can be equated to the extraction of the original message free of noise at the receiver. Because of these similarities, we can apply the same statistical techniques used in communication theory to build a mathematical model and analyze the security of the system. This model helps us in understanding the uncertainty associated with the deciphering of the cryptogram.

2.3 Mathematical model

Let us assume that there are finite message sequences - M_1, M_2 . . . M_n with apriory probabilities $P(M_1)$ $P(M_2)$ . . . P() resulting in finite number of cryptograms E_1, E_2.….E_m. Let T_i be the transformation function used. Then,

$$E = T_i(M)$$

The condition for a perfect secrecy system can be derived from Baye's theorem describing the relation between the a priori and posteriori probabilities associated with a cryptographic system. Baye's theorem is given by,

$$P_E(M) = \frac{P(M)P_M(E)}{P(E)}$$

Where,

P (M) = apriori probability of message M

$P_E(M)$ = a posteriori probability of message M if cryptogram E is intercepted.

$P_M(E)$ = Conditional probability of cryptogram E if message M is chosen

P (E) = Probability of obtaining cryptogram E.

The condition for perfect secrecy can be obtained by solving Baye's theorem with the condition that the conditional probability criterion $P_M(E) = P(E)$ for all E and M.

If $P_M(E) = P(E)$, then from Baye's law it is shown by Shannon [1] that,

$P_E(M) = P(M)$.

The encrypted message or the cryptogram is available for analysis and for decryption. The cryptanalysis attempts to evaluate a posteriori probability of the message M. In the case of side channel attack techniques, the side channel information intercepted will increase the posteriori probability of decrypting the message $P_E(M)$ and also the probability of obtaining the key used for the encryption.

2.4 Side Channel attack techniques

An implementation of a mathematically proven cryptographic algorithm does not necessarily guarantee the security of a system. Physical implementation presents the attacker with important information about the cryptographic information. Numerous side channel attack techniques have been defined employing the implementation dependent characteristics like time, frequency, power, radiation and acoustics.

Among these side channel attacks, power analysis attack and its variants pose the biggest challenge to the system. The following chapters will introduce the main side channel analysis techniques namely timing and power analysis attacks.

2.4.1 Timing attacks

Timing attacks were first described by Kocher in 1996 and is one of the basic side channel attacks [4]. Timing attacks are based on measuring the time taken by the system to

perform encryption or decryption processes. The information that is gathered can be further used to derive more information of the secret key used. Some of the specialized attacks exist for determining Diffe-Hellman exponents and RSA keys [5] [4]. On vulnerable systems, the attacks are inexpensive and only need cipher text to execute the attack.

Cryptosystems often have slightly different delays associated with processing of different inputs. This difference in the delays can be measured easily and then used to establish a statistical map. This statistical map can then be used to arrive at the probable inputs compared to the perfect secrecy system explained in the previous section. The reasons for these timing differences include the inherent differences arising from the t_{plh} (time taken for propagation from low to high) and t_{phl} (time taken for propagation from low to high) differences, rise time and fall time differences in logic devices, performance optimizations in software, conditional statements, RAM (Random Access Memory) and cache access hits, and processor instructions that are used. It has been shown by Kocher et al that the whole cryptographic key can be obtained from a vulnerable system with these variations in timing [1].

The timing variations are easy to measure and can be analyzed using standard statistical methods. The resultant correlation samples can be used to determine the key to certain accuracy. This method of attack is explained in detail by Kocher et al [4].

The attack can be classified as a signal detection problem [4]. The signal is the timing variation which is the desired data and the noise is the inaccuracies in measurement and timing variation due to unknown exponent bits. The properties of signal and noise determine the number of timing measurements that are needed to obtain the information.

Simple statistical variation functions can be used to predict the exponent bits. Generally, error correction techniques can reduce the number of samples that are needed to obtain the

exponent bits, at the cost of increased overhead [7]. Timing attacks can be minimized by having customized logic architecture in the critical path with equal timing variations for all input combinations. In addition to hardware countermeasures, software countermeasures are also possible.

2.4.2 Power analysis

Power analysis was first proposed by Kocher as a side channel attack technique in 1999 [6]. This side channel attack has been one of the most widely studied branches on side channel attacks. The basic idea behind the power analysis attack is similar to timing attacks and is based on the same framework described in Section 2.3 [4].

Most of the cryptographic devices involve a processor implementing the cryptographic library. The Complementary Metal Oxide Semiconductor (CMOS) transistor based processor and the peripherals are active devices and have dynamic power consumption. Power Analysis attacks make use of this principle and try to discover the cryptographic key by monitoring the power consumption and using statistical comparisons.

Of the two types of side channel attack techniques, power analysis attacks can be considered as a passive attack. In passive attack techniques, the information about the key is gathered without any physical damage to the system under attack as compared to active attacks in which some form of damage/tampering is done to the system in examination. Since passive side channel attacks do not leave traces of tampering, they are much more difficult to detect and prevent.

Kocher et al described two important variations of power analysis [6]. For both these techniques, we need to extract the power consumption of the chip during dynamic working conditions. Theoretically, the power consumption can be measured by recording the current

that is passing through the resistor connected between the power supply pin and the circuit [6].

There are two important variations of power analysis, simple power analysis (SPA) and differential power analysis (DPA).

1. Simple power analysis

SPA is a technique in which the power consumption of the circuit under analysis is observed and suitable conclusions can be made about the input transitions. Simple power analysis can also be used to determine the key information and also information about the arithmetic operations of a key.

SPA can is used to reveal a sequence of microprocessor instructions in the case of complete software implementation of the algorithms. The power consumption difference for a simple arithmetic operations and higher order computation blocks is noticeable provided the measurement systems have a good resolution. Thus, SPA can be used to identify permutations, comparisons, multipliers, and exponentiations involved. These operations give information about the algorithm and also some information about the inputs for the algorithms. In many public key algorithms involving mathematical operations, most common method of exponentiation is to use squaring and multiplication operations. For a time efficient implementation, squaring is implemented using a faster separate algorithm than general multiplication. The two algorithms have different power signatures thus enabling the ability to distinguish each operation in a SPA attack [9]. Similarly various DES implementations have different power consumptions during permutation and shifts [9]. Also, conditional branches are an additional source of significant power consumption differences.

Techniques for countering SPA can be easily implemented [10]. Since intermediate keys and branching are sources of SPA, they can be avoided as a countermeasure. In some cases, balanced branch implementation where the branching is absolutely necessary can be implemented. However, the microcode in some microprocessors can cause large operand-dependent power consumption features. For these systems, even constant execution path code can have serious SPA vulnerabilities.

Most of the hardware implementations of symmetric cryptographic algorithms have sufficiently small power consumption variations that such that SPA does not yield key material.

2. Differential power analysis

DPA builds on the Simple Power analysis by using statistical analysis of the power traces collected. In contrast to SPA attack, considerably more traces need to be obtained. While SPA is basically a visual inspection, DPA uses statistical analysis and error correlation techniques [9]. The basic idea of DPA stems from differential timing attacks. Similar statistical differences are used to obtain information as with the timing attacks. Since most of the hardware systems use CMOS devices as logic elements, the power consumption difference between the various transition states can be studied to obtain key information. In many ways, DPA attack is more powerful than the SPA attack.

The technique below gives the skeletal technique for DPA attack as given by [8] [9] [10]. Implementation of DPA consists of two phases - data collection and data analysis. Data collection for DPA can be done similar to SPA attack by collecting the power consumption traces. The following steps explain the technique. The technique is based on known plain text, and is categorized as plaintext attack.

Initially, N cryptographic operations are carried out and for each iteration, only the plaintext input is varied with the key kept constant. For each operation the corresponding power trace is obtained. The traces will be compiled into 2 dimensional arrays for simpler analysis. Let the power data be designated as P_{ij} where i and j are the indexes to the array. P_{ij} is the power value at point i for all time j. The average power is computed using average values at each point i, which is given by,

$$\overline{P_j} = \frac{1}{N}\sum_{i=1}^{N} P_{ij}$$

.

The next step is to select a target bit of the output. Let a bit position 'b' be the target bit. A key dependent function D (K_m, C) (where K_m is some key information of 'm' bits that are needed and C is the cipher text) is selected which is known to affect the bit position. A hypothesis value consisting of m bits is constructed. Once a hypothesis is constructed, the theoretical mean power trace of $\overline{P'_j}$ is obtained. This power trace can be of two categories. One for a bit value resulting in 1 and another resulting in 0. Let the mean curve of the second category be P^0. This trace is compared with the measured mean power curve obtained through the measurements ($\overline{P_j}$). If there is a difference in the curves more than the standard deviation allowed for the noise, then the chosen key bits are correct. Otherwise the hypothesis is termed as incorrect and a new hypothesis involving a different bit value is constructed and the steps are repeated. There are 2^m possible guesses need to be made for the worst case hypothesis.

If the hypothesis is proved to be right, then the next m bits of the key are used to construct the next hypothesis. This process is repeated until sufficient information about the key is gathered.

The DPA can be altered suitably for different cryptographic algorithms. For DES, the procedure concentrates on the first S box and the first 6 bits of the secret key [8]. It is seen that the public key algorithms have stronger power leakage information because of the selective use of crypto-processors. In general, most of the RSA implementations use the Chinese reminder theorem which is particularly vulnerable to DPA attacks. Even SPA can be used to obtain considerable information about the algorithm and operations [12].

Various countermeasures can be used against both SPA and DPA attacks. Many of the countermeasures make the DPA attack difficult to execute. These countermeasures and implementation details are discussed in next chapter.

One more reason that the differential power analysis (DPA) attack is effective is because the attack can reveal the key information with little or no information about the underlying algorithm [8] -- even when the algorithm used is proven to be secure.

CHAPTER THREE
RELATED WORK

3.1. Countermeasures for Differential Power Analysis

Various countermeasures have been proposed for DPA and SPA attacks. These countermeasures can be broadly classified into two categories.

1. Software countermeasures
2. Hardware countermeasures

3.1. 1. Software countermeasures

Most of the countermeasures that have been suggested address the software aspects of the cryptographic system. The software countermeasures can be of three types. The first approach is to reduce the signal size as explained by Kocher et al [6]. This technique attempts to counter the DPA attacks by implementing a constant execution path and balancing the hamming codes associated with the power traces [6]. The hamming weight between the power traces can be reduced by having each data and its complement combined. This has a disadvantage that it takes up double the number of storage registers. Also the register needs to be cleared after each write which increases the computational overhead [15]. Thus, even though these techniques make the DPA analysis tougher, an attacker with sufficiently large number of samples can still be able to carry out DPA attacks on the degraded signal [4]. The signal suppression can also be implemented by introducing a high amount of white noise into the system [19][4]. This method increases the number of samples needed for the attack. This effectively reduces the signal to noise ration of the

cryptographic operation. But the introduction of noise components can be easily countered by using various averaging techniques as described in [18].

Another software based countermeasure approach is to implement Random process interrupts (RPI). The idea behind this technique is the fact that power analysis attacks are possible because operations being attacked occur at a constant place in time [17]. The random interrupts can be of two ways. Either by introducing randomness into timing, or execution order [15]. This is sometimes implemented by introducing random delays using specialized hardware [21]. Timing shifts do not provide complete defense since an attacker can use statistical techniques, such as cross correlation, to realign the power traces. Randomizing execution order is implemented by randomly placing dummy instructions (NOP) between actual instructions. This, apart from randomizing the time for different encryptions, also causes the order of instruction to vary.

Apart from these, countermeasures are designed to mask the key bit used in the encryption process [22]. This method is based on the fact that most of the key symbols used in the public key cryptographic system can be mapped into a different subset. Similar countermeasures are also proposed for Koblitz curves like random rotation of the key and random insertion of redundant symbols (RIRS) [22].

To be useful, these randomizing techniques have to be extensively used to provide an effective countermeasure against various statistical averaging processes and other noise removal techniques [15].

Another form of countermeasure focuses on the instructions which are of critical importance regarding the algorithm implemented. In this technique, some instructions in the implementation are replaced by their secure counterparts such that there is no leakage of

information from their energy consumption [16]. This technique is implemented on a DES system.

In this technique, the secure instructions are concentrated on 4 important operations in DES encryption which are assignment operation, bitwise XOR, shift and indexing operations. All operations are not masked -- only the ones that uses the secret key and the data that is generated by previous secure operations [16].

Data Initial Permutation (L0, R0) = PermuteIP(data)	**Data Initial permutation** (L0, R0) = PermuteIP(data)
Key Permutation (C0, D0)= permuteK1 (key)	**Key permutation** (C0, D0)← permuteK1 (key)
Mth Round **Left side operation** Lm=Rm-1 **Mth key operation** Cm=Rotate (Cm-1,n) Dm=Rotate (Dm-1,n) Km=permutek2 (Cm,Dm) **Right side operation** ER=PermutekE(Rm-1) f(Rm-1,K)=S(E(R)(+)Km) Rm=Lm-1 (+) f(Rm-1,K)	**Mth Round** **Left side operation** Lm←Rm-1 **Mth key operation** Cm←Rotate (Cm-1,n) Dm←Rotate (Dm-1,n) Km←permutek2 (Cm,Dm) **Right side operation** ER←PermutekE(Rm-1) f(Rm-1,K) ←S(E(R)(+)Km) Rm←Lm-1 (+) f(Rm-1,K)
Output Inv Permutation Output = PermuteIP^{-1} (R16, L16))	**Output Inv permutation** Output = PermuteIP^{-1} (R16, L16))

Table 3.1 Secure DES implementation from the data obtained from [16].

From the table 3.1 it can be seen that the secure instructions are substituted only during the key permutation and operations in each round. The secure operations also include secure assignments replacing normal assignments [16]. The XOR instructions are implemented as complementary pre-charged circuits as shown in Fig 3.1.

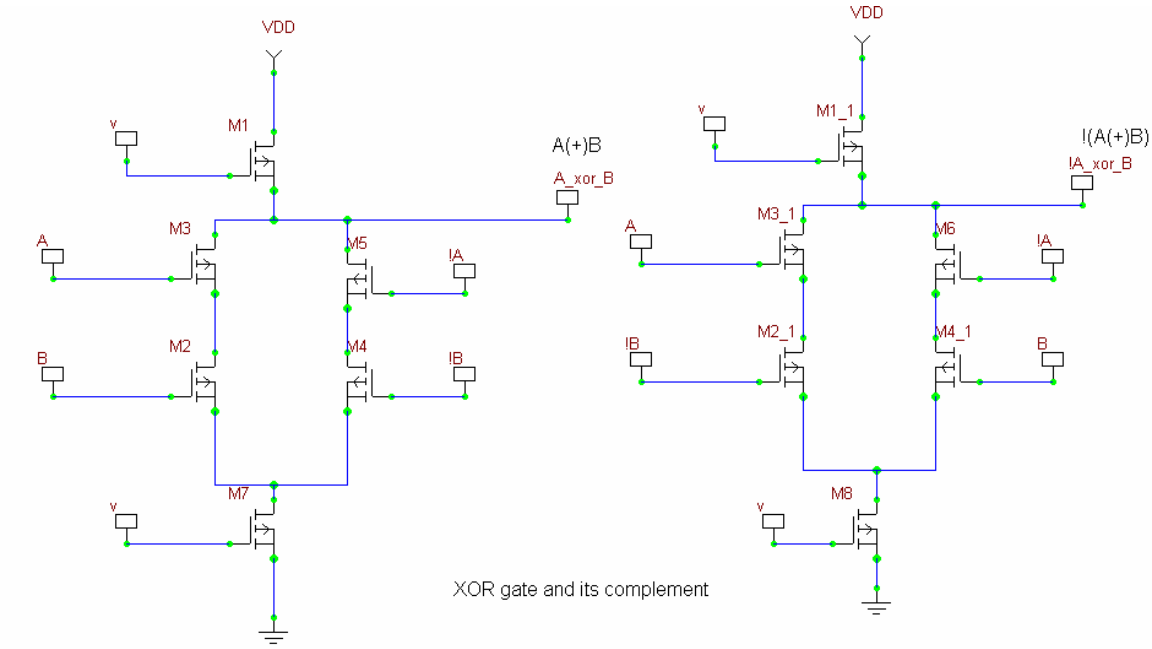

Fig 3.1 XOR gate and its complement [16]

3.1. 2. Hardware countermeasures.

In contrast to software countermeasures, hardware countermeasures concentrate on the gate and logic architecture to counter DPA attacks. This is based on the knowledge that the DPA attacks are feasible because of the leakage present in the hardware circuit. Thus the fundamental countermeasure has to be implemented in the hardware abstraction. Though this countermeasure is simple to understand, there are difficulties involved in the implementation. There are two types of hardware countermeasures. In the first approach, the power to the cryptographic chip is controlled. The second countermeasure involves the use of alternative logic architectures.

In the first approach, the power supply is switched to the circuit under test by the use of capacitors [23] or external chips [24]. In the approach by A.Shamir in [23], the power to the

system is switched using two capacitors C_1andC_2. Another method involves the use of cryptographic circuit to draw power at alternate cycles [24]. Any attempt to make the power consumed by a smart card absolutely uniform by changing the physical design is unsuccessful because sensitive digital oscilloscopes can capture any non-uniformity and then that data can be analyzed to reveal useful information. Also, the attempt to cause every instruction executed to switch the same number of gates is very unnatural and requires twice as much area and total power consumption, along with slowing down the operation of the system. Another method for controlling input power is to use an internal battery to remove the external power pins. This would keep attackers from being able to obtain power traces without tampering with the card. The downside is that batteries small enough to fit onto a chip are expensive and have a short life span. It would not be practical to have easy access to the battery because that would mean that a power trace could easily be obtained and then the usefulness of the approach is circumvented. To avoid having to replace the battery, it has been proposed to use rechargeable batteries.

The reader could be used to recharge them, but rechargeable batteries of this size cannot hold a charge for long amounts of time and thus would have to be recharged at the beginning of each use causing an unreasonable charging delay at each use [18]. Also, rechargeable batteries will wear out after relatively few charges and then must be replaced.

In the second approach, different logic architectures are used to counter power analysis attacks. The idea behind this is that the DPA attacks stem from the power characteristics of the logic circuits. The software countermeasures address the problem at the algorithmic level and the switching power models address the problem at the architecture and packaging level. Thus there is a need to implement a basic solution which can be implemented at the logic

and gate level which is the simplest solution to the problem. The solution can be applied to the whole circuit or to the part of the system which is susceptible to power analysis.

There are different solutions proposed as logic and gate level countermeasures. Of these the current mode logic and dual rail logic are important.

1. Current mode Logic (CML)

   In this type of logic, the circuit draws current continuously from the power supply irrespective of the signal transitions and its logic state determined from the path of the current [25]. This technique has constant power consumption and perfectly draws current from the power supply irrespective of the input and output transitions [3]. It has been found that the Dynamic current mode logic (DCML) is faster and has low communication noise. But the main drawback is that they suffer from static current consumption [3] [25]. In order to implement this technique, special circuit techniques to minimize channel length modulation has to be implemented. This makes it difficult for the embedded processors to use DMCL as the basic logic gate.

2. Dual rail logic

Dual rail logic also called as dynamic and differential logic is based on the concept that the following conditions have to be met for a secure circuit.

1. Constant power consumption for all transitions
2. Constant load capacitance

This can be achieved by combining the differential and dynamic logic architectures. This architecture charges the capacitance for all the four possible logic transitions (0-0,0-1, 1-0 and 1-1). The differential logic masks the input value. Power is consumed irrespective of the input transitions. However there is still a possibility of differentiating the two main classes.

The 0-1 and 1-0 transitions consume power whereas 1-1 and 0-0 do not. Thus the algorithmic methods making use of Hamming weight balancing techniques is not entirely successful against DPA attacks [26].

The dynamic logic breaks the input sequence so that there is no difference between the 1-0 and 1-1 transitions. Independent of the input transitions, power is consumed only when the load capacitance charges.

This logic architecture requires about twice as much space and energy than a standard CMOS implementation [26]. Another issue is that since SABL has the pre-charge element of dynamic logic, fan-out issues limit the number of gates that can be cascaded together. Cascading has to be done by either inserting an inverter between each gate or by alternating gates with *n* pull-down networks with gates having *p* pull-up networks. The use of inverters limit the number of gates that can be cascaded together to the number that can be evaluated in one clock period, and only non-inverting logic can be used. The second alternative eliminates the disadvantages of the first, but the speed of the circuit will be degraded because of the mobility of holes in the *p* transistor gates. Finally, the use dynamic circuits requires extra design effort to ensure correct operation under all circuit conditions including timing sequences, charge leakage, and noise sensitivity .

Wave dynamic differential logic (WDDL) overcomes the issue of dynamic logic gates by using static CMOS gates to implement the differential logic. This has an advantage of secure logic circuit comparable to the SDDL logic but has a reduced power signature [3].

The basic structure of a WDDL gate shown in Fig 3.2 consists of two positive complementary gates. One of the gates computes the correct logic where as the second complementary gate implements the complement of the output. The complementary gate computes the false output by using the complementary gate and the complementary inputs.

The proof for this can be given by De Morgan's theorem as below. Let us assume that A and B are the inputs and $\overline{A}$ and $\overline{B}$ be the inputs to the compound gate. The outputs are denoted by Y and $\overline{Y}$.

AND operation:

$$Y = A \bullet B$$

$$\overline{Y} = \overline{A \bullet B}$$

$$\overline{Y} = \overline{A} + \overline{B}$$

Similarly, a WDDL OR gate can be realized as a combination OR-AND gates.

Thus any logic can be implemented as a combination of AND, OR and inverters. The inverted output can be further used for the next cascading logic block. The fig shows the implementation of WDDL gates (as a derivation of basic SDDL gate) using static CMOS gates.

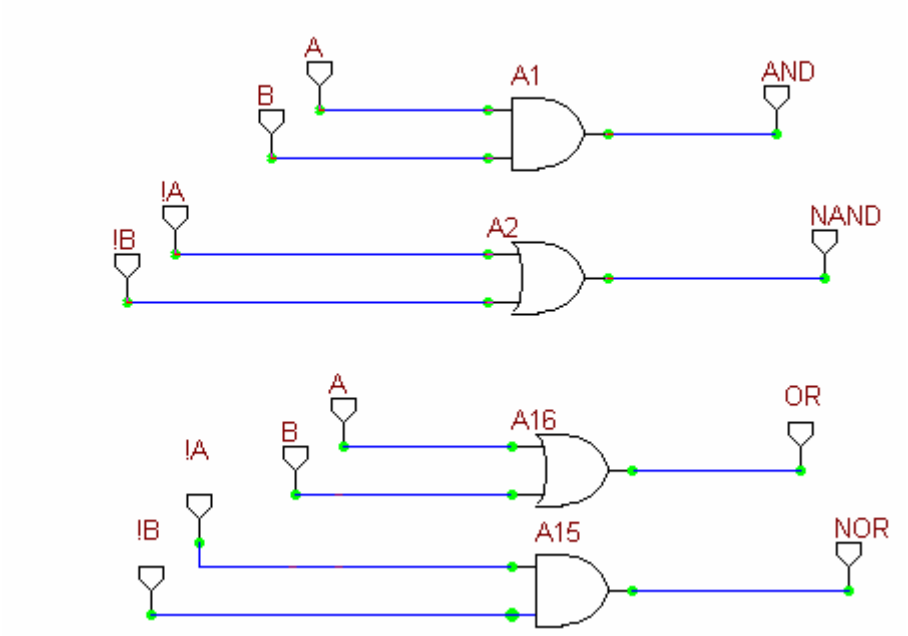

Fig 3.2 Dual rail implementation from the data obtained from [3].

Both timing and value of the inputs influence the number of switching events. SDDL can never achieve input signal independent power consumption. Restricting the problem to the conception of a secure version of the and OR operator resolves this.

Wave dynamic differential logic gates can be constructed by connecting these SDDL gates together keeping the complementary structure intact.

The dual rail logic consists of a compound cell with both the true and complementary logic. The use of static CMOS logic in WDDL instead of NMOS logic provides an advantage for both ASIC and FPGA implementation. Universal gates are realized based on the dual rail principle.

In verilog, NOR and NAND gate primitives are used to generate the universal gate logic. In order to have a uniform implementation, all the primitives used were two input edge triggered logic gate instantiations with default rise and fall times.

The truth-table for the secure NOR gate is similar to the NOR gate but has both the normal and complementary outputs and the outputs are valid only when the pre-charge input is high.

A	B	Pre-charge	NOR	OR
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	0	1
X	X	0	0	0

Table 3.2 Truth table for secure dual rail logic comparable with data obtained from [3].

These universal gates were implemented on Xilinx Spartan 2E board. These modules were then used in the development of an exponentiation module using verilog. For the verilog fpga implementation, Xilinx Spartan 2E boards were chosen and the Verilog implementations were tested on both Icarus compiler [29] and also on the Xilinx ® ISE 8.1i [30].

CHAPTER FOUR
FPGA IMPLEMENTATION

This chapter discusses the implementation of a secure multiplier circuit on an FPGA (Field Programmable Gate Array) core. As an evaluation circuit, an exponentiation circuit using the basic gates and secure gates was implemented on Xilinx Spartan 2E FPGA board. The implementation consists of a top level sequential module and a multiplier designed using generic and secure methodologies. The evaluation circuits are then compared to obtain the area consumption results.

4.1 Evaluation Circuit

The multiplier is an 8x8 bit multiplier with AND encoded input bits. This multiplier architecture is a carry save array multiplier based on a 16X16 bit multiplier described in ISCAS [14]. The multiplier uses AND gates to generate the inputs to adder blocks and the carry is propagated to the next stages.

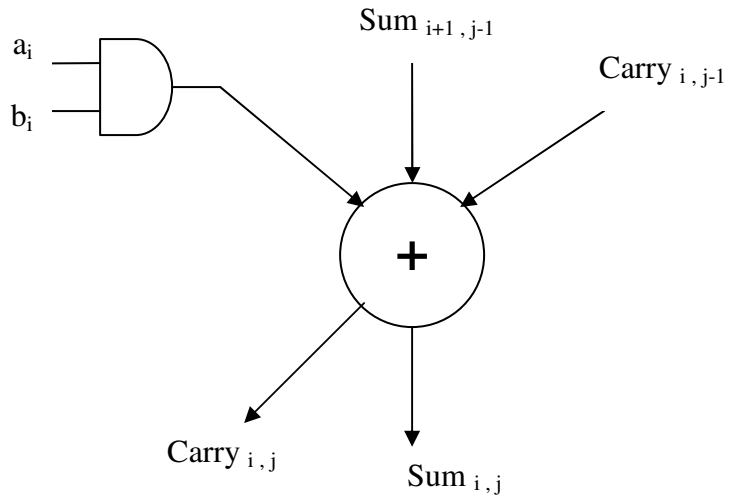

Fig 4.1 Adder implementation.

Here the inputs are denoted by a_i, b_i. The inputs are initially given to an AND gate as shown in fig 4.1 which are then fed to the adder circuit. The first level of the multiplier is a half adder block and the rest are full adders. The complete architectural diagram of the multiplier is shown in Fig 4,2.

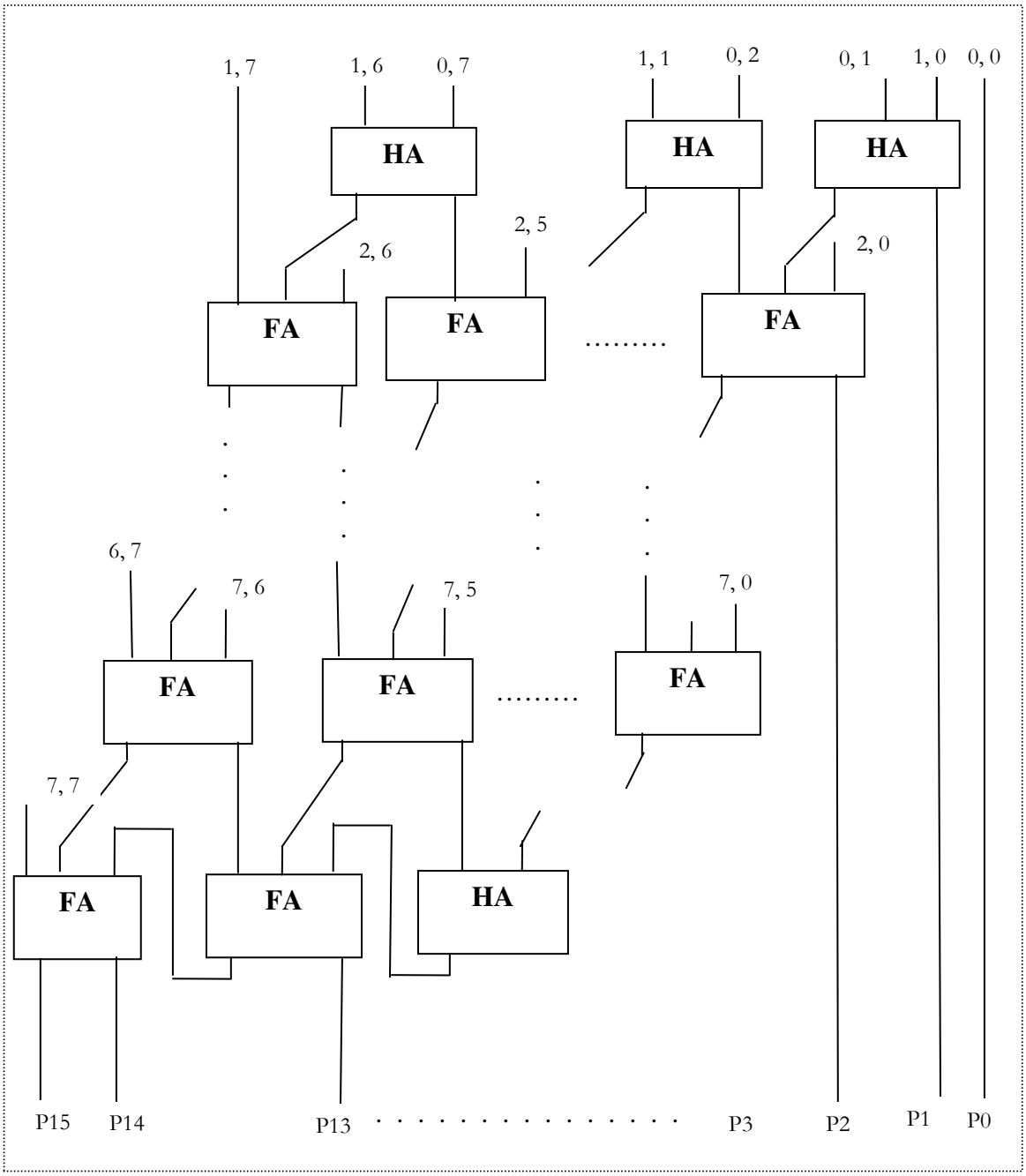

Fig 4.2 Multiplier architecture.

4.1.1 Exponentiation design

The exponentiation circuit is comprised of the structural multiplier component and a sequential control unit which implements the binary exponentiation algorithm by passing on the inputs to the multiplier blocks as shown in fig 4.3.

In this project, two instances of the multiplier are used. One of the multipliers is for normal multiplication, and another is used for the squaring operation. The sequential control is implemented as a state machine with three prominent states.

1. Initialization

The multiplier blocks are initialized with the appropriate input values such that the squared output is the input and multiplier output is 1.

2. Square and multiply

The inputs to the Square blocks and multiplier block are passed on depending on the exponent bit 1 or 0.

3. Result stage

Once all the exponent bits are shifted out, the result is sent to the output port.

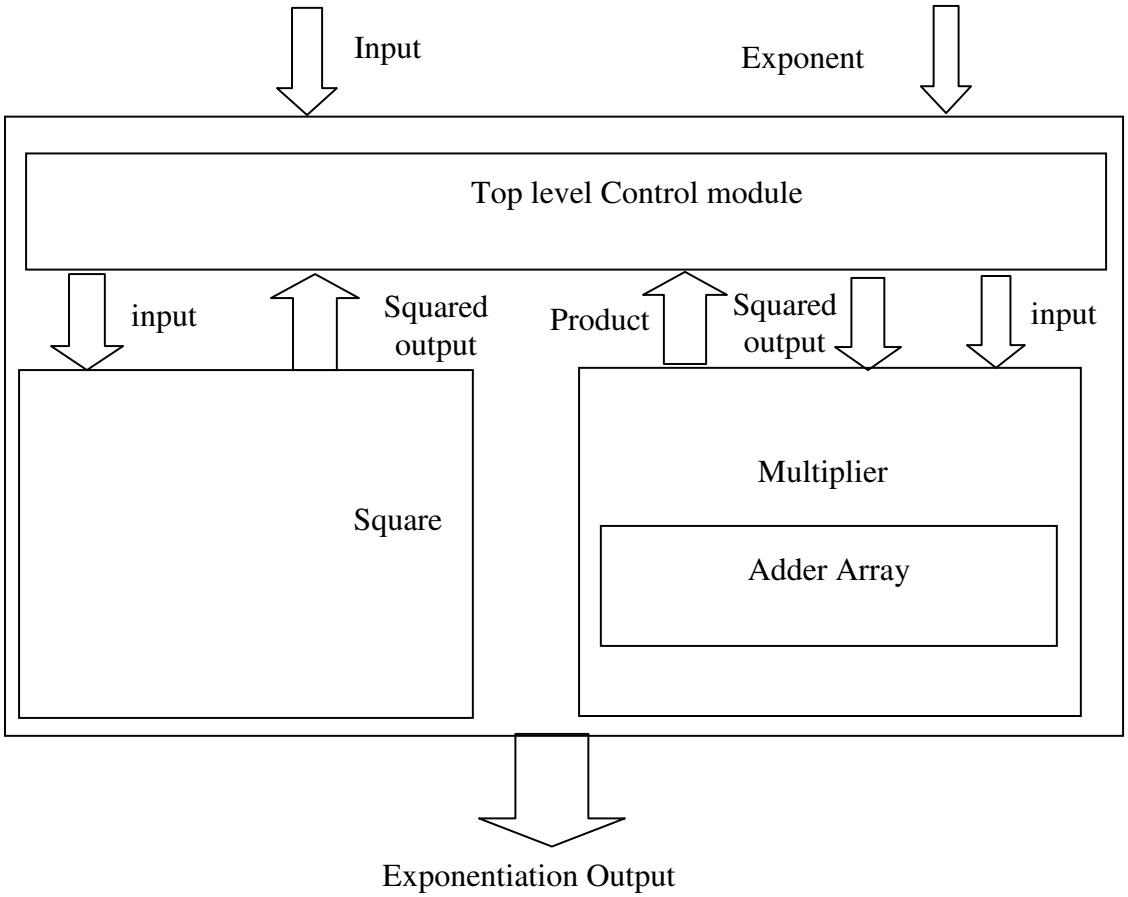

Fig 4.3 Exponentiation block diagram.

The exponentiation output is of the form modulo 2^8. This is because only the lower 8 bits of the multiplier and square modules are used for the next cycle of exponentiation. And the reminder is nothing but the last 8 bits. The division module can be used to derive reminders for values which are not a power of 2.

4.1.2 Test-bench architecture

The text bench is designed to pass on the inputs to the differentiator circuit which has been instantiated as the unit under test. The test bench consists of an initial block and an always block. The initial block initializes the inputs and the control inputs. Appropriate control signals and clock are passed. The clock generation is done using an *always* block, which inverts the clock signal every specified period of time. In this implementation, the clock period is 100ns. This is to make sure that the multiplier output is available before the start of next clock cycle. The output of the execution of this test bench can be observed via display commands or can be stored as a value change dump (vcd) output file to be viewed through a wave display tool.

4.1.3 Simulation

In Xilinx Verilog ISE simulation environment, once the project with the desired name is created, the Verilog module is added to the sources list. This Verilog code is checked for syntax in the synthesis stage. Once the synthesis is complete, the circuit is implemented. The implementation stage consists of floor-planning and placing and routing.

After successful run of synthesis and Implement stages, the test bench added as another source to the project is checked for syntax and a post place and route simulation is done to obtain the result display.

Of the complete exponentiation circuit, we are interested in the structural component of the exponentiation which was implemented using static CMOS and WDDL. The top level behavioral part just passes on the inputs to the multiplier. This behavioral module cannot be considered as the implementation is natively handled by Xilinx. The FPGA implementation

gives us a fair idea of the area needed for the static and WDDL implementations for the core components in cryptographic chips.

The size consideration in Xilinx ISE can be estimated using the number of LUTs and the slice flip-flops needed for the implementation. The Fig 4.4 shows the utilization information for the implementations.

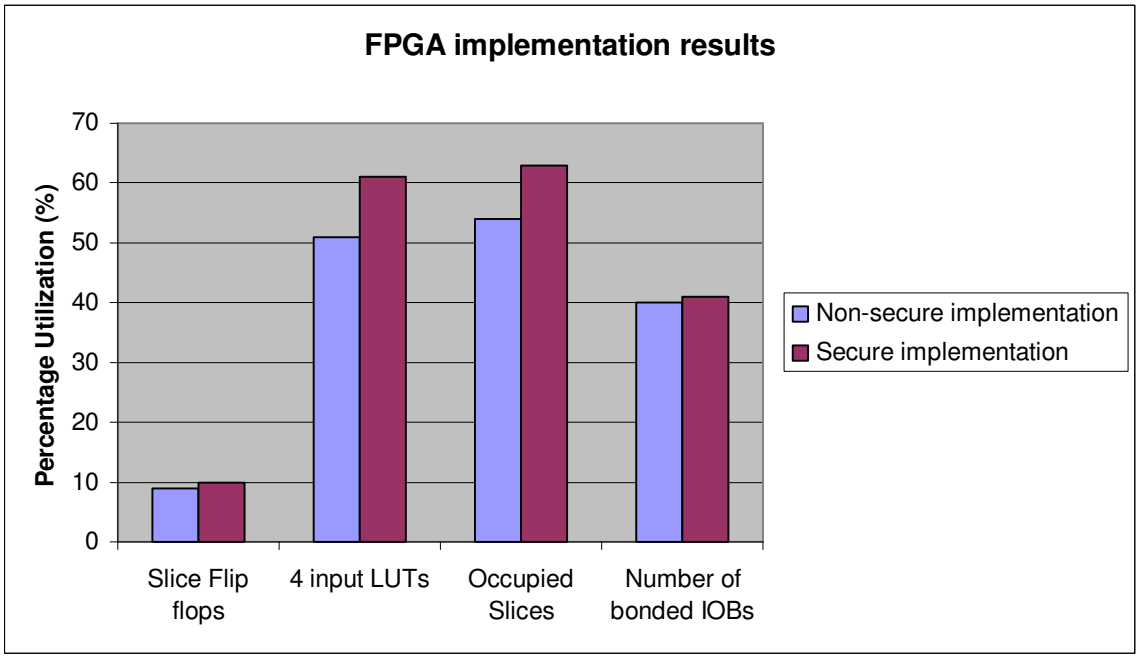

Fig 4.4 Plot of FPGA utilizations.

A comparison of the size is done for both non secure and secure implementations. The comparison is based on the actual structural logic and also the number of LUTs needed for the implementation. It can be seen that even though the Verilog implementation used doubles the number of gates instantiated, the increase in the logic utilization is marginal. This is because Xilinx auto routing and mapping will automatically distribute the logic uniformly. Hence the actual on-chip utilization of the secure implementation is about 10% more than the non secure implementation.

CHAPTER 5
WDDL IMPLEMENTATION USING TRANASMISSION GATES

It is seen that the WDDL implementation of the basic gates takes up almost double the amount of gates compared to the insecure implementation [26]. This increase in the number of gates also increases the area of the cryptographic chip which might be a disadvantage. This increase in area prompts the use of alternative gate level logic architecture to replace the static gate architecture. In this section, we will introduce the application of transmission gate wave pipelined architecture [28] to replace WDDL.

5.1 CMOS Transmission gate

The transmission gate is one of the most commonly used structures for most of the multiplexer designs and also as logic structures. The basic transmission gate consists of a pMOS and nMOS transistors connected in parallel as shown in fig 5.1.

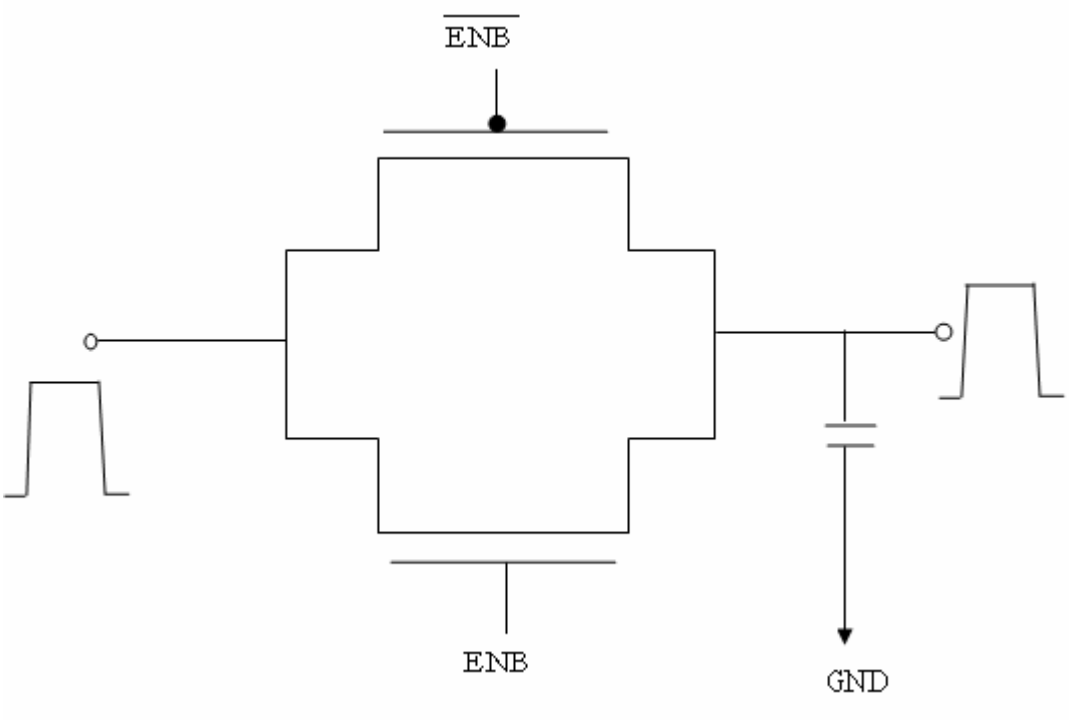

Fig 5.1 Transmission gate as switching device.

The control signal is applied to the gate of the nMOS and the complement $\overline{ENB}$ is applied to the gate of the pMOS. The bulk of the nMOS and pMOS are connected to ground and VDD respectively. The CMOS transmission gate acts as a bi-directional switch between the input and the output [27]. The operation of the transmission gate can be explained by considering the characteristics of nMOS and pMOS separately.

The nMOS transistor operation can be analyzed by removing the pMOS from the circuit shown in Fig 5.1. With ENB = 0, the output voltage is 0 irrespective of the input voltage. When ENB is 1 and input is high, the nMOS begins to conduct and charges the capacitor to V_{DD}. When the output voltage approaches $V_{DD} - V_{tn}$, the nMoS begins to turn off. Thus the transmission of logic 1 is degraded. With input voltage low, the nMOS transistor begins to conduct and discharges the output to V_{SS}.

The pMOS acts in a complementary approach to the nMOS pass transistor. Here, the logic level 1 is not degraded but logic 1 is not transmitted accordingly. Table 5.1 summarizes the characteristics.

Device	**Transmission of 1**	**Transmission of 0**
NMOS	Poor	Good
PMOS	Good	Poor

Table 5.1 Transmission gate characteristics from data obtained from [13].

The ON resistance of the transmission gate for the input between Vtn and Vtp is constant. The ON resistance is as shown in Fig 5.2.

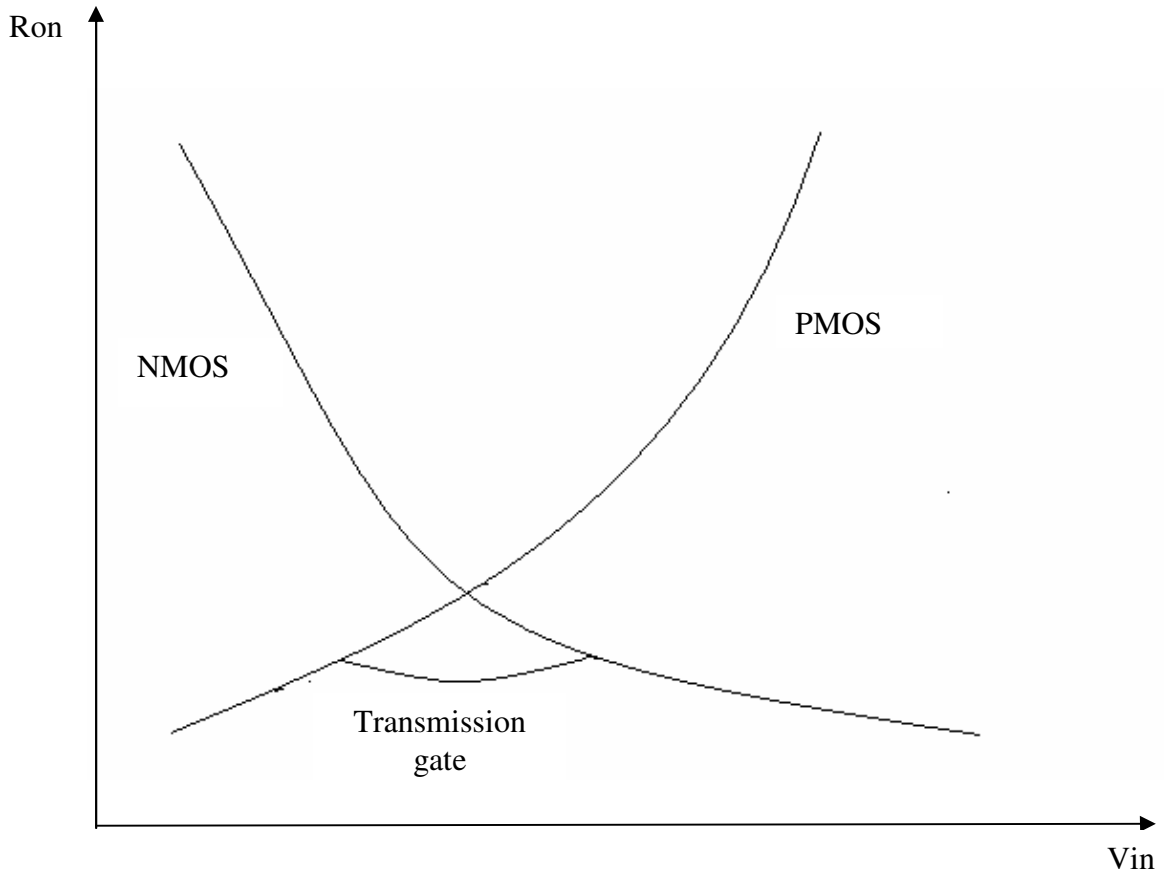

Fig 5.2 ON resistance of CMOS transmission gate [From data obtained from [13].

Because of the balanced nature, steady ON resistance and the use of complementary enable signals, the transmission gate based logic can be used to realize faster circuits with less area overhead.

5.2 Dual rail implementation

The transmission gate based wave pipelined structures have been used to generate high speed digital systems with considerable speed and area reductions [28]. The implementation of CMOS transmission gates in this thesis uses the circuit structure as a solution for DPA resistant logic architecture.

The transmission gate based logic can be developed using the multiplexer based logic structure. Fig 5.3 shows the implementation of the basic gates using Transmission gates.

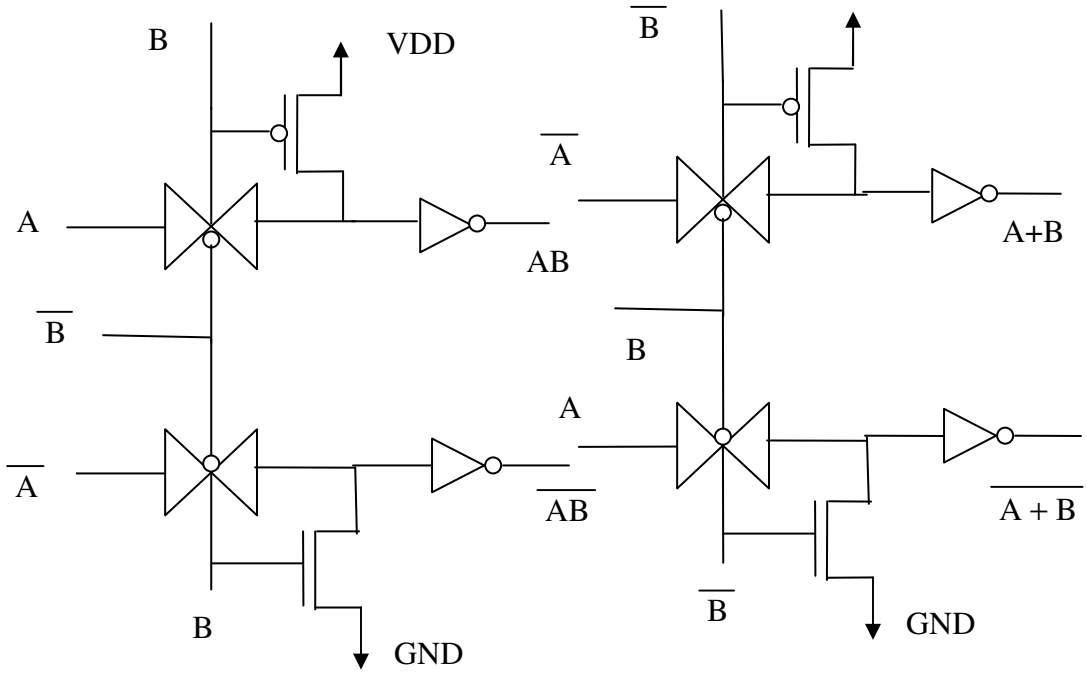

Fig 5.3 AND-NAND and OR-NOR implementation using transmission gates from data obtained from [28].

The transmission gate based structure is similar to the WDDL logic, but instead of the complementary inputs provided to the false output gate, only one complementary input is provided to the false gate. The other complementary gate is given for both the true and false gates. Also the AND-OR gate have the same architecture. It is only the inputs are exchanged to get the OR-NOR implementation.

Unlike the generic CMOS implementation of WDDL, the transmission gate architecture uses multiplexer based XOR gate. In static CMOS, the XOR gate is implemented using either basic gates or universal gates. Using basic gates, XOR can be implemented using the relation XOR = $\overline{A}\,B + A\,\overline{B}$.

The WDDL XOR implementation using static CMOS and Transmission gates is shown Fig 5.4.

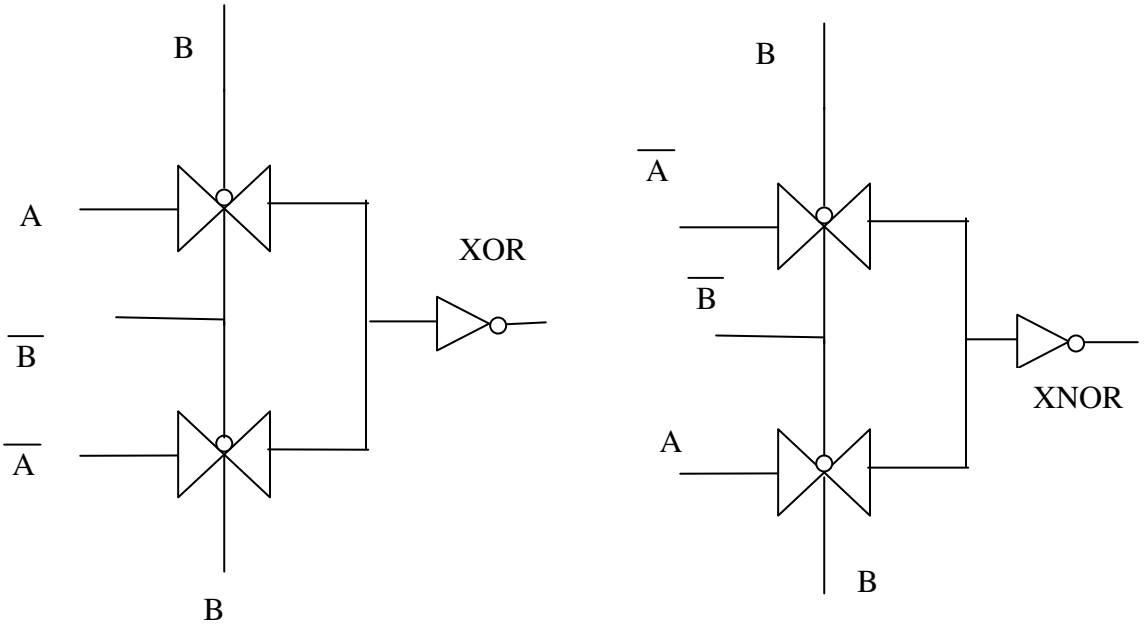

Fig 5.4 WDDL XOR implementations using transmission gates from data obtained from [28].

5.3 Reference implementation and comparisons

The transmission gate logic was used to implement a set of full adders which are the basic modules of the multiplier and the exponentiation circuit. Full adders were chosen because of the simplicity in comparing it with the static CMOS WDDL implementation. The number of gates for the adder is sufficient enough to establish the secureness of the circuit as they will be repeatedly used in digital logic circuit. Also the basic gates are compared thus making sure that any logic implemented using these gates are secure, thus the complete circuit. The implementation of these circuits was done in B^2SPICE v5 [11] simulator. The Bsim3v3 model MOS transistor library was used for all the implementations on B^2SPICE.

The initial inputs are generated using non ideal voltage sources with the input signals having a rise time of 10ns and a fall time of 10ns. The CMOS devices are implemented using 0.35um technology node. The transistor parameters are obtained from MOSIS website. The pMOS – nMOS ration for the static gates was chosen as 2.22 and for the transmission gate based design, the ratio was chosen to be 1.7. These values of the ratios are optimized for maximum speed, equal t_{plh} and t_{phl}. For the optimization process, the optimum fan-out value was determined to be 2. The typical parameters used for the MOS are given in Table 5.2.

Model = Bsim v3
Level 8
Capmod 2
Mobmod 1
Tox =7.6e-9
Xj=1e-7
Vth0 = -0.66(PMOS)
Vth0 = 0.51 (nmos)

Table 5.2 CMOS device parameters used.

5.3.1 Comparison of basic gates

The current consumption plot for the adder using WDDL and transmission gate is shown in Fig 5.5 and in Fig. 5.6 respectively. It can be seen that the transition currents for the 1-0 and 0-1 are similar for both forms of the secure circuit.

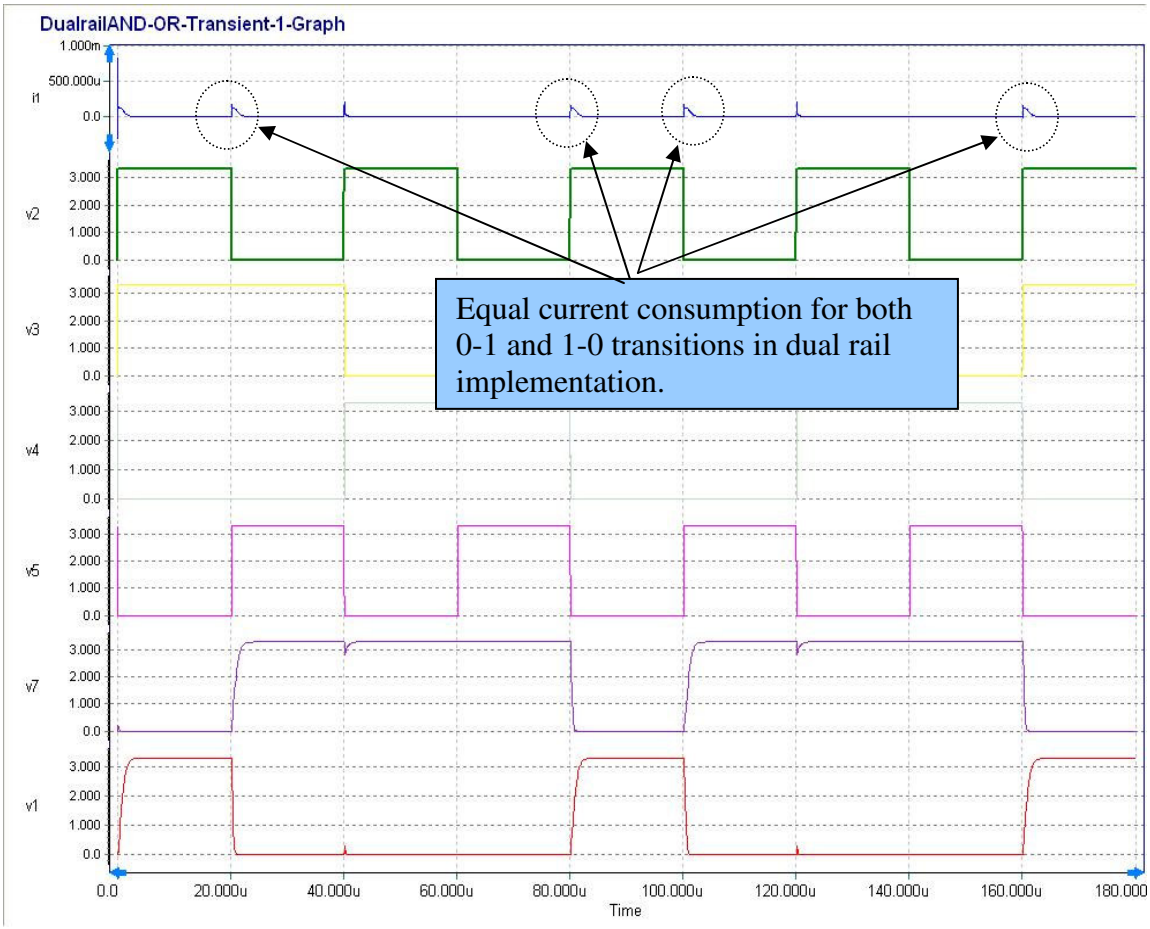

Fig 5.5 Current characteristics of AND-OR gate in WDDL.

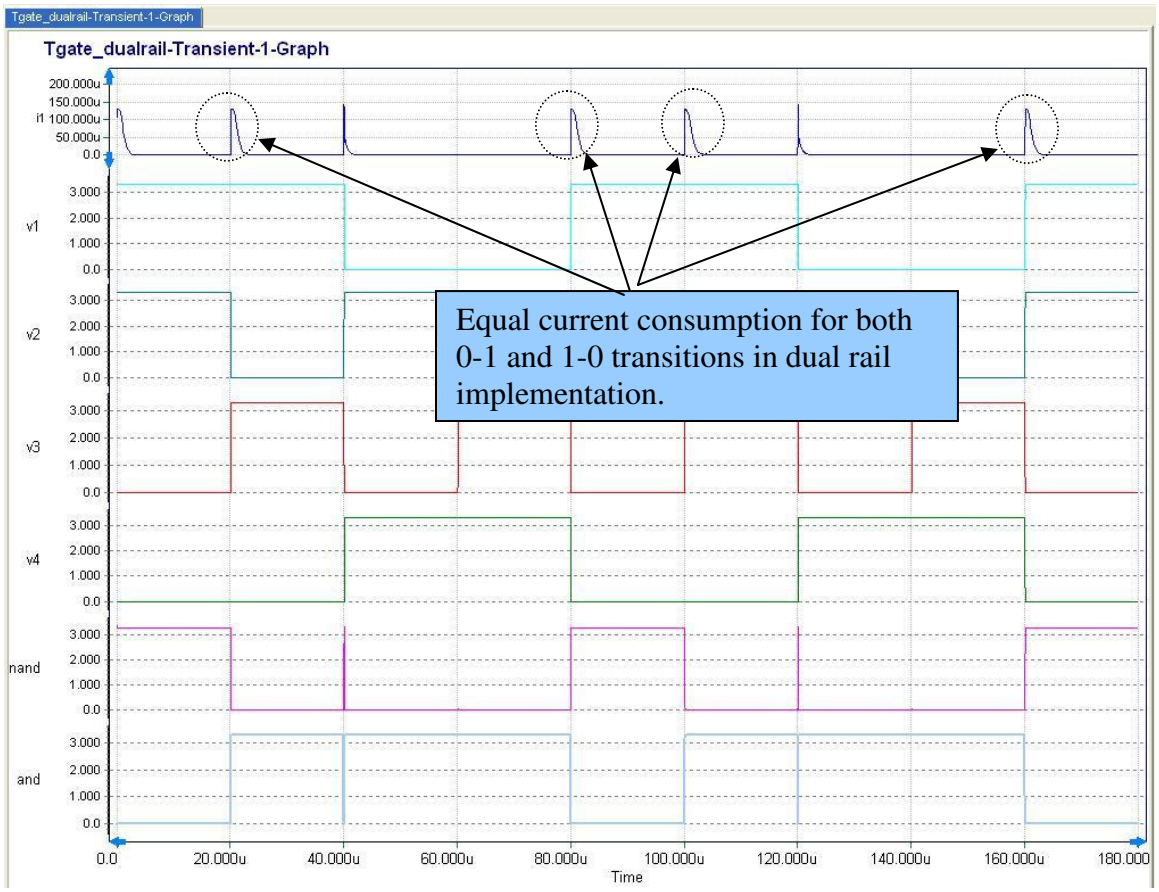

Fig 5.6 AND –OR characteristics for Transmission Gate wave pipeline.

The current characteristics plot for the non secure implementation in Fig. 5.7 shows that the current consumption is not symmetric for the 1-0 and 0-1 transitions.

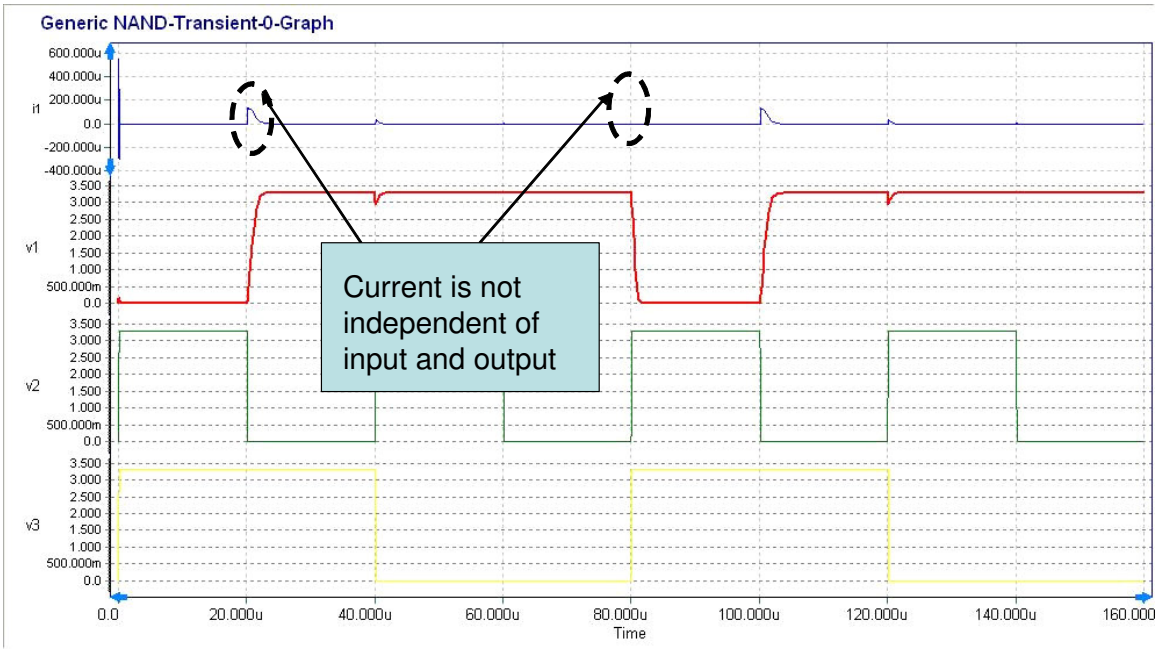

Fig 5.7 Non secure current characteristics.

It can be seen from the visual observation that during 1-0 transition, a significant amount of current is drawn from the power supply. This is needed to charge the output capacitors for output high. But no current is drawn during the 0-1 transition which makes the circuit vulnerable to both simple and differential power analysis attacks. In order to implement larger circuits using the basic secure gates, a simple design methodology can be used.

5.3.2 Secure design Methodology

The steps used for the secure design methodology are given below

1. The desired circuit is realized using generic CMOS gate level schematic.
2. Each generic gate is replaced by its compound gate equivalent. For both WDDL and transmission gate based WDDL, the compound gate consists if AND-OR and OR-AND pairs. While the generic WDDL uses AND-OR pair to realize the XOR

implementation, transmission gate uses the balanced XOR implementation given in Fig 5.4.

3. The gate level circuit is then realized using the NMOS and PMOS devices

5.3.3 Comparison of full adders

Using the secure design methodology presented in section 5.4, adder structures were simulated in both generic WDDL and transmission gate WDDL architectures in B^2SPICE simulation environment. The schematic for generic non-secure full adder is shown in Fig 5.8.

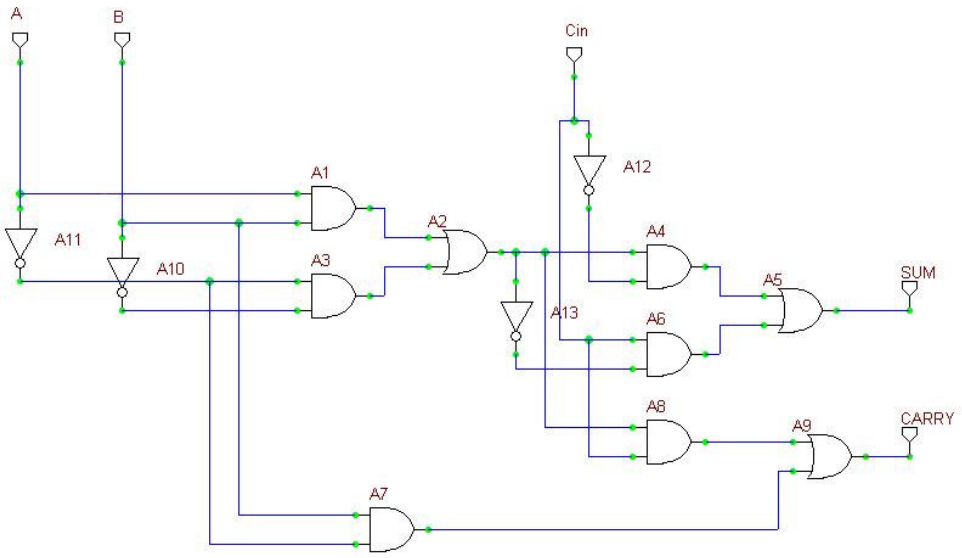

Fig 5.8 Full adder schematic.

The XOR and XNOR are implemented using generic CMOS gates. The design methodology given in section 5.5 was followed to arrive at the logic implementations. For transmission

gate based WDDL implementation, XOR implementation given in Fig 5.4 was used. The full adder sum implementation using the wave pipelined transmission gates is shown in Fig 5.9

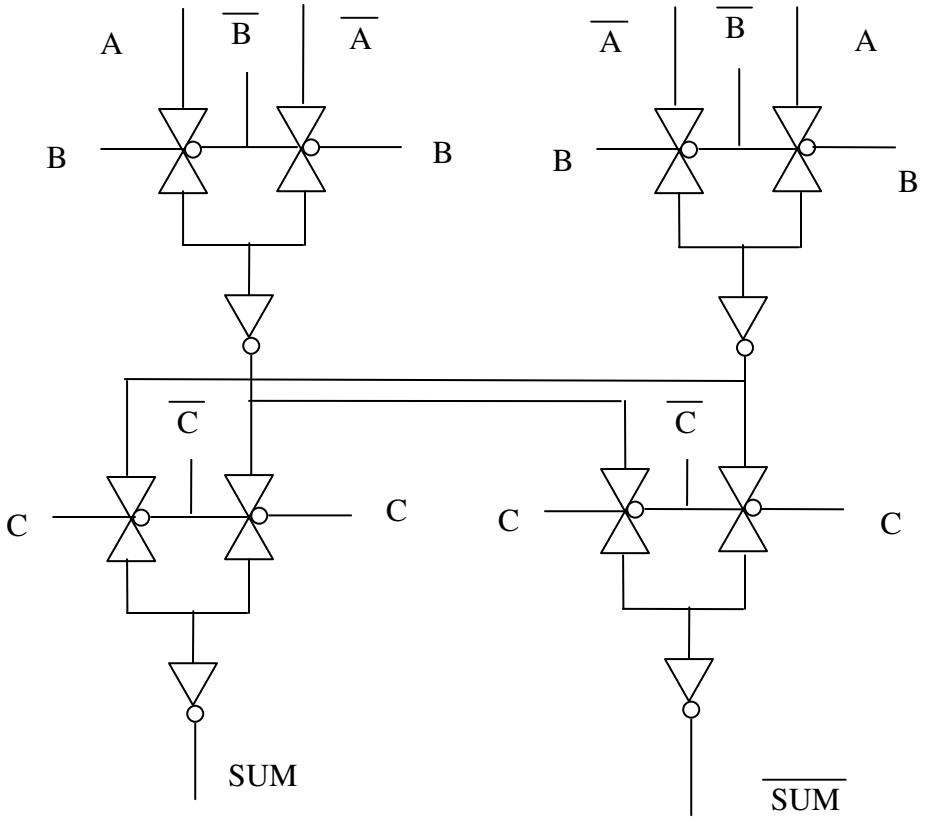

Fig 5.9 SUM implementation using Wave Pipelined Transmission Logic.

The carry is implemented in the similar way using the basic AND–NAND gates and OR-NOR gates.

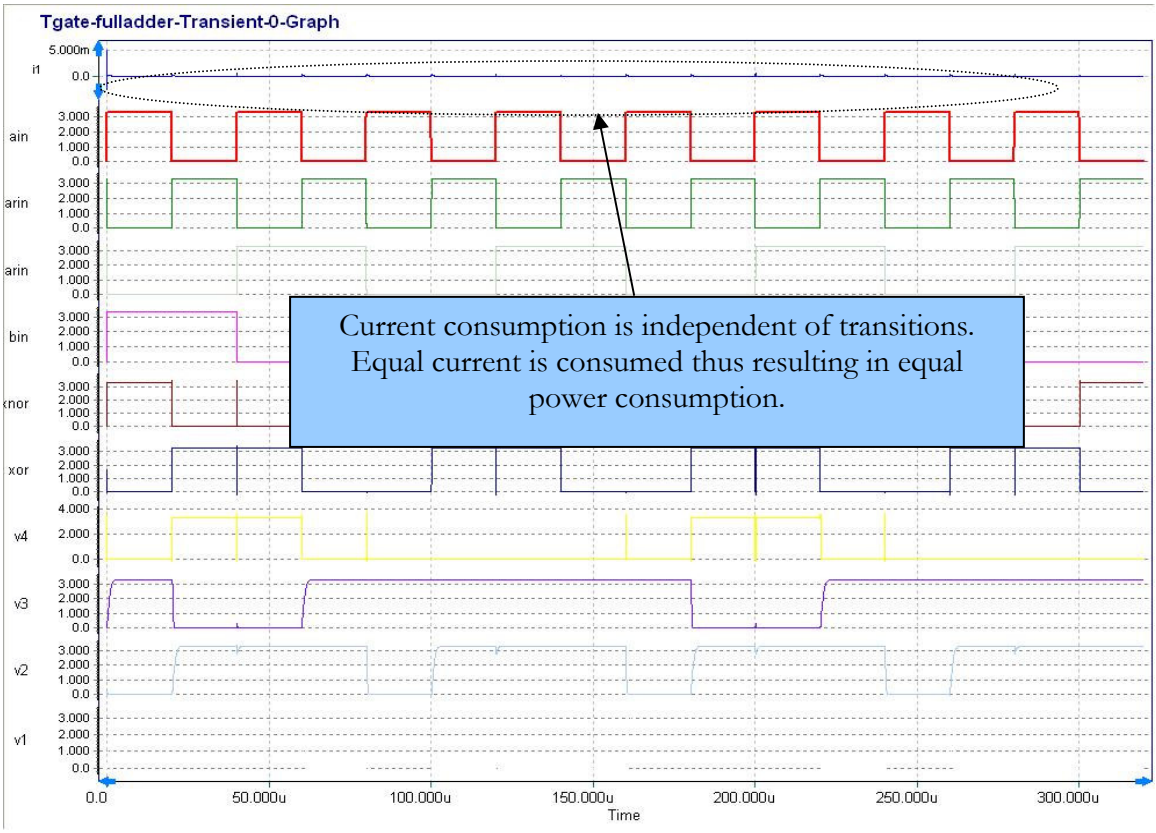

Fig 5.10 Current characteristics of transmission gate dual rail full-adder implementation.

The transmission gate current characteristics for the transitions are given in the Table 5.3. The peak current occurs at the point where the input logic changes the state. The peak current is denoted by Δi. The power consumption is determined by the time taken for the output capacitors to charge or discharge. The time taken for the transmission gate dual rail logic is also measured and is denoted by Δt. These measurements are done for each transition in the cycle.

Transition/ Parameter	1	2	3	4	5	6	7	8
Δi (uA)	130.339	130.323	130.054	130.336	130.009	130.052	130.406	130.199
Δt (us)	4.008	4.053	4.026	3.956	3.93	4.015	4.039	4.007

Table 5.3. Dynamic current characteristics in one cycle.

5.3.4 Statistical comparison

The statistical comparison can be done by determining the dynamic current consumption during the transitions. The power dissipation is not considered as this will be averaged out during the DPA analysis. The dynamic current and normalized deviation (represented by ND) is given by,

$$\text{NED} = \frac{\max - \sigma}{\max}$$

Where,

max = maximum value obtained (In this case current)

σ = Average value of dynamic current

The statistical analysis was done for both generic CMOS based WDDL and transmission gate based WDDL. The dynamic current and the normalized deviation for the implementations are shown in fig 5.11.

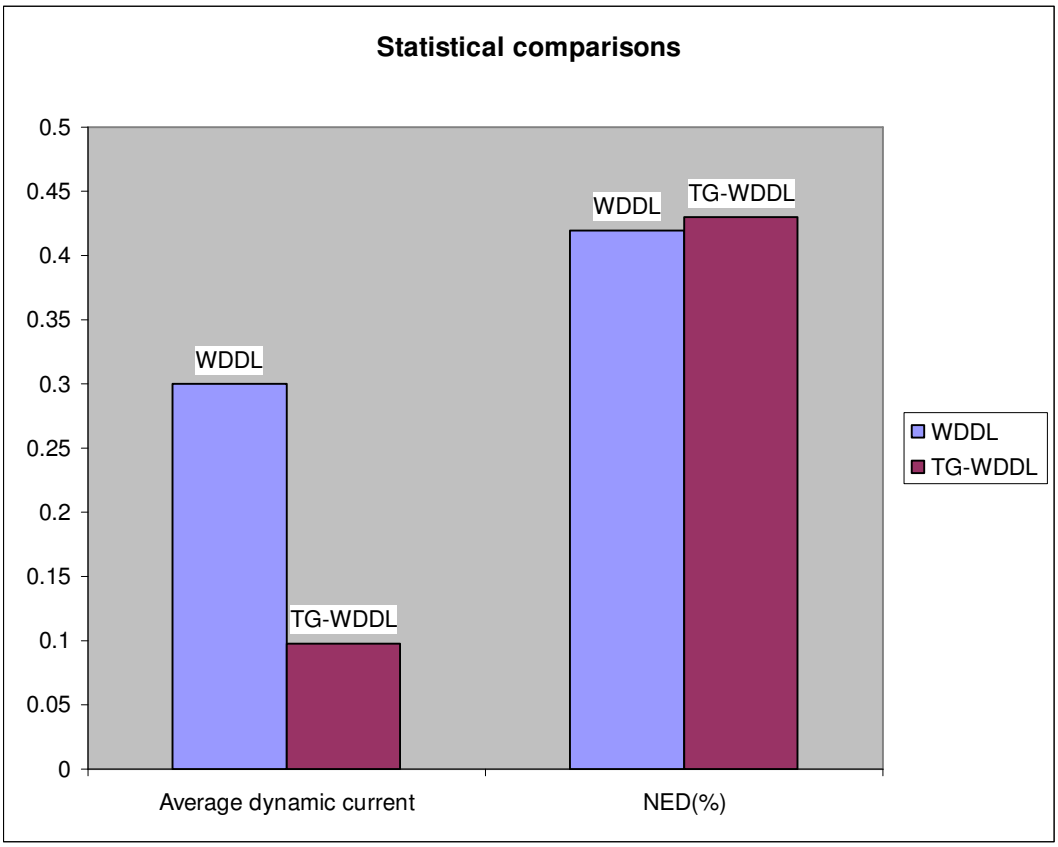

Fig 5.11 Statistical comparison.

5.3.5 Size comparison

The table below shows the number of gates required for the implementation of the logic circuits using normal, WDDL and TG WDDL. Table 5.4 compares the number of gates taken for the implementations.

Logic Implementation	AND-OR	XOR	FULL ADDER
Generic CMOS	6 transistors for each gate	18 MOS transistors (6 transistors for area optimized logic)	54 MOS transistors (26 transistors for area optimized circuit)
WDDL	12 transistors	36 transistors (12 transistors for optimized XOR-XNOR pair)	60 transistors
TG-WDDL	10 transistors	12 transistors	54 transistors

Table 5.4 Comparison of gates for different logic implementations.

It can be seen from the count for the number of transistors required for transmission gate based circuit is half that is needed for WDDL implementation using basic gates. For an optimized adder implementation, the reduction is about 10%.

CHAPTER SIX
CONCLUSIONS

The thesis research included the implementation and testing of transmission gate wave pipelined circuits as possible countermeasures to the power analysis attacks. The comparison of the transmission gate circuit with the static CMOS (Complementary metal oxide semiconductor) implementation showed that the transmission gate based wave pipelined circuits have a great potential as a hardware countermeasure. It is seen that the number of gates needed for the secure implementations in static CMOS gates require almost double the number required for non-secure implementation. The transmission gate based design reduces the transistor count by about 10% for optimized implementation and 50% for un-optimized adder implementations. The reduced number of transistors also means that the amounts of routing and metallization requirements are reduced. This reduction will considerably decrease the area of the secure cryptographic implementation.

The transmission gate based design is modular and can be easily implemented using existing methodologies for CMOS design. The gate and the architecture is still the CMOS implementation which is relevant and current proven technology. This implementation is thus suited for large scale ASIC (Application specific integrated circuit) design compared to pseudo-NMOS or domino logic previously proposed. The transmission gate based circuits can be easily interfaced with static CMOS logic thus eliminating the need for interfacing circuitry between secure and non-secure components.

The transmission gate based design is inherently a multiplexer based design. Because of this characteristic, secure LUT implementations can be realized in FPGA without major modifications to the hardware.

APPENDIX
CMOS POWER CHARACTERISTICS

Complementary metal oxide semiconductor (CMOS) digital circuits are the enabling technology for communication and cryptographic circuits. Since we are looking at designing logic architecture with CMOS gates, a brief idea of the power characteristics of CMOS gates is presented in this section.

CMOS technology provides two types of devices, n type transistors (also called as gates) and p-type transistors. The basic structure of a CMOS gate is shown in Fig A.1. The device consists of a source and a drain which are lightly doped with impurities on a lightly doped n-type or a p-type silicon substrate called bulk. A polysilicon layer called the gate is deposited between the drain and the source regions. The drain, source and the gate are connected to metal contacts.

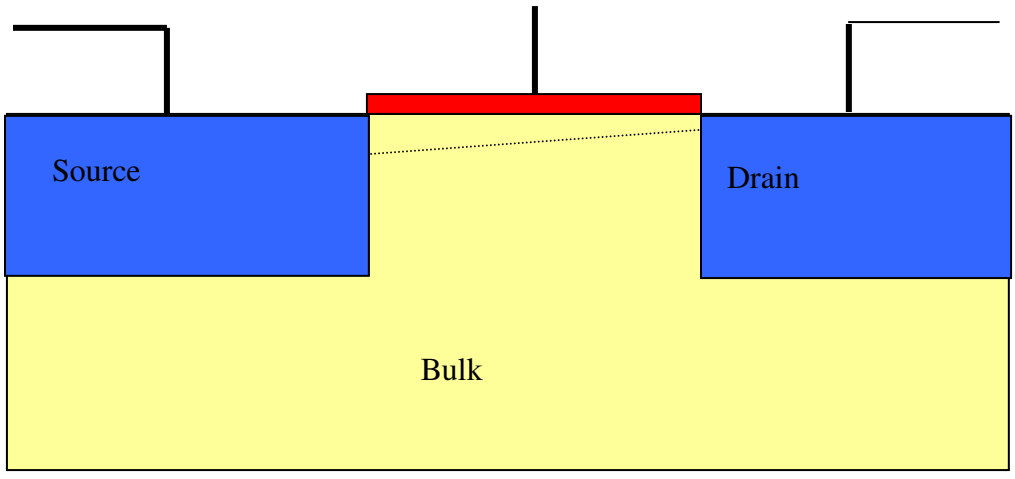

Fig A.1 Basic structure of a CMOS gate.

The gate is the control input and affects the flow of the current between the source and the drain regions. This results in the formation of a channel which is the conduction path for electrons or holes.

There are two components of power [13] in CMOS that affect the total power dissipation in a CMOS circuit. These are,

1. Static dissipation – Leakage current drawn from the power supply continuously
2. Dynamic consumption due to
    a. Switching transition current
    b. Charging and discharging of loads

For a CMOS inverter, if the input is 0 then the nMOS is OFF and the pMOS is ON, thus driving the output to high or logic state 1 (VDD). When the input is 1, the associated output is 0 or low (VSS). During any state of input, only one transistor is ON and the other is not conducting. There is no current path from VDD to VSS through the transistors. Thus the quiescent current or the steady state current (P_s) is zero.

But even when the MOS gate is not conducting, there is a small static dissipation due to reverse bias leakage between diffusion regions and the substrate. Also the sub-threshold conduction contributes to the static current. The leakage current in the MOS device can be described using a diode model. The leakage current is given by

$$i_0 = i_s\left(e^{qV \div kT} - 1\right)$$

Where,

i_s = reverse saturation current

V = Diode voltage

q =electronic charge

k = Boltzmann's constant

T = temperature

The static power dissipation is the product of leakage current and the supply voltage. The total static power dissipation P_s is given by,

$$P_S = \sum_{1}^{n} i_0 \times V_{DD}$$

Where

n = number of devices

The typical static dissipation is considerably less so that it can be easily ignored in normal circuits.

Whenever the output signal changes from $0 \rightarrow 1$ or from $1 \rightarrow 0$, moth n and p transistors are on for a short duration of time. Current is also dissipated in the form of charging and discharging of output capacitive load. The current spike resulting when both the pMOS and nMOS devices are ON is resulting in the short circuit current dissipation. This short circuit current depends on the rise and fall times, the load capacitance and the gate design. More than the static current dissipation, the implantation of DPA is based on the difference in the dynamic current characteristics (both short circuit and load capacitance) of a CMOS circuit.

It is found that at no load condition the short circuit current dominates. As the rise and fall times increase, the short circuit current also increases [13].

The dynamic power model can be represented that the rise and fall time is comparatively lesser than the period of the signal. Let us assume that the square wave input of frequency f. Let the average dynamic power dissipated be P_d. Then P_d is given by,

$$P_d = \frac{1}{t_p}\int_0^{\frac{t_p}{2}} i_n(t)V_{out}dt + \frac{1}{t_p}\int_{\frac{t_p}{2}}^{t_p} i_p(t)\left(V_{DD} - V_{out}\right)dt$$

Where

i_n = nMOS transient current

i_p = pMOS transient current

If we consider a step input with C_L as the load capacitance, we can find [13] that

$$P_d = C_L V^2_{DD} f_p$$

From equations 3.3 and 3.4 it can be seen that the transition is independent of device parameters.

The short circuit dissipation is given by,

$$P_{sc} = I_{mean} \times V_{DD}$$

For an input waveform with finite rise and fall times, the mean short circuit current in an unloaded inverter can be given [13] as

$$I_{mean} = \frac{2}{T}\int_{t_1}^{t_2} I(t)dt + \frac{2}{T}\int_{t_2}^{t_3} I(t)dt$$

If we can assume that the nMOS and pMOS devices have same β and the behavior is symmetrical it is shown [13] that

$$P_{sc} = \frac{\beta}{12}(V_{DD} - 2V_t)\frac{3t_{rf}}{t_p}$$

Total power dissipation is the sum of the static and dynamic power consumptions. The total power is given by

$$P_{total} = P_s + P_d + P_{sc}$$

This total power dissipation equation can be used to determine the power consumption for CMOS circuits. The differential power analysis methods need to consider the dynamic power consumption and thus need to consider both dynamic and short circuit current for the analysis. The power analysis attacks use the difference in dynamic current consumptions inherent in basic CMOS gates.

In a static CMOS device, every $1 \rightarrow 0$ transition and $0 \rightarrow 1$ transition has a different current characteristic. For example, let us consider the AND gate using static CMOS devices as shown in Figure A.2.

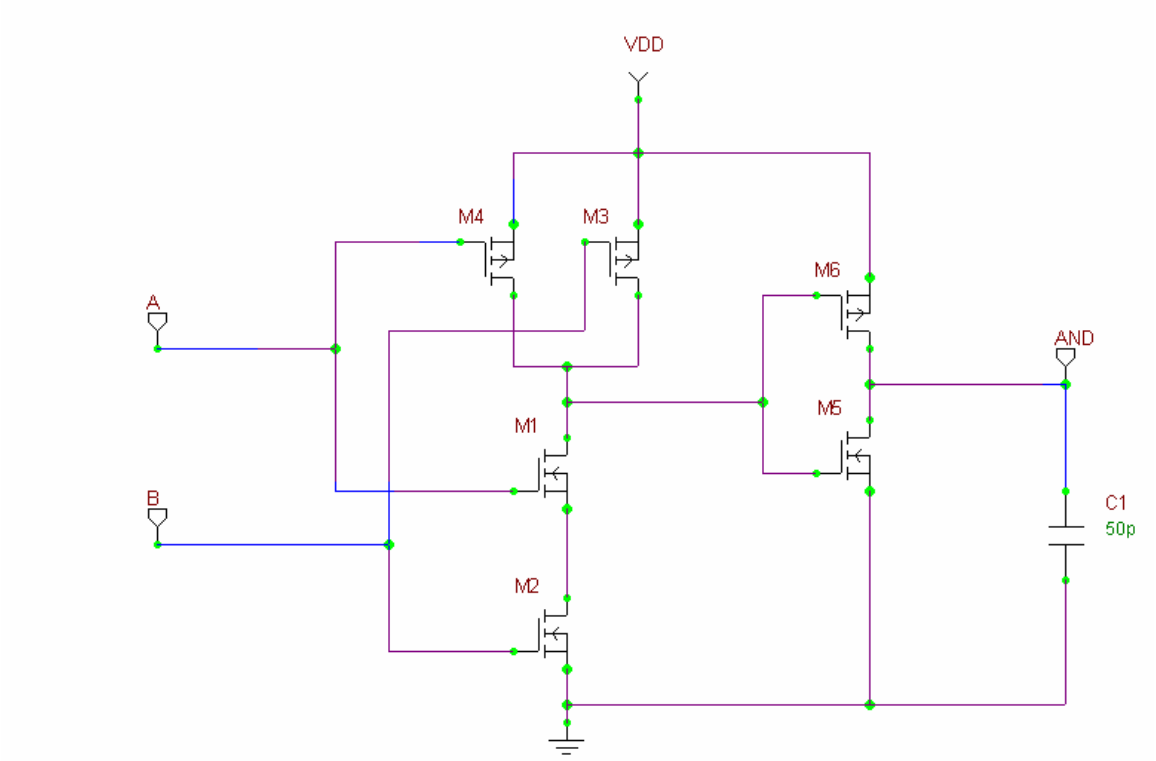

Fig A.2 AND gate using CMOS.

The current consumption for the 0→ 1 transition is shown in fig A.3. It can be seen that similar to the inverter, the current is drawn during the charging of the capacitor and is discharged during the complementary output.

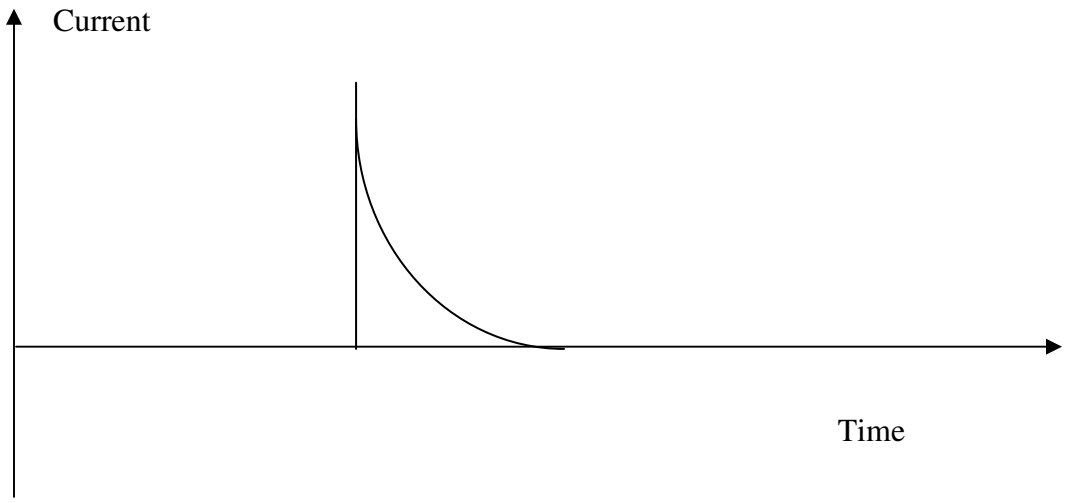

Fig A.3 Typical Current consumption curve for AND gate from data obtained from [13].

Contrary to the AND gate, OR gate (shown in fig A.4) has a complementary current consumption. The current (and hence the power) characteristics of an OR gate is shown in fig A.5.

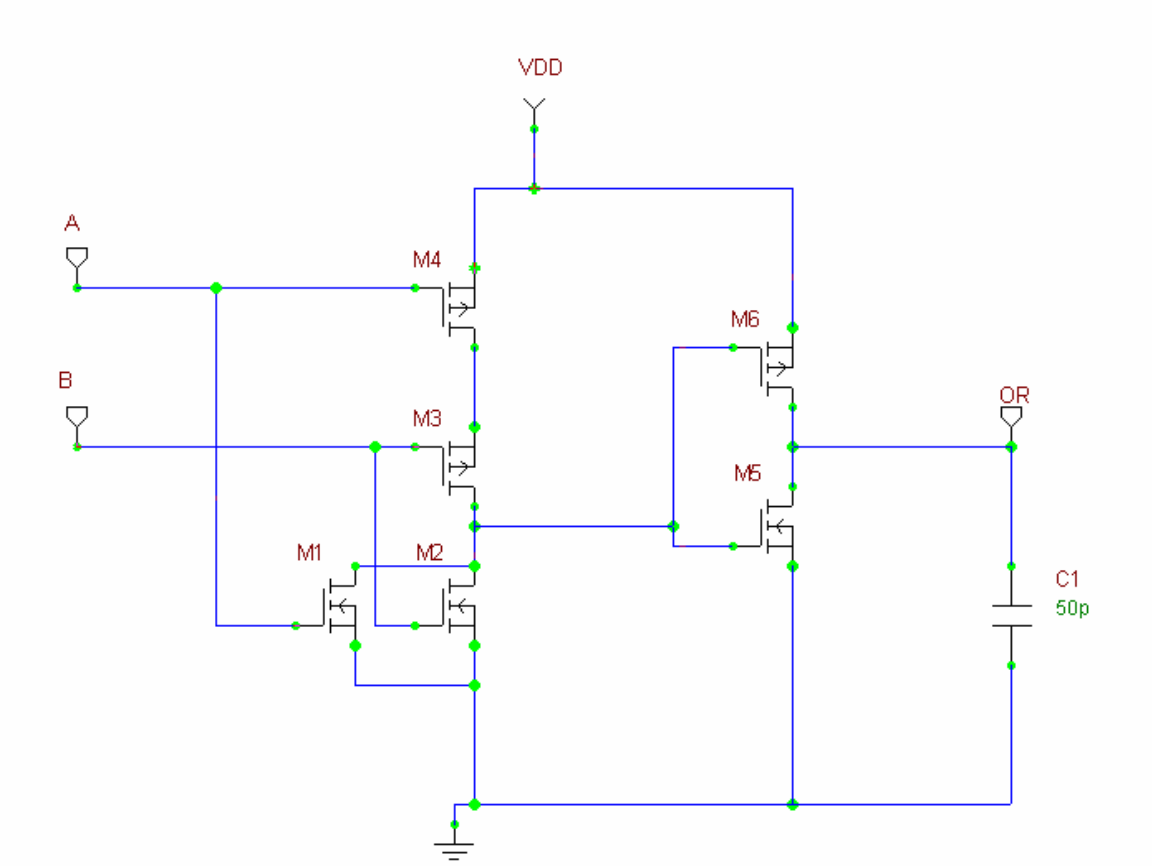

Fig A.4 OR gate using CMOS devices.

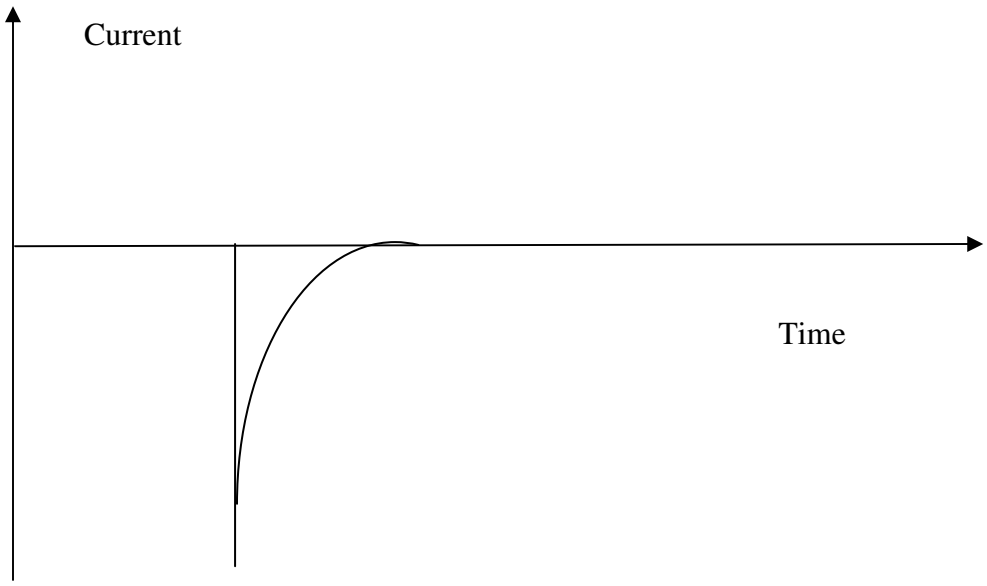

Fig A.5 Current consumption for OR gate from the data obtained from [13].

These current consumptions are inherently different thus making the power analysis attacks more feasible to implement. This difference combined with the statistical analysis described in chapter 2 form the fundamental factors affecting power analysis attacks.

REFERENCES

1. C. E. Shannon, "Theoretical analysis of cryptographic systems: [Communication Theory of Secrecy Systems]".

2. Yong Bin Zhou, DengGuo Feng, "Side channel attacks : Ten years after its publication and the impacts on cryptographic module security testing" available csrc.nist.gov/cryptval/physec/papers/physecpaper19.pdf

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

4. P. Kocher, "Timing Attacks on Implementation of Diffie-Hellman, RSA, DSS, and Other Systems," *Proceedings of Advances in Cryptology (CRYPTO '96)*, 1996, pp 104-113.

5. Discretix , "Introduction to side channel attacks" available hppp://www.discretix.com/PDF/Introduction%20to%20Side%20Channel%20Attacks.pdf.

6. P. Kocher, J. Jaffe and B. Jun. *Differential Power Analysis*. In Advances in Cryptology, CRYPTO '99, Springer LNCS 1666, pp 388--397, 1999.

7. Jean-Francois Dhem, Francois Koeune, Philippe-Alexandre Leroux, Patrick Mestre, JeanJacques Quisquater, and Jean-Louis Willems. "A practical implementation of the timing attack". In CARDIS, pages 167--182, 1998

8. L. Goubin, J. Patarin, "DES and Differential Power Analysis---The Duplication Method," CHES'99, Springer-Verlag, 1999, pp.158—172

9. P. Kocher, J. Jaffe, and B. Jun, "Introduction to Differential Power Analysis and Related Attacks," http://www.cryptography.com/dpa/technical, 1998.

10. P. Kocher, J. Jaffe, and B. Jun, Differential power analysis, Cryptography Research Inc, p4.

11. B^2SPICE from Beige bag available at http://www.beigebag.com/

12. R. Novak, "On the Security of RSA Capable Smart Cards," *Proceedings of the* 55 *10th Electrotechnical and Computer Science Conference*, Vol. B, September, 2001, pp. 135-138.

13. Neil H.E Weste, Kamran Eshragian, "Principles of CMOS VLSI design, A systems perspective" pp 86-90.

14. ISCAS 16X16 bit multiplier architecture available online at http://www.eecs.umich.edu/~jhayes/iscas.restore/c6288.html

15. N. Smart, "Physical Side-Channel Attacks on Cryptographic Systems," *Software Focus*, Vol. 1, Issue 2, pp. 6-13, December 2000.

16. Saputra, H., Vijaykrishnan, N., Kandemir, M., M.J. Irwin, R. Brooks, S. Kim, and W. Zhang. *Masking the Energy Behavior of DES Encryption*. DATE 2003 pp. 84-89

17. T. Messerges, E. Dabbish, and R. Sloan, "Examining Smart-Card Security under the Threat of Power Analysis Attacks," *IEEE Transactions on Computers*, Vol. 51, No. 5, pp. 541-552, May 2002.

18. T. Messerges, E. Dabbish, and R. Sloan, "Investigations of Power Analysis Attacks on Smartcards" available online at, http://www.usenix.org/events/smartcard99/full_papers/messerges/messerges_html/index.html

19. Girish B. Ratanpal, Ronald D. Williams, Travis N. Blalock, "An On-Chip Signal Suppression Countermeasure to Power Analysis Attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 01, no. 3, pp. 179-189, Jul-Sept, 2004.

20. Peter Wayner, "Code breaker cracks smart card's digital safe" available online at http://www.nytimes.com/library/tech/98/06/biztech/articles/22card.html

21. M. Bucci, R. Luzzi, M.Guglielmo, A. Trifiletti , "A Countermeasure against Differential Power Analysis based on Random Delay Insertion" IEEE international symposium on circuits and systems, 2005, ISCAS 2005, volume 4, pp 3547-3550.

22. M.A. Hasan, "Power Analysis Attacks and Algorithmic Approaches to Their Countermeasures for Koblitz Curve Cryptosystems," *IEEE Transactions on Computers*, vol. 50, no. 10, pp. 1071-1083, Oct., 2001.

23. A. Shamir, "Protecting Smart Cards from Passive Power Analysis with Detached Power Supplies," *Proceedings of Cryptographic Hardware and Embedded Systems (CHES 2000),* Lecture Notes in Computer Science, vol. 1965, August 2000, pp. 71-77.

24. S.Almanei, " Protecting smart cards from power analysis attacks" available online at islab.oregonstate.edu/koc/ece679/project/2002/almanei.pdf

25. F. Mace, F.-X. Standaert, I. Hassoune, J.-D. Legat, J.-J. Quisquater "A Dynamic Current Mode Logic to Counteract Power Analysis Attacks" *DCIS 2004 - 19th Conference on Design of Circuits and Integrated Systems*, pages 186 - 191, November 2004.

26. K. Tiri, M. Akmal, I. Verbauwhede. *A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Differential Power Analysis on Smart Cards*. ESSCIRC 2002 pp. 403-406

27. Sung-Mo Kang, Yusuf Leblebici, "CMOS digital Integrated circuits, Analysis and Design", pp 307-314.

28. X.Zhang and R Sridhar, "CMOS wave pipelining using Transmission gates", proc IEEE international ASIC conference and Exhibit, 1994, pp 92-95.

29. Xilinx ISE 8.1i, http://www.xilinx.com/ise/logic_design_prod/foundation.htm

30. Icarus verilog compiler available online at, http://icarus.com/eda/verilog/Release_Notes_for_Icarus_Verilog_0_8.html