DESIGN OF POWER ATTACK RESISTANT CIRCUITS FOR CRYPTOGRAPHY

Partha De

DESIGN OF POWER ATTACK RESISTANT CIRCUITS FOR CRYPTOGRAPHY

Thesis submitted in partial fulfillment
of the requirements for the award of the degree

of

Masters in Science

by

Partha De

Under the supervision of

Dr. Chittaranjan Mandal

Department of Computer Science and Engineering

Indian Institute of Technology, Kharagpur

August 2014

APPROVAL OF THE VIVA-VOCE BOARD

Certified that the thesis entitled **"Design of Power Attack Resistant Circuits for Cryptography,"** submitted by **Partha De** to the Indian Institute of Technology, Kharagpur, for the award of the degree of Master In Science has been accepted by the external examiners and that the student has successfully defended the thesis in the viva-voce examination held today.

Prof Dpankar Sarkar
(Member of the DAC)

Prof Santanu Chattopadhay
(Member of the DAC)

Prof Anupam Basu
(Member of the DAC)

Prof Chittaranjan Mandal
(Supervisor)

(External Examiner)

(Chairman)

Date:

ACKNOWLEDGMENTS

I would like to express my heartiest gratitude to my thesis supervisor Prof Chittaranjan Mandal for guiding me through the Master in Science program. I acknowledge his constant technical and moral support and guidance throughout my MS period. At the time of joining, I was a novice in the field of VLSI, but in the course of my work, I have learned a lot about the art of conducting research, solving problems, designing circuits, using complex cad tools and art of taping-out IC from my supervisor. I consider myself extremely lucky for getting the opportunity to work under him. I would also express my thanks to Prof Debdeep Mukhopadhay for helping me in my work.

I want to thank my lab mates Aritra, Antara, Kunal, Sandipan, Chandan, Rajoshree, Subhadip, Tamal, Satya Goutam, Dhiman Gargi, Devleena, Sudip-da, Maunendra, Sumana, Arindam, Shiladitya, Soumyadeep and Debjit for making the laboratory environment most enjoyable. Special thanks go to Kunal for his association with my work.

I also thank Sayan, Santa, Debasis Kundu, Sankar for making my stay in Kharagpur extremely memorable. I also thanks to D-304 mess for taking care of my food habits. Special thanks to Bappa, Prasun-da, Durga-da and Sibu-da for their assistance in the laboratories.

I also thank my brother Papai who stays home and taking care of our parents. Last but not the least, I express my gratitude to my parents who has sacrificed much and provided me continuous support and encouragement without which I could not come out with this work. I would also like to thank my family members for their encouragement and for the confidence they reposed on me.

Partha De

Indian Institute of Technology Kharagpur

Certificate by the Supervisor

Date: 20-08-2014

This is to certify that the thesis entitled,

"Design of Power Attack Resistant Circuits for Cryptography"

submitted by PARTHA DE(10CS70P03) to the Indian Institute of Technology Kharagpur, is a record of bonafide research work carried under my supervision and is worthy of consideration for the award of the Master of Science of the Institute.

Signature of the Supervisor(s):

_____

Prof. CHITTARANJAN MANDAL COMPUTER SCIENCE & ENGINEERING

Indian Institute of Technology Kharagpur

Declaration by the Student

Date: 20-08-2014

Title of the Thesis:

Design of Power Attack Resistant Circuits for Cryptography

I Certify that

a. the work contained in the thesis is original and has been done by me under the guidance of my Supervisor;

b. the work has not been submitted to any other Institute for any degree or diploma;

c. I have followed the guidelines provided by the Institute in preparing the thesis;

d. I have conformed to ethical norms and guidelines while writing the thesis and;

e. whenever I have used materials (data, models, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis, giving their details in the references, and taking permission from the copyright owners of the sources, whenever necessary.

Signature of the Student: _____

Name of the Student: PARTHA DE (10CS70P03)

ABSTRACT

Power analysis attacks (PAAs) have been found to be extremely effective on cryptographic systems to derive the cryptographic secrets from these traces.

In this work, Binary Decision Diagram (BDD) based dual-rail logic circuit schemes have been developed to counter power analysis attacks (PAAs). The hallmark of our circuit schemes is that an identical number of switchings is ensured on each circuit path. The transistors are interconnected to create pull-up and pull-down paths to outputs by way of binary decisions based on the input variables, so as to realise the required Boolean function. This principle of operation has directly permitted the use of BDD based logic synthesis to design the required pull-up and pull-down networks of transistors. The operation of these circuit schemes feature novel pre-charge generation, voltage scaling with leakage power minimization and early propagation effect resistance mechanism. In particular, we have developed and explored top pre-charging, top-bottom pre-charging, bottom pre-charging and symmetric NMOS bottom pre-charging logics. A simple synthesis algorithm for mapping a given Boolean functions to such BDD based circuits is also presented.

Extensive experimentation has been carried out to establish resistance of our circuits to PAAs. Objective of the experimentation is two fold, to demonstrate resistance to power attacks and to highlight the low power characteristics. Towards the first objective, differential power attacks such as, difference of mean (DoM) and correlation power attack (CPA) have been carried out. Resilience to the the early propagation effect (EPE) is also demonstrated. Six 2-input basic cell and two 4×4 S-boxes are used for experimental benchmark. Experimental results on circuits with bottom pre-charge logic demonstrate a significant reduction by 99.68% and 88.55% in peak power variance (PPV) over two chosen competing designs, for the basic cell. The reduction in PPV is recorded to be greater than 99.9% for the S-box implementations for both those design. A reduction of about 30% to 67% in both average power and average current consumption is observed while comparing with the chosen techniques. Experimental results on circuits with various other features such as top pre-charge and top-bottom pre-charge also demonstrate a large reductions in PPV. Significant reduction for average power and average current for both the pre-charge logics is also achieved. Circuits using top-pre charging required less transistors and demonstrated lower PPV in comparison with others. Symmetric NMOS bottom pre-charge logic was more resilient to EPE due to its symmetric nature. Bottom pre-charge and symmetric NMOS bottom pre-charge logic were also effective in avoiding timing attacks along with top-bottom pre-charge logic.

Keywords: Side channel attack; Power analysis attack; Binary Decision Diagram; Early propagation effect; Voltage scaling; Pre-charge logic;

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Digital communications have become a major part of modern day life. With increase in reliance on the information transmitted through the web and other communication media, it has become a major challenge to keep those information safe and resist unauthorized accesses. A sensitive component of such transactions involve electronic banking ranging from internet banking to the use of smart debit/credit cards. We often talk of speed on the internet access, another significantly more important concern is that of security of information that is sent across the internet or between electronic devices. Smart cards include electronic circuits that authenticate the identity of the card holder. While data going through the internet is protected by encryption, there are several, relatively new and counter initiative, issues regarding the secure use of smart cards. The work done in this thesis is directly applicable to the vulnerability of such transactions done using smart cards.

Modern cryptography has the four objective: confidentiality, integrity, non-repudiation and authentication [5].

- Confidentiality: information can be only understood by intended persons
- Integrity: information should be unaltered
- Non-repudiation: sender of the information cannot deny at a later stage his or her intentions in the creation or transmission of the information

- Authentication: sender or receiver have to confirm their identity on time of sending and receiving the data

A cryptographic algorithm is a complex mathematical function that uses a secret key to encrypt information and without the knowledge of the required secret key, decrypting this information would be infeasible. During the past years, there has been a lot of research on cryptography and as a result there are several algorithms that provide data security and authenticity, such as RSA [37], ECC [23, 29], AES [31], DES [18], and DSA [32].

Cryptographic systems involve complex mathematical functions to make it difficult for intruders to discover the systems' secret keys. However, conventional hardware proves to be inadequate to process these functions. To overcome this issue, dedicated cryptographic hardware is used and still evolving. While these devices aim to achieve both security and efficiency, operational behaviour and corresponding data being handled by these hardware can be deciphered by intruders at the time of their execution by statistically analyzing their various properties like power, execution time, electromagnetic leaks, sound, etc. This phenomenon of acquiring sensitive data by exploiting the information gained from the physical implementation of a cryptographic system is called *side channel attack*.

The information available to the attacker is a set of public transactional messages processed by the cryptographic device and the corresponding running time, power dissipation, etc. and his goal is to recover the secret parameters being used inside the device using the available information. Different forms of side channel attacks are as follows:

- Timing attacks: attacks based on measuring how much time is taken to perform various computations [19, 24].
- Power monitoring attacks: attacks which make use of the varying power consumption by the hardware during computation [26, 30].
- Electromagnetic attacks: attacks based on leaked electromagnetic radiation which can directly provide information [1, 21].
- Acoustic cryptanalysis: attacks which exploit sound produced during computations [49].

- Differential fault analysis: in which secrets are discovered by introducing faults in a computation [8, 50].

- Data remanence: in which sensitive data are read after supposedly having been deleted [38].

A comprehensive study of side channel attacks can be found in [26, 30]. Among the various physical properties, the power consumption model is predominantly used to identify the secret keys because power traces are easily available. In general, the hardware synthesis flow attempts to optimize area, average power consumption and propagation delay. However in those flows security is mostly neglected.

Power dissipation of a circuit is proportional to its switching activity which, in turn, depends on the data that is being handled. The data dependent power consumption can be exploited to leak away the secret information, specifically, distribution of 0's and 1's. This is explained by means of table 1.1 which shows the signal transitions of single rail static circuit. Let P_1 and P_2 be the powers consumed during the transitions $1\rightarrow0$ and $0\rightarrow1$, respectively. The amount of power consumed for these two transitions being different (i.e., $P_1 \neq P_2$), the asymmetry can be exploited to mount power analysis attacks. Depending upon how the power traces are monitored, their analysis can be further classified.

- Simple Power Analysis (SPA): Only a single power trace is examined to extract the secret key.

- Differential Power Analysis (DPA): It involves collecting large number of power traces and performing statistical analysis of the power variation with respect to changes in data values to extract the secret key.

Table 1.1: Power consumed due to switching

Transitions	Power consumed
$1\rightarrow0$	P_1
$0\rightarrow1$	P_2
$0\rightarrow0$	0
$1\rightarrow1$	0

There are broadly two types of countermeasures of DPA based on their level of application: *(i)* algorithmic level, and *(ii)* cell level. Our focus is on cell level countermeasures for DPA. A cell level approach ensures algorithmic agnostic DPA resistance and also enables speedy completion of the encryption and the decryption processes. Moreover, it avoids extra circuitry to the extent that is needed for implementing reported algorithmic level DPA countermeasures [4, 33].

Prevalent circuit level countermeasures for DPA rely on customized transistor level circuit designs. The concept of Dynamic Differential Logic (DDL) has been used in Differential Cascade Voltage Switch Logic (DCVSL) [22]. However, circuit asymmetries in the gates realized using DCVSL can cause large variations in power consumption. Sense Amplifier Based Logic (SABL) [42, 45, 46], has been developed to overcome this drawback of DCVSL. SABL uses fixed amount of charge for every transition, even when a gate does not change its switching state. In every cycle, a SABL gate charges a total capacitance with a constant value. The emergence of WDDL [47] was an important development against DPA. WDDL has been found to exhibit resistance against power attacks for deployment on ASIC, FPGA [47], and AES coprocessor [44] fabricated on 0.18μm CMOS technology.

A digital design flow for producing secure integrated circuits using WDDL is described in [48]. While WDDL circuits have the advantage of being realizable using standard cells, the resulting circuits lack the optimization possible for circuits specifically designed for DPA resistance. WDDL also suffers from the *early propagation effect* (EPE) which is caused when input signals of a WDDL gate have different delay times. The leakage due to EPE can be a potential source of data-dependent power consumption that may lead to more sophisticated power analysis attacks [27].

To overcome EPE, Dual-rail Pre-charge circuit with Binary Decision Diagram (DP-BDD) architecture was proposed in [3]. In a DP-BDD based circuit, it is ensured that the input signals always pass through the same number of AND-OR gates, thus countering the early propagation effect. Recently, Secure Differential Multiplexer Logic using Pass transistor (SDMLp) [35] has been used to provide DPA resistance comparable to that of WDDL while requiring lesser area, power and current, and having lower peak power variation. SDMLp is based on Reduced Ordered Binary Decision Diagram (ROBDD) which is capable of representing a logic function more succinctly (i.e., requiring lesser nodes) than normal BDD, thus saving on layout area.

However, it is important to evaluate the effectiveness of the countermeasures. Some techniques for systematically analysing DPA leakage have been developed in the literature [7, 15, 16]. Constructing a power consumption model is an important step for analysis of the effects of countermeasures. For instance, the model based on analog characteristics of CMOS circuits [15], the model based on the Hamming weight [16], and the simplification model of [15] based on transition of data registers [7] were proposed in 1999, 2000 and 2002, respectively.

1.1 Motivation

Reported DPA countermeasures are found not to ensure an identical number of transistors switches for all possible inputs. In this work we developed a design and synthesis scheme where the underlying transistor circuitry necessarily has the same number of transistors on all paths from the inputs to the output. The transistors are interconnected to create pull-up and pull-down paths to output by way of binary decisions based on the input variables, so as to realise the required Boolean function. This principle of operation has directly permitted the use binary decision diagram (BDD) [2, 11] based logic synthesis to design the required pull-up and pull-down networks of transistors. While the reduced order binary decision diagram (ROBDD) [2, 13] based synthesis mechanism yields optimised logic functions, it does not ensure identical path lengths. This problem is overcome through the insertion of *dummy* transistor for path length equalisation.

It is well known that dual rail pre-charge logic discharges parasitic capacitance periodically and thus enhances the power invariant characteristics of the circuit [41, 45–47]. In this work we explore the potential benefits of positioning the pre-charge generation logic at various places in the overall circuit – an aspect that is practically missing in the literature. In particular, we consider four pre-charge configurations and evaluate the resulting circuit behaviour with respect to characteristics such as power, current, delay and also evaluate the effectiveness against various power attacks.

1.2 Problem statement

Variance in power dissipation is a key factor in determining the success or failure of a power analysis attack. Dynamic power which is major component of the total power consumption depends on the switching activity of the transistors which, in turn, depends on the input combinations applied to the transistors. The current flows from the *voltage-source* to the *ground* through intermediate capacitors (but not directly) of transistor networks depending on the switching activity of transistors. Outputs are produced by charging or discharging the output capacitors. Existing dual-rail complementary circuit realizations of logic functions designed for power attack resistance have asymmetry in the critical paths between points through which there is flow of charge. This asymmetry leads to different power consumption and also different propagation delays for different input combinations – making the circuit vulnerable to power attacks as well as timing attack, and the early propagation effect.

In this work, our objective is to identify power attack, timing attack and early propagation effect resistant circuit structures towards achieving satisfactory attack resistance with the property of identical critical path lengths of all possible switching paths. This property also has the potential to give immunity to timing attacks unless the excecution of the underlying algorithm itself is data dependent and opens the possibility of launching of timing attacks. Necessary pull-up and pull-down circuits are constructed based on the Boolean function of the input variables. Therefore, BDD based logic synthesis can be applied to design such pull-up and pull-down networks of transistors. ROBDD based mechanism reduces the logic functions by changing the order of the input variables. However, critical paths from parent to leaf nodes vary in such designs. To overcome this issue, dummy nodes have been inserted as required for path balancing.

1.3 Summary of contributions

The contribution of this work is the development of a BDD based logic synthesis approach to counter power analysis attacks along with two different pre-charge generation logics styles. Each logic style consists of two different pre-charge generation

schemes. The operation of this logic (for each of the pre-charge generation scheme) has four aspects, viz.:

- Pre-charging circuitry specially designed to work with BDD based directed acyclic graph (arising from folding the BDD tree)
- BDD based normal and complementary function realizations with identical critical path length of all possible switching paths and dual-rail complementary functions
- Output generation with proper voltage level
- Low power techniques (voltage scaling, leakage reductions) to reduce overall power dissipation without hampering DPA resistance

Based on our experience of designing power analysis attack (PAA) resistant circuits, we also provide an automated synthesis process of such circuits which involves the following steps:

- ROBDD based logic minimization with normal and complementary functions
- Insertion of dummy nodes for path balancing, pre-charge nodes for pre-charge logic and regenerative nodes for fanout
- Partitions of larger BDD structures into smaller realisable structures without compromising attack resistance
- Converting the resulting BDD to transistor-level Verilog

The operation of these customized designs is first described using a basic cell supporting fourteen logic functions including AND, OR, XOR, NOT, NAND, NOR. While any logic can be constructed using this basic cell, more optimized circuit realization is possible by utilizing the ROBDD based normal and complementary function realizations aspects of this logic synthesis approach. This is illustrated through the design of the 2-bit adder and different S-boxes [9, 40]. Experimental results have been gathered for the basic cell, the adder and the different S-box realizations. These results have demonstrated that our logic outperforms competing methods in terms of peak power variation, average power and average current and also repelled strong power attacks.

1.4 Thesis outline

The remaining thesis chapters are as follows:

Chapter 2: Preliminaries This chapter presents some preliminary concepts relevant to the thesis which are as follows:

- Binary Decision Diagrams: This section describes BDD principles and the basic mechanism to represent a Boolean function as a BDD or ROBDD
- ASIC design flow: This section describes basic flow of digital system design
- Pass transistor logic: This section describes pass transistor logic principles
- Power consumption of CMOS logic: This section describes how power dissipation happen in the CMOS circuits
- Cryptographic preliminaries: This section describes basic principle of cryptography
- Side channel attacks: This section describes basic concepts of side channel attacks, different type of power attacks

Chapter 3: Basic BDD based circuits with bottom pre-charge This chapter describes BDD based logic synthesis and circuit design methods to counter power attack with the specific feature that the pre-charging is controlled via the leaf nodes of the transistor network realising the BDD.

- Bottom pre-charge
- Symmetric NMOS transistor based pre-charge

Chapter 4: BDD based circuits with various other features This chapter describes BDD based logic synthesis and circuit design methods to counter power attack with other pre-charging schemes such as:

- Top pre-charge
- Top-bottom pre-charge

Chapter 5: Automated synthesis scheme This chapter describes automated synthesis schemes for circuits designed using the above pre-charging techniques along with generation of the synthesisable Verilog code for these.

Chapter 6: Experimental results with different process technology This chapter describes the experimentation and presents experimental results to establish resistance of our circuits to power attacks

Chapter 7: Conclusions and future work Summary of work done in this thesis and conclusions so derived are presented. We also consider how this work can be further extended.

Chapter 2

Preliminaries

The work in this thesis involves use of CMOS circuits, PTL circuits, ASIC design, binary decision diagrams, low power techniques, use of cryptographic algorithms, side channel attacks. This chapter covers some elementary concepts and topics from these diverse areas that are relevant to this thesis.

This chapter is organized as follows. In the section 2.1 we introduce the binary decision diagram. Different ASIC design flows are introduced in section 2.2. Power dissipation in CMOS circuits is described in the section 2.4. Basic cryptographic concepts and the notion of side channel attacks, specifically for CMOS circuits are given in section 2.5.

2.1 Binary Decision Diagrams and Reduced Ordered Binary Decision Diagrams

We first introduce binary decision diagrams (BDD) and then explain how reduced ordered binary decision diagrams (ROBDD) are derived from those. We also discuss the additional properties of ROBDDs over BDDs.

Binary Decision Diagrams (BDDs) are decision trees based on Shannon's expansion. BDDs are extensively used for circuit design. A brief definition of the BDD is as follows: A binary decision diagram (BDD) is a rooted decision tree having the

following properties.

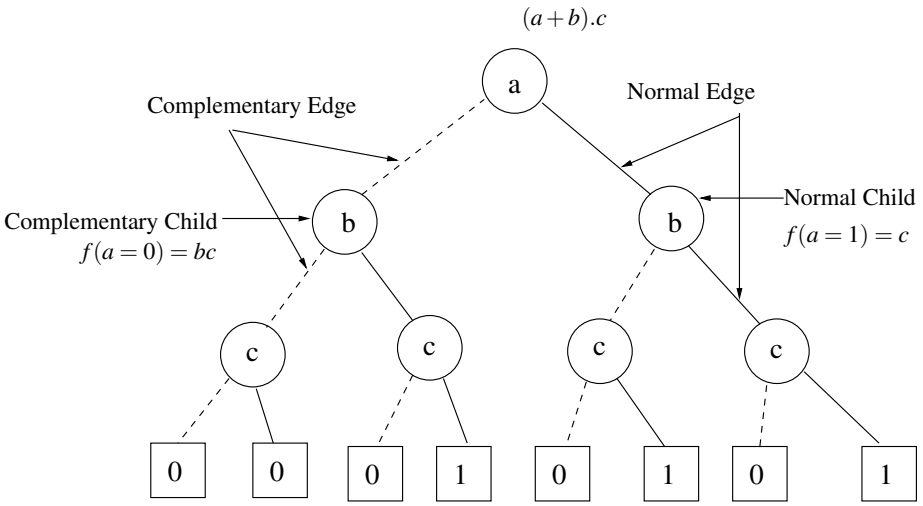


Figure 2.1: Decision tree for $f = (a \vee b) \wedge c$.

- Internal nodes of a BDD are labelled with variable names
- Leaf nodes of a BDD are labelled with either 0 or 1
- Each internal node in the DAG has exactly two children; one of the two arcs connecting a parent node to one child node is labelled by 0 (represented by a dashed line) while the other arc is labelled 1 (represent as a solid line)
- Nodes on every path in the graph have unique labels; different nodes on a single path are labeled by distinct variables
- Left and right sub-DAGs of every node are distinct
- Every pair of sub-DAGs rooted at two different nodes n_1, n_2 are non-isomorphic

The last property ensures that the BDD is reduced.

Given a Boolean function $f(x_1, x_2 ..... x_k)$, by Shannon's decomposition around variable x_1, $f(x_1, x_2 ..... x_k)$, this may be written as

$$f(x_1, x_2 ..... x_k) = (x_1 \wedge f_{x_1 \leftarrow 1}) \vee (\neg x_1 \wedge f_{x_1 \leftarrow 0}) \tag{2.1}$$

Here $f_{x_1 \leftarrow 1}$ denotes f with x_1 substituted by 1 and $f_{x_1 \leftarrow 0}$ denotes f with x_1 substituted by 0. $f_{x_1 \leftarrow 1}$ and $f_{x_1 \leftarrow 0}$ are also called the positive and negative cofactors to f with respect to x_1.

Suppose the Boolean function $f(x_1, ...x_k)$ is decomposed around variable x_i represented by a DAG node labelled x_i, then that node will have two children – the 0-child, representing $f_{x_1 \leftarrow 0}$ and the 1-child representing $f_{x_1 \leftarrow 1}$. The edge connecting the node labelled x_1 to the 1-child is called the normal edge, while the edge connecting that to the 0-child is called the complementary edge – refer to the picture of a BDD shown in Fig. 2.1. These cofactors may be further decomposed recursively, terminating at the Boolean constant 0 or 1, giving rise to the BDD representing the given function.

Given a valuation of the variables, the value of the function can be determined as follows:

- Starting from the root of the DAG follow either the normal or the complementary edges depending on the value of the decision variable at the node
- Continue this process until leaf node is reached
- For the given valuation of the variables, the value of the leaf node gives the value of the function

The variable ordering along a path from the root node to a leaf node is the sequence in which the variables appear along that path staring from the root node; it is the sequence in which variables are chosen for carrying out Shannon's decomposition. If the variable order is the same on all paths of the tree it is called an ordered decision tree. A BDD that is both ordered and reduced is called a Reduced Ordered Binary Decision Diagram (ROBDD). The ROBDD for a given Boolean function is unique and so the ROBDD representation of a Boolean function is canonical.

The steps of reducing BDD is following [17] :

- If two nodes represent the same function, then we merge them
- If a node has the same 0 -child and 1-child, then that node represents a "don't care" variable, and is removed. Formally, it follows from Shannon's decomposition that f is independent of x_i whenever $f_{x_i \leftarrow 0} = f_{x_i \leftarrow 1}$

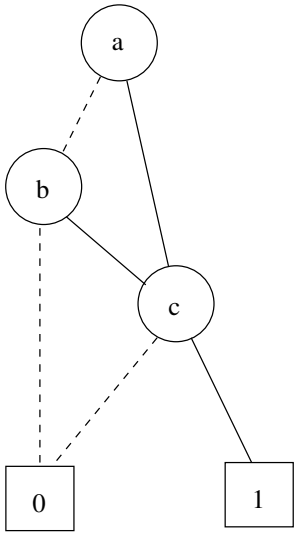


Figure 2.2: Reduced BDD for $f = (a \vee b) \wedge c$

Applying these rules one obtains the Reduced Ordered Binary Decision Diagrams (ROBDD) which is canonical in nature. The task of finding the optimal variable ordering for a function is a computationally difficult problem. However, a wide variety of heuristics are available for finding a good variable ordering. Existing BDD packages such as CUDD often produce the optimum solution. The ROBDD for $f = (a \vee b) \wedge c$ is shown in Fig. 2.2.

There are mainly two types of variable ordering available:

Static variable ordering: Static variable ordering techniques attempt to establish the optimal ordering of variables prior to constructing the actual BDD. A simple heuristic is that input variables that are topologically close together within the circuit should be relatively close together within the variable ordering for the resulting BDD. This is found to work well for tree-like circuits but does not generalise to most other circuits. Another principle is that the most influential of the primary inputs to the circuit (such as control inputs) are placed earlier on in the ordering. A detailed survey on static varaible ordering is available in [36].

Dynamic variable ordering: In case of dynamic variable ordering, the ordering heuristics are used during the construction of BDDs. A circuit may have multiple subcircuits which may have different optimal variable ordering. In such situations, instead

of a uniform global variable ordering, the ordering may vary from one sub-circuit to another. In a dynamic situation, shifting variables up and down and evaluating the impact is an important dynamic variable ordering mechanism. Sifting is not relevant to the current work.

2.2 ASIC design flow

Application specific integrated circuit is popularly termed as ASIC. ASIC design flow is given in diagram 2.3.

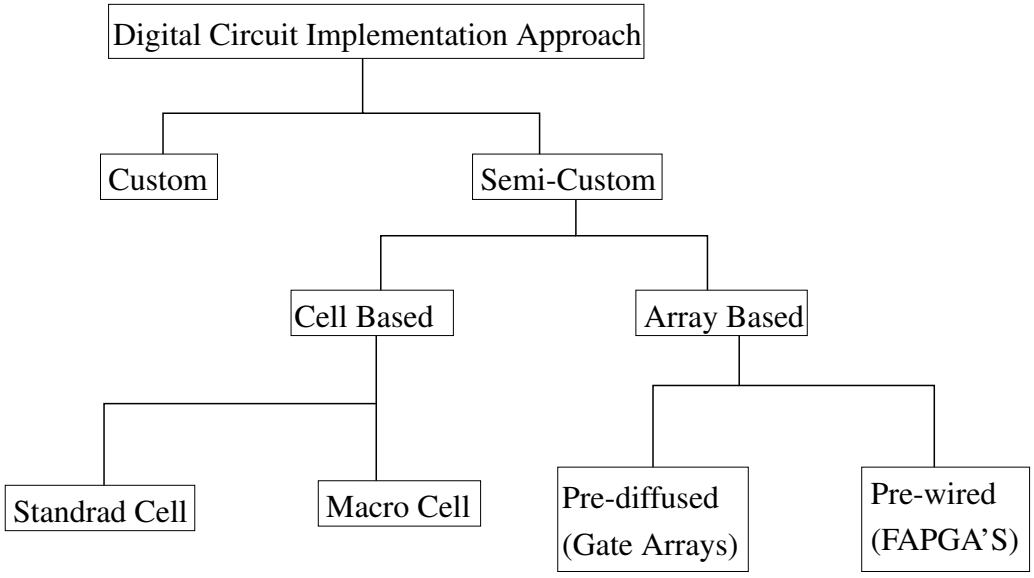


Figure 2.3: Digital design flow.

There are broadly two methods that are followed for designing ASICs, using modern CAD tools, one is the full custom design flow, the other is the semi-custom design flow. Semi-custom design are two types: cell based and array based. Standard cell based and macro based design comes under cell based design flow. On the other hand, gate arrays and FPGA comes under array based design flow.

2.2.1 Full custom design flow

In the full custom design flow, designers start designing from the scratch. Designers meet specification by designing transistor level circuits with optimized transistor widths. Layout, routing and synthesis is done at the transistor level by the designer. Most of the components are individually designed to give best overall performance, consuming lesser area and power in comparison with standard cell based designs. It is highly expensive and consumes huge man hours. It is generally used for designing new components for use with the other design styles. Specialised algorithms may also generate full custom design for particular applications. This is the design style we used for designing our circuits.

2.2.2 Semi-custom design flow

Semi-custom design flow is the main area of focus of advanced CAD tools. We would basically emphasis on ASIC design and omit discussion on the FPGA design flow as that is not relevant to our current work. Details of semi-custom design flow is given below.

Standard cell based design flow

Here designers design and fabricate their design by using pre-designed basic gates, commonly known as standard cell. For a given a technology, standard cells are provided by the chip fabrication facility which are called standard cell design kits. These design kits reduce the cost of ASIC development. Standard cells of a particular family have the same height but vary in width making it suitable for placing in rows side by side. Gaps between rows are channels that are used for routing. Standard cell design flow is shown in Fig. 2.4.

Macro cell based design flow

More complex design modules may be provided as macros. For use with CAD tools these are given at a high level of abstraction. Macro cell based design can be sub-

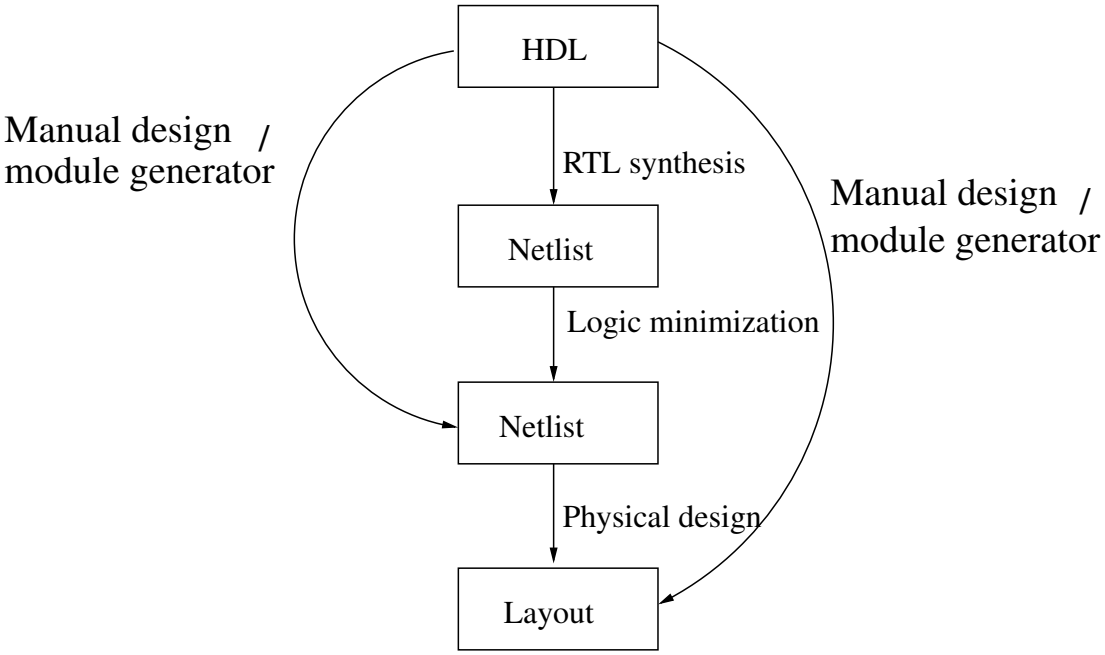


Figure 2.4: Standard cell design flow.

classified into three parts.

- Hard macros: These are pre-complied design components which can be instantiated in a large design. For use at various levels of simulations, detailed abstract models of the component are made available for the designers, for protection of intellectual property right. Hard macros do not provide detailed internal layout information. Disadvantages of hard macros is their association with particular fab which results in lower flexibility. However these do have the benefit of optimised performance often with human optimisation.

- Firm macros: These are pre-complied design components which can be instantiated in the large design. Here the netlist is provided for use with the design tool. The physical design is done at the designer end. Protection of intellectual properties is less compared to hard macros. Physical design is more flexible with firm macros.

- Soft macros: These are also pre-complied design components which can be instantiated in the large design. Here the RTL is provided for use with the design tool. Logic design is done at designer end. Protection of intellectual properties is even less but design options are more. Optimisation is done by the CAD tool

and that is usually lower than that offered by hard macros.

2.3 Pass transistor logic

The basic idea behind pass transistor logic, commonly known as PTL, is multiplexing. Functionally, pass transistors behave like a switches. The source of the pass transistor is connected with some input signal. Generally, it is connected with power supply rail for other logics. Either NMOS or PMOS transistor is sufficient to perform the logic operation. This results in smaller number of transistors and smaller input loads, especially when NMOS transistor networks are used. However, there is a voltage drop at the output, $V_{out} = VDD - V_{thN}$ due to the threshold voltage of the NMOS transistor while passing a logic '1'. To maintain the output voltage an acceptable range, swing restoration at the gate output is necessary. Pass transistor logic network whose basic structure is that of a multiplexer requires complementary control signals. Thus, dual-rail logic which produces both normal and complementary output, is usually used in order to provide necessary signals in both normal and complemented form. To provide acceptable output driving capabilities inverters are attached with the gate outputs. Only single paths of each network must be active at a time to avoids short circuits [51].

Advantages of pass transistor logic:

- Ratio-less: In conventional CMOS logic, the width to length (W/L) ratio of the pull up device is generally few times greater than the pull down devices. As a result the geometrical dimension of the transistor is not minimal always. However, pass transistor can be realize with minimum dimension of a particular technology thus making it more area efficient.
- Lesser power: In a pass transistor logic realization of a Boolean function there is no DC path from supply to ground. So, standby power dissipation is small. Each additional input requires only minimum geometry transistor which results in minimal increase in power dissipation.
- Lower area: Only a few NMOS and PMOS transistors are sufficient to realize any logic function using pass transistor logic which ensures smaller input load and smaller area. Thus lower power consumption is achieved.

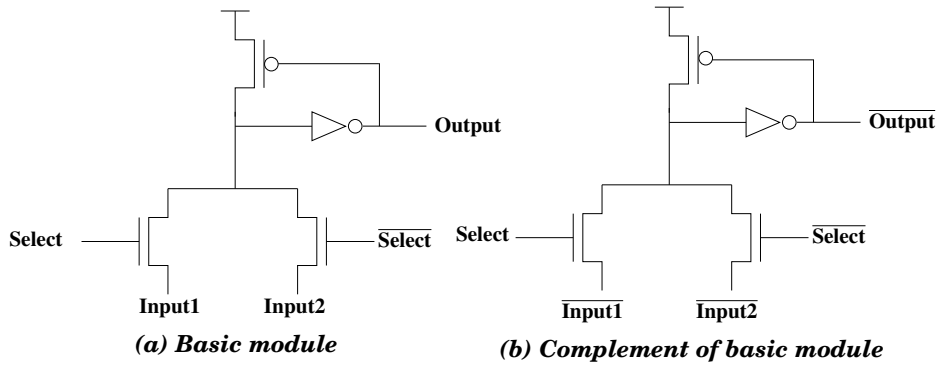


Figure 2.5: PTL based basic cell design.

There are many different types of pass transistor logic styles in use, such as:

- Complementary pass transistor logic (CPL)
- Swing restored pass-transistor logic (SRPL)
- Double pass transistor logic (DPL)
- Single-rail pass transistor logic (LEAP)
- The differential pass transistor logic (DPTL)

We use the complementary pass transistor logic with swing restorations. The basic multiplexer structure is given in Fig. 2.5. It is to be noted that pass transistor logic synthesis from BDDs is a well-studied domain [6, 14, 20].

2.4 Power consumption of CMOS logic

Complementary Metal Oxide Semiconductor (CMOS) logic is most widely used for realization of ICs in modern times. CMOS consist of a pull up and a pull down network. The pull up network is realized with PMOS transistors while the pull down network is realized with NMOS transistors. Pull up and pull down networks are functionally complementary in nature. Only one of this network is conducting at a time.

Three major components of power consumption are:

- Dynamic power: Dynamic power which is the major component of the of total power consumption depends on charging and discharging of load capacitance. Dynamic power consumption can be calculated by the following equation:

$$P_{dynamic} = \alpha C_{load} V_{DD}^2 f \tag{2.2}$$

where α is the switching activity factor of the circuit, C_{load} is the load capacitance including the parasitic capacitance, V_{DD} is the supply voltage and f is the circuit's operating frequency. α the switching activity of the transistors, in turn, depends on the input combinations applied to the transistors which is measured by $0 \rightarrow 1$ transitions on the output of CMOS gates. Dynamic power directly depends on the input combinations applied to the transistors if the supply voltage and frequency are constant.

- Short circuit power: Short circuit power dissipation happens when gate voltage is applied to CMOS gate and both the transistors are changing state, then both the pull up network and pull down network are conducting simultaneously for a short period of time and there exists a direct connection between the voltage source to ground. Short circuit power is also dependent on switching activity factor α .

- Static power (leakage power) : The static power or leakage power consumption of a circuit is given by the following equation:

$$P_{static} = I_{static} V_{DD} \tag{2.3}$$

where I_{static} is the current that flows between the supply rail when circuit is in idle mode that means there is zero switching activity in the circuit. Leakage power consumption emerges as a major portion of total power consumption for sub-micron technology, as CMOS technology scales down leakage power increases. A higher value of the threshold voltage helps to reduce the leakage current and therefore the lekage power.

2.5 Vulnerability of cryptosystems to side channel attacks

A cryptographic algorithm is a complex mathematical function that uses a secret key to encrypt information. The process by which message is encrypted by the secret key is referred to as encryption. The process by which an encrypted message is recovered in its earlier from is called decryption. The input of the encryption process is termed as plain text and the resultant output of the encryption process is termed as cipher text.

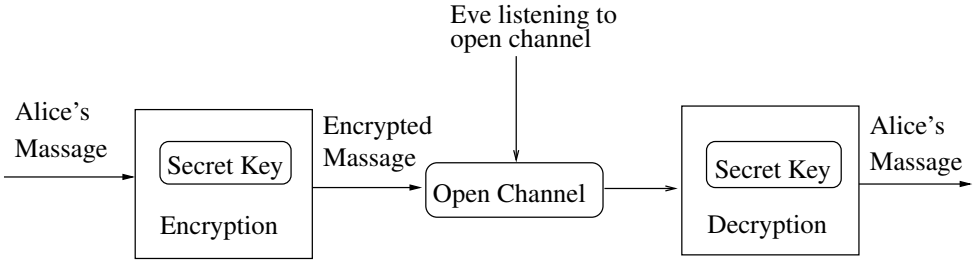


Figure 2.6: Conventional Process of cryptography.

Consider the famous cryptographic scenario where Alice has to send a secret message to Bob and does not want anyone else other then Bob to see this message. However, Alice sends the message through an insecure channel. Eve, a third person, is interested in knowing the content of the message. To protect the message from Eve, Alice send a cryptographic message to Bob. The entire mechanism is shown in Fig. 2.6. Depending upon encryption and decryption technique, this mechanism can be further classified in two ways.

2.5.1 Symmetric-Key encryption

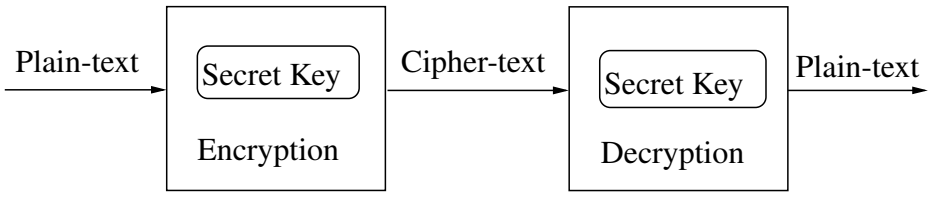


Figure 2.7: Symmetric-Key cryptography.

Alice encrypts the message with a secret key using an encryption algorithm and sends it to Bob. Bob after receiving the message, decrypts it using the same secret key (which Alice has used) with the help of a decryption algorithm. Here a single secret key has been used for both encryption and decryption. Let P be the plain text which Alice has sent to Bob, K be the secret key by which Alice encrypted P with encryption algorithm $E_k(x)$ and produced cipher text C. Bob decrypts C with the decryption algorithm $D_K(x)$ and regenerates P.

So, during encryption: $C = E_K(P)$ and during decryption $P = D_K(C)$
such that $D_K(E_k(x)) = E_k(D_K(x))$
In a nutshell, Alice: $C = E_K(P)$ Bob: $P_1 = D_K(C) = D_K(E_K(P)) = P$

The mechanism of the Symmetric-Key encryption is showing in Fig. 2.7. Depending upon the algorithm symmetric-key cryptography can be further classified into block ciphers and stream ciphers.

Block cipher

A block cipher takes an n-bit plain-text as a input and generates an n-bit cipher-text as output, where n is the block size. Diffusion and confusion techniques are used to encrypt data. The process by which redundancy in the plain-text and secret key are dissipated in the cipher-text is called diffusion. Resultant change in single input bit will be diffused over several cipher-text bits making it difficult for the attacker to gain knowledge of the plain text by analyzing the cipher-text. On the other hand, confusion is a process which makes the relationship between input and the cipher-text complex to make it difficult for the attacker to predict the patterns.

A product cipher combines two or more simple operations in a way that the resulting cipher is more secure than the individual components. These simple operations are meant to increase confusion or diffusion. An iterated block cipher is a cipher which involves sequential repetition of an internal functions which referred as a round functions. Two well known schemes for designing block ciphers are Substitution-Permutation (SP) networks and Feistel networks. We have experimented with the Lucifer and Present S-boxes which are all block ciphers.

Substitution-Permutation (SP) Substitution-Permutation (SP) networks is a product ciphers which generate after substitutions and permutations in different number of stages. The data is separated into smaller blocks during substitutions. For increasing the confusion, the values in these blocks are substituted for others. This method uses a look-up table which is referred as S-box. The influence of data from one part of the plain-text is diffused through the whole cipher-text by using swapping bits or combining values.

Feistel networks Feistel networks are a subset of SP networks. It also generates cipher after substitutions and permutations in different number of stages. We do not use this type of cipher in this thesis.

Stream cipher

A stream cipher operates on smaller units of plain-text, usually some bits. Stream cipher does not need fixed length of data, it operates on any length. It works on a continuous stream of data with a random number generator for encryption of the plain text. Cipher-text generated by a stream cipher will vary depending on when they are encountered during the encryption process. Stream cipher are generally faster and less complex in nature compared to block cipher. We do not use this type of cipher in this thesis.

2.5.2 Asymmetric-Key encryption

Key distribution is an important problem with symmetric cryptography. Take the classical example of cryptography where Alice has to securely communicate with Bob. So, in symmetric cryptography Alice and Bob has to exchange the secret key with other and they have to ensure the confidentiality of the key. But the channel is insecure. If there is a scenario where Alice has to communicate separately with a hundred different users he has to exchange that many secret keys before communicating. Here Asymmetric-Key Encryption turns out to be useful.

Here two keys are used; one is a public key and other is a private key. Alice

encrypts the message using Bob's public key. Bob decrypts the message using his own private key. The mechanism of Asymmetric-Key encryption is showing in Fig. 2.8. We do not use this method in this thesis.

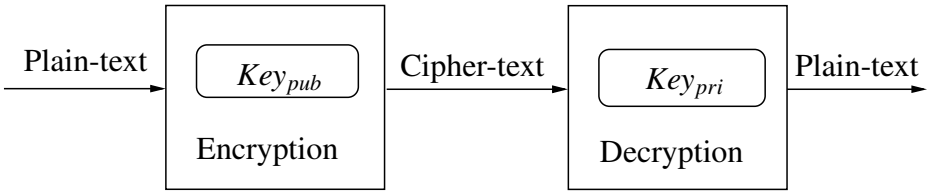


Figure 2.8: Asymmetric-Key cryptography.

2.6 Side channel attacks on cryptographic devices

The term *cryptanalysis* often refers to the study of ciphers, It is based on observing and analysing inputs and outputs of the cryptographic device. The cryptanalyst would attempt to extract the secret key based on these observations along with analysis and some knowledge of the implemented algorithms. Modern day cryptographic algorithms are mathematically more secure. It is almost impossible to mathematically break those systems and extract the systems' secret keys. However, their hardware implementations may be targeted and exploited for deducing the secret keys in use inside the system. Depending upon methods of the attacks it can be classified into two broad different category – (*i*) Invasive attacks and (*ii*) Non-invasive attacks

Invasive attacks are those which leave a physical evidence of tampering on the device. Common techniques are de-packing smart card chips, memory reverse engineering, and micro probing.

Non-invasive attacks are those that do not physically tamper with the device, instead they use information that is leaked from the device to attack. Non-invasive attacks are also commonly referred to as side channel attacks. The ability to attack devices such as smart cards on the fly, when those are being used, without leaving a trace of the attack has made side channel attacks very attractive. For this reason, securing devices from such attacks has gained significant practical importance.

2.7 Side channel attacks

A special category of non-invasive attacks are called side channel attacks where the physical implementation of a cryptographic system can be monitored during its execution and the traces obtained can be examined by an attacker to discover the the secret key used in the system. Timing information, power consumption electromagnetic leaks, fault injection and sound can provide additional information by which cryptographic systems can be exploited. A comprehensive study of side channel attacks can be found in [26, 30]. Side channel attacks can be classified into invasive, non-invasive, active and passive attacks. The information available to the attacker is a set of messages processed by the cryptographic device and the corresponding running time, power dissipation etc. and his goal is to recover the device's secret parameters using the available information. Different forms of side channel attacks are as follows.

- Timing attacks: attacks based on measuring how much time is taken to perform various computations [19, 24].
- Power monitoring attacks: attacks which make use of the varying power consumption by the hardware during computation [26, 30].
- Electromagnetic attacks: attacks based on leaked electromagnetic radiation which can directly provide information [1, 21].
- Acoustic cryptanalysis: attacks which exploit sound produced during computations [49].
- Differential fault analysis: in which secrets are discovered by introducing faults in a computation [8, 50].
- Data remanence: in which sensitive data are read after supposedly having been deleted [38].

Among all these forms of attacks, the power monitoring attacks are the most prominent threat to the cryptographic systems since power traces of operations can be easily obtained. Those power traces can be mathematically analysed to reveal the secret keys quite easily. In general, power dissipation of a circuit is proportional to its switching activity which, in turn, depends on the data that is being handled. The data

Table 2.1: Power consumed due to switching

Transitions	Power consumed
1→0	P_1
0→1	P_2
0→0	0
1→1	0

dependent power consumption can be exploited to leak away the secret information, specifically, distribution of 0's and 1's. This is explained by means of table 2.1 which shows the signal transitions of single rail static circuit. Let P_1 and P_2 be the powers consumed during the transitions 1→0 and 0→1, respectively. The amount of power consumed for these two transitions being different (i.e., $P_1 \neq P_2$), this asymmetry can be exploited to mount power analysis attacks. Depending upon how the power traces are monitored, their analysis can be further classified.

- Simple Power Analysis (SPA): Only a single power trace is examined to extract the secret key.
- Differential Power Analysis (DPA): It involves collecting large number of power traces and performing statistical analysis of the power variation with respect to changes in data values to extract the secret key.

There are broadly two types of countermeasures of DPA based on their level of application: (i) algorithmic level, and (ii) cell level. Our focus is on cell level countermeasures for DPA. A cell level approach ensures algorithmic agnostic DPA resistance and also enables speedy completion of the encryption and the decryption processes. Moreover, it avoids extra circuitry to the extent that is needed for implementing reported algorithmic level DPA countermeasures [4, 33].

2.7.1 Differential power analysis

DPA exploits the correlation between the data and the instantaneous power consumption of the cryptographic device. Though the correlation is very small, statistical

method is used to increase the efficiency. In this process the attacker uses a hypothetical model of the device under attack and then statistically analyses the correlation of power consumption from the actual device to the hypothetical model in order to find the secret key in use in the system. In DPA, bits of the key are deduced in stages. The choice of bits that are attacked first is guided by the attacker's knowledge of the device and the cryptographic algorithm in use. These key bits are usually referred to as a subkey. A DPA attack on a cryptographic module performing encryption is described below:

- The power consumption of the cryptographic device is recorded while it encrypts N different plain-text inputs with the same key and is denoted as a matrix $P_{1...N,1...T}$, where T is the number of points that are recorded per encryption. The number N is usually referred to as the number of traces.
- The attacker chooses an intermediate result of the executed algorithm that is a function of the plain-text and the subkey. Based on the plain texts and all possible values for the sub-key, hypothetical values for the intermediate results are calculated as a matrix $I_{1..2^k,1...N}$ where K is the number of subkey bits and 2^k is the number of possible values of the subkey.
- The attacker then determines a hypothetical power consumption value $H_{K,n}$ for every $I_{K,n}$
- The attacker reveals the correct subkey by correlating the hypothetical power consumption $I_{1..2^k,1...N}$ with the power traces $P_{1...N,1...T}$

2.7.2 Difference of means method

The working principle of this technique is to split the power traces into two groups for each key hypothesis based on a selection function. First the captured traces are partitioned $P_{1...N,1...T}$ into two sets, based on a selection function. The means of the power traces in both sets are calculated and the means of one set are subtracted from those of the other set (eg. if the LSB bit is 1, add current trace to set one else the other is set). A threshold α can also be used to partition the traces based on $H_{1..2^k,1...N}$. The equation of difference of means is given below [25]

$$R_{2^K,T} = P1_{(\forall N|H_{2^k,N}>\alpha),T} - P1_{(\forall N|H_{2^k,N}\leq\alpha),T} \tag{2.4}$$

The resulting difference of means matrix, $H_{1..2^k,1...T}$. will have a difference of mean trace for every key hypothesis. The difference of mean trace for correct key hypothesis will have significantly visible peaks when compared to the the other result traces.

2.7.3 Correlation power analysis

An attacker using CPA will acquire a set of N power consumption traces while a given algorithm is being computed $(w_i for 1 \leq i \leq N)$, and attempt to predict the Hamming weight of the computer word being manipulated at a chosen point in time for each acquired trace $(h_i for 1 \leq i \leq N)$ [10]. The correlation between these predictions H and the instantaneous power consumption of the set of acquired traces W, i.e.

$$\rho_{W,H} = \frac{cov(W,H)}{\rho_W \rho_H} \tag{2.5}$$

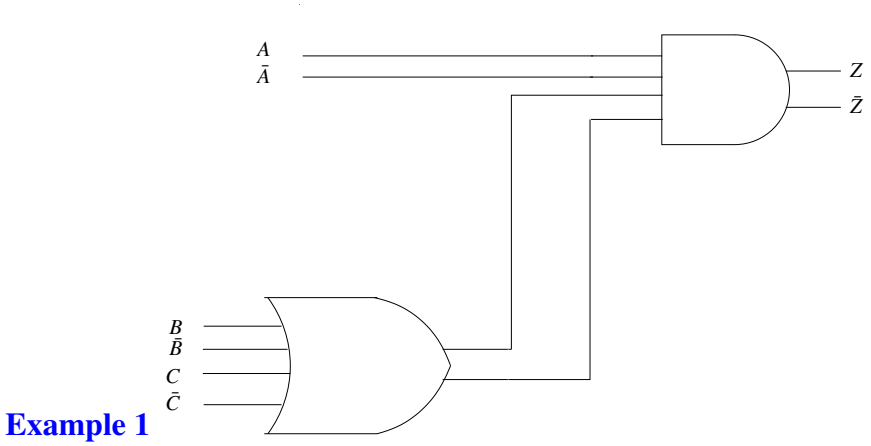
can be calculated to deduce where in the traces the chosen point in time appears. This involves generating a CPA trace that represents the correlation between H and W at each point in the acquired power trace

2.7.4 Early propagation effect

A physical gate can produce the correct output even before all its inputs change. Output of a gate may be determined by the logic values acquired by a subset of its inputs (say X). Changes on the other inputs may not have an effect on the outputs. However redundant transitions will effect the power consumption. Which can be monitored and these may help to deduce possible logic values of the lines in X.

This property can result in data dependent power consumption even for circuits implemented with balanced gates and with balanced routing [27]. This is illustrated in example 1 with a combinational circuit depicted in figure 2.9.

The data dependence can be observed in the timing differences of the gate transitions. Using a simple delay and power consumption model, the number of gate



Example 1

Figure 2.9: A example data-dependent power consumption due to early propagation.

Table 2.2: Power consumed due to switching

	Number of gate transitions	
A B C	After ΔT propagation delay	After 2ΔT propagation delay
0 0 0	2	0
0 0 1	2	0
0 1 0	2	0
0 1 1	2	0
1 0 0	1	1
1 0 1	1	1
1 1 0	1	1
1 1 1	1	1

transitions with respect to time for all the possible input combinations are shown in Table 2.2.

As seen in table 2.2, the number of gate transitions (and hence the power consumption) after "ΔT" propagation delay is dependent on the logical value of the A input. If A is a logical-zero there are always two gate transitions at "ΔT" time, while if it is a logical one there will be one logical transition at "ΔT" propagation delay time and another at "2ΔT" propagation delay time.

Chapter 3

Basic BDD based circuits with bottom pre-charge

In this chapter, a novel BDD based logic synthesis approach to counter power analysis attacks with two different bottom pre-charge logics is presented. We have devised a hardware countermeasure in the form of a Binary Decision Diagram (BDD) based dual rail circuits with two different pre-charging schemes, namely bottom pre-charge and symmetric NMOS based bottom pre-charge logic. For the first time, bottom pre-charge logic has been used in the design of such a cell.

The operation of this logic has four aspects, viz (*i*) A pre-charging phase. two pre-charging schemes have been presented (*a*) bottom pre-charge and(*b*) symmetric NMOS based bottom pre-charge. Both these circuits are specially designed to work with BDD based mechanism (*ii*) BDD based normal and complementary function realisations (*iii*) use of swing restoration for producing outputs with proper voltage level and (*iv*) use of voltage scaling and leakage power minimization to reduce overall power dissipation without hampering PAA resistance.

The operation of this logic is first described using a basic cell supporting fourteen logic functions including AND, OR, XOR, NOT, NAND, NOR. While any logic can be constructed using this basic cell, more optimized circuit realization is possible by utilizing the second aspect of this logic synthesis approach. This is illustrated through the design of two different S-boxes – Lucifer [40] and Present [9].

The rest of the chapter is organized as follows. In section 3.1, our BDD based synthesis technique with bottom pre-charge logic and symmetric NMOS based pre-charge logic is elaborated. Design of a basic cell with fourteen logic functions, a 1 bit adder, a 2 bit adder along with the Lucifer and the Present S-boxes, using our bottom pre-charge logic technique is given in section 3.2, and our symmetric NMOS based pre-charge technique, is given in section 3.3. The chapter is concluded in section 3.4.

3.1 Basic BDD based circuits with bottom pre-charge

This section elaborates our BDD based logic synthesis approach for countering power analysis attacks with symmetric NMOS and bottom pre-charge logic. The organization of such circuit is indicated in the Figs. 3.2 and 3.1. The operation of this scheme has four aspects which are describe below.

3.1.1 Aspect-1: Pre-charge generation logic

Pre-charge generation logic has been used extensively in pass transistor logic (PTL) based circuit designs. Pre-charging has long been used to reduce the number of transistors in logic gates and also to reduce the power dissipation. Pre-charging also helps to counter the skew in power dissipation of pre-charging free circuits when a long stream of 1s or 0s is produced in output. We propose two different types of pre-charging circuits, each having its own merits, as described in the following subsections.

Bottom pre-charge generation logic

The bottom pre-charge logic of aspect 1 consists of a pair of PMOS and NMOS transistor T_1 and T_2 Fig. 3.1, respectively. As shown in the Fig. 3.1 drains of transistor T_1 and T_2 are connected and their gates are also connected and driven by the pre-charging signal 'Pre'. The source of transistor T_1 is connected to VDD while the source of transistor T_2 is connected to the input. To minimize early propagation effect PMOS transistor width is taken few times higher than the width of NMOS transistor.

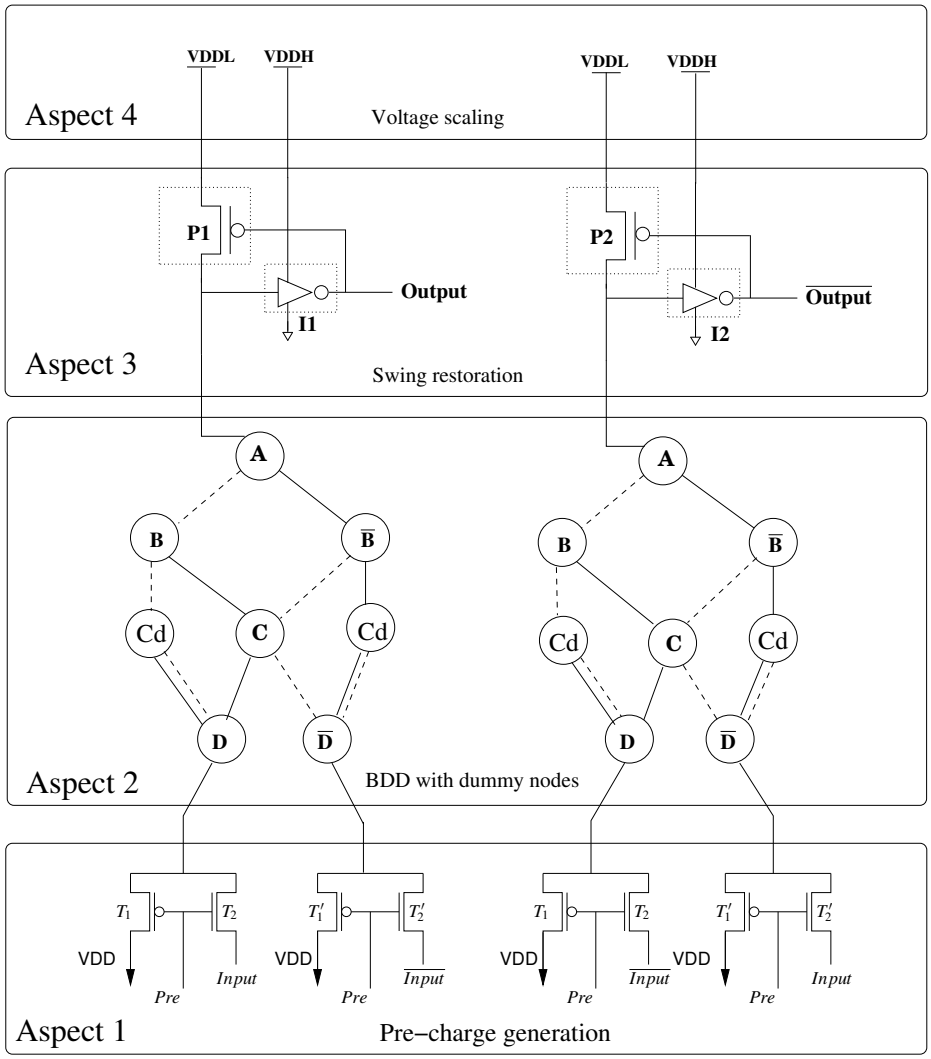


Figure 3.1: The four aspects of BDD based logic synthesis with bottom pre-charge.

Bottom pre-charge logic operates in two phases, namely, pre-charging phase and evaluation phase which are described below.

Pre-charging phase: When the pre-charge is low, the PMOS transistor T_1 in the pre-charge logic circuit shown in Fig. 3.1 is ON, and the NMOS transistor T_2 is OFF. This ensures blocking of the external input signal in the pre-charge logic circuit. The current from VDD of the pre-charge logic circuit flows through the BDD network to reach the swing restoration parts of the circuit. In the swing restoration circuit, P_1 which is designed as a weak PMOS transistor, is ON because voltage coming to the inverter through the BDD network drives it low, there by discharging the intrinsic capacitor at the output through the NMOS transistor of the inverter. So, charge from VDD at swing restoration circuit will enhance the signal strength and make the output

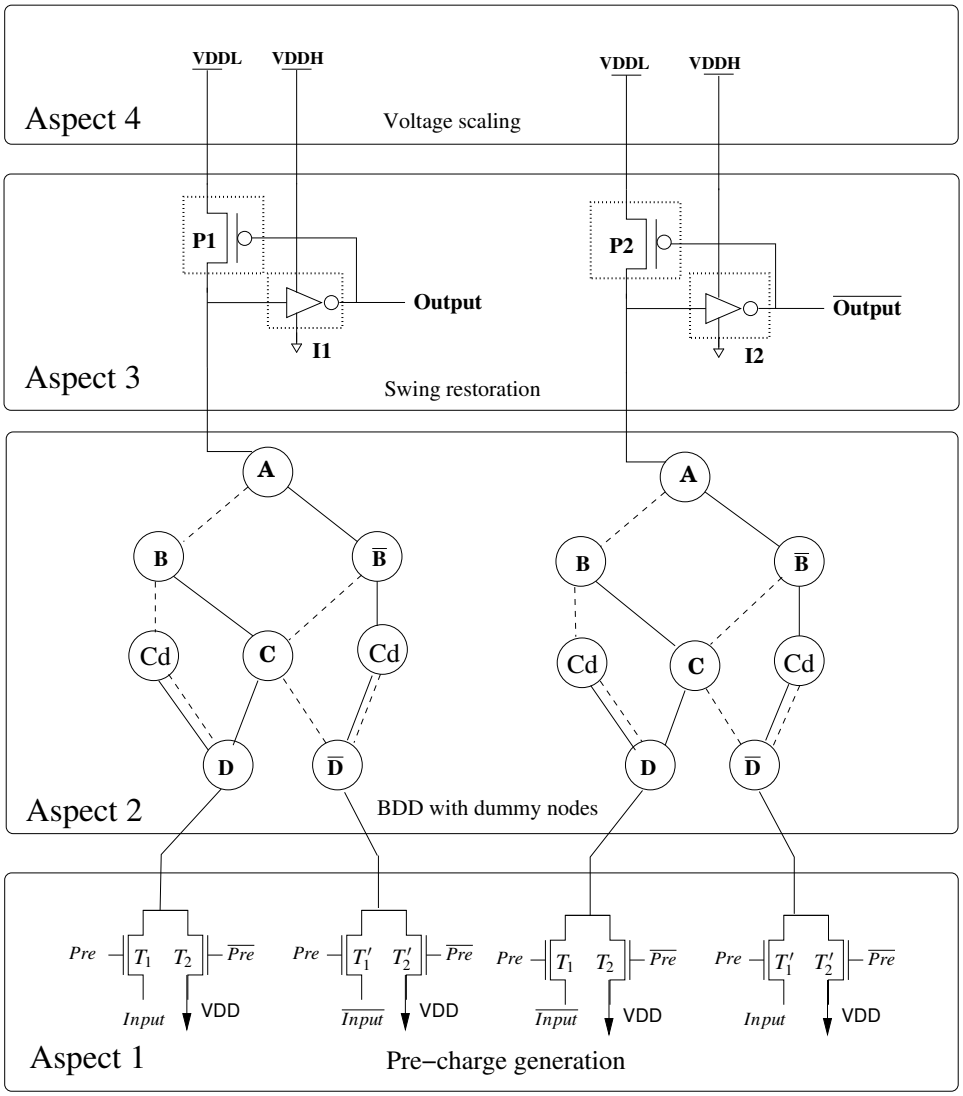


Figure 3.2: The four aspects of BDD based logic synthesis with symmetric NMOS based pre-charge.

signal stable at zero.

Evaluation phase: When pre-charge is high, the PMOS transistor T_1 in the pre-charge logic circuit is OFF, and the NMOS transistor T_2 is ON. So, the external input will go through the BDD network and reach the swing restoration network producing the output.

The bottom pre-charge logic is similar to the technique proposed in [47]. In BDD based architecture input comes from the bottom, i.e., input driving voltage signals are applied at the bottom nodes and then they progress upward in a BDD tree. Bottom pre-charging ensures that a constant capacitance is charged or discharged independent

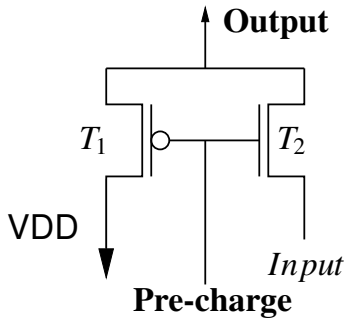


Figure 3.3: Design of the Bottom pre-charge logic.

of the data that is being processed. Complementary BDDs have also been used with the associated pre-charge circuitry.

Symmetric NMOS based bottom pre-charge generation logic

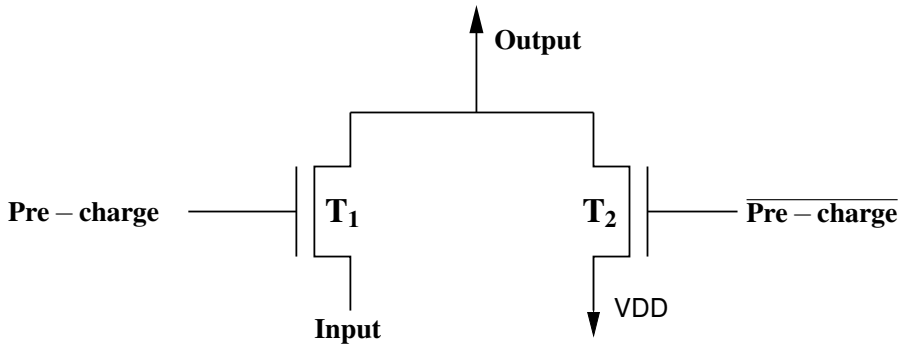


Figure 3.4: Design of the symmetric NMOS based bottom pre-charge logic.

The functionality of the pre-charge generation logic is generally achieved using PMOS and NMOS connected networks. However, this may lead to early propagation effects which may turn out to be a source of security vulnerability [27]. We attempt to overcome this vulnerability by way of our symmetric NMOS bottom pre-charge generation mechanism, described below with the help of Fig. 3.4. Here, the two NMOS transistors connected at their drains with symmetrical widths can be considered as the two complementary (True and False) children of a BDD node.

The gate of transistor T_1 is connected with the **Pre – charge** signal and its source is connected to the **Input** signal, whereas for transistor T_2, its gate is connected with the complement of the **Pre – charge** signal ($\overline{\textbf{Pre – charge}}$) and its source is connected to

the Voltage source (VDD). At a time only one of the two NMOS transistor can remain conducting, thus, preventing the logic circuit from short circuiting. The pre-charging ensures that a constant capacitance is charged or discharged independent of the data that is being processed.

The pre-charge logic operates in two phases — pre-charging phase and evaluation phase.
Pre-charging phase: This phase remains operational when the **Pre − charge** is 'LOW', when transistor T_1 is in the OFF state and transistor T_2 is in the ON state. So, VDD, i.e.a high voltage, will go through the circuit; this leads to the output going high irrespective of the value of the **Input** line.
Evaluation phase: This phase remains operational when the **Pre − charge** is 'HIGH', then the transistor T_1 stays ON and the transistor T_2 stays OFF. The **Input** value then passes through the transistor T_1 into the circuit and produces the same value at the output.

Advantages of symmetric NMOS based bottom pre-charge generation logic

There are several advantages of symmetric NMOS based pre-charge generation logic over PMOS and NMOS connected pre-charge generation logic, which are as follows:

- Total power consumed by BDD based design is less than that of PMOS and NMOS connected design of equal width [22] (our experiments show a reduction by 15% in power consumption when BDD based NMOS-NMOS pre-charge generation logic is used).
- Due to the same width and length of the NMOS transistor, symmetric NMOS design has less EPE compared to PMOS and NMOS connected design.
- Two NMOS transistors connected in parallel can be easily represented using a finger pattern in the layout, leading to real estate savings as compared to the other designs.
- Multiple fan-out or single fan-out can be easily realized by changing the source value of the transistor T_2 . For single fan-out use VDD at the source of transistor T_2; for multiple fan-out use GND instead and increase one extra inverter in the

swing restoration part. This is a significant advantage because only one extra inverter is needed for increasing the fan-out.

3.1.2 Aspect-2: BDD based tree network to realize logic functions

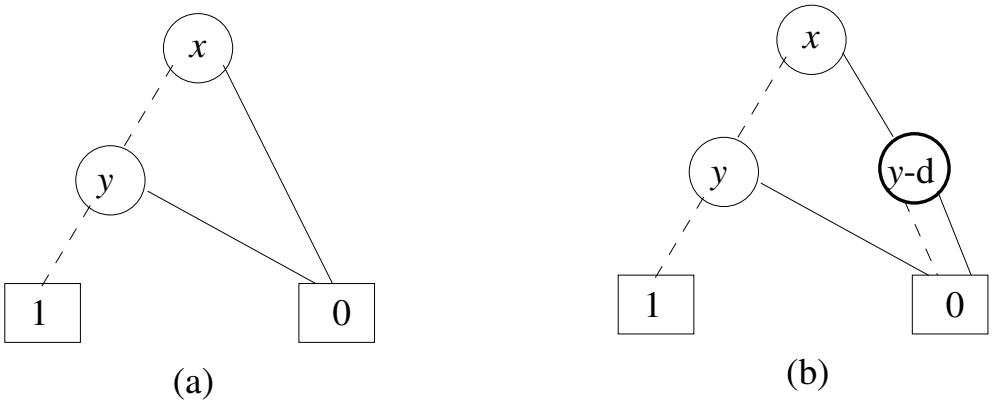


Figure 3.5: (a) BDD for $\overline{x+y}$. (b) BDD for $\overline{x+y}$ after dummy node insertion.

Any Boolean logic function can be realized by a BDD. Here ROBDD-based design principles turn out to be useful since they minimize the logic function and produce a smaller BDD. The CUDD tool [39] is used to generate the ROBDDs from specific Boolean functions. For realization of circuits, each decision branch is replaced by two NMOS transistors with complementary gate voltages. To make the circuit DPA attack resistant, three different measures have been incorporated.

Effective variable ordering: A good variable ordering reduces the size of the BDD. Due to the limited number of functional variables *static variable ordering* mechanism is used for variable ordering of the BDDs.

Complementary tree: Power consumption occurs due to charging and discharging of capacitors in a circuit. To make the total charging and discharging constant, the complementary logic have been used; thus, while one BDD tree produces the $output$, the other produces $\overline{output}$. The complementary BDD is realized by inverting the leaf node values of the original BDD, thus the circuit geometries remain invariant.

Dummy nodes: The ROBDD generated, however, may not have all paths of the same length. Therefore, if the input signals pass through different number of stages of the same BDD-based circuit, then there will be a difference in their delay times, and consequently, in the arrival of the outputs. To avoid the difference in time delays, dummy nodes are added so that timing delays of all the paths in the BDD are equalized. As a

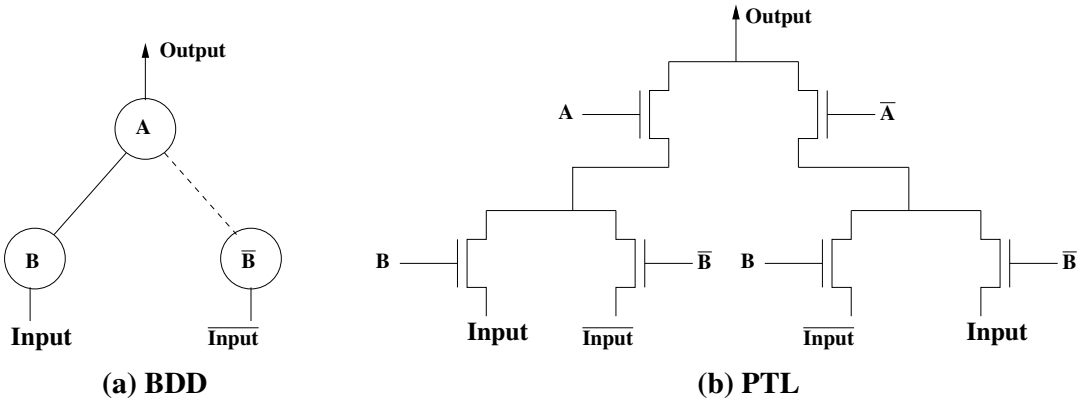


Figure 3.6: Pass transistor logic based circuit realization from a BDD.

result, computation along each decision branch will be through same number of transistors giving rise to identical delay. This is shown in Fig. 3.5(b), where *y*-d represents the node *y* as a dummy node.

The BDDs thus obtained are realized using PTL. An example of PTL based circuit realization of a BDD node BDD is shown in Fig. 3.6. Each BDD node is replaced by two connected NMOS transistor which are gated with complementary driving signals. The source of the transistor is driven either by a input (in normal or complemented forms) or by the drains of child circuits. Identical width NMOS transistor are used for providing identical delay for any identical number of transistor paths. At any point of time there exists a single path between root node to ground in the corresponding realized circuits.

3.1.3 Aspect-3: Swing restoration logic

Swing restoration ensures good output driving capabilities of the proposed circuit mechanism. Swing restoration consists of a weak PMOS **P1** transistor which is minimal width in nature (for a certain process technology) and an inverter **I1**. It ensures that the outputs always stay at a proper voltage level. In our design, for each BDD decision node, the signal strength is degraded by V_{th} of the NMOS transistor. Thus swing restoration at the gate outputs is necessary. When the BDD network produces 1, the inverter **I1** in Fig. 3.1 and Fig. 3.2 which has re-generative property produces output 0; this, in turn, sets PMOS **P1** to ON state and VDDL boosts the signal strength.

When the BDD network produces 0, the inverter produces output 1, **P1** is set to OFF state and VDDL does not reach the network.

3.1.4 Aspect-4: Voltage scaling and leakage power minimization

Voltage scaling: The voltage scaling approach has been used to reduce the total power dissipation. Energy dissipation of a single transistor is $\frac{1}{2}C_L(VDD)^2$, where C_L is the load capacitance of the transistor and VDD is the supply voltage. Thus, the total energy is directly proportional to the square of the supply voltage. In our design, the main transistor network works on a lower supply voltage; only inverter in the swing restoration parts works on a higher supply voltage. Therefore, the circuit achieves lower power dissipation without impeding the data invariant power dissipation properties.

Leakage reduction: Note that in sub-90nm technology, a significant portion of the overall power consumption is accredited to leakage power. Leakage current is inversely proportional to the threshold voltage V_{th} of the transistor. To minimize the leakage current of the circuit and consequently, the overall power consumption, the transistors **P1** and **P2**, and the load transistors of the inverters **I1** and **I2** in Fig. 3.1 are chosen to have high threshold voltage V_{th}. Similarly the transistors **P1** and **P2**, and the load transistors of the inverters **I1** and **I2** in Fig. 3.2 are chosen to have high threshold voltage V_{th}.

3.1.5 Circuit synthesis of bottom pre-charge logic by combining four aspects

The operational steps of the circuit in Fig. 3.1 synthesized by combining the four aspects is elaborated below. First, consider the normal (left) circuit of Fig. 3.1. When pre-charge (*Pre*) is zero, then the supply voltage VDD flows through the transistor corresponding to T_2 NMOS of pre-charge circuit and thus the pre-charge circuit produces 1. This signal flows through the multiplexer logic (BDD tree of Aspect 2) and reaches the swing restoration part where it is inverted producing 0 at the **Output**. Thus, the

output is always 0 (independent of the input value) when pre-charge is zero. Since the signal strength reduces during propagation, the VDDH is connected to the inverter to restore the signal strength at the time of output. However, the rest of the circuit is driven by a lower supply voltage VDDL, thus, reducing the total power requirement of the circuit. When pre-charge is one, the transistor corresponding to *Pre* remains closed and the **Input** signal comes through the multiplexer logic and produces the output after swing restoration. The complementary (right) circuit of Fig. 3.1 operates in a similar manner.

3.1.6 Synthesis of symmetric NMOS based bottom pre-charge logic by combining four aspects

The operational steps of the circuit in Fig. 3.2 synthesized by combining the four aspects is elaborated below. First, we consider the normal (left) circuit of Fig. 3.2. When pre-charge (*Pre*) is zero, then the supply voltage VDD flows through the transistor corresponding to $\overline{Pre}$ and the pre-charge circuit produces 1. This signal flows through the multiplexer logic (BDD tree of Aspect 2) and reaches the swing restoration part where it is inverted producing 0 at the **Output**. Thus, the output is always 0 (independent of the input value) when pre-charge is zero. Since signal strength reduces during propagation, the VDDH is connected to the inverter to restore the signal strength at the time of output. However, the rest of the circuit is driven by a lower supply voltage VDDL, thus, reducing the total power requirement of the circuit. When pre-charge is one, the transistor corresponding to *Pre* remains closed and the **Input** signal comes through the multiplexer logic and produces the output after swing restoration. The complementary (right) circuit of Fig. 3.2 operates in a similar manner.

3.2 Applications of bottom pre-charge logic

In this section designing of circuits with NMOS and PMOS transistors based bottom pre-charge logic has been elaborated. Designing of the basic cell is discussed first, there after complex circuit structures such as 2-bit adder and different substitution boxes are elaborated. Output power and current waveform which are statistically analyzed by attackers to reveal the system key are also plotted. Power and current

Table 3.1: Basic cell functions using multiplexing

Input1	Input2	Select	*output*	$\overline{output}$
$\overline{A}$	$\overline{B}$	B	$A.B$	$\overline{A.B}$
$\overline{B}$	$\overline{A}$	B	$A+B$	$\overline{A+B}$
A	$\overline{A}$	B	$A.\overline{B}+\overline{A}.B$	$\overline{A.\overline{B}+\overline{A}.B}$
$\overline{B}$	$\overline{A}$	S	$A.\overline{S}+B.S$	$\overline{A.\overline{S}+B.S}$
A	$\overline{B}$	B	$\overline{A}.B$	$A+\overline{B}$
$\overline{B}$	A	B	$\overline{A}+B$	$A.\overline{B}$
$\overline{A}$	$\overline{A}$	A	A	$\overline{A}$

waveforms are generated using simulating the schematic capture. Details of the tool used is given below.

- Design Tool: Cadence Virtuoso IC design tool
- Technology: UMC 65nm process technology
- Version : 5.1.41
- Process technology specification: mixed mode/RF
- Operating temperature : $30^{\circ}C$
- Supply Voltage : 0.9 – 1.1 volt
- Operating frequency : 500 MHz

3.2.1 Basic cell design using bottom pre-charge logic

Our first objective is to design a dual rail basic cell resistant to the DPA attacks. A simple logic cell has been designed to realize fourteen logic functions including 2-input AND, OR, XOR, NOT, NAND, NOR, etc. using multiplexing based on the aspects mentioned above. To ensure DPA resistance, the current/power characteristics of the cell must remain invariant to their inputs. This is achieved by constructing a basic module and a *complement* of the basic module within a single basic cell. Both the modules have the same circuitry, however, the basic module is fed with the original

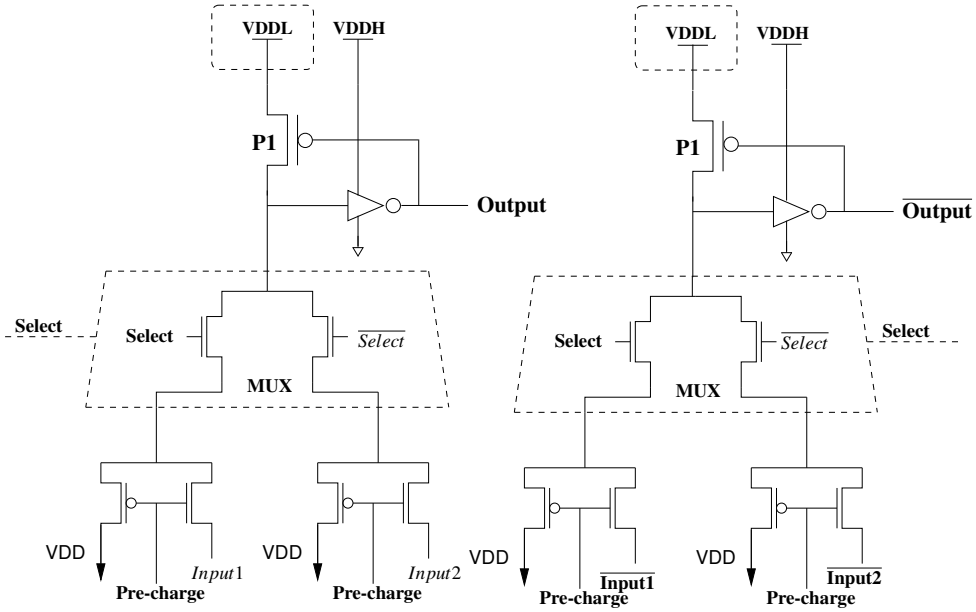


Figure 3.7: Design of the basic cell with bottom pre-charge logic.

inputs, whereas, the complementary module is fed with the inverted inputs. Design of the basic cell is given in Fig. 3.7. In the design, PTL-based logic has been used for the NMOS transistor circuitry, which is basically multiplexing in nature and it is referred to as a MUX. Swing restoration design is used at the output. Depending on the select line of the MUX and the input parameters, fourteen logic functions can be realized as given in table 3.1. Complementary logic has been used during basic cell design so that at a time only one transistor is open, thus preventing short circuit. In the experimentation done with the above mentioned tool, all possible input combinations are applied to the basic multiplexer circuit. Glitch free outputs with acceptable voltage level are generated. The current and power consumption with inclusion of bottom pre-charge (NMOS and PMOS transistor) logic for multiplexer circuit can be found in Fig. 3.10 where circuit simulation period is 600 ns. This current waveform is symmetric in nature which makes it hard for the attackers to analyze the variance of power with different input combinations.

3.2.2 Adder design with bottom and BDD based pre-charge logic

The adder is one of the most widely used circuit. Here we choose a 2-bit adder as a basic building block because it consumes less power in comparison with two single

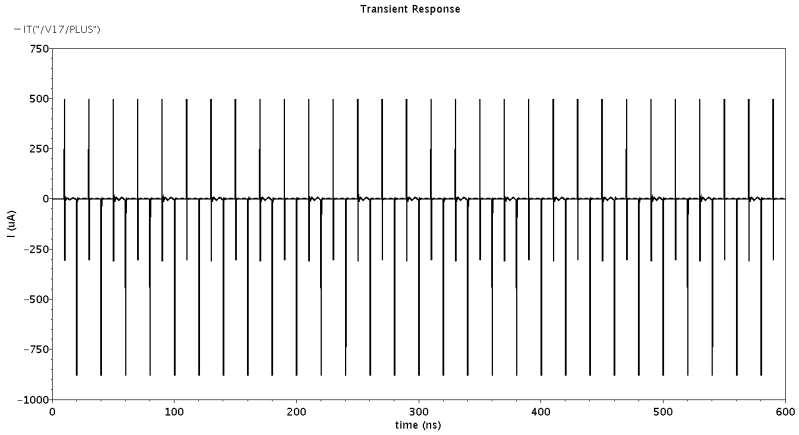


Figure 3.8: Current waveform :time (ns) vs current (μA).

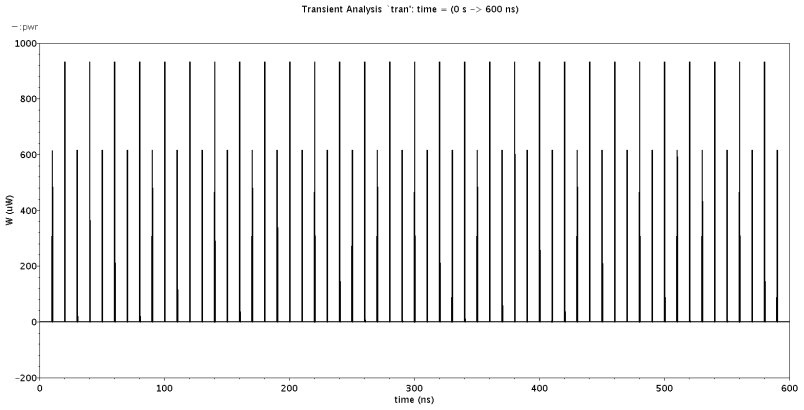


Figure 3.9: Power waveform :time (ns) vs power (μW).

Figure 3.10: Waveforms for the basic cell with bottom pre-charge

bit adders. While designing a DPA resistant adder, it has to be ensured that its path lengths are balanced. Here ROBDD-based design principles turn out to be useful. The CUDD tool [39] is used to generate the ROBDDs.

Since each node of the BDD can be represented using a MUX, same basic cell design principles are used here. Pass transistor based implementation has been done where each node is represented by 2 transistors. Basic equations for the 2 bit adder are as follows [34]:

$$S_0 = Pre[C_{in}(A_0B_0 + \overline{A_0B_0}) + \overline{C_{in}}(\overline{A_0}B_0 + A_0\overline{B_0})] \tag{3.1}$$

$$S_1 = Pre\{[C_{in}(A_0 + \overline{A_0}B_0) + \overline{C_{in}}(A_0B_0)](A_1B_1 + \overline{A_1B_1}) + [C_{in}(\overline{A_0B_0}) + \overline{C_{in}}(\overline{A_0} + A_0\overline{B_0})](\overline{A_1}B_1 + A_1\overline{B_1}\} \quad (3.2)$$

$$C_{out} = Pre\{[C_{in}(A_0 + \overline{A_0}B_0) + \overline{C_{in}}(A_0B_0)](A_1 + \overline{A_1}B_1) + [C_{in}(\overline{A_0B_0}) + \overline{C_{in}}(\overline{A_0} + A_0\overline{B_0})](A_1B_1)\} \quad (3.3)$$

$$\overline{C_{out}} = Pre\{[C_{in}(A_0 + \overline{A_0}B_0) + \overline{C_{in}}(A_0B_0)](\overline{A_1B_1}) + [C_{in}(\overline{A_0B_0}) + \overline{C_{in}}(\overline{A_0} + A_0\overline{B_0})](\overline{A_1} + A_1\overline{B_1})\} \quad (3.4)$$

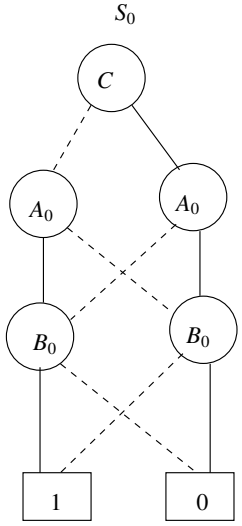


Figure 3.11: Resultant BDD after dummy node insertion of the corresponding 3.1 equation.

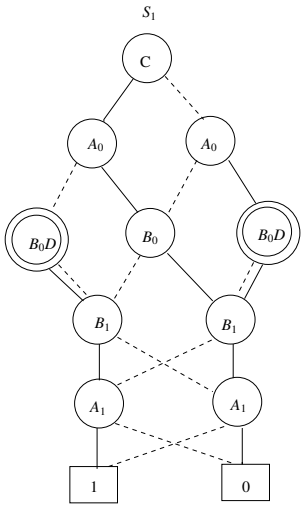


Figure 3.12: Resultant BDD after dummy node insertion of the corresponding 3.2 equation.

The corresponding BDD for the above four equation are given in Figs. 3.11 , 3.12, 3.13 and 3.14. While constructing the 2-bit adder using these equations, it is decomposed into four sub parts, namely, `Sum0` circuit (Fig. 3.15) and `Sum1` circuit (Fig. 3.16) which compute the sum of the first bit and the second bit respectively, along with carry and complementary carry circuits (Fig. 3.17). Both the pre-charging technique can be

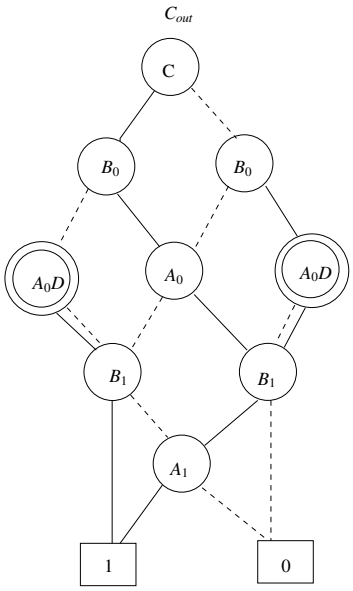


Figure 3.13: Resultant BDD after dummy node insertion of the corresponding 3.3 equation.

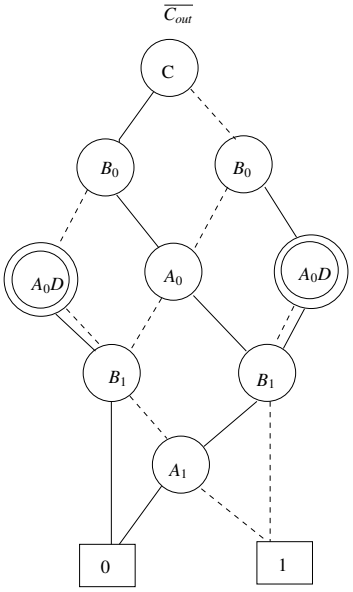


Figure 3.14: Resultant BDD after dummy node insertion of the corresponding 3.4 equation.

applied to the pre-charge generation box. The current and the power waveforms of the adder with bottom pre-charge logic given in Fig. 3.20.

Experimentation is done with the above mentioned tool. All possible input combinations are applied to the adder circuit. Glitch free outputs with acceptable voltage level are generated. The current and power consumption with inclusion of bottom pre-charge (NMOS and PMOS transistor) logic for adder circuit can be found in Fig. 3.10, where circuit simulation period is 600 ns. This current waveform is symmetric in nature which makes it hard for the attackers to analyze the variance of power with different input combinations.

3.3 Applications of symmetric NMOS based pre-charge logic

In this section, design of circuits with symmetric NMOS transistor based bottom pre-charge logic has been elaborated. Design of the basic cell are discussed first thereafter complex circuit structures such as different substitution boxes, Lucifer and Present,

have been elaborated. Output power and current waveforms are statistically analyzed by attacking schemes to reveal the system key are also plotted. Power and current waveforms are generated by simulating schematic capture. Details of the tool are given below.

- Design Tool: Cadence Virtuoso IC design tool
- Technology: UMC 65nm process technology
- Version : 5.1.41
- Process technology specification: mixed mode/RF
- Operating temperature : $30°C$
- Supply Voltage : 0.9 – 1.1 volt
- Operating frequency : 500 MHz

3.3.1 Basic cell design with symmetric NMOS based pre-charge logic

A single basic cell has been designed to realize several logic functions using multiplexing based on the aspects mentioned above. The current/power characteristics of the cell must remain invariant to the inputs to ensure PAA resistance. Depending on the select line of the MUX and the input parameters, fourteen logic functions including 2-input AND, OR, XOR, NOT, NAND, NOR can be realized as shown in Table 4.1. In this table, *A* and *B* denote the inputs and *S* denotes the select line of the MUX.

Design of the basic cell with voltage scaling is given in Fig. 3.21. Circuitry with the contour is needed only when voltage scaling is used. The layout of the basic cell is shown in Fig. 3.22. The current and the power waveforms of the basic cell with symmetric NMOS transistor based pre-charge generation logic and dual voltage source is given in Figure 3.23 and Figure 3.24, respectively, where circuit simulation period is 1 μs. Experimentation is done with the above mentioned tool. All possible input combinations are applied to basic multiplexer circuit. Glitch free outputs with acceptable voltage level are generated. This current waveform is *symmetric* in nature

Table 3.2: Lucifer and Present S-box functions

Lucifer	x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	S[x]	C	F	7	A	E	D	B	0	2	6	3	1	9	4	5	8
Present	x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	S[x]	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

which makes it hard for the attackers to analyze the variance of power with different input combinations.

3.3.2 BDD based S-box design symmetric NMOS based pre-charge logic

Lucifer is the earliest block cipher [40], whereas, Present is a modern lightweight block cipher [9]. S-box used in the Lucifer and the Present is 4-bit $\times$ 4-bit; function S: $\mathbb{F}_4^2 \rightarrow \mathbb{F}_4^2$. The action of the S-box in hexadecimal notation is given in Table 3.2. Each hexadecimal input `x` represents the four input bits of the S-box, numbered as `V0`, `V1`, `V2` and `V3`. The corresponding output `S[x]` represents the four output bits of the S-box, numbered as `out0`, `out1`, `out2`, and `out3`. Thus, each output bit is a function of the four input bits.

The ROBDDs obtained from the CUDD tool are unbalanced, hence they need to be balanced prior to implementation. The normal and complementary circuits corresponding to the balanced BDDs for the output bits of the Present S-box are shown in Fig. 3.29. In this figure, the dummy nodes have been highlighted with dashed boxes.

Experimentation is done with the above mentioned tool. All possible input combinations are applied to both S-box circuits. Glitch free outputs with acceptable voltage level are generated. The current and power consumption with inclusion of bottom pre-charge (symmetric NMOS) logic for both Lucifer and Present circuit can be found in Figures 3.32, 3.33, 3.30 and 3.31, where circuit simulation period is 1 μs. This current waveform is symmetric in nature which makes it hard for the attackers to analyze the variance of power with different input combinations.

3.4 Conclusion

DPA has been known to pose serious challenges in designing secured systems. In this chapter, ROBDD based dual rail circuit designs of DPA resistant basic cell and a 2-bit adder have been presented. For the first time, bottom pre-charge logic has been used in the design of the basic cell. It ensures that a constant capacitance is charged or discharged independent of the data that is being processed ensuring identical delays and power consumption. The ROBDD based architecture, along with dummy node insertion for path balancing, serves minimize the early propagation effect. The use of ROBDD also results in an optimised circuit. Bottom pre-charge is expensive in terms of area but it provides highest form of resistance from power attacks, timing attacks and early propagation effects.

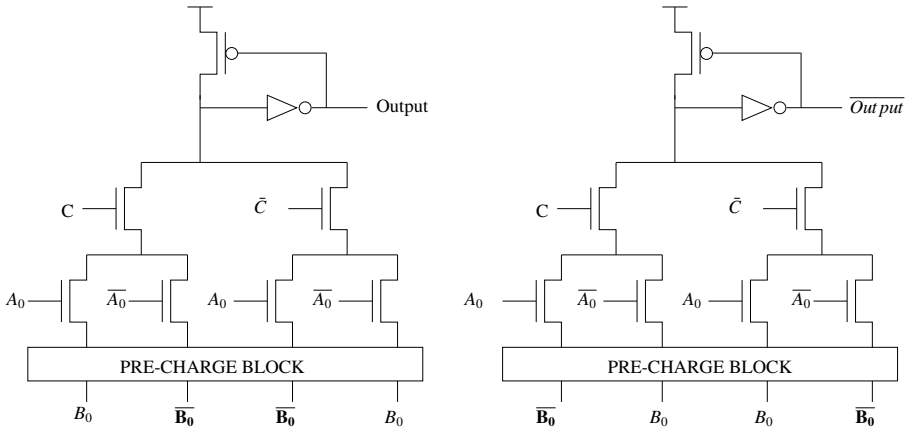


Figure 3.15: Sum0 circuit.

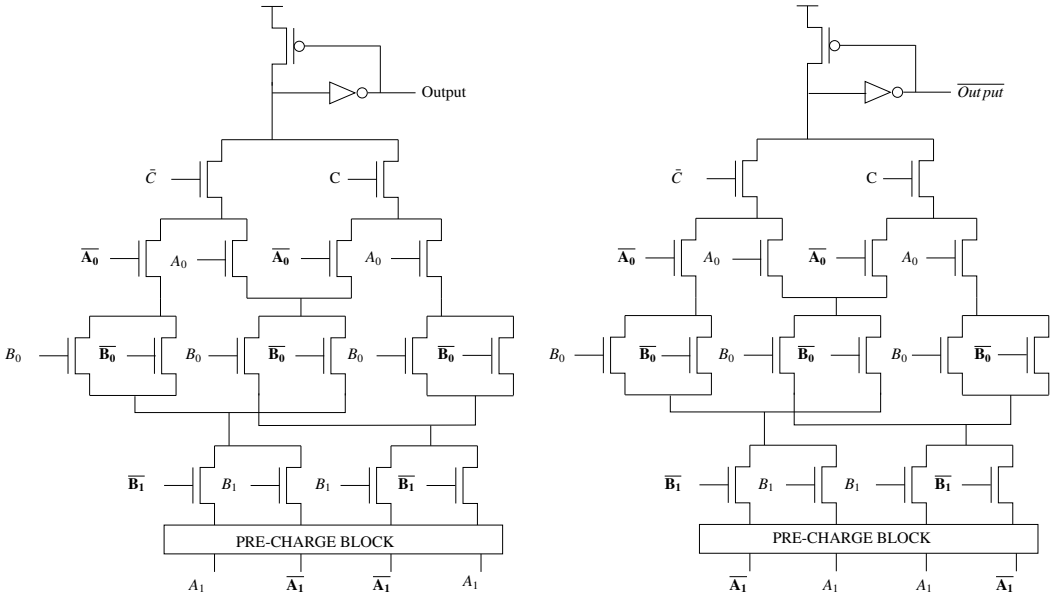


Figure 3.16: Sum1 circuit.

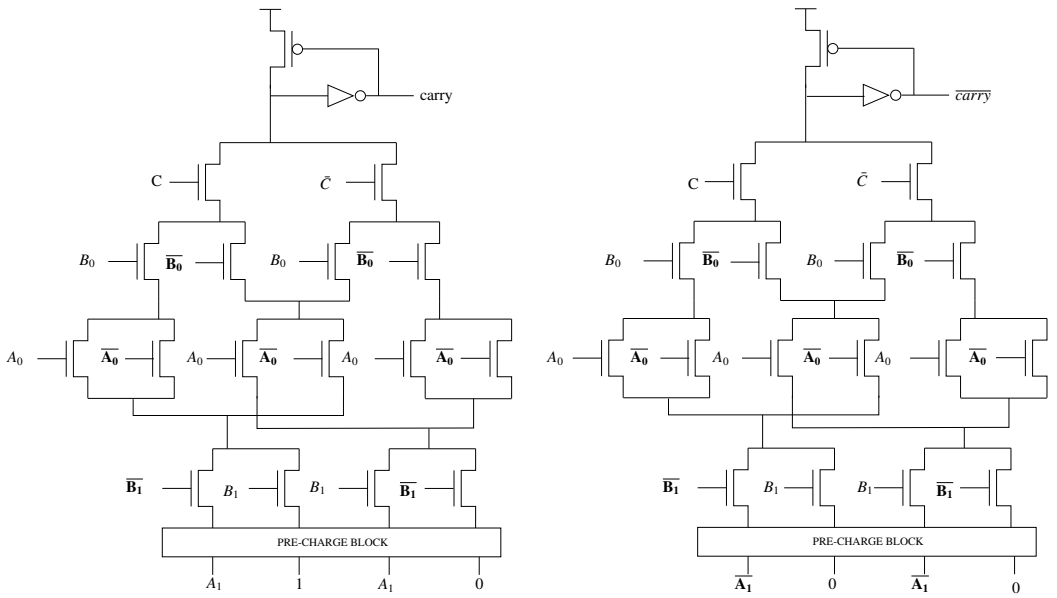


Figure 3.17: Carry and complementary carry circuits.

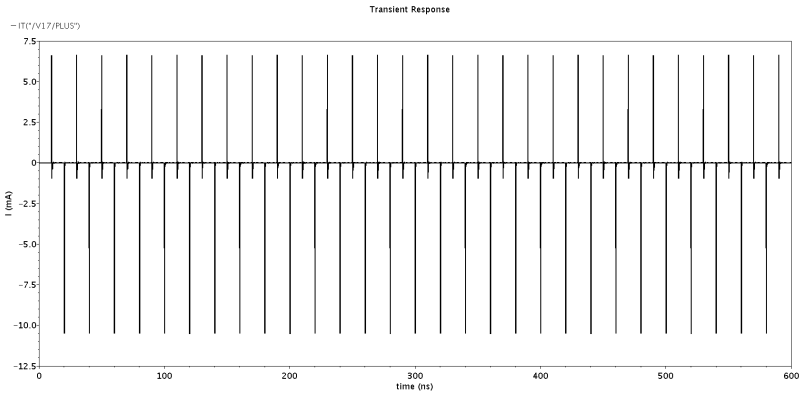


Figure 3.18: Current waveform for the 2 bit adder :time (ns) vs current (μA).

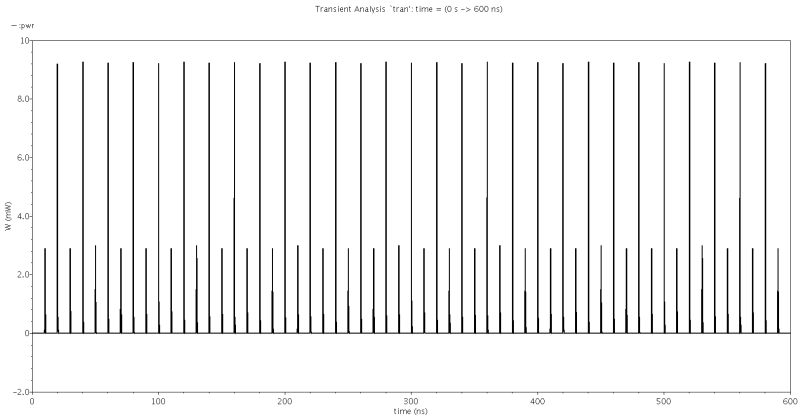


Figure 3.19: Power waveform for the 2 bit adder :time (ns) vs power (μW).

Figure 3.20: Power and current waveform for the 2 bit adder with bottom pre-charge logic.

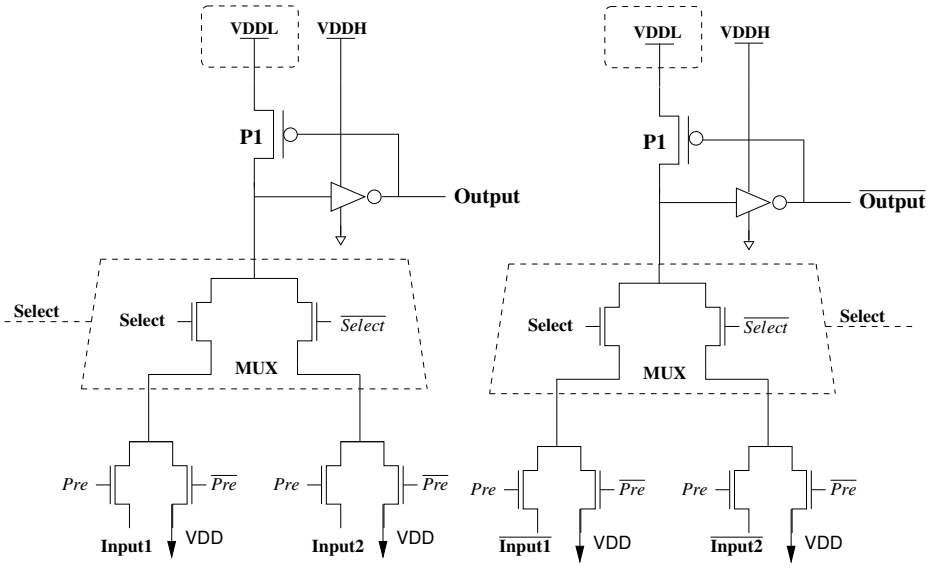


Figure 3.21: Design of the basic cell with voltage scaling using symmetric NMOS based pre-charge logic.

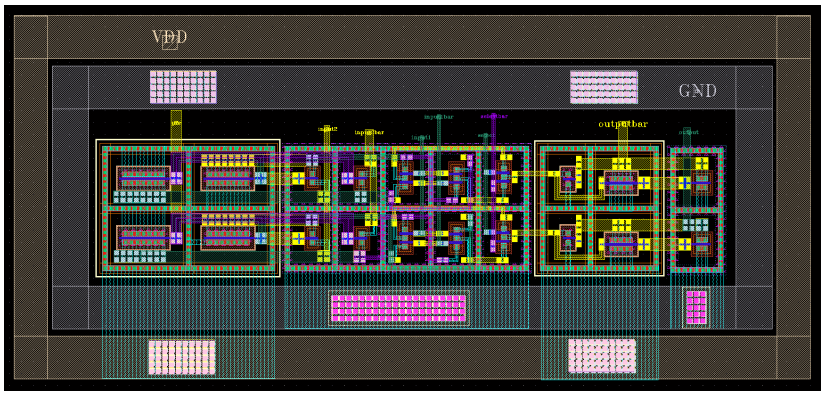


Figure 3.22: Layout of the basic cell using symmetric NMOS based pre-charge logic.

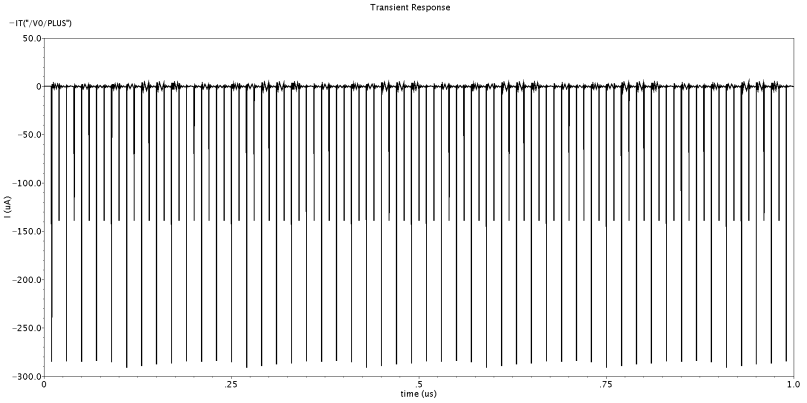


Figure 3.23: Current waveform characteristics of the basic cell with the symmetric NMOS based pre-charge generation logic and the dual voltage source: time (ns) vs current (μA).

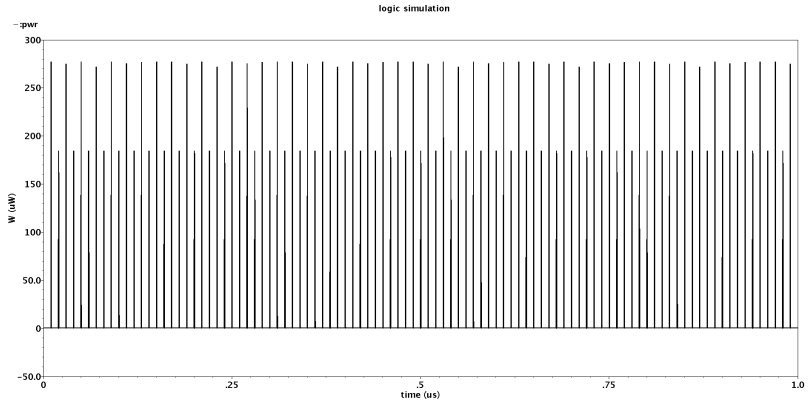


Figure 3.24: Power waveform characteristics of the basic cell with the symmetric NMOS based pre-charge generation logic and the dual voltage source: time (ns) vs power (μW).

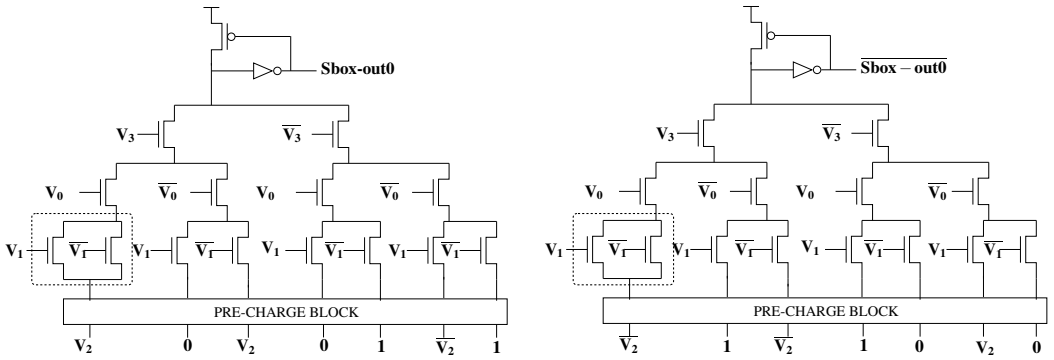


Figure 3.25: Normal and complementary circuits for out0.

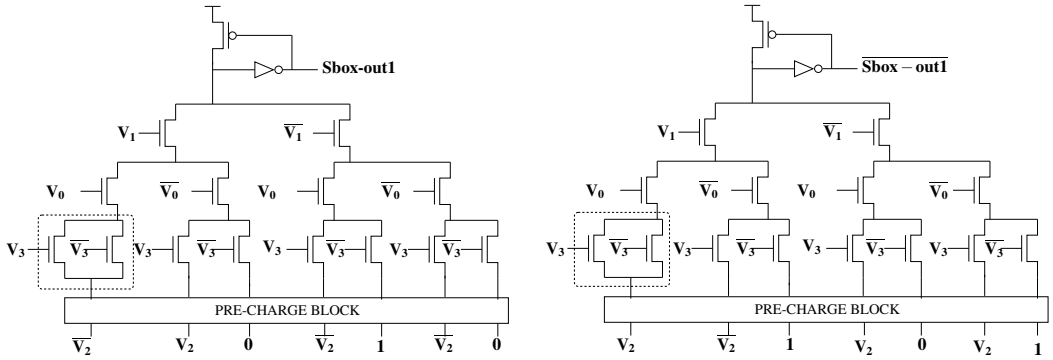


Figure 3.26: Normal and complementary circuits for out1.

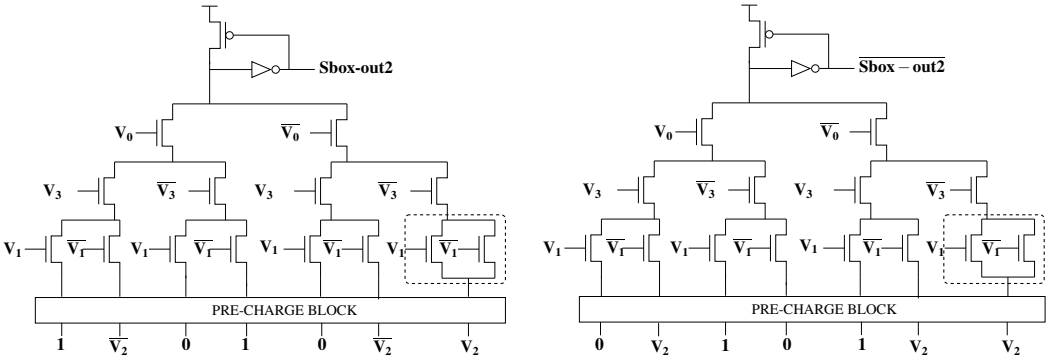


Figure 3.27: Normal and complementary circuits for out2.

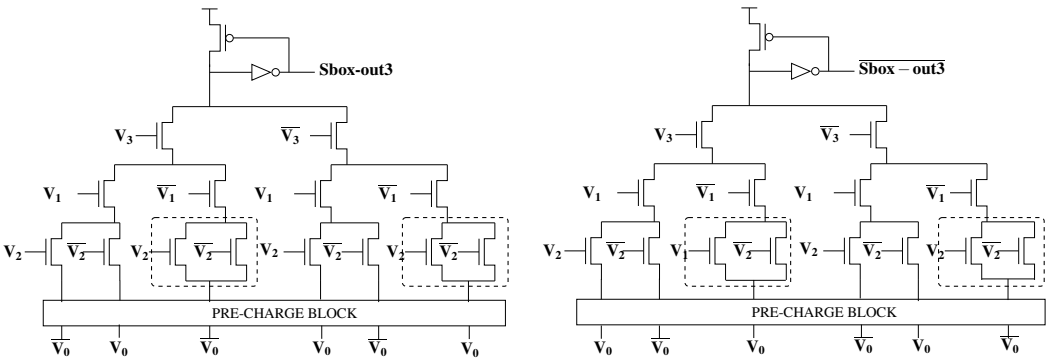


Figure 3.28: Normal and complementary circuits for out3.

Figure 3.29: Normal and complementary circuits for the output bits of Present S-box with the dummy nodes highlighted using dashed boxes.

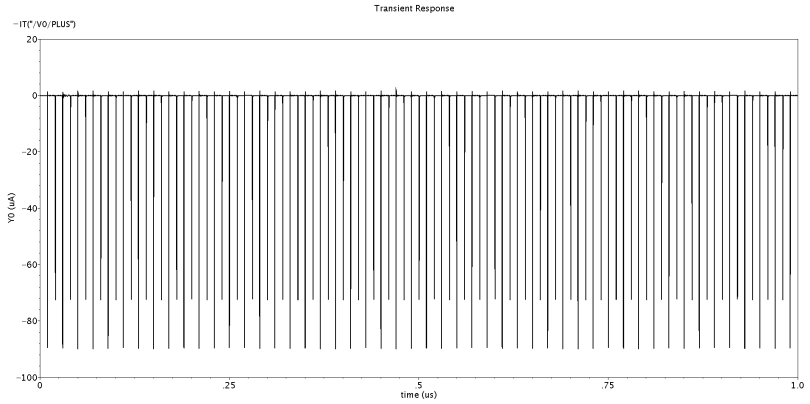


Figure 3.30: Current waveform characteristics of the Present S-box with the symmetric NMOS based pre-charge generation logic : time (ns) vs current (mA).

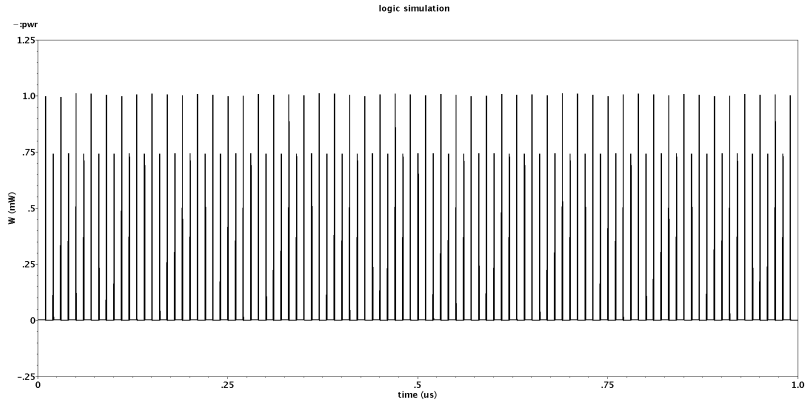


Figure 3.31: Power waveform characteristics of the Present S-box with the symmetric NMOS based pre-charge generation logic : time (ns) vs power (mW).

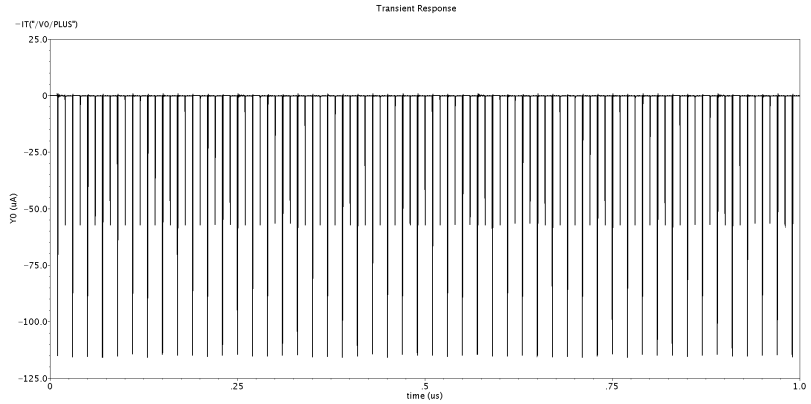


Figure 3.32: Current waveform characteristics of the Lucifer S-box symmetric NMOS based pre-charge generation logic : time (ns) vs current (mA).

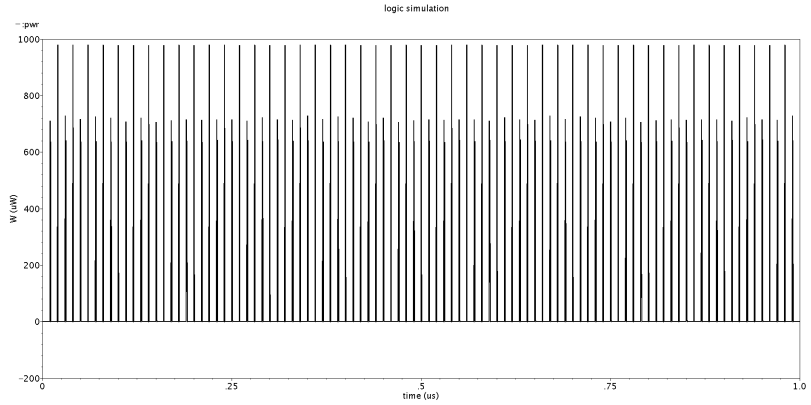


Figure 3.33: Power waveform characteristics of the Lucifer S-box symmetric NMOS based pre-charge generation logic : time (ns) vs power (mW).

Chapter 4

BDD based circuits with various other features

In this chapter, two BDD based dual-rail logic circuit schemes have been developed to counter PAAs. These circuit schemes feature novel pre-charge generation, voltage scaling with leakage power minimization and early propagation effect (EPE) resistance mechanisms. The basic scheme features low power circuitry and extremely low peak power variation. A particular variation of this scheme features superior EPE characteristics at the cost of marginal increase in power and area over the basic scheme. The previous chapter dealt with pre-charge generation from the bottom. In this chapter, other two kinds of pre-charge generations will be explored. Bottom pre-charge is expensive in terms of area but it provides highest form of resistance from power attacks, timing attacks and early propagation effects. In this chapter, two area efficient pre-charge logic are proposed although bottom pre-charge is slightly superior in terms of resistance of overall security.

Te two pre-charge logics described here are, top pre-charge logic and top-bottom pre-charge logic. The operation of the customized designs has three aspects, viz. (*i*) pre-charge generation; (*ii*) realization of normal (un-complemented) and complemented functions with path balanced BDDs; (*iii*) voltage scaling and leakage power minimization for reducing total power consumption.

The operation of the customized designs is first described using a basic cell supporting fourteen logic functions including AND, OR, XOR, NOT, NAND, NOR.

While any logic can be constructed using this basic cell, more optimized circuit realization is possible by utilizing the normal and the complemented function realisations with path balanced BDDs. This is illustrated through the design of two different S-boxes:- Lucifer [40] and Present [9]. Experimental results obtained with the basic cell and the two S-box realisations demonstrate that our logic outperforms DP-BDD and SDMLp in terms of peak power variation, average power consumption and average current consumption while exhibiting comparable propagation delay.

In section 4.1, our BDD based synthesis technique has been elaborated. Design of a basic cell with fourteen logic functions and two different S-boxes, all constructed using our technique, is given in section 4.3 and section 4.3. The chapter is concluded in section 4.4.

4.1 BDD based circuits with various other features

Here a BDD based logic synthesis approach for countering PAAs with two different pre-charge generation logics are describe As indicated in Fig. 4.1 and Fig. 4.2, our approach has three aspects.

4.1.1 Aspect-1: Pre-charge generation

We propose two different types of pre-charge circuits, each having their own merits, as described next.

Top pre-charge logic

A PMOS transistor **P1** and an NMOS transistor **N1**, such that their gates are tied together and the drain of **P1** is connected to the drain of **N1**, which drives the input of the inverter **I1**, to regenerate the output of the BDD network which feeds to the source of **N1**. This circuitry is shown in the Fig. 4.1 in the box labeled Aspect 1. It operates in two phases.
Pre-charging phase: When pre-charge is 0, transistor **N1** is OFF; so, no output from

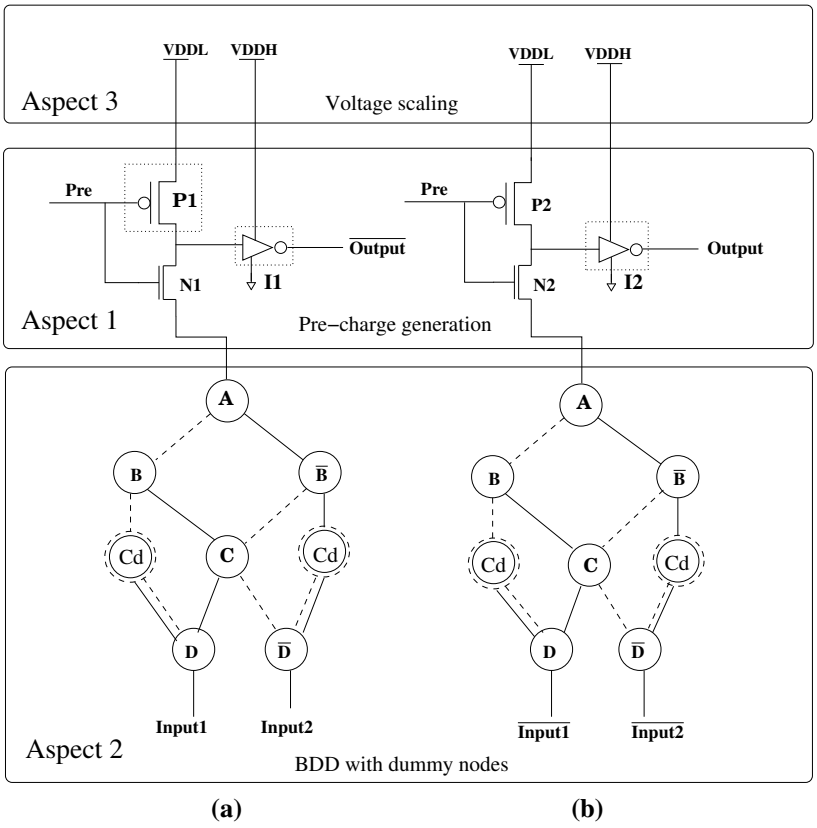


Figure 4.1: The three aspects of BDD based logic operation with top pre-charge logic.

the BDD network reaches the pre-charge circuit. However, since transistor **P1** is ON, voltage from VDD comes to the pre-charge circuit and after inversion produces 0 as output.

Evaluation phase: When pre-charge is 1, transistor **N1** is ON and transistor **P1** is OFF; consequently, BDD network output reaches the inverter **I1** and produces the inverted output.

Top-bottom pre-charge logic

It consists of a transistor PMOS (**P1**) along with an inverter (**I1**) connected to the top of the BDD network and two NMOS transistor(**N1** and **N2**) connected to the input nodes of the BDD network as shown in Fig. 4.3 (b). It operates in the following two phases.

Pre-charging phase: When pre-charge is 0, transistor **N1** and **N2** are OFF; so, no input reaches the BDD network. Since transistor **P1** remains ON, voltage comes from the

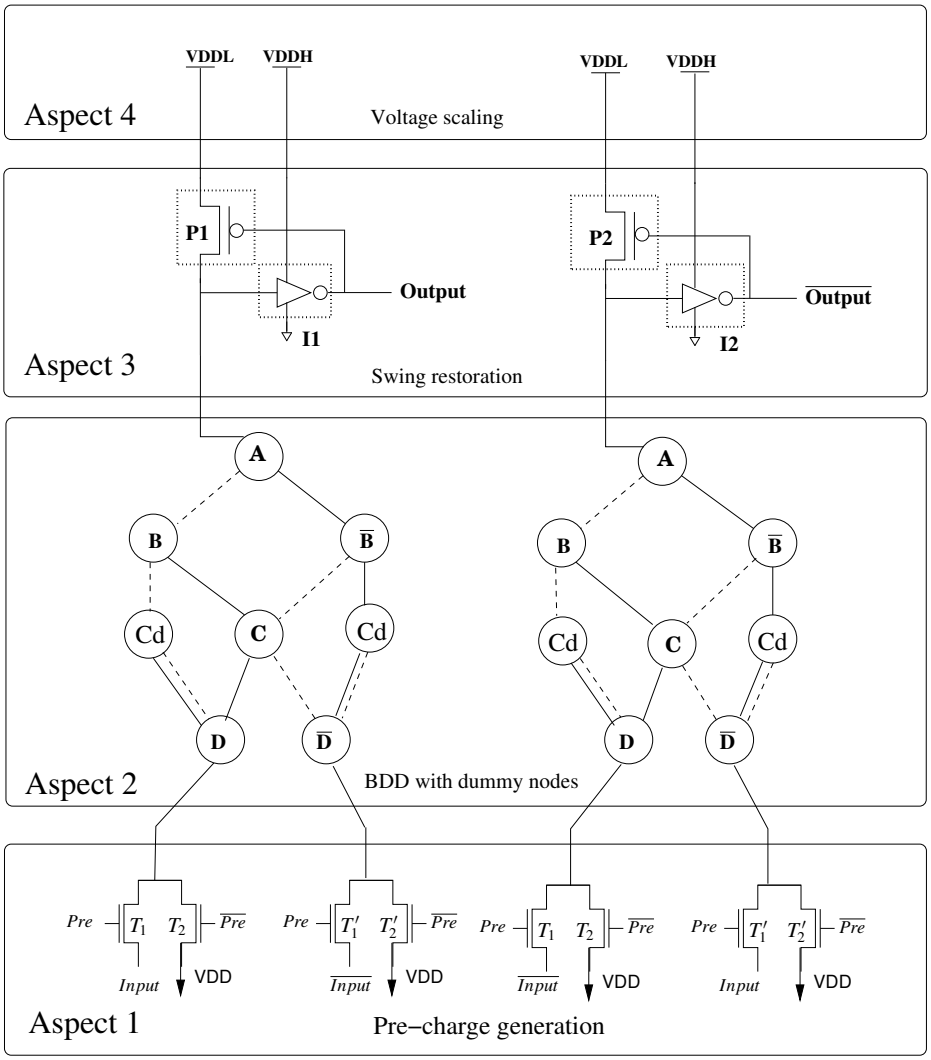


Figure 4.2: The three aspects of BDD based logic operation with top-bottom pre-charge logic.

VDD and after inversion produces the output 0.

Evaluation phase: When pre-charge is 1, transistor **N1** and **N2** are ON and **P1** is OFF; so, input comes through the BDD network, reaches the inverter **I1** and produces the inverted output.

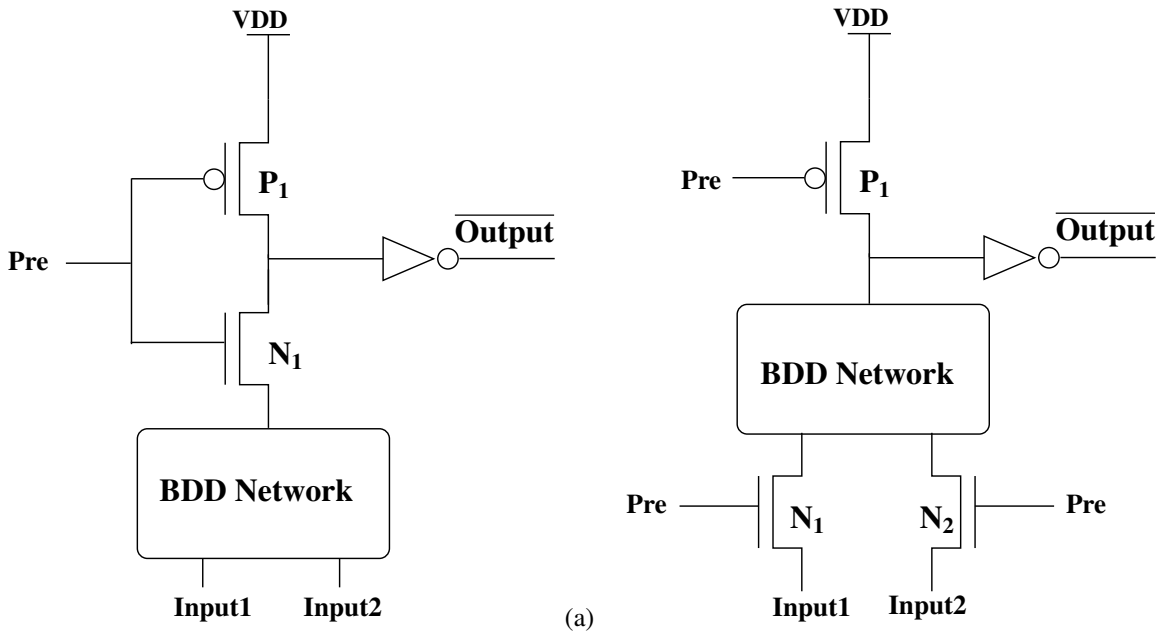


Figure 4.3: Basic structure of the basic cell with (a) top pre-charge logic, (b) top-bottom pre-charge logic.

4.1.2 Aspect-2: BDD based realisation of logic functions through the network

A BDD [12] is a graphical representation of a Boolean function. It is a folded binary tree where the input variables appear as the intermediary nodes and 0 and 1 as the terminal nodes, and each edge is labeled with 0 or 1 to denote the possible valuations of the variable appearing at the node from which the edges emanate. Given an assignment of 0s and 1s to the Boolean variables, one starts at the root node of the BDD and

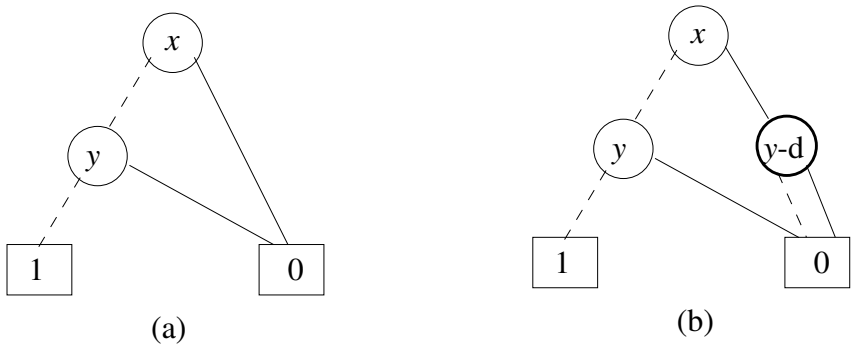


Figure 4.4: (a) BDD for $\overline{x+y}$ (b) BDD for $\overline{x+y}$ after dummy node insertion.

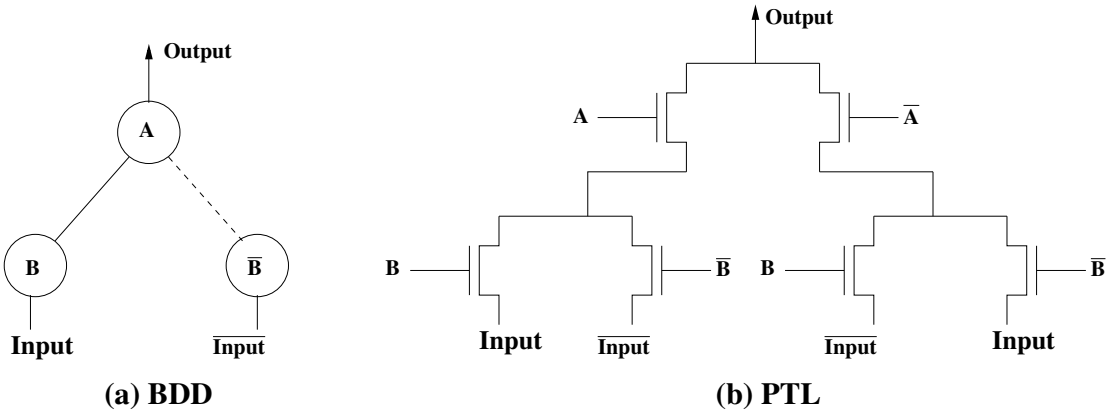


Figure 4.5: Pass transistor logic based circuit realization from a BDD.

travels down the tree according to the assigned values; the value of the terminal node reached gives the corresponding Boolean function's value. For example, Fig. 4.4(a) shows the BDD corresponding to the Boolean function $\overline{x+y}$.

For realization of circuits, pass transistor logic (PTL) has been used and each decision branch is replaced by two NMOS transistors with complemented gate voltages. PTL based circuit realization from a BDD is shown in Fig. 4.5. To increase the efficacy of the circuits against PAAs, the following measures have been taken.

Effective variable ordering: A good variable ordering reduces the size of the BDD.

Dummy nodes: The ROBDD generated, however, may not have all the paths of the same length. Therefore, if the input signals pass through different number of stages of the same BDD based circuit, then there will be a difference in their delay times, and consequently, in the arrival of the outputs. This may lead to EPE which, in turn, may lead to sophisticated data dependent attacks [27]. To avoid the difference in delay times, dummy nodes are added so that delay timings of all the paths in the BDD are equalized. As a result, computation along each decision branch will be through same number of transistors giving rise to identical delays. This is shown in Fig. 4.4(b), where y-d represents the node y as a dummy node.

Complemented tree: Power consumption occurs due to charging and discharging of capacitors in a circuit. To make the total charging and discharging power constant, the complemented tree is constructed; therefore, while one BDD tree produces the **Output**, the other produces $\overline{\textbf{Output}}$. The complemented BDD is realized by inverting the leaf node values of the original BDD, thus the circuit geometries remain invariant.

4.1.3 Aspect-3: Voltage scaling and leakage power minimization

Voltage scaling: The voltage scaling approach has been used to reduce the total power dissipation. Energy dissipation of a single transistor is $\frac{1}{2}C_L(VDD)^2$, where C_L is the load capacitance of the transistor and VDD is the supply voltage. Thus, the total energy is directly proportional to the square of the supply voltage. In our design, the circuit works on a lower supply voltage when pre-charge is 0 (pre-charge phase) and works on a higher supply voltage when pre-charge is 1 (evaluation phase). Therefore, the circuit achieves lower power dissipation without impeding the data invariant power dissipation properties.

Leakage reduction: Note that in sub-90nm technology, a significant part of the overall power consumption is due to leakage power. Leakage current is inversely proportional to the threshold voltage V_{th} of the transistor. To minimize the leakage current of the circuit and thereby the overall power consumption, the transistors **P1** and **P2**, and the load transistors of the inverters **I1** and **I2** in Fig. 4.2 are chosen to have high value V_{th}. Similarly the transistors **P1** and **P2**, and the load transistors of the inverters **I1** and **I2** in Fig. 4.1 are chosen to have high value V_{th}.

The operational steps of the circuit in Fig. 4.2 synthesized by combining the three aspects is elaborated below.

When pre-charge (**Pre**) is zero, the transistors **N1** and **N2** remain OFF, so no input reaches the BDD network. However, the transistor **P1** remains in ON state, and thus the supply voltage VDDL flows through the transistor and after getting inverted produces 0 at $\overline{\textbf{Output}}$. Thus, the output is always 0 (independent of the input value) when pre-charge is zero. When **Pre** is one, **P1** remains OFF while **N1** and **N2** remain ON. Hence, the input signals flow through the BDD tree of Aspect 2 and produces the inverted output at $\overline{\textbf{Output}}$. Since signal strength reduces during propagation, the VDDH is connected to the inverter **I1** to restore the signal strength at the time of output. Note that the circuit is driven by a lower supply voltage VDDL when **Pre** is 0, thus reducing the total power requirement of the circuit. The complementary circuit Fig. 4.2(b) operates in a similar manner.

In Fig. 4.1(a), the transistor **P1** remains ON and **N1** remains OFF when **Pre** is 0, thus producing 0 at $\overline{\textbf{Output}}$; whereas, when **Pre** is 1, the states of the transistors

reverse and the inverted output is produced at $\overline{\textbf{Output}}$. Thus, the basic functionality of the circuits in Fig. 4.1 and Fig. 4.2 remains the same.

4.2 Applications of BDD based logic with top-bottom pre-charge

In this section, design of circuits with NMOS and PMOS based top-bottom pre-charge logic has been elaborated. Design of basic cell has been discussed first then complex circuit structures of two different substitution boxes have been elaborated. Output power and current waveforms are statistically analyzed with the objective of revealing the system key are also plotted. Power and current waveforms are generated by simulating schematic capture. Details of the tool used are given below.

- Design Tool: Cadence Virtuoso IC design tool
- Technology: UMC 65nm process technology
- Version : 5.1.41
- Process technology specification: mixed mode/RF
- Operating temperature : $30°C$
- Supply Voltage : 0.9 – 1.1 volt
- Operating frequency : 500 MHz

4.2.1 BDD based basic cell design

A single basic cell has been designed to realize several logic functions using multiplexing based on the aspects mentioned above. The power/current characteristics of the cell must remain invariant to the inputs to ensure PAA resistance. Depending on the select line of the MUX and the input parameters, fourteen logic functions including AND, OR, XOR, NOT, NAND, NOR can be realized as shown in Table 4.1. In

Table 4.1: Basic cell functions using multiplexing

Input0	Input1	Select	$out\,put$	$\overline{out\,put}$
$\overline{A}$	$\overline{B}$	B	$A.B$	$\overline{A.B}$
$\overline{B}$	$\overline{A}$	B	$A+B$	$\overline{A+B}$
A	$\overline{A}$	B	$A.\overline{B}+\overline{A}.B$	$\overline{A.\overline{B}+\overline{A}.B}$
$\overline{B}$	$\overline{A}$	S	$A.\overline{S}+B.S$	$\overline{A.\overline{S}+B.S}$
A	$\overline{B}$	B	$\overline{A}.B$	$A+\overline{B}$
$\overline{B}$	A	B	$\overline{A}+B$	$A.\overline{B}$
$\overline{A}$	$\overline{A}$	A	A	$\overline{A}$

this table, A and B denote the inputs and S denotes the select line of the MUX. Design of the basic cell with voltage scaling and top-bottom pre-charge logic is shown in Fig. 4.6 and its layout is given in Fig. 4.7. The power and the current waveforms of the basic cell with top-bottom pre-charge logic is given in Fig. 4.8 and Fig. 4.9, respectively.

Experimentation is done with the above mentioned tool. All possible input combinations are applied to the basic multiplexer circuit. Glitch free outputs with acceptable voltage level are generated. This current waveform is *symmetric* in nature which makes it hard for the attackers to analyze the variance of power with different input combinations; simulation period is 1 μs.

4.2.2 BDD based S-box designs with top-bottom pre-charge logic

Lucifer is the earliest block cipher [40], whereas, Present is a modern lightweight block cipher [9]. The S-box used in both is 4bit $\times$ 4bit; function S: $\mathbb{F}_4^2 \rightarrow \mathbb{F}_4^2$. The actions of these S-boxes in hexadecimal notation is given in Table 4.2. Each hexadecimal input `x` represents the four input bits of the S-box, numbered as `V0`, `V1`, `V2` and `V3`. The corresponding output `S[x]` represents the four output bits of the S-box, numbered as `out0`, `out1`, `out2`, and `out3`. Thus, each output bit is basically a function of the four input bits. The unbalanced ROBDDs obtained from the CUDD tool for the output bits of the Lucifer and the Present S-boxes and their corresponding balanced BDDs,

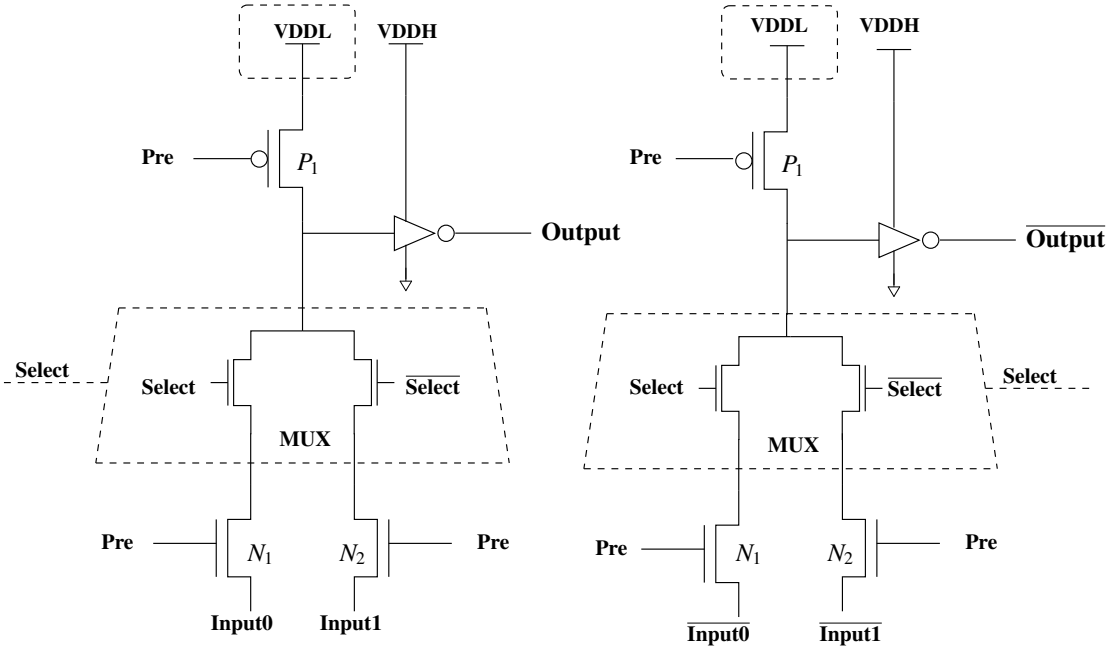


Figure 4.6: Design of the BDD based basic cell with voltage scaling and top-bottom pre-charge logic.

generated by our tool, are already shown in Fig. 5.1 and Fig. 5.2, respectively. The power waveforms of our Lucifer and Present S-box implementations can be found in Fig. 4.12 and Fig. 4.13, respectively.

Table 4.2: Present and Lucifer S-box functions

Lucifer	x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	S[x]	C	F	7	A	E	D	B	0	2	6	3	1	9	4	5	8
Present	x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	S[x]	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Experimentation is done with the above mentioned tool. All possible input combinations are applied to two S-box circuits. Glitch free outputs with acceptable voltage level are generated. This current waveform is *symmetric* in nature which makes it hard for the attackers to analyze the variance of power with different input combinations; simulation period is 1 μs.

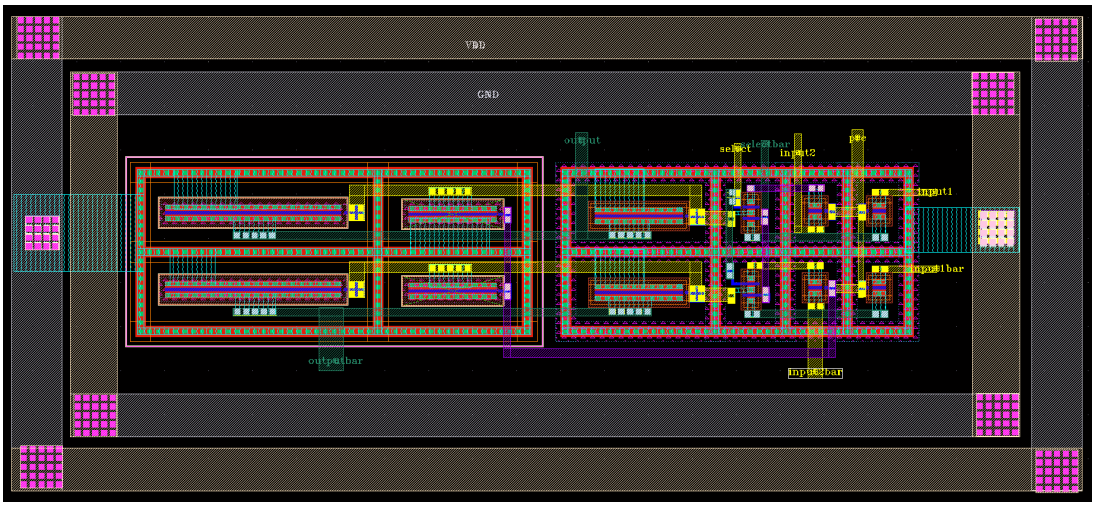


Figure 4.7: Layout of the basic cell with top-bottom pre-charge generation logic.

4.3 Applications of BDD based logic with top pre-charge

In this section, design of circuit with NMOS and PMOS based top pre-charge logic has been elaborated. Design of basic cell has been discussed first then complex circuit structures of two different substitution boxes have been elaborated. Output power and current waveform which are statistically analyzed by attackers to reveal the system key are also plotted. Power and current waveforms are generated by simulating schematic capture. Details of the tool are given below.

- Design Tool: Cadence Virtuoso IC design tool
- Technology: UMC 65nm process technology
- Version : 5.1.41
- Process technology specification: mixed mode/RF
- Operating temperature : $30°C$
- Supply Voltage : 0.9 – 1.1 volt
- Operating frequency : 500 MHz

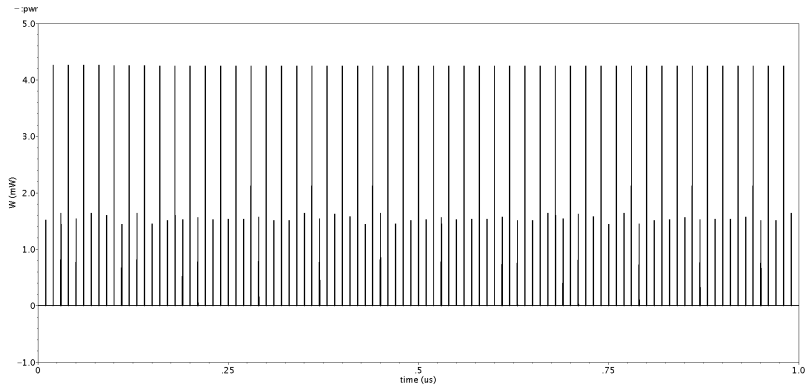


Figure 4.8: Power waveform characteristics of the basic cell with top-bottom pre-charge generation logic: time (ns) vs power (μW).

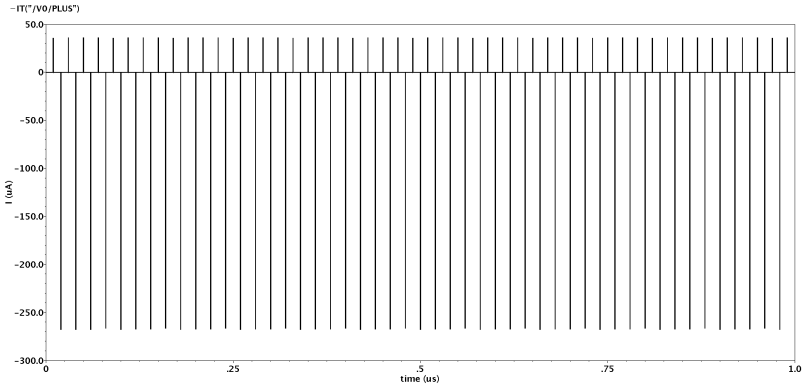


Figure 4.9: Current waveform characteristics of the basic cell with top-bottom pre-charge generation logic: time (ns) vs current (mA).

4.3.1 Basic Cell design using top pre-charge logic

Depending on the select line of the MUX and the input parameters, fourteen logic functions including AND, OR, XOR, NOT, NAND, NOR can be realized as shown in Table 4.1. In this table, *A* and *B* denote the inputs and *S* denotes the select line of the MUX. Design of the basic cell with voltage scaling of top pre-charge logic is shown in Fig. 4.14. The power and the current waveforms of the basic cell with top-bottom pre-charge logic is given in Fig. 4.15 and Fig. 4.16, respectively.

Experimentation is done with the above mentioned tool. All possible input combinations are applied to the basic multiplexer circuit. Glitch free outputs with acceptable voltage level are generated. This current waveform is *symmetric* in nature which makes it hard for the attackers to analyze the variance of power with different input

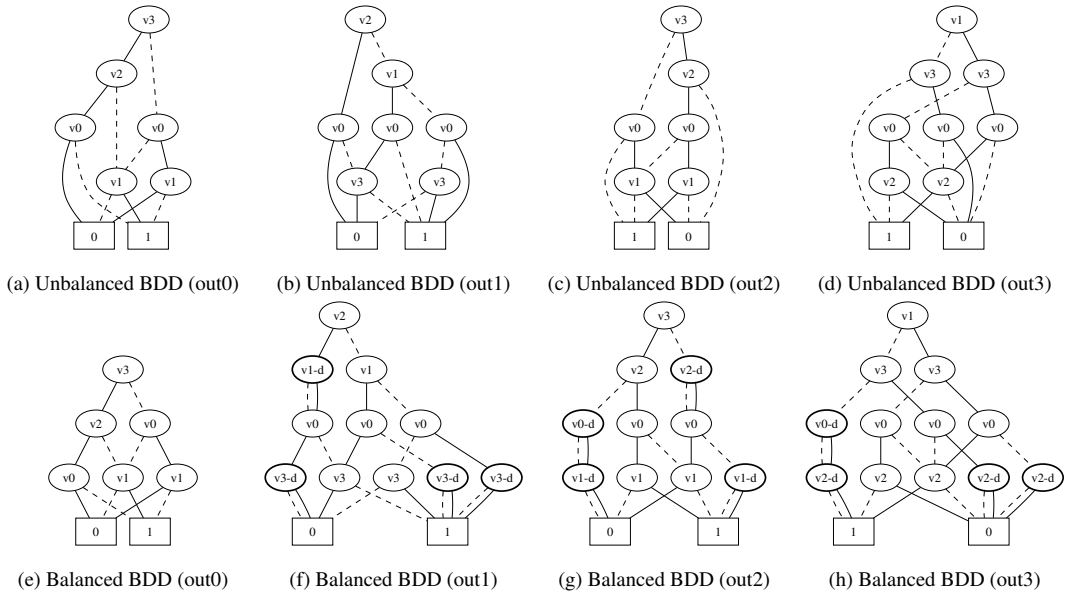


Figure 4.10: Unbalanced and balanced BDDs for the output bits of the Lucifer S-box highlighting the dummy nodes inserted.

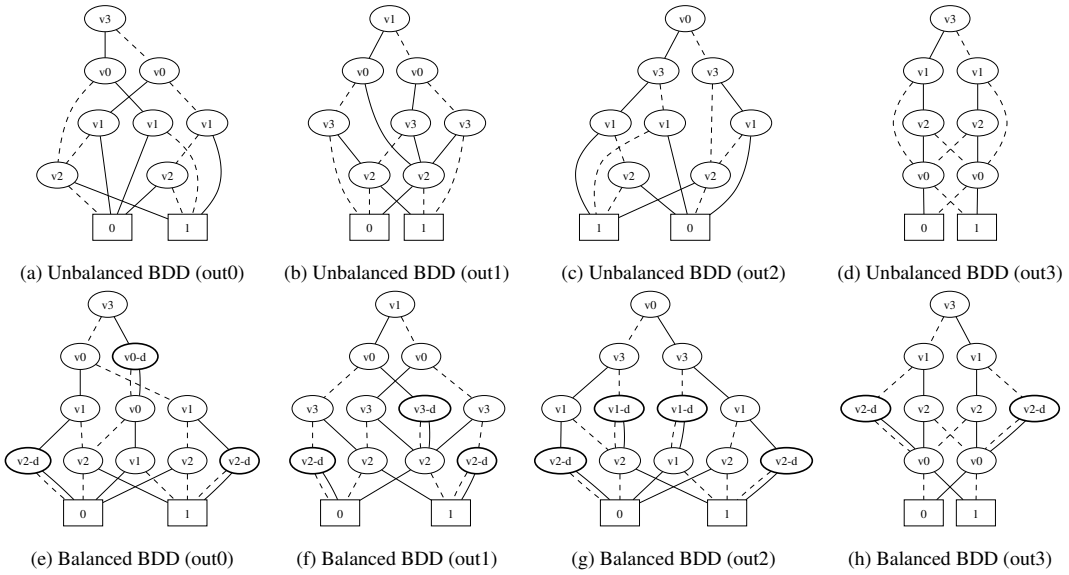
combinations; simulation period is 1 μs.

4.3.2 BDD based S-box designs with top pre-charge logic

Lucifer [40] and Present [9] are well-known block ciphers. The S-box used in both is 4bit $\times$ 4bit; function S: $\mathbb{F}_4^2 \rightarrow \mathbb{F}_4^2$. Each input is represented by the four input bits of the S-box, numbered as `V0`, `V1`, `V2` and `V3`. The corresponding output is represented by the four output bits of the S-box, numbered as `out0`, `out1`, `out2`, and `out3`. Thus, each output bit is basically a function of the four input bits.

The power waveforms of our Lucifer and Present S-box implementations can be found in Fig. 4.17 and Fig. 4.18, respectively.

Experimentation is done with the above mentioned tool. All possible input combinations are applied to two S-box circuits. Glitch free outputs with acceptable voltage level are generated. This current waveform is *symmetric* in nature which makes it hard for the attackers to analyze the variance of power with different input combinations; simulation period is 1 μs.



(a) Unbalanced BDD (out0) (b) Unbalanced BDD (out1) (c) Unbalanced BDD (out2) (d) Unbalanced BDD (out3)

(e) Balanced BDD (out0) (f) Balanced BDD (out1) (g) Balanced BDD (out2) (h) Balanced BDD (out3)

Figure 4.11: Unbalanced and balanced BDDs for the output bits of the Present S-box highlighting the dummy nodes inserted.

4.4 Conclusion

In this chapter, a novel area efficient balanced BDD based dual-rail circuit design technique with two different types of pre-charge generation mechanisms along with voltage scaling is proposed. Both the designs feature low power and extremely low peak power variation. Of the two designs, the one involving top-bottom pre-charge provides greater security against EPE than that of top pre-charge at the cost of marginal increase in power and area.

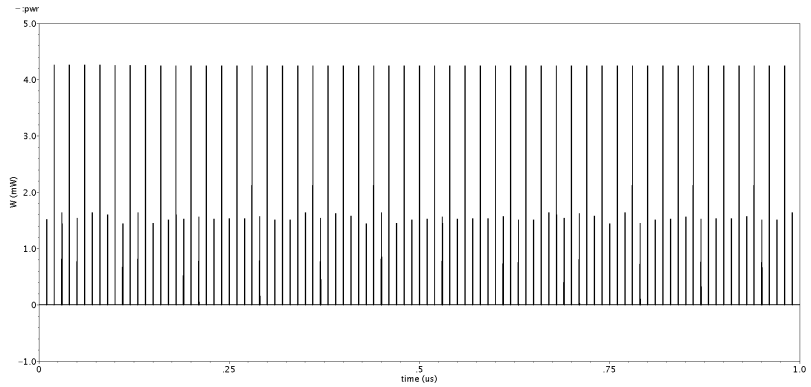


Figure 4.12: Power waveform characteristics of the Lucifer S-box with top bottom pre-charge generation logic: time (ns) vs power (μW).

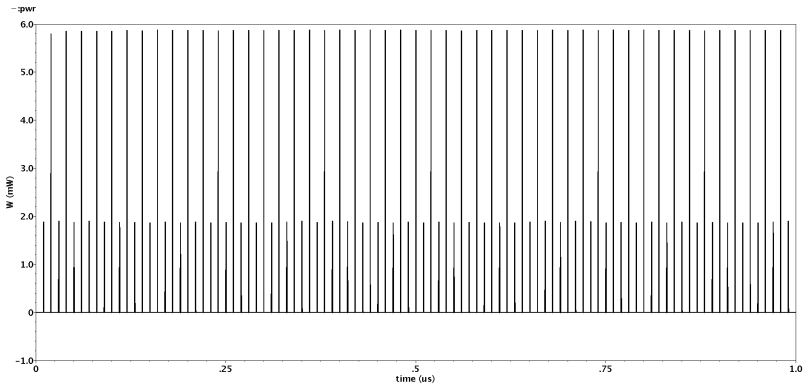


Figure 4.13: Power waveform characteristics of the Present S-box with top bottom pre-charge generation logic: time (ns) vs power (μW).

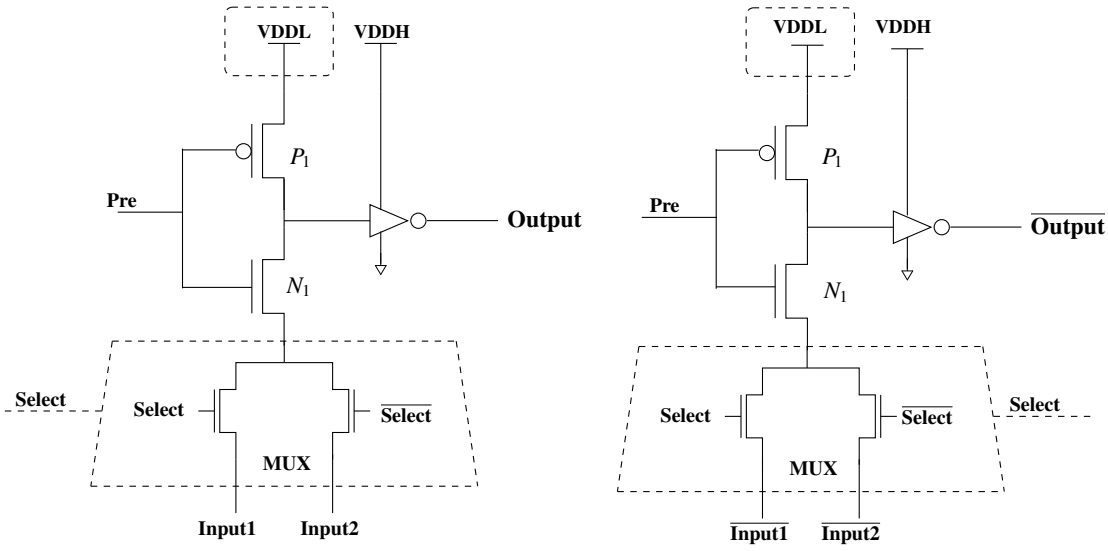


Figure 4.14: Design of the BDD based basic cell with voltage scaling and top precharge logic.

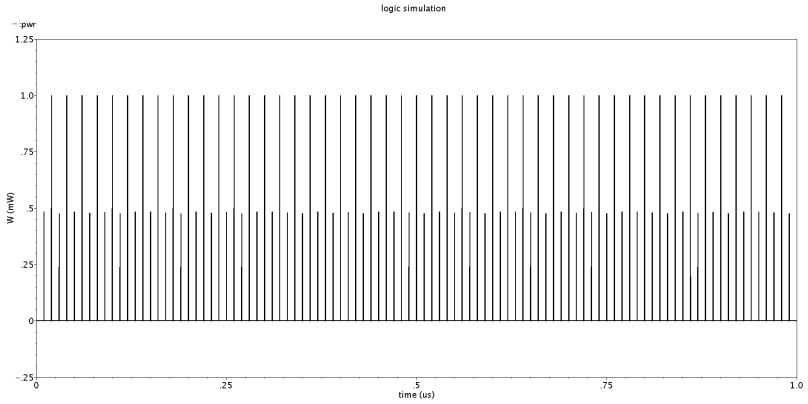


Figure 4.15: Power waveform characteristics of the basic cell with top pre-charge generation logic: time (ns) vs power (μW).

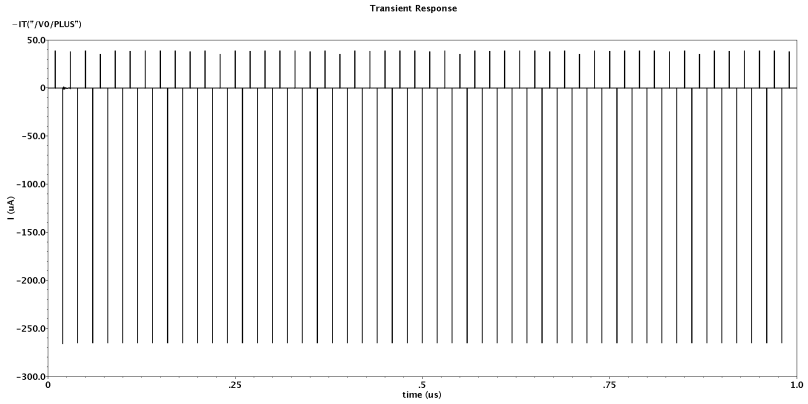


Figure 4.16: Current waveform characteristics of the basic cell with top pre-charge generation logic: time (ns) vs current (mA).

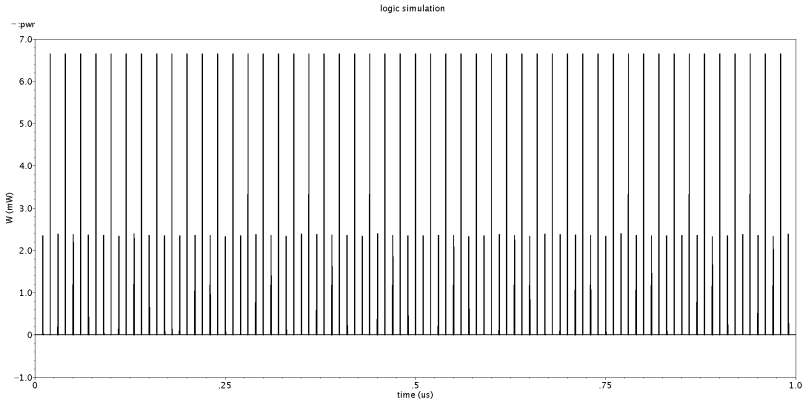


Figure 4.17: Power waveform characteristics of the Lucifer S-box with top pre-charge generation logic: time (ns) vs power (μW).

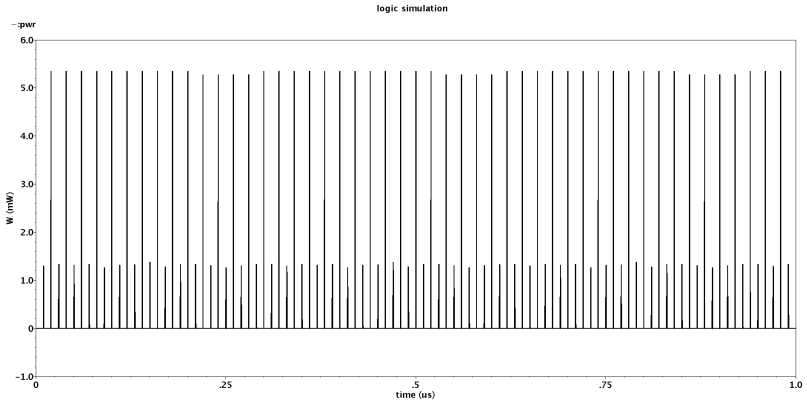


Figure 4.18: Power waveform characteristics of the Present S-box with top pre-charge generation logic: time (ns) vs power (μW).

Chapter 5

Automated synthesis scheme

A simple synthesis algorithm for mapping given Boolean functions to PAA resistant BDD based circuits is presented in this chapter. The automated synthesis process of such circuits involves the following steps: (*i*) ROBDD based logic minimization with normal and complemented functions; (*ii*) insertion of dummy nodes for path balancing, pre-charge nodes for pre-charge logic and regenerative nodes for fanout; (*iii*) converting the resulting BDD structure to transistor-level Verilog code. The methodology which described in the previous two chapters has been automated to generate transistor-level Verilog code; the key steps of automation have been described in section 5.1. If the depth of the BDD is too high for direct realisation by PTL a partition procedure is followed. This is described in section 5.2. Application of this process is demonstrated through the realisation of the AES S-box in section 5.3. The chapter is concluded in section 5.4

5.1 Automatic synthesis of Verilog code

In this section, we describe the automatic synthesis process of Verilog code from a given Boolean function. The key steps involved are as follows:

Step 1 - ROBDD Generation: The input to this step is a Boolean function and its output is the corresponding ROBDD. This step is carried out using the CUDD tool [39] which first generates the BDD from the given function and then reorders the involved

Algorithm 1 *balanceBDD (BDD_node* node*)*

Inputs: BDD graphs;

Outputs: BDD graphs with *dummy nodes*;

1: **if** *node* is a leaf node – representing 0 or 1 **then**
2: return 0 or return 1, respectively;
3: **end if**
4: *lh*:= Height of the left sub-tree of *node*;
5: *rh*:= Height of the right sub-tree of *node*;
6: **if** $lh > rh$ **then**
7: Insert $(lh - rh)$ dummy nodes in the left sub-tree of *node*;
8: **else if** $lh < rh$ **then**
9: Insert $(rh - lh)$ dummy nodes in the right sub-tree of *node*;
10: **end if**
11: **if** *node.leftChild* and *node.rightChild* are leaf nodes **then**
12: **if** *node.leftChild*= 0 and *node.rightChild*= 0 **then**
13: Replace *node* by 0;
14: **else if** *node.leftChild* = 1 and *node.rightChild*= 1 **then**
15: Replace *node* by 1;
16: **else if** *(node.leftEdge).label* $\neq$ *node.leftChild* **then**
17: Replace *node* by $\overline{node}$; /* i.e., feed the complement of *node* if its branches (corresponding to 0 and 1) terminate in oppositely labeled leaf nodes */
18: **end if**
19: **else**
20: return *balanceBDD (*node.leftChild*)*;
21: return *balanceBDD (*node.rightChild*)*;
22: **end if**

variables using in-built heuristics to get the final ROBDD. The ROBDD generated by by CUDD may have path of different lengths.

Step 2 - Balanced BDD Generation: This step takes the ROBDD produced in the previous step as input and produces the corresponding path length balanced BDD. The outline of the algorithm used for balancing is given in Algorithm 1. It is to be noted that whenever a difference in heights of the left and the right sub-trees for some node in a BDD is found, the algorithm tries to insert in the shorter sub-tree a dummy node which corresponds to the variable that occurs at the same depth in the longer sub-tree. This is realized with minimal hardware overhead via the fingering mechanism of implementing transistors. This is done in steps 4 to 10 of Algorithm 1.

The number of transistors required to realize a balanced BDD can be further re-

duced if the input variable (and/or its complement) which occurs at the lowest level of the BDD is directly fed to the implementing circuit. Specifically, a BDD node v whose both the emanating branches terminate in the leaf node 0 can be replaced by the input 0, a similar convention is used if both the branches terminate in 1; if the emanating branches corresponding to 1 and 0 terminate in the leaf nodes 1 and 0, respectively. In that case the node can be replaced by v itself, whereas, if its branches terminate in oppositely labeled leaf nodes, then the BDD node can be replaced by $\bar{v}$ (steps 14 to 21 of Algorithm 1). This modification is illustrated in Fig. 5.3 which is obtained from the balanced BDD of Fig. 5.2(h). It is to be noted that the graph shown in Fig. 5.3 is not strictly a BDD but conforms to Shannon's decomposition, nevertheless. These are done in steps 12 –18 of algorithm 1.

The BDD resulting from 1 has the property that all paths from the root to the leaf nodes are of identical length. This ensures that computation along each decision branch of the BDD will be through the same number of transistors, thus giving rise to identical delays. This ensures that the prime requirement for DPA resistance of the synthesised transistor network is satisfied. For the physical circuits to be secure, it is also important that the physical design of the circuits should not disturb the delay equalisation achieved at the transistor level design.

Step 3 - Verilog Generation: Verilog netlist generation has of four parts. (*i*) Verilog generation for pre-charge node. (*ii*) Verilog generation for normal BDD node. (*iii*) Verilog generation for dummy nodes. (*iv*) Verilog generation for swing restorations. (For Bottom pre-charge)

Here modular structural Verilog code is generated. At the top-level, instances of the four modules have been generated; these instances are generated according to the generated BDDs.

The modules in Verilog corresponding to a BDD node, an inverter and a top-bottom pre-charge node are given below.

```
module bdd_node (out, input0, input1, selectBar, select);
output out;
input  input0, input1, selectBar, select;
nmos ( out, input0, selectBar );
```

```
nmos ( out, input1, select );
endmodule

module inverter (out, in);
output out;
input  in;
supply1 power;
supply0 ground;
pmos ( out, power,  in );
nmos ( out, ground, in );
endmodule

module precharge_topBottom (out, pre, input0,
                      input1, selectBar, select);
output out;
input  input0, input1 , selectBar, select, pre;
wire inpBDD0, inpBDD1, outpre;
inverter inv1 ( outpre, out );
bdd_node B1   ( outpre, inpBDD0, inpBDD1,
                          selectBar, select );
pmos ( outpre,  power,  pre );
nmos ( inpBDD0, input0, pre );
nmos ( inpBDD1, input1, pre );
endmodule
```

Having defined these modules, a BDD node can be realized as an instance of the module `bdd_node`, such as the one defined below. Once we have generated the code corresponding to a BDD tree, we reproduce the same code with the final output (corresponding to the root node) and the original inputs V_{DD} and GND (corresponding to the leaf nodes 1 and 0, respectively) inverted; this gives us the complement of the original BDD tree.

```
bdd_node node_v3 ( out_v3, inp0_v3, inp1_v3,
                   selectBar_v3, select_v3 );
```

The modules in Verilog corresponding to a BDD node and a top pre-charge node

are given below. In the latter code, the module `bdd_inv` corresponds to the inverter whose code has been given next.

```
module bdd_node (out, inp0, inp1, selectBar, select);
output out;
input  inp0, inp1, selectBar, select;
nmos ( out, inp0, selectBar );
nmos ( out, inp1, select );
endmodule


module bdd_pre_top ( outpre, inpre, pre );
output outpre;
input   inpre, pre;
wire out;
bdd_inv inv1 ( outpre, out );
pmos ( out, power,  pre );
nmos ( out, inpre, pre );
endmodule
```

For the symmetric NMOS the modules in Verilog corresponding to a BDD node, an inverter, a pre-charge node and a node for swing restoration and voltage scaling combined are given below.

```
module bdd_MUX ( out, inp0, inp1, selectBar, select );
output out;
input  inp0, inp1, selectBar, select;
nmos ( out, inp0, selectBar );
nmos ( out, inp1, select );
endmodule

module bdd_inv ( out, in );
output out;
input  in;
supply1 powerHigh;
supply0 ground;
pmos ( out, powerHigh, in );
```

```
nmos ( out, ground, in );
endmodule

module bdd_pre ( outBDD, inpBDD, pre, preBar );
output outBDD;
input  inpBDD, pre, preBar;
nmos ( outBDD, inpBDD, pre );
nmos ( outBDD, power, preBar );
endmodule

module bdd_swing_scaling ( outswing, inpBDDnet, powerLow );
output outswing;
input  inpBDDnet;
inout  powerLow;
bdd_inv inv1 ( outswing, inpBDDnet );
pmos ( outswing, powerLow, inpBDDnet );
endmodule
```

Having defined these modules, a BDD node can be realized as an instance of the module `bdd_node`, such as the one defined below. Once we have generated the code corresponding to a BDD tree, we reproduce the same code with the final output (corresponding to the root node) and the original inputs VDD and GND (corresponding to the leaf nodes 1 and 0, respectively) inverted; this gives us the complementary of the original BDD tree.

The time complexity for Algorithm 1 is $O(n^2)$ in the worst case, where n is the number of nodes in the BDD tree. One of the scenarios where the worst case occurs is when all the non-leaf nodes is the left child of its parent node and they have one of the leaf nodes (0 or 1) as their right child. The unbalanced and balanced BDDs for the output bits of the Lucifer and the Present S-boxes with dummy nodes added are given in Fig. 5.1 and Fig. 5.2, respectively; the balanced BDDs are obtained by applying Algorithm 1 and given in Fig. 5.1 and Fig. 5.2 respectively

5.2 Partitioning the large BDDs

If the depth of the BDD is too much say greater than five then the direct PTL realisation does not work. Since signal strength decreases with every node traversed,

Algorithm 2 *balanceBDDlarge(BDD_node* node*,Integer emphhop)*

Inputs: BDD graphs;

Outputs: BDD graphs with *hops* and *dummy node*;

1: **if** *node* is a leaf node – representing 0 or 1 **then**
2: return 0 or return 1, respectively;
3: **end if**
4: *lh*:= Height of the left sub-tree of *node*;
5: *rh*:= Height of the right sub-tree of *node*;
6: **if** *lh*> *rh* **then**
7: Insert (*lh* −*rh*) dummy nodes in the left sub-tree of *node*;
8: **else if** *lh* < *rh* **then**
9: Insert (*rh* − *lh*) dummy nodes in the right sub-tree of *node*;
10: **end if**
11: **if** (*node.depth* % *hop*) = 0 **then**
12: Insert two repeaters as the left and the right child of *node*;
13: **end if**
14: **if** *node.leftChild* and *node.rightChild* are leaf nodes **then**
15: **if** *node.leftChild*= 0 and *node.rightChild*= 0 **then**
16: Replace *node* by 0;
17: **else if** *node.leftChild* = 1 and *node.rightChild*= 1 **then**
18: Replace *node* by 1;
19: **else if** *(node.leftEdge).label* ≠ *node.leftChild* **then**
20: Replace *node* by $\overline{node}$; /* i.e., feed the complement of *node* if its branches (corresponding to 0 and 1) terminate in oppositely labeled leaf nodes */
21: **end if**
22: **else**
23: return *balanceBDD (*node.leftChild*)*;
24: return *balanceBDD (*node.rightChild*)*;
25: **end if**

in such situation repeaters double inverters have to be added to boost the signal after a certain number of nodes (parameterised as hop) along a path as shown in Algorithm 2.

Inserting regenerative nodes after certain number of transistors acts like BDD partitioning. Output is generated for small BDDs with manageable number of transistors in a chain. The inserted regenerative node acts like buffer. It passes the same logic value with acceptable signal strength for driving the output capacitor. Note that repeater insertion is due in this 11-13 in the algorithm 2.

5.3 Automated synthesis of AES

The Advanced Encryption Standard (AES) is a standard encryption techniques developed by the U.S. National Institute of Standards and Technology (NIST) in 2001. AES uses Rijndael S-box. It is a block cipher which provide a secure encryption mechanism. AES is a symmetric block cipher based on SP networks. Each hexadecimal input `x` represents the eight input bits of the S-box, numbered as `V0`, `V1`, `V2 V3`, `V4`, `V5`, `V6` and `V7`. The corresponding output `S[x]` represents the four output bits of the S-box, numbered as `out0`, `out1`, `out2`, `out3`, `out4`, `out5`, `out6` and `out7`. Thus, each output bit is a function of the eight input bits. Our structure of AES-out0 is shown in the Fig. 5.5, Though for PTL implementation signal strength is degraded by V_{th} value of the NMOS transistor after traversing single BDD node. To maintain outputs with the proper voltage level regenerative node is necessary. In this synthesis mechanism user can specify the number of transistor path after which regenerative node would be inserted. Here R represent repeater node. And d represent dummy nodes. Regenerative node is acting like a buffer.

Automated Verilog code of out0 with top pre-charge logic is given. Codes for other modules are similar and so are not given here.

```
 module bdd_MUX ( out, inp0, inp1, selectBar, select );

output out;
input  inp0, inp1, selectBar, select;

nmos ( out, inp0, selectBar );
nmos ( out, inp1, select );

endmodule



module bdd_inv ( out, in );
```

```
output out;
input  in;

supply1 power;
supply0 ground;

pmos ( out, power,  in );
nmos ( out, ground, in );

endmodule


module bdd_pre_top ( outpre, insbox, pre );

output outpre;
input  insbox, pre;

wire out;

bdd_inv inv1 ( outpre, out );

pmos ( out, power,  pre );
nmos ( out, insbox, pre );

endmodule


module bdd_Sbox ( uOUTPUT, cOUTPUT, INPUT0, INPUT1, pre );

output uOUTPUT, cOUTPUT;
input  INPUT0, INPUT1, pre;

/* Detail Connection is Omitted */
endmodule
```

5.4 Conclusion

An automatic synthesis tool based on our all design mechanism has also been developed which generates transistor level Verilog code. It has there key steps: (*i*) ROBDD Generation (*ii*) Balanced BDD Generation (*iii*) Partitioning the large BDDs into smaller BDD with manageable height. (*iv*)Verilog Generation. This has been successfully tested by synthesising the Verilog netlist of the AES stream cipher.

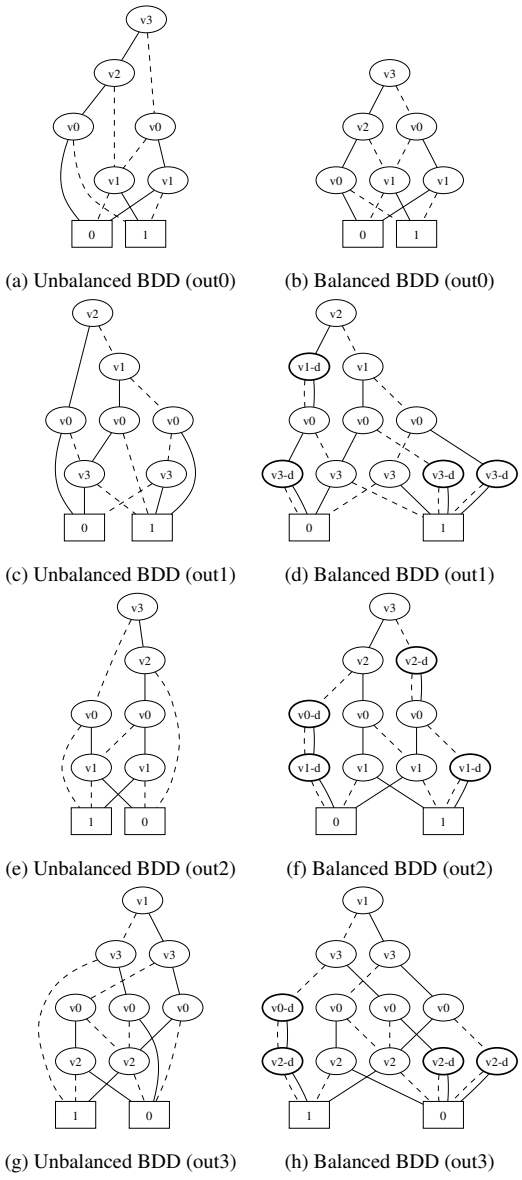


Figure 5.1: Unbalanced and balanced BDDs for the output bits of the Lucifer S-box highlighting the dummy nodes inserted.

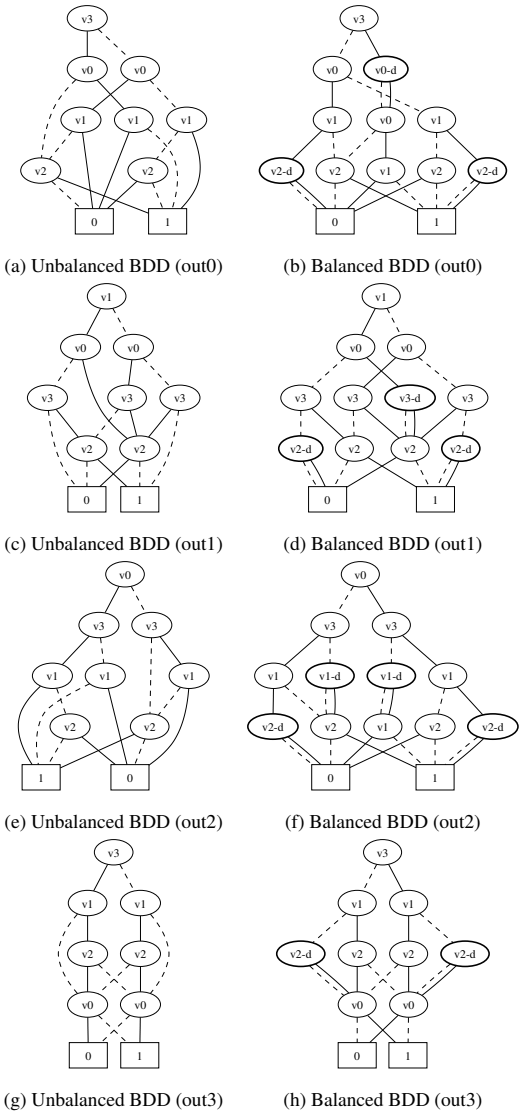


Figure 5.2: Unbalanced and balanced BDDs for the output bits of the Present S-box highlighting the dummy nodes inserted.

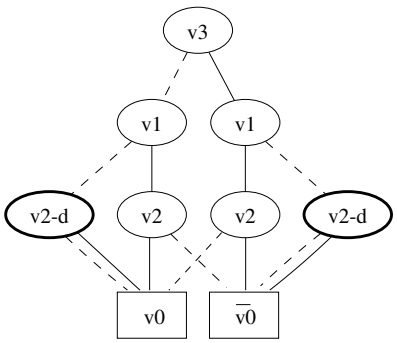


Figure 5.3: Reducing the balanced BDD of Fig. 5.2(h).

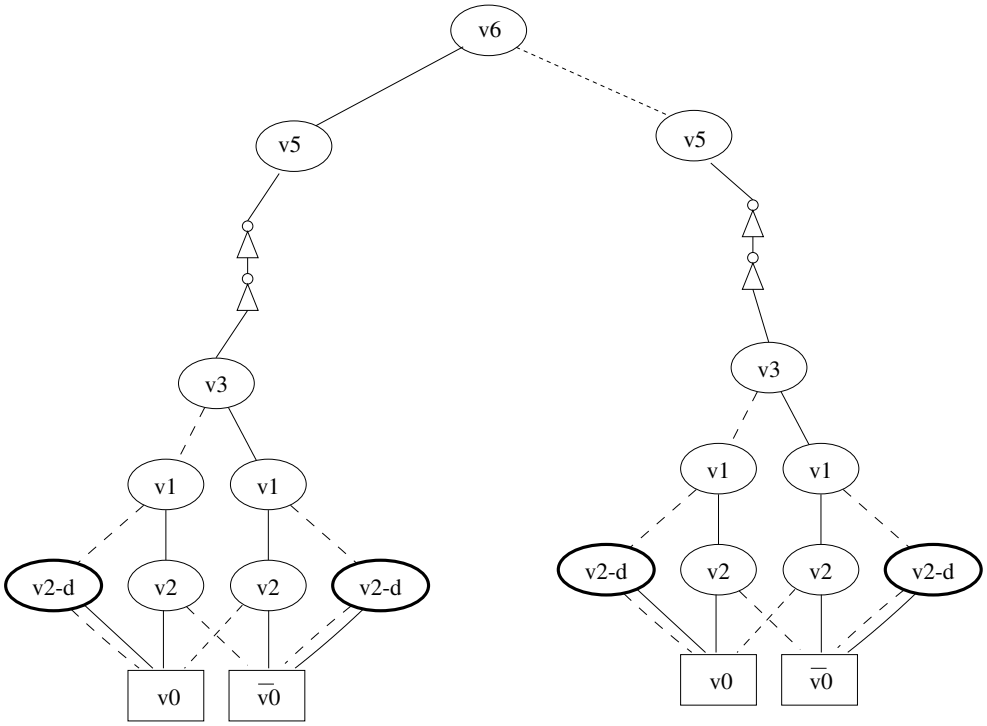


Figure 5.4: Partitioning the large BDDs

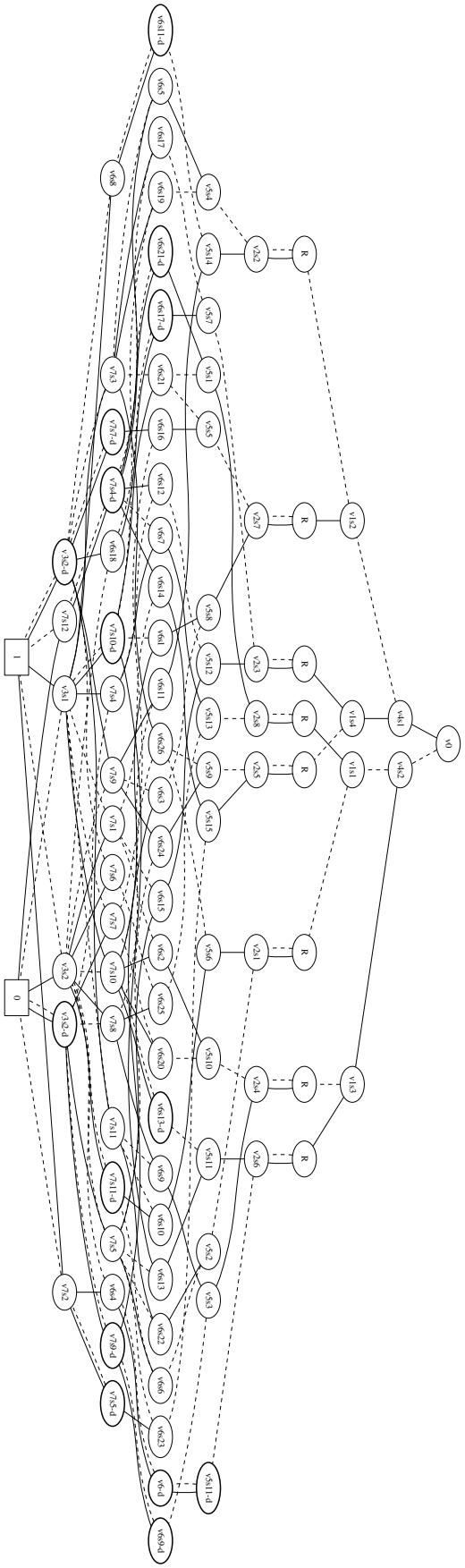


Figure 5.5: Reducing the balanced BDD of AES out0.

Chapter 6

Experimental results with different process technology

The objective of our experimentation is two fold, (*i*) to demonstrate resistance to power attacks and (*ii*) to highlight the low power characteristics. Towards the first objective we have carried out differential power attacks such as, difference of mean (DoM) and correlation power attack (CPA). We demonstrate resilience against the the early propagation effect (EPE). Six 2-input basic cell and two 4×4 S-boxes are used for experimental benchmark.

We further our experimentation by constructing two encryption system, viz Lucifer and Present. Each of these require four copies of an S-box specific to each system. DoM based DPA attack [26] and CPA attack [28] are performed on our S-box implementations with all our pre-charging schemes. We used 40,000 random vectors to launch a DoM attack on this system and obtained 700,000 power traces. For each of the possible 16 keys, we plot the difference of mean of power dissipation for each of the output bits of the S-box. CPA attacks exploit the correlation factor between the power samples and the hamming weights of the handled data [28]. Accordingly, we plot the power dissipation for all the possible 256 combinations of the plain texts and the keys. Detail of experimentation for our various pre-charging schemes are given next. The details of the tool used for the designs and the simulations are listed below.

– Design Tool: Cadence Virtuoso IC design tool

- Technology: UMC 65nm process technology
- Version : 5.1.41
- Process technology specification: mixed mode/RF
- Operating temperature : 30°C
- Supply Voltage : 0.9 – 1.1 volt
- Operating frequency : 500 MHz

The experiments were carried out cover the following aspects.

a) Comparison of our technique with that of DP-BDD and SDMLp in terms of standard attributes

b) Establish DPA attack resistance of our S-box implementations

c) Establish CPA attack resistance of our S-box implementations

d) Comparison of our S-box designs with the other two design styles in terms of normalized attributes defined in [43]

e) Establish EPE attack resistance of our S-box implementations

The normalized energy deviation (NED) and normalized standard deviation (NSD) [43] which are defined as follows.
NED = (max(E) − min(E)) / max(E)
NSD = $\sigma_E/\overline{E}$, where σ and $\overline{E}$ are standard deviation and mean respectively, calculated per complete clock cycle. Energy consumption is given by the formula $E = V_{DD} \cdot \int_0^T I_{DD}(t)dt$ this is computed by the cad tool.

All possible input combinations are applied to both S-box circuits and the basic cells. Glitch free outputs with acceptable voltage level are generated. For the purpose of comparison, competing methods are also implemented with identical parameters used for our circuits.

Rest of the chapter oranised as follows: Experimentation for bottom-pre charge logic in 180nm technology is given in section 6.2. Experimentation with 65nm technology for symmetric NMOS based pre-charge logic, top-bottom pre-charge logic and top pre-charge logic is given in section 6.3, 6.4 and section 6.5 respectively.

6.1 Experimentation for bottom-pre charge logic in 180 nm technology

The details of the tool used for the designs and the simulations are listed below.

- Design Tool: Cadence Virtuoso IC design tool
- Technology: UMC 180nm process technology
- Version : 6.1.4
- Process technology specification: mixed mode/RF
- Device : 180nm mixed mode
- Operating temperature : $30°C$
- Supply Voltage : 1.8 volt
- Operating frequency : 500 MHz

Schematic capture, layout design and post layout simulations have been performed. Current and power waveforms are plotted. Comparative analysis with a competing method SDMLp is also done. Moreover, a test circuit using our designs has been devised and analyzed for DPA resistance.

We further extend our experiment by constructing a test circuit as shown in Fig. 6.1. Difference of mean (DoM) power attack [28] has been performed on this circuit. For DoM power attack, power traces and their corresponding cipher text values are collected. The attacker guesses a byte of the key (key hypothesis), and computes the intermediate value. The attacker then partitions the power traces into two disjoint sets depending upon whether the targeted computed bit is 0 or 1. The mean for each set is

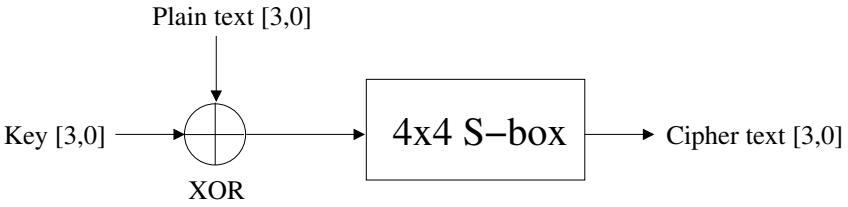


Figure 6.1: Evaluation of DPA resistance by computing DoM.

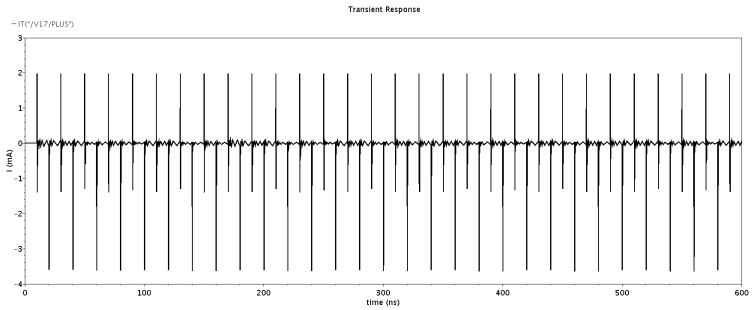


Figure 6.2: Current waveform generated by the attack.

computed. The key hypothesis is considered to be correct, if there occurs a significant difference between the means. This process is then repeated for the rest of the bytes of the key.

In Fig. 6.1, a plain text and a (constant) secret key have been XOR-ed and sent through the `S` function. The function `S` is a typical 2 bit addition function. Every time it reads 6 input bits and produces 4 outputs, sum_0, sum_1, $carry$, and $\overline{carry}$. Continuous different plain text has been sent for 600ns where each text duration (evaluation period) is 20ns and pre-charge phase is 10ns. Variations of power for 0, and separately for 1, in the least significant bit of the plain text have been monitored. The power trace obtained is found to be a repetitive power waveform with small peak power variation of 0.58mW. Hence, the DoM power attack fails in revealing the secret key, signifying that the test circuit is indeed DPA resistant. The current and the power waveforms generated by the attack are given in Figs. 6.2 and 6.3, respectively.

A comparative analysis is done with a competing method SDMLp [35]. It is to be noted that in [35] 90nm technology has been used, but for our experimentation their basic cell has been reproduced using 180nm technology. The results are given in table 6.1. It has been found that the peak power variance (PPV) for SDMLp is 39.7mW while for our method it is 0.03mW. Thus, the PPV for our cell is lesser

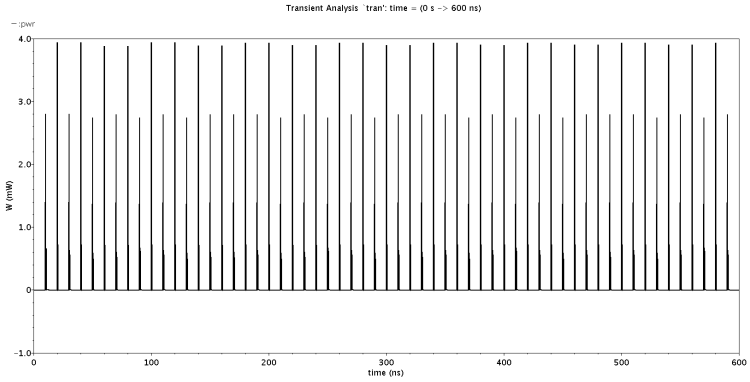


Figure 6.3: Power waveform generated by the attack.

Table 6.1: Comparison between SDMLp and Our method

Parameter	SDMLp	Our
PPV (mW)	39.7	0.03
Area (#transistors)	12	18
Delay (ns)	0.11	0.18

than that of SDMLp by *1300 times*. The area taken by a basic cell of SDMLp is 12, whereas for us it is 18 in terms of the number of transistors used. The propagation delay for SDMLp is 0.11ns while it is 0.18ns for us. Since PPV directly implies the DPA resistance of a circuit (the lesser being the better), it can be concluded that the proposed design outperforms that of SDMLp by large, at the cost of minimal increase in area and propagation delay.

6.2 Experimentation for bottom-pre charge logic in 65 nm technology

We describe the experiments carried out to establish the PAA resistance of our implementation with PMOS and NMOS transistor based bottom pre-charge logic of the basic cell and Present S-box [9]. Through the experiments described below, we characterize peak power variance, average power, average current and propagation delay for our design and make comparison of these parameters with that of DP-BDD [3] and SDMLp [35]. Schematic capture and standard spectra simulations have been performed, and current and power waveforms are plotted. The results are given in

Table 6.2: Comparison with other methods with bottom pre-charge logic

Circuit	Peak Power Variance (μW)			Average Power (μW)			Average Current (μA)			Propagation Delay (ns)		
/ S-box	DPBDD	SDMLp	Our	DPBDD	SDMLp	Our	DPBDD	SDMLp	Our	DPBDD	SDMLp*	Our
AND	835.33	15.82	0.41	3.91	8.52	2.90	3.68	7.64	2.63	0.05	0.11×2	0.13
OR	1092.11	31.55	3.13	3.86	8.48	3.95	3.63	7.61	3.65	0.05	0.11×2	0.13
XOR	594.33	13.21	1.54	4.30	8.61	3.93	4.04	7.73	3.57	0.06	0.11×2	0.16
NAND	835.33	15.82	0.41	3.91	8.52	2.90	3.68	7.64	2.63	0.05	0.11×2	0.13
NOR	1092.11	31.55	3.13	3.86	8.48	3.95	3.63	7.61	3.65	0.05	0.11×2	0.13
XNOR	594.33	13.21	1.54	4.30	8.61	3.93	4.04	7.73	3.57	0.06	0.11×2	0.16
MUX	564.16	31.01	0.41	4.51	11.00	2.90	4.22	10.10	2.63	0.05	0.10×2	0.13
Avg	775.95	21.74	1.37	4.14	8.88	3.42	3.89	8.01	3.12	0.05	0.11×2	0.13
Present	41991.22	6965.04	17.1	96.44	226.83	30.51	92.26	206.79	27.73	0.14	0.19×2	0.33

*It may be noted that SDMLp has a twin cycle operation as described in section 1 .

Table 6.2.

6.2.1 Comparison in terms of standard attributes

Experimental results in Table 6.2 demonstrate a significant reduction by 99.82% and 93.69% in peak power variance (PPV) with respect to DP-BDD and SDMLp, respectively, for the basic cell. A reduction of about 17.39% in both average power and average current consumption is observed for the DP-BDD implementation of the basic cell whereas, for SDMLp, these parameters are found to be reduced by 61.48%. A greater reduction in average power and average current consumption is observed for the S-box implementations. While determining the propagation delay, we found that our method exhibits considerable more delay than that of DP-BDD but it is 40% less compared to that of SDMLp.

6.2.2 DPA attack resistance

We further our experiment by constructing a cryptographic system as shown in Fig. 6.1. Difference of mean (DoM) based DPA attack [26] and CPA attack [28] are performed on both of our S-box implementations. We used 30,000 random vectors to launch a DoM attack on this system and obtained 500,000 power traces. For each of the possible 16 keys, Fig. 6.4 plots the difference of mean of power dissipation for each of the output bits of the Present S-box.

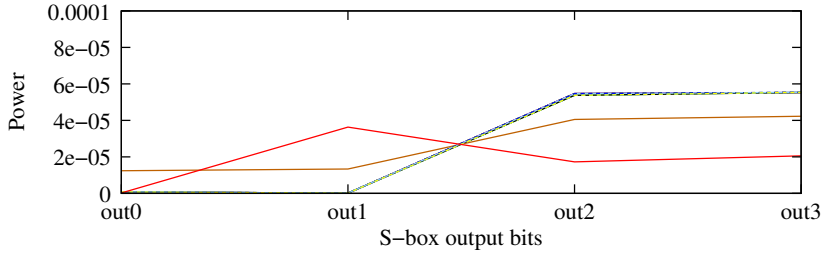


Figure 6.4: DPA attack on bottom pre-charge logic design: Present S-box output bits vs power (μW).

6.2.3 CPA attack resistance

CPA attacks exploit the correlation factor between the power samples and the hamming weights of the handled data [28]. Accordingly, we plot the power dissipation for all the possible 256 combinations of the plain texts and the keys as shown in Fig. 6.5 where points belonging to different hamming weights (ranging from 0 to 4) are highlighted with different colours; all the input combinations are found to lie on the same plane which indicates that they exhibit identical power dissipation.

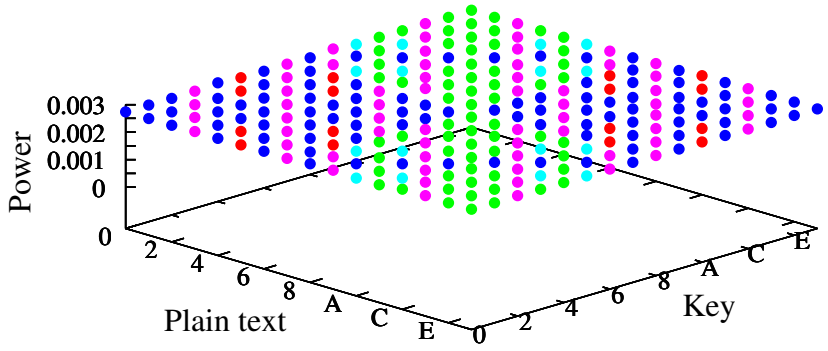


Figure 6.5: CPA attack on bottom pre-charge logic Present S-box design: plain text vs key vs power (μW).

Table 6.3: Comparison with respect to NED and NSD

	Normalized energy deviation (NED)			Normalized standard deviation (NSD)		
	DPBDD	SDMLp	bottom	DPBDD	SDMLp	bottom
Present	0.77242	0.08021	0.0168	0.34301	0.35216	0.0022

Table 6.4: Delay in output generation for the basic cell by symetric-NMOS based pre-charge logic

Pre	Input0	Input1	Select	Time
0	0	0	0	1
0	0	0	1	1
⋮	⋮	⋮	⋮	1
0	1	1	1	1
1	0	0	0	2
⋮	⋮	⋮	⋮	2
1	1	1	1	2

6.2.4 Comparison in terms of normalized attributes

In addition, we compare our S-box designs with those of DP-BDD and SDMLp based on the parameters normalized energy deviation (NED) and normalized standard deviation (NSD) [43]. It can be seen from Table 6.3 that NED and NSD of our S-box implementations are much less than that of the others, which establish the superior PAA resistance of our hardware designs.

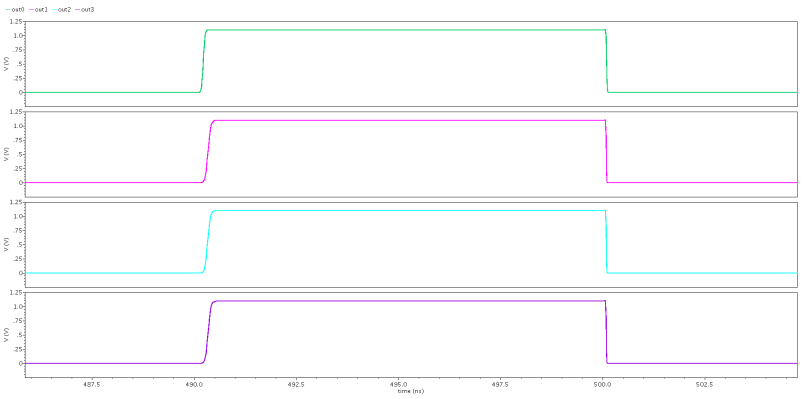


Figure 6.6: Timing response of the four output bits (all 1s) generated by the Present S-box desin using bottom pre-charge logic: time (ns) vs voltage (V).

6.2.5 EPE attack resistance

We established that our entire synthesis scheme is EPE attack resistant as follows. The input signals come from the bottom and produce outputs at the top (swing restoration part) by traversing paths (which are of equal length) of the BDD network depending on the switch value of the NMOS transistor. In Table 6.4, we tabulate the delays in output generation for the basic cell in terms of number of transistor switching. The basic cell undergoes one transistor switching whenever the pre-charge is 0 and two transistor switching whenever the pre-charge is 1; recall that whenever pre-charge is 0, the output produced is also 0 irrespective of the inputs provided. Furthermore, we plot the transient response of the four output bits generated by the Present S-boxe in Fig. 6.6; we deliberately chose the input combination which produces 1s at all the output bits of the Present S-box to reveal that the four outputs are generated simultaneously and hence, launching EPE attack by distinguishing among the delays in output generation is not viable.

6.3 Experimentation for symmetric NMOS based pre-charge logic in 65 nm technology

Table 6.5: Comparison with other methods with symetric-NMOS based pre-charge logic

Circuit / S-box	Peak Power Variance (μW)			Average Power (μW)			Average Current (μA)			Propagation Delay (ns)		
	DPBDD	SDMLp	Symetric-NMOS	DPBDD	SDMLp	Symetric-NMOS	DPBDD	SDMLp	Symetric-NMOS	DPBDD	SDMLp*	Symetric-NMOS
AND	835.33	15.82	1.42	3.91	8.52	2.95	3.68	7.64	2.65	0.05	0.11×2	0.18
OR	1092.11	31.55	3.21	3.86	8.48	2.96	3.63	7.61	2.67	0.05	0.11×2	0.18
XOR	594.33	13.21	3.62	4.30	8.61	2.96	4.04	7.73	2.67	0.06	0.11×2	0.18
NAND	835.33	15.82	1.42	3.91	8.52	2.95	3.68	7.64	2.65	0.05	0.11×2	0.18
NOR	1092.11	31.55	3.21	3.86	8.48	2.96	3.63	7.61	2.67	0.05	0.11×2	0.18
XNOR	594.33	13.21	3.62	4.30	8.61	2.96	4.04	7.73	2.67	0.06	0.11×2	0.18
MUX	564.16	31.01	0.93	4.51	11.00	2.96	4.22	10.10	2.65	0.05	0.10×2	0.16
Avg	775.95	21.74	2.49	4.14	8.88	2.96	3.89	8.01	2.66	0.05	0.11×2	0.18
Lucifer	37841.10	6711.00	1.43	83.33	223.40	41.82	79.72	203.51	38.01	0.15	0.19×2	0.45
Present	41991.22	6965.04	5.60	96.44	226.83	26.56	92.26	206.79	24.14	0.14	0.19×2	0.34

*It may be noted that SDMLp has a twin cycle operation as described in section 1.

We describe the experiments carried out to establish the PAA resistance of our implementation with symmetric NMOS transistor based pre-charge logic of the basic cell, Lucifer S-box [40] and Present S-box [9]. Through the experiments described below, we characterize peak power variance, average power, average current and propagation delay for our design and make comparison of these parameters with that of DP-BDD [3] and SDMLp [35].

Schematic capture and standard spectra simulations have been performed, and current and power waveforms are plotted. The results are given in Table 6.5.

6.3.1 Comparison in terms of standard attributes

Experimental results in Table 6.5 demonstrate a significant reduction by 99.68% and 88.55% in peak power variance (PPV) with respect to DP-BDD and SDMLp, respectively, for the basic cell. The reduction in PPV is recorded to be greater than 99.9% for the S-box implementations for both DP-BDD and SDMLp. A reduction of about 30% in both average power and average current consumption is observed for the DP-BDD implementation of the basic cell whereas, for SDMLp, these parameters are found to be reduced by 67%. A greater reduction in average power and average current consumption is observed for the S-box implementations. While determining the propagation delay, we found that our method exhibits considerable more delay than that of DP-BDD but it is 18% less compared to that of SDMLp.

6.3.2 DPA attack resistance

We further our experiment by constructing a cryptographic system as shown in Fig. 6.1. Difference of mean (DoM) based DPA attack [26] and CPA attack [28] are performed on both of our S-box implementations. We used 40,000 random vectors to launch a DoM attack on this system and obtained 700,000 power traces. For each of the possible 16 keys, Fig. 6.7 plots the difference of mean of power dissipation for each of the output bits of the Present S-box.

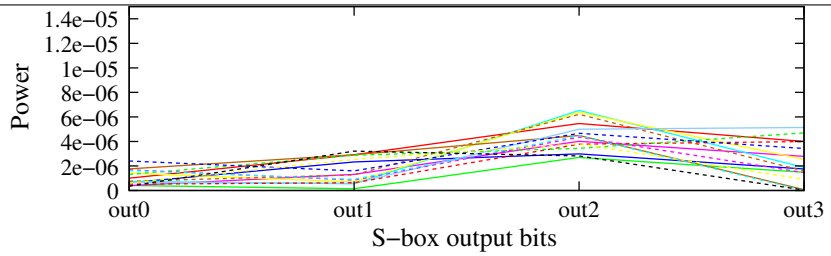


Figure 6.7: DPA attack on symetric-NMOS based pre-charge logic design: Present S-box output bits vs power (μW).

6.3.3 CPA attack resistance

CPA attacks exploit the correlation factor between the power samples and the hamming weights of the handled data [28]. Accordingly, we plot the power dissipation for all the possible 256 combinations of the plain texts and the keys as shown in Fig. 6.8 where points belonging to different hamming weights (ranging from 0 to 4) are highlighted with different colours; all the input combinations are found to lie on the same plane which indicates that they exhibit identical power dissipation. The mean variances of power dissipation for the Lucifer and the Present S-boxes are found to be $1.6 \times 10^{-5}\mu$W and $2.2 \times 10^{-5}\mu$W, respectively. As evident form these statistics, the required information to identify the correct key (by way of separation) is not available.

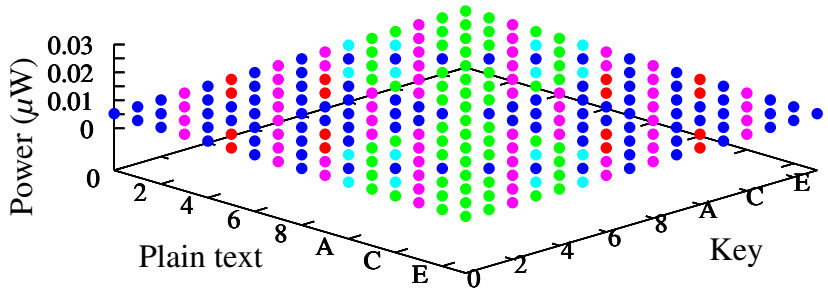


Figure 6.8: CPA attack on symetric-NMOS based pre-charge logic Present S-box design: plain text vs key vs power (μW).

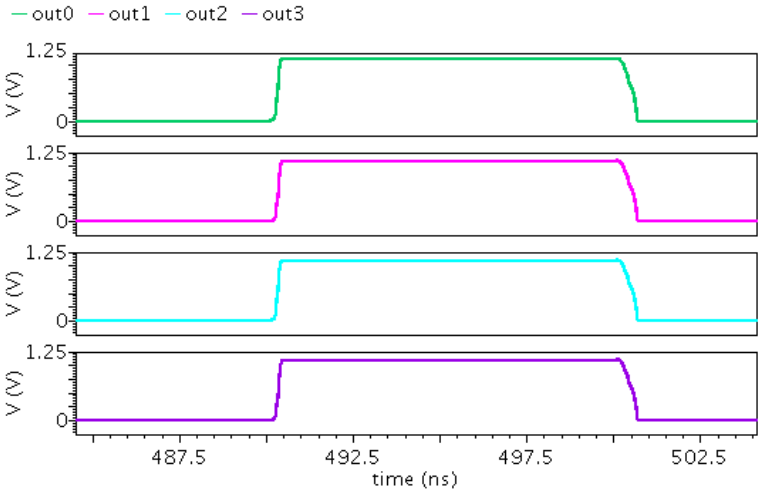


Figure 6.9: Timing response of the four output bits (all 1s) generated by the Present S-box desin using symetric-NMOS based pre-charge logic: time (ns) vs voltage (V).

Table 6.6: Comparison with respect to NED and NSD

	Normalized energy deviation (NED)			Normalized standard deviation (NSD)		
	DPBDD	SDMLp	Sym-NMOS	DPBDD	SDMLp	Sym NMOS
Lucifer	0.68215	0.70102	0.0015	0.32311	0.33182	0.0014
Present	0.77242	0.08021	0.0415	0.34301	0.35216	0.0056

6.3.4 Comparison in terms of normalized attributes

In addition, we compare our S-box designs with those of DP-BDD and SDMLp based on the parameters normalized energy deviation (NED) and normalized standard deviation (NSD) [43]. It can be seen from Table 6.6 that NED and NSD of our S-box implementations are much less than that of the others, which establish the superior PAA resistance of our hardware designs.

6.3.5 EPE attack resistance

Our entire synthesis scheme is EPE attack resistant; the input signals come from the bottom and produce outputs at the top (swing restoration part) by traversing paths (which are of equal length) of the BDD network depending on the switch value of the

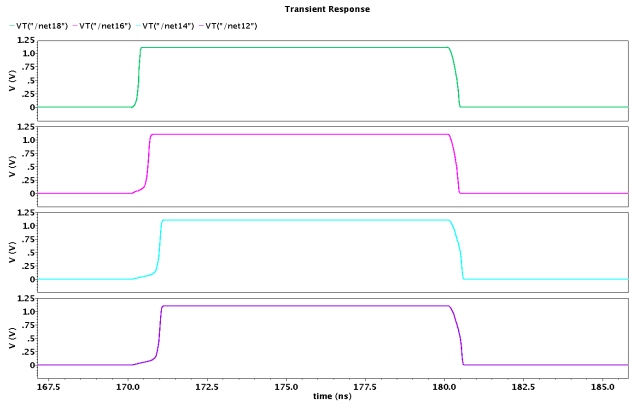


Figure 6.10: Timing response of the four output bits (all 1s) generated by the lucifer S-box desin using symetric-NMOS based pre-charge logic: time (ns) vs voltage (V).

Table 6.7: Delay in output generation for the basic cell by symetric-NMOS based pre-charge logic

Pre	Input0	Input1	Select	Time
0	0	0	0	1
0	0	0	1	1
⋮	⋮	⋮	⋮	1
0	1	1	1	1
1	0	0	0	2
⋮	⋮	⋮	⋮	2
1	1	1	1	2

NMOS transistor. In Table 6.7, we tabulate the delays in output generation for the basic cell in terms of number of transistor switching. The basic cell undergoes one transistor switching whenever the pre-charge is 0 and two transistor switching whenever the pre-charge is 1; recall that whenever pre-charge is 0, the output produced is also 0 irrespective of the inputs provided. Furthermore, we plot the transient response of the four output bits generated by the two S-boxes in Fig. 6.10 and Fig. 6.9; we deliberately chose the input combination which produces 1s at all the output bits of the Present S-box to reveal that the four outputs are generated simultaneously and hence, launching EPE attack by distinguishing among the delays in output generation is not viable.

6.4 Experimentation for top-bottom pre-charge in 65 nm technology

We elaborate the experiments carried out to establish the DPA and CPA resistance of our implementations of the basic cell and the Lucifer and the Present S-boxes. The uniform timing behaviour of our circuits which counters EPE attacks is also highlighted. Through the experiments described below, we characterize peak power variance, average power consumption, average current consumption and propagation delay for our design and make comparison of these parameters with that of DP-BDD [3] and SDMLp [35].

Table 6.8: Comparison with other methods with top-bottom pre-charge logic

Circuits	Peak Power Variance (μW)			Average Power (μW)			Average Current (μA)			Propagation Delay (ns)		
	DPBDD	SDMLp	TopBtm	DPBDD	SDMLp	TopBtm	DPBDD	SDMLp	TopBtm	DPBDD	SDMLp*	TopBtm
AND	835.33	15.82	0.03	3.91	8.53	3.82	3.68	7.64	3.47	0.05	0.09×2	0.10
OR	1092.11	31.55	0.03	3.86	8.48	3.82	3.63	7.61	3.47	0.05	0.08×2	0.10
XOR	594.33	13.21	0.02	4.30	8.61	3.82	4.04	7.73	3.47	0.06	0.09×2	0.10
NAND	835.33	15.82	0.03	3.91	8.53	3.82	3.68	7.64	3.47	0.05	0.09×2	0.10
NOR	1092.11	31.55	0.03	3.86	8.48	3.82	3.63	7.61	3.47	0.05	0.08×2	0.10
XNOR	594.33	13.21	0.02	4.30	8.61	3.82	4.04	7.73	3.47	0.06	0.09×2	0.10
MUX	564.16	30.89	1.44	4.51	11.28	3.81	4.22	10.92	3.44	0.05	0.10×2	0.11
Avg	775.95	22.87	0.37	4.14	9.22	3.82	3.89	8.46	3.45	0.05	0.09×2	0.10
Lucifer	37841.10	6711.00	2.11	83.33	223.40	13.70	79.72	203.51	12.45	0.15	0.19×2	0.11
Present	41991.22	6965.04	5.08	96.44	226.83	27.29	92.26	206.79	24.82	0.14	0.19×2	0.13

6.4.1 Comparison in terms of standard attributes

The experimental results given in Table 6.8 demonstrate a reduction by a 99.9% and 98.3% in peak power variance for the basic cell with top-bottom pre-charge logic with respect to DP-BDD and SDMLp, respectively. Significant reduction for average power and average current is also achieved. Although the propagation delay for the basic cell is found to be almost twice of that of DP-BDD, it is nearly half of that of SDMLp. The area required (in terms of number of transistors) to realize our basic cell with the pre-charge logic is 14 whereas, the basic cell of SDMLp requires 12; this number varies for different gate realizations using DP-BDD because it does not use a

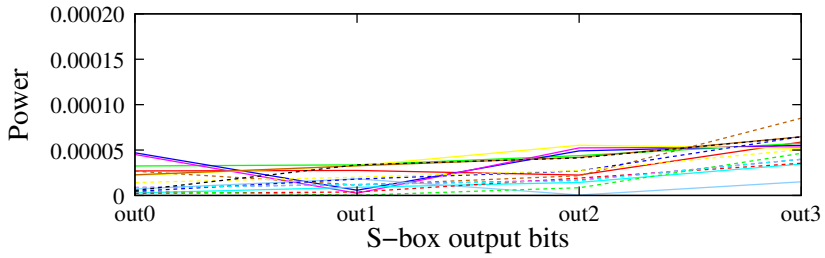


Figure 6.11: DPA attack on our Lucifer S-box design with top-bottom pre-charge logic: S-box output bits vs power (μW).

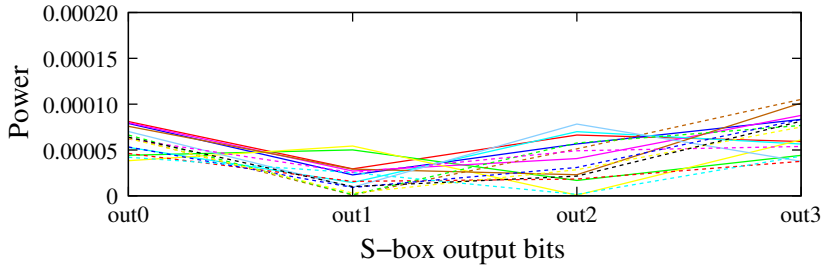


Figure 6.12: DPA attack on our Present S-box design with top-bottom pre-charge logic: S-box output bits vs power (μW).

multiplexing scheme, eg., for AND it requires 12 transistors and for XOR it requires 20 transistors.

6.4.2 DPA attack resistance

We further our experiment by constructing a cryptographic system as shown in Fig. 6.1. Difference of mean (DoM) based DPA attack [26] and CPA attack [28] are performed on both of our Lucifer and Present S-box implementations. We used 25,000 random vectors to launch a DoM attack on this system and obtained 500,000 power traces. For each of the possible 16 keys, Fig. 6.11 and Fig. 6.12 plot the difference of mean of power dissipation for each of the output bits of the Lucifer and the Present S-boxes, respectively. It can be easily seen from these figures that the correct key is not distinguishable from the others.

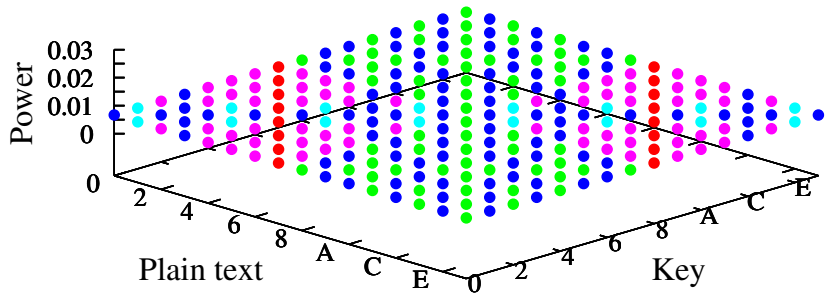


Figure 6.13: CPA attack on our Lucifer S-box design with top-bottom pre-charge logic: plain text vs key vs power (μW).

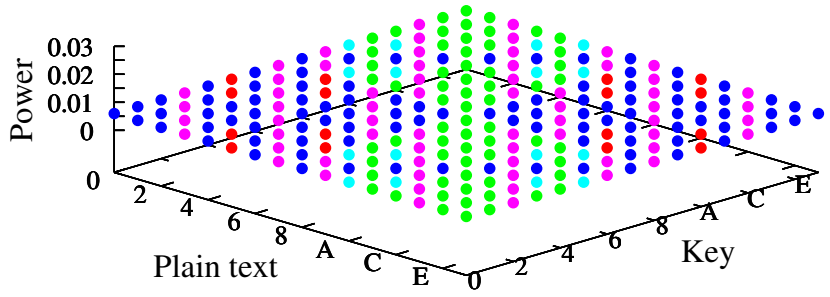


Figure 6.14: CPA attack on our Present S-box design with top-bottom pre-charge logic: plain text vs key vs power (μW).

6.4.3 CPA attack resistance

CPA attacks exploit the correlation factor between the power samples and the hamming weights of the handled data [28]. Accordingly, we plot the power dissipation for all the possible 256 combinations of the plain texts and the keys as shown in Fig. 6.13 and Fig. 6.14, where points belonging to different hamming weights are highlighted with different colours; all the input combinations are found to lie on the same plane which indicates that they exhibit identical power dissipation. The mean variances of power dissipation for the Lucifer and the Present S-boxes with top-bottom pre-charge logic are found to be 1.7E-5μW and 1.4E-5μW, respectively. As evident form these statistics, the required information to identify the correct key (by way of separation) is not available.

Table 6.9: Comparison with respect to NED and NSD

	Normalized energy deviation (NED)			Normalized standard deviation (NSD)		
	DPBDD	SDMLp	TopBtm	DPBDD	SDMLp	TopBtm
Lucifer	0.68215	0.70102	0.00102	0.32311	0.33182	0.00003
Present	0.77242	0.08021	0.01130	0.34301	0.35216	0.00090

6.4.4 Comparison in terms of normalized attributes

In addition, we compare our S-box designs with those of DP-BDD and SDMLp based on the parameters normalized energy deviation (NED) and normalized standard deviation (NSD) [43]. It can be seen from Table 6.9 that NED and NSD of our S-box implementations are much less than that of the others, which establish the superior PAA resistance of our hardware designs.

Table 6.10: Delay in output generation for the basic cell

Pre	Input0	Input1	Select	Time
0	0	0	0	1
0	0	0	1	1
⋮	⋮	⋮	⋮	1
0	1	1	1	1
1	0	0	0	2
⋮	⋮	⋮	⋮	2
1	1	1	1	2

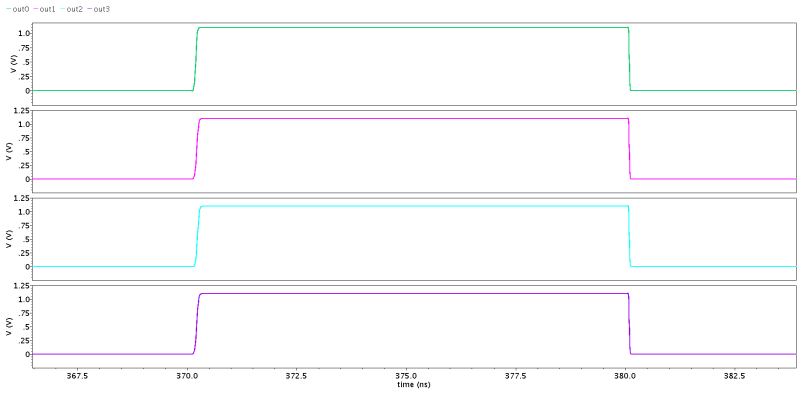


Figure 6.15: Transient response of the four output bits (all 1s) generated by the Lucifer S-box: time (ns) vs voltage (V).

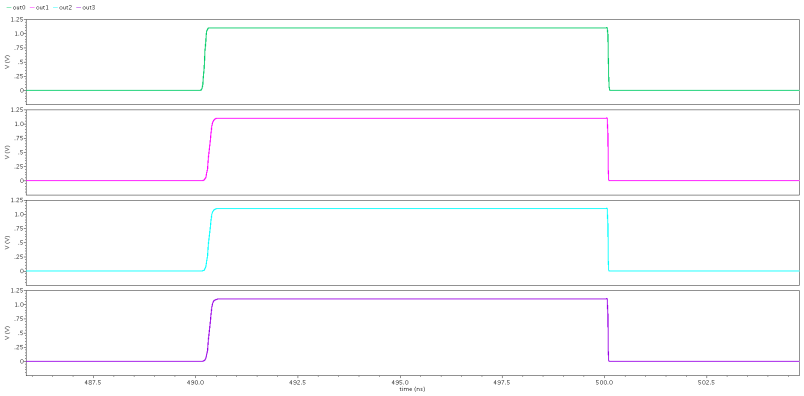


Figure 6.16: Transient response of the four output bits (all 1s) generated by the Present S-box: time (ns) vs voltage (V).

6.4.5 EPE attack resistance

The timing characteristics of our circuit designs are analyzed next. In Table 6.10, we tabulate the delays in output generation for the basic cell in terms of number of transistor switchings. The basic cell undergoes one transistor switching whenever the pre-charge is 0 and two transistor switchings whenever the pre-charge is 1; recall that whenever pre-charge is 0, the output produced is also 0 irrespective of the inputs provided. Furthermore, we plot the transient response of the four output bits generated by the two S-boxes in Fig. 6.15 and Fig. 6.16; we deliberately chose the respective input combinations which produce 1s at all the output bits of the S-boxes to reveal

that the four outputs are generated simultaneously and hence launching EPE attack by distinguishing among the delays in output generation is not viable.

6.5 Experimentation for Top pre-charge logic in 65 nm technology

Table 6.11: Comparison with other methods

Circuits	Peak Power Variance (μW)			Average Power (μW)			Average Current (μA)			Propagation Delay (nS)		
	DPBDD	SDMLp	Top	DPBDD	SDMLp	Top	DPBDD	SDMLp	Top	DPBDD	SDMLp*	Top
AND	835.33	15.82	0.01	3.91	8.53	3.79	3.68	7.64	3.45	0.05	0.09×2	0.08
OR	1092.11	31.55	0.01	3.86	8.48	3.79	3.63	7.61	3.45	0.05	0.08×2	0.09
XOR	594.33	13.21	0.01	4.30	8.61	3.78	4.04	7.73	3.45	0.06	0.09×2	0.09
NAND	835.33	15.82	0.01	3.91	8.53	3.79	3.68	7.64	3.45	0.05	0.09×2	0.08
NOR	1092.11	31.55	0.01	3.86	8.48	3.79	3.63	7.61	3.45	0.05	0.08×2	0.09
XNOR	594.33	13.21	0.01	4.30	8.61	3.78	4.04	7.73	3.45	0.06	0.09×2	0.09
MUX	564.16	30.89	0.06	4.51	11.28	3.79	4.22	10.92	3.45	0.05	0.10×2	0.08
Avg	775.95	22.87	0.02	4.14	9.22	3.79	3.89	8.46	3.45	0.05	0.09×2	0.08
Lucifer	37841.10	6711.00	0.08	83.33	223.40	33.53	79.72	203.51	30.48	0.15	0.19×2	0.14
Present	41991.22	6965.04	7.01	96.44	226.83	24.77	92.26	206.79	22.39	0.14	0.19×2	0.17

*It may be noted that SDMLp has a twin cycle operation as described in section 1.

We elaborate the experiments carried out to establish the DPA and CPA resistance of our implementation of the basic cell and the two S-boxes. Through the experiments described below, we characterize peak power variance, average power consumption, average current consumption and propagation delay for our design and make comparison of these parameters with that of DP-BDD [3] and SDMLp [35].

Schematic capture and standard spectra simulations have been performed, and current and power waveforms have been plotted. Experimental results are given in Table 6.11.

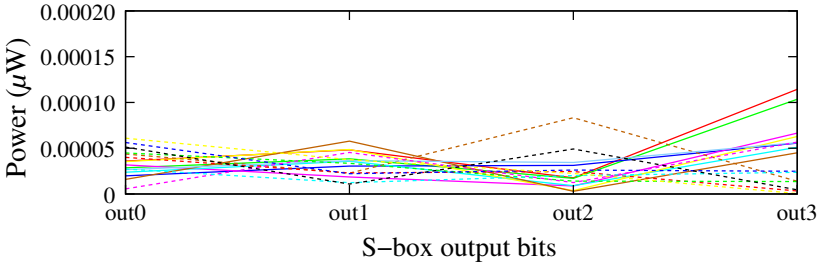


Figure 6.17: DPA attack on our Present S-box design with top pre-charge logic: S-box output bits vs power (μW).

6.5.1 Comparison in terms of standard attributes

Experimental results on circuits with various other features demonstrate a 99.9% and 99% reduction in PPV for the basic cell with top pre-charge logic with respect to DP-BDD and SDMLp, respectively; Significant reduction for average power and average current for both the pre-charge logics is also achieved. Although the propagation delay for the basic cell is found to be almost twice of that of DP-BDD, it is comparable with that of SDMLp.

6.5.2 DPA attack resistance

We further our experiment by constructing a cryptographic system as shown in Fig. 6.1. DoM based DPA attack [26] and CPA attack [28] are performed on both of our Present S-box implementations, each involving one of the two pre-charge logics. We used 25,000 random vectors to launch a DoM attack on this system and obtained 500,000 power traces. For each of the possible 16 keys, Fig. 6.17 and Fig. 6.18 plot the difference of mean of power dissipation for each of the output bits of the two S-boxes.

6.5.3 CPA attack resistance

CPA attacks exploit the correlation factor between the power samples and the hamming weights of the handled data [28]. Accordingly, we plot the power dissipation for all the possible 256 combinations of the plain texts and the keys as shown in Fig. 6.19, where points belonging to different hamming weights are highlighted with different

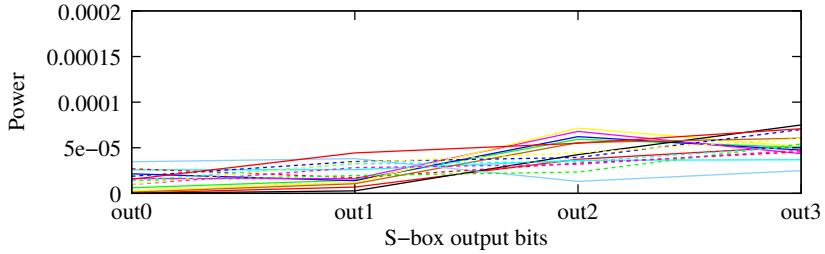


Figure 6.18: DPA attack on our Lucifer S-box design with top pre-charge logic: S-box output bits vs power (μW).

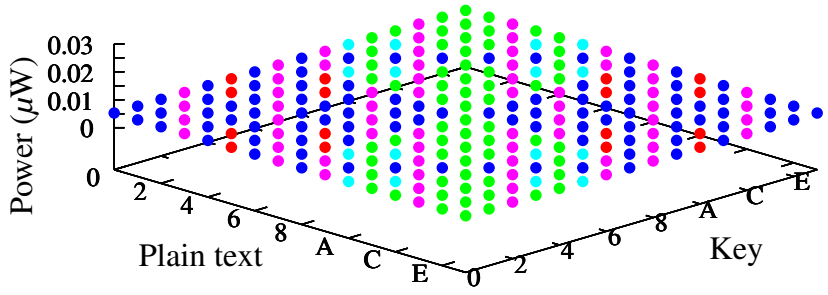


Figure 6.19: CPA attack on our Present S-box design with top pre-charge logic: plain text vs key vs power (μW).

colours. The mean variances of power dissipation for the Present S-box with top pre-charge and top-bottom pre-charge are found to be $3.2 \times 10^{-5}\mu$W and $1.4 \times 10^{-5}\mu$W, respectively. As evident form these statistics, the required information to identify the correct key (by way of separation) is not available adequately.

Table 6.12: Comparison with respect to NED and NSD

	Normalized energy deviation (NED)			Normalized standard deviation (NSD)		
	DPBDD	SDMLp	Top pre-charge	DPBDD	SDMLp	Top pre-charge
Lucifer	0.68215	0.70102	0.00032	0.32311	0.33182	0.0001
Present	0.77242	0.08021	0.03667	0.34301	0.35216	0.03103

6.5.4 Comparison in terms of normalized attributes

We compare our S-box designs with those of DP-BDD and SDMLp based on the parameters normalized energy deviation (NED) and normalized standard deviation

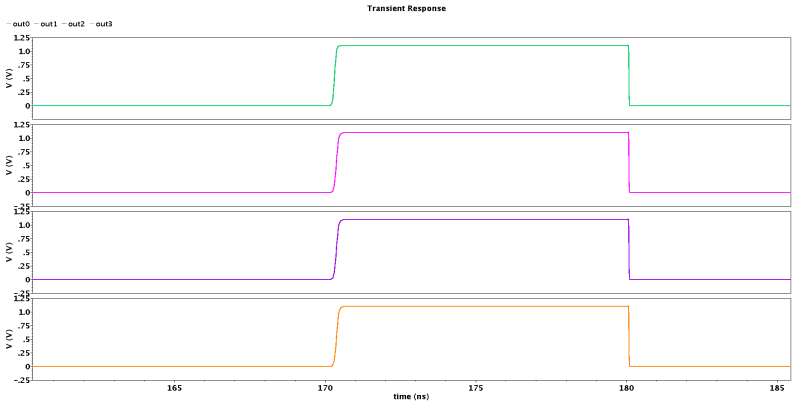


Figure 6.20: Transient response of the four output bits (all 1s) generated by the Lucifer S-box with top pre-charge: time (ns) vs voltage (V).

(NSD) [43]. It can be seen from Table 6.12 that NED and NSD of our S-box implementations are significantly lesser than those of the others.

Table 6.13: Delay in output generation for the basic cell

Pre	Input0	Input1	Select	Time
0	0	0	0	1
0	0	0	1	1
⋮	⋮	⋮	⋮	1
0	1	1	1	1
1	0	0	0	2
⋮	⋮	⋮	⋮	2
1	1	1	1	2

6.5.5 EPE attack resistance

The timing characteristics of our circuit designs are analyzed next. In Table 6.13, we tabulate the delays in output generation for the basic cell in terms of number of transistor switchings. The basic cell undergoes one transistor switching whenever the pre-charge is 0 and two transistor switchings whenever the pre-charge is 1; it may be recalled that whenever pre-charge is 0, the output produced is also 0 irrespective of the inputs provided. Furthermore, we plot the transient response of the four output

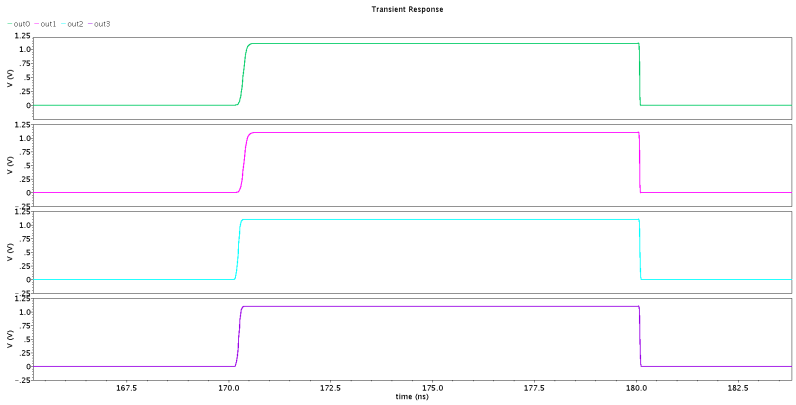


Figure 6.21: Transient response of the four output bits (all 1s) generated by the Present S-box with top pre-charge: time (ns) vs voltage (V).

bits generated by the two S-boxes in Fig. 6.20 and Fig. 6.21; we deliberately chose the respective input combinations which produce 1s at all the output bits of the S-boxes to reveal that the four outputs are generated simultaneously and hence establishing that launching EPE attack by distinguishing among the delays in output generation is not viable.

6.6 Conclusion

Experimental results on circuits with bottom pre-charge logic demonstrate a significant reduction by 99.82% and 93.69% in peak power variance (PPV) with respect to DP-BDD and SDMLp, respectively, for the basic cell. A reduction of about 17.39% in both average power and average current consumption is observed for the DP-BDD implementation of the basic cell whereas, for SDMLp, these parameters are found to be reduced by 61.48%.

Experimental results on circuits with symmetric NMOS transistors based bottom pre-charge logic demonstrate a significant reduction by 99.68% and 88.55% in peak power variance (PPV) with respect to DP-BDD and SDMLp, respectively, for the basic cell. The reduction in PPV is recorded to be greater than 99.9% for the S-box implementations for both DP-BDD and SDMLp. A reduction of about 30% in both average power and average current consumption is observed for the DP-BDD implementation of the basic cell whereas, for SDMLp, these parameters are found to

be reduced by 67%.

Experimental results on circuits with various other features demonstrate a 99.9% and 99% reduction in PPV for the basic cell with top pre-charge logic with respect to DP-BDD and SDMLp, respectively; for the basic cell with top-bottom pre-charge logic, the results demonstrate a 99.9% and 98.3% reduction. Significant reduction for average power and average current for both the pre-charge logics is also achieved.

Circuits using top pre-charging required less transistors and demonstrated lower PPV in comparison with others. Symmetric NMOS bottom pre-charge logic was more resilient to EPE due to its symmetric nature. Bottom pre-charge and symmetric NMOS bottom pre-charge logic were also effective in avoiding timing attacks along with top-bottom pre-charge logic.

Chapter 7

Conclusions and future work

In this work, *balanced* BDD based dual rail circuit design technique relying on (*i*) BDD based pre-charging to counter EPE, (*ii*) complementary path balanced BDD logic network for delay equalization and countering EPE, and (*iii*) voltage scaling and leakage power minimization for overall power reduction is presented. The design mechanism also supports extra fan-out with the use of only one additional inverter (whereas, all other logics require two). A synthesis tool to automate our design mechanism has also been developed which generates transistor-level Verilog code.

Circuit development is done with schematic capture and layout design. The standard spectra simulations have been performed. Current and power waveforms are plotted. The objective of our experimentation was two fold, (*i*) to demonstrate resistance to power attacks (*ii*) to highlight the low power characteristics. Towards the first objective we have carried out differential power attacks such as, difference of mean (DoM) and correlation power attack (CPA). We demonstrate resilience against the the early propagation effect (EPE). Our designs also outperformed other competing methods in terms of DPA attack resistance metrics along with average power and current. Six 2-input basic cell and two 4×4 S-boxes are used for experimental benchmark. Summary of experimentation for our various pre-charging schemes are given next.

Experimental results on circuits with bottom pre-charge logic demonstrate a significant reduction by 99.82% and 93.69% in peak power variance (PPV) with respect to DP-BDD and SDMLp, respectively, for the basic cell. A reduction of about 17.39% in both average power and average current consumption is observed for the DP-BDD

implementation of the basic cell whereas, for SDMLp, these parameters are found to be reduced by 61.48%.

Experimental results on circuits with symmetric NMOS transistors based bottom pre-charge logic demonstrate a significant reduction by 99.68% and 88.55% in peak power variance (PPV) with respect to DP-BDD and SDMLp, respectively, for the basic cell. The reduction in PPV is recorded to be greater than 99.9% for the S-box implementations for both DP-BDD and SDMLp. A reduction of about 30% in both average power and average current consumption is observed for the DP-BDD implementation of the basic cell whereas, for SDMLp, these parameters are found to be reduced by 67%.

Experimental results on circuits with various other features demonstrate a 99.9% and 99% reduction in PPV for the basic cell with top pre-charge logic with respect to DP-BDD and SDMLp, respectively; for the basic cell with top-bottom pre-charge logic, the results demonstrate a 99.9% and 98.3% reduction. Significant reduction for average power and average current for both the pre-charge logics is also achieved.

Through our experimentation we have established that circuits developed using our technique achieve significant reductions in peak power variance, average power and average current consumption in comparison with those of competing techniques. Strong power analysis attacks carried on our S-box implementations were successfully repelled, thereby further establishing PAA resistance of our secure hardware design circuits and the supporting synthesis technique.

7.1 Future work

A counter measure for repelling power attacks is to randomize the intermediate results occurring during the execution of the cryptographic algorithm. Masking logic exploits this idea by ensuring that the power consumption of operations on randomized data is not correlated with the actual plain intermediate data. However, the various design models based on masking have neglected the possibility of multiple switching of the outputs of the gates in a single clock cycle although this phenomenon, termed as *glitching*, is typical for CMOS circuits. Binary Decision Diagram (BDD) based circuit synthesis mechanisms have been demonstrated to successfully safeguard se-

cret information against power analysis attack by maintaining identical critical path lengths from the root to all leaf nodes, thereby ensuring that computation along each decision branch of the BDD will be through same number of transistors, thus giving rise to identical delays. In future work, one may explore that this property of BDDs to eliminate glitches in circuits. A secure hardware design flow that combines BDD based circuit synthesis technique with masked dual-rail pre-charge logic can also be targeted.

Our BDD based dual-rail synthesis mechanism to counter power attacks can be further extended to the dual-rail three phase mechanism. Working principle of three phase mechanism can be classified like pre-charge phase, evaluation phase and pre-discharge phase. By maintaining identical critical path lengths from the root to all leaf nodes, thereby ensuring that computation along each decision branch of the BDD will be through same number of transistors, security of the three phase circuit can be further enhanced. This seems to be a promising future endeavour.

Appendix A

BDDs of AES generated by automated synthesis tool

Each hexadecimal input `x` represents the eight input bits of the S-box, numbered as `V0`, `V1`, `V2` `V3`, `V4`, `V5`, `V6` and `V7`. The corresponding output `S[x]` represents the four output bits of the S-box, numbered as `out0`, `out1`, `out2`, `out3`, `out4`, `out5`, `out6` and `out7`. Thus, each output bit is basically a function of the eight input bits.

Our structure of AES `out0`, `out1`, `out2`, `out3`, `out4`, `out5`, `out6` and `out7` is shown in the Figs. A.1, A.2, A.3, A.4, A.5, A.6, A.7 and A.8 respectively.

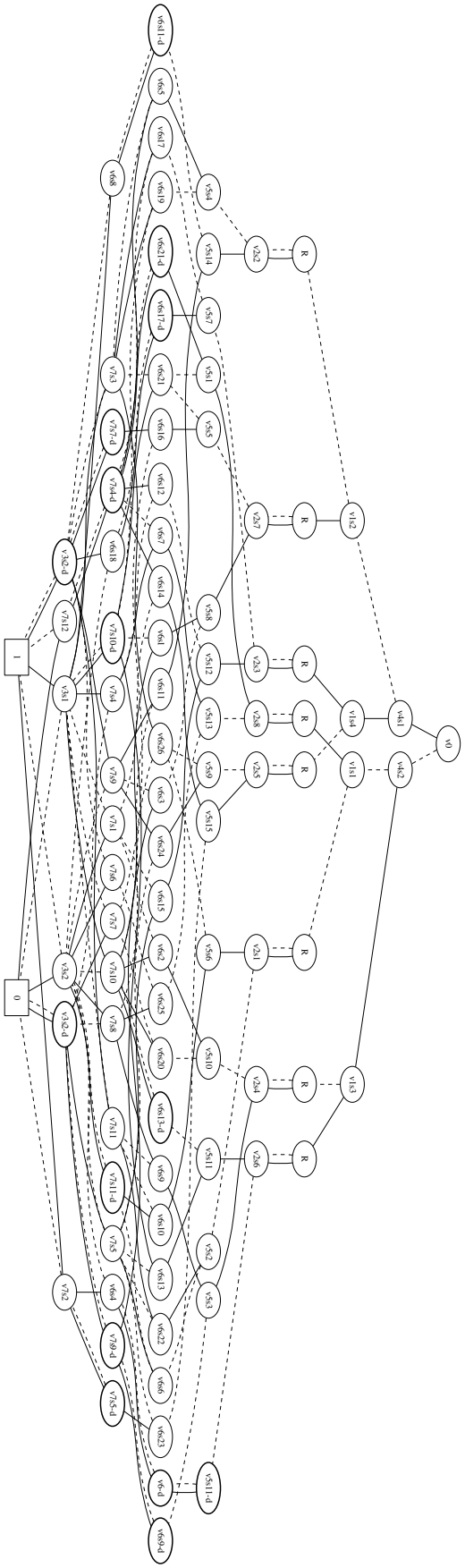


Figure A.1: The balanced BDD of AES out0 with repeaters.

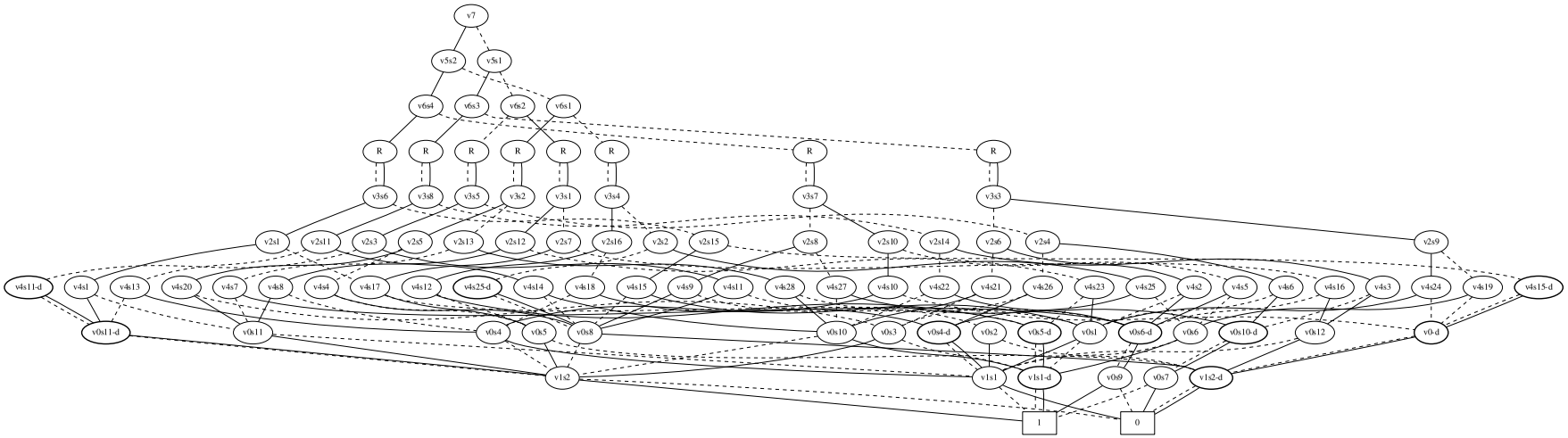


Figure A.2: The balanced BDD of AES out1 with repeaters.

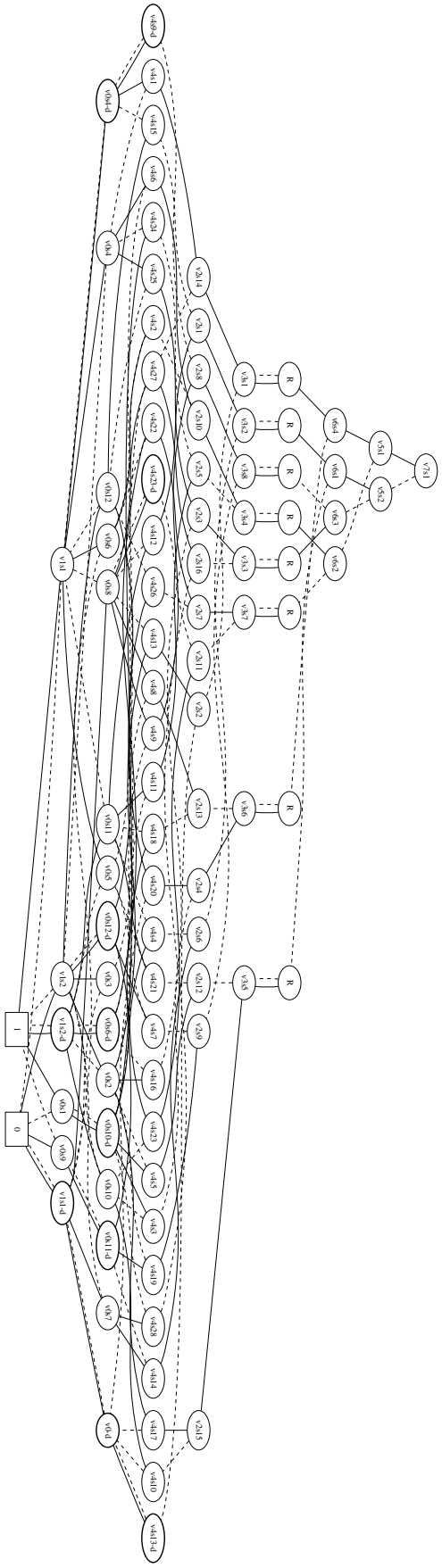


Figure A.3: The balanced BDD of AES out2 with repeaters.

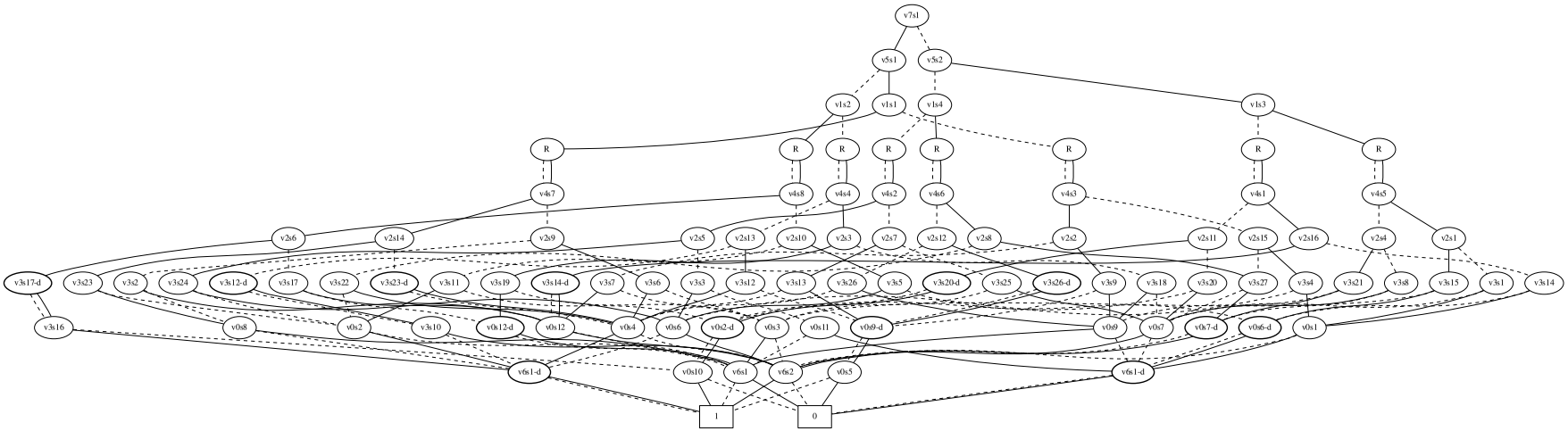


Figure A.4: The balanced BDD of AES out3 with repeaters.

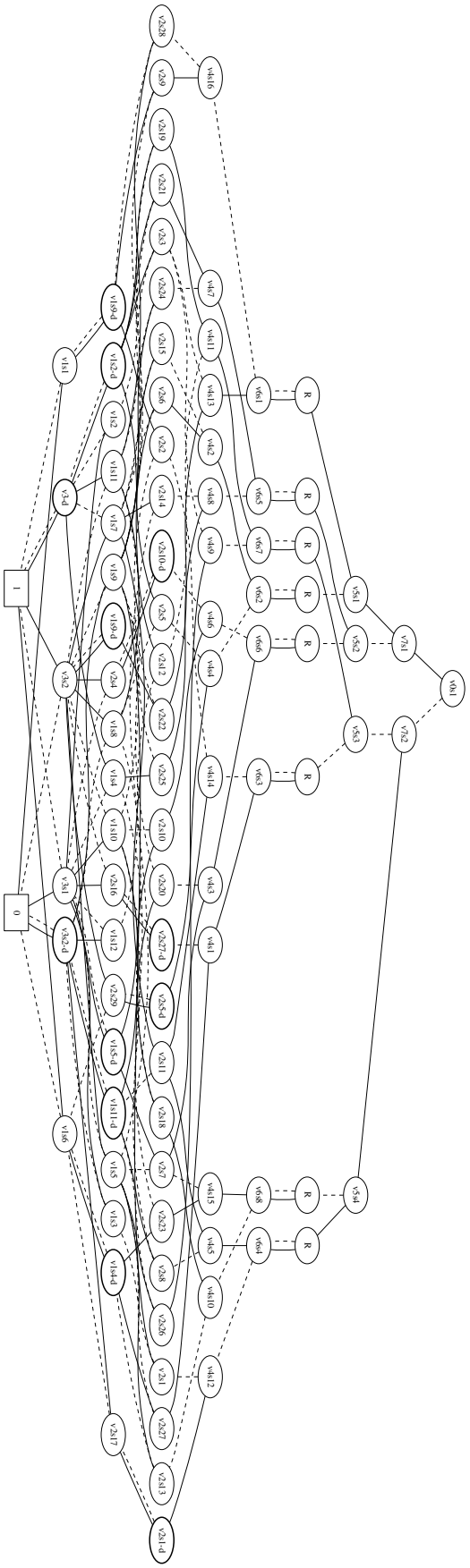


Figure A.5: The balanced BDD of AES out4 with repeaters.

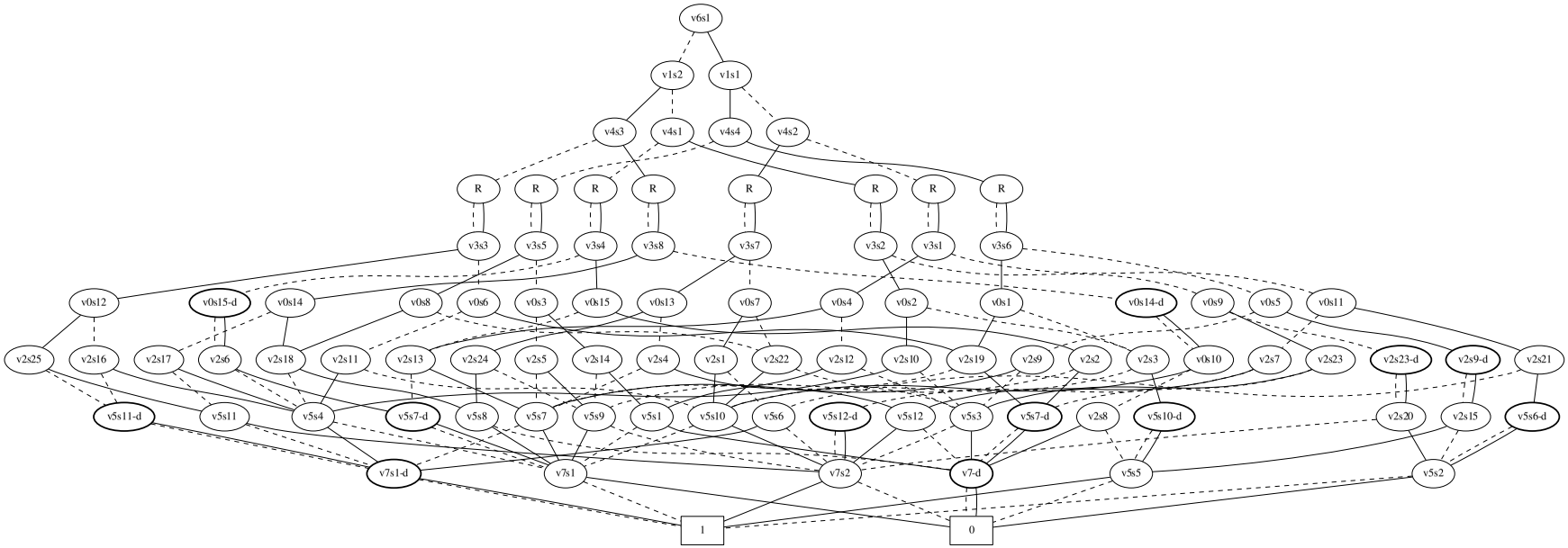


Figure A.6: The balanced BDD of AES out5 with repeaters.

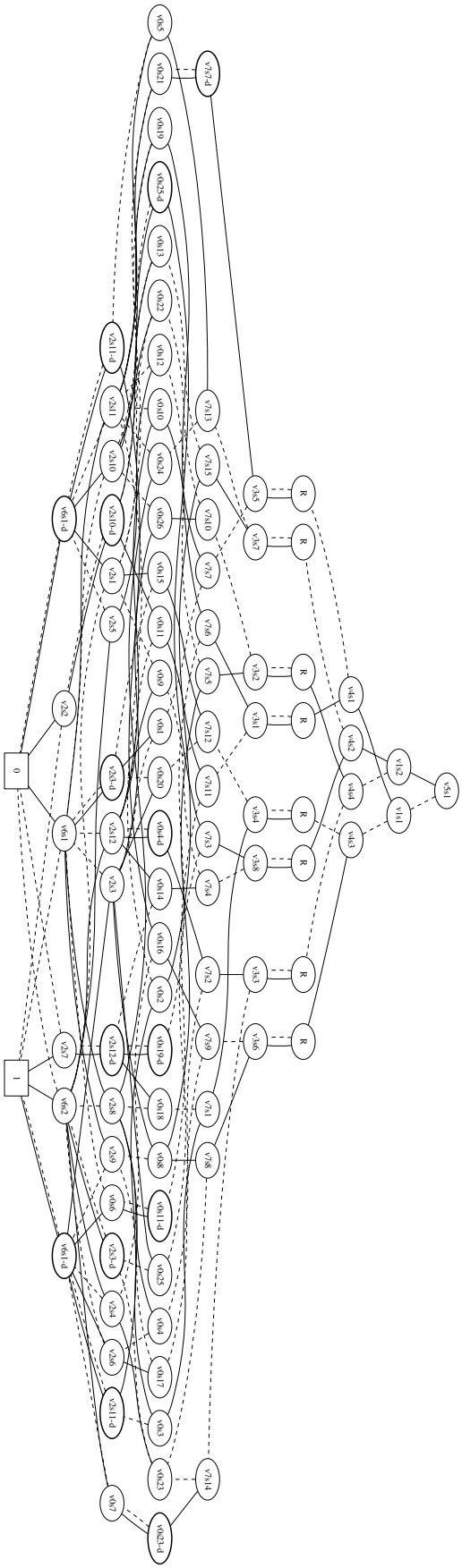


Figure A.7: The balanced BDD of AES out6 with repeaters.

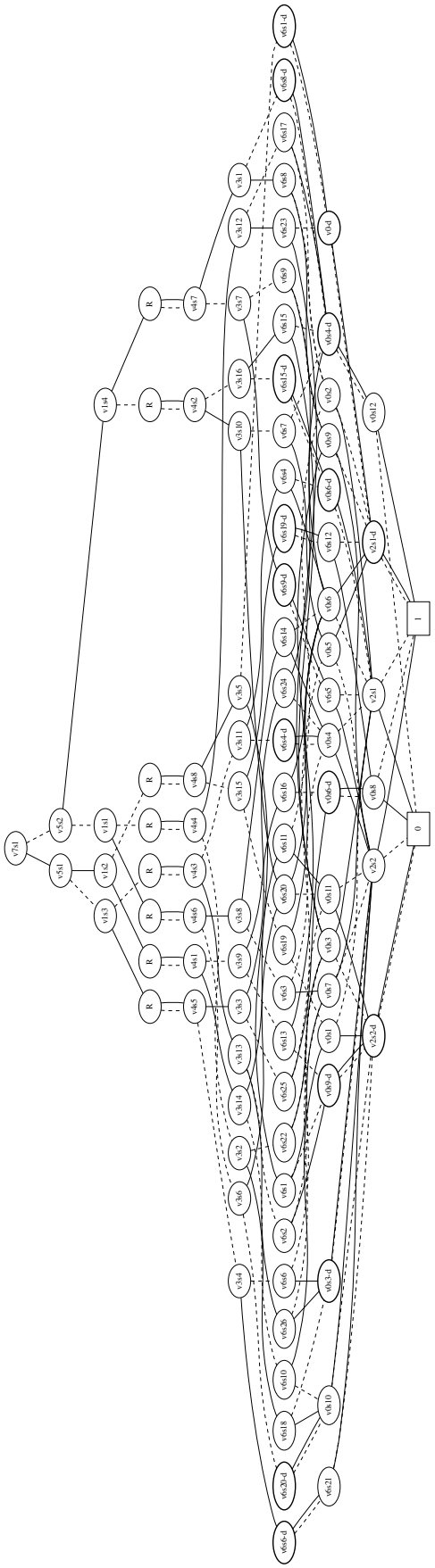


Figure A.8: The balanced BDD of AES out7 with repeaters.

Bibliography

[1] Dakshi Agrawal, Bruce Archambeault, Josyula R. Rao, and Pankaj Rohatgi. The EM Side-Channel(s). In *CHES*, pages 29–45, 2002. 2, 25

[2] S. B. Akers. Binary Decision Diagrams. *IEEE Trans. Comput.*, 27(6):509–516, June 1978. 5

[3] Toru Akishita, Masanobu Katagi, Yoshikazu Miyato, Asami Mizuno, and Kyoji Shibutani. A Practical DPA Countermeasure with BDD Architecture. In *CARDIS*, pages 206–217, 2008. 4, 93, 98, 102, 107

[4] Mehdi-Laurent Akkar and Christophe Giraud. An Implementation of DES and AES, Secure against Some Attacks. In *Proceedings of the Third International Workshop on Cryptographic Hardware and Embedded Systems*, CHES '01, pages 309–318. Springer-Verlag, 2001. 4, 26

[5] Karthik Baddam. *Hardware level countermeasures against differential power analysis*. PhD thesis, University of Southampton, 2012. 1

[6] V. Bertacco, S. Minato, P. Verplaetse, L. Benini, and G. De Micheli. Decision diagrams and pass transistor logic synthesis. In *In Int'l Workshop on Logic Synth*, 1997. 19

[7] Régis Bevan and Erik Knudsen. Ways to Enhance Differential Power Analysis. In *ICISC*, pages 327–342, 2002. 5

[8] Eli Biham and Adi Shamir. Differential Fault Analysis of Secret Key Cryptosystems, 1997. 3, 25

[9] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In *CHES*, pages 450–466, 2007. 7, 31, 47, 58, 65, 69, 93, 98

[10] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In *CHES*, pages 16–29, 2004. 28

[11] Randal E. Bryant. Graph-Based Algorithms for Boolean Function Manipulation. *IEEE Trans. Comput.*, 35(8):677–691, August 1986. 5

[12] Randal E. Bryant. Graph-Based Algorithms for Boolean Function Manipulation. *IEEE Trans. Computers*, 35(8):677–691, 1986. 61

[13] Randal E. Bryant. Symbolic Boolean Manipulation with Ordered Binary-decision Diagrams. *ACM Comput. Surv.*, 24(3):293–318, September 1992. 5

[14] Premal Buch, Amit Narayan, A. Richard Newton, and Alberto L. Sangiovanni-Vincentelli. Logic synthesis for large pass transistor circuits. In *ICCAD*, pages 663–670, 1997. 19

[15] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In *CRYPTO*, pages 398–412, 1999. 5

[16] Christophe Clavier, Jean-Sébastien Coron, and Nora Dabbous. Differential Power Analysis in the Presence of Hardware Countermeasures. In *CHES*, pages 252–263, 2000. 5

[17] P. Dasgupta. *A Roadmap for Formal Property Verification*. Springer, 2007. 13

[18] Des. Data Encryption Standard. In *In FIPS PUB 46, Federal Information Processing Standards Publication*, pages 46–2, 1977. 2

[19] Jean-FranÃ§ois Dhem, FranÃ§ois Koeune, Philippe-Alexandre Leroux, Patrick Mestre, Jean-Jacques Quisquater, and Jean-Louis Willems. A practical implementation of the timing attack. In Jean-Jacques Quisquater and Bruce Schneier, editors, *Proceedings of the Third Working Conference on Smart Card Research and Advanced Applications (CARDIS 1998)*, volume 1820 of *LNCS*. Springer-Verlag, 1 1998. 2, 25

[20] Fabrizio Ferrandi, Alberto Macii, Enrico Macii, Massimo Poncino, Riccardo Scarsi, and Fabio Somenzi. Symbolic algorithms for layout-oriented synthesis of pass transistor logic circuits. In *ICCAD*, pages 235–241, 1998. 19

[21] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic Analysis: Concrete Results. In *Proceedings of the Third International Workshop on Cryptographic Hardware and Embedded Systems*, CHES '01, pages 251–261, London, UK, UK, 2001. Springer-Verlag. 2, 25

[22] A. Chandrakasan J. M. Rabaey and B. Nikolic. *Digital Integrated Circuit Design*. Prentice Hall, 2nd edition, 2003. 4, 36

[23] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209, 1987. 2

[24] Paul C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In *CRYPTO*, pages 104–113, 1996. 2, 25

[25] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In *CRYPTO*, pages 388–397, 1999. 27

[26] Paul C. Kocher, Joshua Jaffe, Benjamin Jun, and Pankaj Rohatgi. Introduction to differential power analysis. *J. Cryptographic Engineering*, 1(1):5–27, 2011. 2, 3, 25, 89, 94, 98, 103, 108

[27] Konrad J. Kulikowski, Mark G. Karpovsky, and Er Taubin. Power Attacks on Secure Hardware Based on Early Propagation of Data. In *IOLTS*, pages 10–12, 2006. 4, 28, 35, 62

[28] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks - Revealing the Secrets of Smart Cards*. Springer, 2007. 89, 91, 94, 95, 98, 99, 103, 104, 108

[29] Victor S Miller. Use of elliptic curves in cryptography. In *Lecture Notes in Computer Sciences; 218 on Advances in cryptology—CRYPTO 85*, pages 417–426, New York, NY, USA, 1986. Springer-Verlag New York, Inc. 2

[30] Amir Moradi and Axel Poschmann. Lightweight Cryptography and DPA Countermeasures: A Survey. In *Financial Cryptography Workshops*, pages 68–79, 2010. 2, 3, 25

[31] National and N. I. S. T. Technology. *Announcing the Advanced Encryption Standard (AES)*, 2001. 2

[32] National Institute of Standards and Technology. In *Announcing the Standard for DIGITAL SIGNATURE STANDARD (DSS)*, page 197, 1994. 2

[33] Elisabeth Oswald, Stefan Mangard, Norbert Pramstaller, and Vincent Rijmen. A Side-Channel Analysis Resistant Description of the AES S-Box. In *FSE*, pages 413–423, 2005. 4, 26

[34] Gopal Paul, Rohit Reddy, Chittaranjan A. Mandal, and Bhargab B. Bhattacharya. A bdd-based design of an area-power efficient asynchronous adder. In *ISVLSI*, pages 29–34, 2010. 43

[35] Lakshmi Narasimhan Ramakrishnan, Manoj Chakkaravarthy, Antarpreet Singh Manchanda, Mike Borowczak, and Ranga Vemuri. SDMLp: On the Use of Complementary Pass Transistor Logic for Design of DPA Resistant Circuits. In *HOST*, pages 31–36, 2012. 4, 92, 93, 98, 102, 107

[36] Michael Rice and Sanjay Kulhari. A Survey of Static Variable Ordering Heuristics for EfficientBDD/MDD Construction. *University of California, Tech. Rep*, 2008. 14

[37] R.L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21:120–126, 1978. 2

[38] Sergei Skorobogatov. Low temperature data remanence in static RAM. Technical Report UCAM-CL-TR-536, University of Cambridge, Computer Laboratory, June 2002. 3, 25

[39] Fabio Somenzi. CUDD: CU Decision Diagram Package. http://vlsi.colorado.edu/~fabio/CUDD/, 2012. 37, 43, 75

[40] Arthur Sorkin. Lucifer: A Cryptographic Algorithm. *Cryptologia*, 8(1):22–42, 1984. 7, 31, 47, 58, 65, 69, 98

[41] Daisuke Suzuki and Minoru Saeki. Security Evaluation of DPA Countermeasures Using Dual-Rail Pre-charge Logic Style. In *CHES*, pages 255–269, 2006. 5

[42] K. Tiri, M. Akmal, and I. Verbauwhede. A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards. In *Solid-State Circuits Conference, 2002. ESSCIRC 2002. Proceedings of the 28th European*, pages 403–406, 2002. 4

[43] K. Tiri, M. Akmal, and I. Verbauwhede. A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards. In *ESSCIRC 2002*, pages 403–406, Sept 2002. 90, 96, 100, 105, 110

[44] Kris Tiri, David Hwang, Alireza Hodjat, Bo-Cheng Lai, Shenglin Yang, Patrick Schaumont, and Ingrid Verbauwhede. Prototype IC with WDDL and Differential Routing - DPA Resistance Assessment. In *CHES*, pages 354–365, 2005. 4

[45] Kris Tiri and Ingrid Verbauwhede. Securing Encryption Algorithms against DPA at the Logic Level: Next Generation Smart Card Technology. In *CHES*, pages 125–136, 2003. 4, 5

[46] Kris Tiri and Ingrid Verbauwhede. A Dynamic and Differential CMOS Logic Style to Resist Power and Timing Attacks on Security ICs. *IACR Cryptology ePrint Archive*, 2004:66, 2004. 4

[47] Kris Tiri and Ingrid Verbauwhede. A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation. In *DATE*, pages 246–251, 2004. 4, 5, 34

[48] Kris Tiri and Ingrid Verbauwhede. A digital design flow for secure integrated circuits. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 25(7):1197–1208, 2006. 4

[49] Eran Tromer. Acoustic cryptanalysis: on nosy people and noisy machines. *Eurocrypt2004 Rump Session, May*, 2004. 2, 25

[50] Michael Tunstall, Debdeep Mukhopadhyay, and Subidh Ali. Differential Fault Analysis of the Advanced Encryption Standard Using a Single Fault. In *WISTP*, pages 224–233, 2011. 3, 25

[51] Reto Zimmermann and Wolfgang Fichtner. Low-power logic styles: Cmos versus pass-transistor logic. *IEEE J. Solid-State Circuits*, 32:1079–1090, 1997. 18

List of Publications out of this work

1. "Designing DPA Resistant Circuits Using BDD Architecture and Bottom Precharge Logic," **Partha De**, Kunal Banerjee, Chittaranjan Mandal, Debdeep Mukhopadhyay. 16^{th} Euromicro Conference on Digital System Design (DSD), September 2013 (accepted as poster).

2. "A BDD Based Circuit Synthesis Approach to Counter Power Analysis Attacks," **Partha De**, Chittaranjan Mandal, Kunal Banerjee. 27^{th} IEEE International Conference on VLSI Design, 2014 (accepted as poster).

3. "A BDD based Secure Hardware Design Method to Guard Against Power Analysis Attacks," **Partha De**, Kunal Banerjee, Chittaranjan Mandal. 18^{th} International Symposium on VLSI Design and Test (VDAT), 2014 (accepted as poster).

4. "Circuits and Synthesis Mechanism for Hardware Design to Counter Power Analysis Attacks," **Partha De**, Kunal Banerjee, Chittaranjan Mandal, Debdeep Mukhopadhyay. 17^{th} Euromicro Conference on Digital System Design (DSD), August 2014.