Secured-by-Design FPGA against Side-Channel Attacks based on Power Consumption

by

Ziyad Mohammed Almohaimeed

B.Sc., Qassim University , 2009

M.A.Sc., University of Victoria, 2013

A Dissertation Submitted in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Electrical and Computer Engineering

Secured-by-Design FPGA against Side-Channel Attacks based on Power Consumption

by

Ziyad Mohammed Almohaimeed

B.Sc., Qassim University , 2009

M.A.Sc., University of Victoria, 2013

Supervisory Committee

Dr. Mihai Sima, Supervisor

(Department of Electrical and Computer Engineering)

Dr. Stephen Neville, Departmental Member

(Department of Electrical and Computer Engineering)

Dr. Florin Diacu, Outside Member

(Department of Mathematics and Statistics)

Supervisory Committee

Dr. Mihai Sima, Supervisor

(Department of Electrical and Computer Engineering)

Dr. Stephen Neville, Departmental Member

(Department of Electrical and Computer Engineering)

Dr. Florin Diacu, Outside Member

(Department of Mathematics and Statistics)

ABSTRACT

Power Analysis Attacks pose serious threats to hardware implementations of cryptographic systems. To retrieve the secret key, the attackers can exploit the mutual information between power consumption and processed data / operations through monitoring the power consumption of the cryptosystems. Field Programmable Gate Arrays (FPGA) have emerged as attractive implementation platforms for providing hardware-like performance and software-like flexibility for cryptosystem developers. These features come at the expense of larger power consumption, which makes FPGAs more vulnerable to power attacks. Different countermeasures have been introduced in the literature, but as they have originally been developed for Application-Specific Integrated Circuits (ASIC), mapping them onto FPGAs degrades their effectiveness. In this work we propose a logic

family based on pass transistors, which essentially consists of hardware replication, that can be used to build FPGAs with constant power consumption. Since the power consumption is no longer related to processed data and operations, a quadruple robustness to attacks based on dynamic power consumption, static power consumption, glitches, and early evaluation effect is achieved. Such a secured-by-design FPGA will relieve the cryptosystems developers from doing advanced analog design to secure the cryptosystem implementation. Our pass-transistor logic family can also be used in implementing ASICs. The silicon area overhead costs are shown to be less than prior art, which makes our FPGA attractive to cryptosystems developers.

ملخص الرسالة

الطاقة المستهلكة في الدوائر الكهربائية تنقسم إلى قسمين: قسم الطاقة الساكنة وقسم الطاقة المتغيرة. كلا النوعين من الطاقة المستهلكة تعتمدان اعتمادا مباشراً على نوعية الدالة المطبقة بالاضافة إلى البيانات المعالجة. هذه الحقيقة اعطت الفرصة للقراصنة للوصول إلى معلومات سرية (كلمة السر المستخدمة) في معظم انواع الخوارزميات المعترف بها عالمياً على سبيل المثال AES و DES. يعمد القراصنة في قرصنة الاجهزة الألكترونية إلى دراسة العلاقة بين نوع الطاقة المستهلكة والبيانات المعالجة. نظراً لسرعة الوصل إلى الهدف (كلمة السر) ولقلة تكلفة تنفيذ هذه العملية يشكل هذا النوع من القرصنة خطورة بالغة. البوابات المنطقية القابلة للبرمجة (FPGA) وفرت البيئة المناسبة لتصميم انظمة التشفير بأنو اعها نظراً لصغر حجمها وإمكا نية إعادة برمجتها، وباعتبارها دائرة كهربائية في الاصل فهي ليست بمنأى من خطر هذه الهجمات. وقد اثبنا واثبت ذلك في عدة دراسات أخرى أن الطاقة المستهلكة بنوعيها الساكنة والمتغيرة في الدائرة الكهربائية تقوم لا إراديا بتسريب معلومات مهمة حول الدالة المطبقة والبيانات المعالجة في الدائرة نفسها. في هذا البحث قمنا بدراسة تفصيلية لأسباب تسرب البيانات في الـ (FPGA) من خلال الطاقة المستهلكة. ووجدنا أن جميع أنواع الطاقة: الساكنة والمتغيرة والطاقة المعتمدة على الخطأ اللحظي في النظام والطاقة المعتمدة على معالجة بعض المدخلات قبل وصول إجمالي المدخلات تقوم بتسريب معلومات مهمة عن الدائرة المطبقة والمدخلات المعالجة. اعتمادا على الملاحظات والنتائج عمدنا إلى الحد من إمكانية الربط بين أي نوع من أنواع الطاقة المستهلكة والبيانات المعالجة عن طريق توحيد الطاقة المستهلكة مع جميع الدوال والبيانات المعالجة. لتوحيد الطاقة قمنا بإعادة بناء الدوائر الكهربائية لمكونات الـ (FPGA) مثل الدائرةالكهربائية للـ (LUT) والدائرة الكهربائية لصناديق توصيل الـ(LUTs) ببعضها البعض لتكوين نظام كامل. بالحد من هذه الهجمات، وفرنا عائلة جديدة من البوابات المنطقية القابلة للبرمجة تتميز بالامان ضد أنواع القرصنة التي تعتمد على تحليل البيانات المتسربة عبر الطاقة المستهلكة بجميع أنواعها.

Contents

List of Tables

List of Figures

List of Abbreviations

DES Data Encryption Standard

AES Advanced Encryption Standard

RSA Rivest-Shamir-Adleman

ECC Elliptic Curve Cryptography

FPGA Field Programmable Gate Array

ASIC Application-Specific Integrated Circuit

DRL Dual-Rail Logic

LUT Look-Up Table

MUX MultiPleXer

2TB Two-Transistor Branch

CLB Configurable Logic Block

SB Switch Box

CB Connection Box

FET Field Effect Transistor

SNR Signal-to-Noise Ratio

BLEs Basic Logic Elements

NRE Non-Recurrent Engineering

PTL Pass-Transistor Logic

SPA Simple Power Analysis

DPA Differential Power Analysis

CPA Correlation Power Analysis

NIST National Institute of Standards and Technology

RIP Random Initial Point

VLIW Very Long Instruction Word

MDPL Masked Dual-rail Precharge Logic

iMDPL improved Masked Dual-rail Precharge Logic

DRSL Dual-rail Random Switching Logic

PMRML Pre-charge Masked Reed-Muller Logic

FPRM Fixed Polarity Reed-Muller

SABL Sense Amplifier Based Logic

LBDL LUT-Based Differential Logic

WDDL Wave Dynamic Differential Logic

DWDDL double WDDL

iWDDL isolated WDDL

AWDDL Asynchronous WDDL

DBWDDL Double backend WDDL

BCDL Balanced Cell-based Differential Logic

DPL-noEE Dual Rail Precharge Logic without Early Evaluation

PA-DPL Precharge-Absorbed Dual-rail Precharge Logic

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to:

my parents, Mohammed Almohaimeed and Sharifah Alhassan for their encouragement and motivation throughout my life.

my wife, Manahil Almuqbil, for her patience, endless love, and support.

my son, Tariq, for his love and understanding.

my supervisor, Dr. Mihai SIMA, for his directions, patience, motivation, enthusiasm, and immense knowledge throughout this work, which provided me with precious enlightenment toward my work obstacles.

the committee members, Dr. Stephen Neville and Dr. Florin Diacu, for taking part of their time reading my dissertation and providing me with their valuable feedback and suggestions.

my sponsor, the Kingdom of Saudi Arabia represented by Qassim University, for funding me with a scholarship.

شكر وعرفان

(رَبِّ أَوْزِعْنِي أَنْ أَشْكُرَ نِعْمَتَكَ الَّتِي أَنْعَمْتَ عَلَيَّ وَعَلَىٰ وَالِدَيَّ وَأَنْ أَعْمَلَ صَالِحًا تَرْضَاهُ وَأَدْخِلْنِي بِرَحْمَتِكَ فِي عِبَادِكَ الصَّالِحِينَ)
النمل (19)

الحمد لله الذي بنعمته تتم الصالحات. أشكر الله سبحانه وتعالى أن وفقني لإكمال هذا البحث العلمي ولأتمام شكر الله لا بد من شكر أهل الفضل اعترافاً وتقديراً لفضلهم.

يشرفني شكر والديَ محمد بن عبدالله المحيميد و شريفة بنت عبدالعزيز الحسن على جهدهم المتواصل في حث الهمم والتشجيع على رسم الاهداف النبيلة والسعي لتحقيقها. أسأل الله ان يعينني على البر بهما وأن أكون عند حسن ظنهم.

كما يشرفني شكر زوجتي مناهل بنت محمد المقبل على تخفيف معاناة الغربة وتهيئة الأجواء المناسبة لأتمام هذا المشروع.

ابني طارق اعتذر منك لأنشغالي عنك خلال مراحل المشروع البحثي، واشكر لك تفهمك ذلك.

كما يشرفني شكر الدكتور/ ميهاي سيما (Dr. Mihai SIMA) على دعمهم المتواصل من خلال تقديم الاقتراحات البحثية مما انعكس ايجابا على رفع مستوى البحث كما اشكره على منحي الفرصة للأستفادة من خبرته العلمية والعملية.

كذلك يشرفني شكر الاساتذة أعضاء اللجنة التحكيمية للرسالة الدكتور/ ستيفن نيفيل (Dr. Stephen Neville) والدكتور / فلورن دياكو (Dr. Florin Diacu) على صرف جزء من وقتهم الثمين للأطلاع على البحث وتقديم الا قتراحات التي أثرت الاطروحة فلهم مني جزيل الشكر.

في الختام الشكر لدولتي المملكة العربية السعودية ممثلة بجامعة القصيم على الدعم المادي والمعنوي.

DEDICATION

To my parents, **Mohammed Almohaimeed and Sharifah Alhassan**, my source of inspiration and motivation.

To my lovely wife, **Manahil Almuqbil**.

To my beloved son, **Tariq**.

To my brothers and sisters.

Chapter 1

Introduction

Beyond any doubt, people rely on digital communications everyday. Online banking, e-Health and e-Government services enhance our day-to-day activities. However, they face privacy and security issues regarding confidential information. To securely exchange information between endpoints, it is essential to encrypt the communications. This chapter outlines the threats and difficulties in implementing cryptographic systems (also referred to as cryptosystems).

1.1 Motivation

A number of cryptographic algorithms such as Data Encryption Standard (DES) [111], Advanced Encryption Standard (AES) [83], Rivest-Shamir-Adleman (RSA) [100], and Elliptic Curve Cryptography (ECC) [56, 85] are in use today. All these algorithms perform complex operations (e.g. modular operation) on long operands (e.g. 256-bit integers). In order to encrypt/decrypt data in real time, such operations require high-performance implementations. Software implementations are flexible, but they are generally slow. In contrast, hardware implementations, e.g., Application-Specific Integrated Circuit (ASIC), are fast, but they are expensive and not flexible. Between these

two extremes, Field Programmable Gate Arrays (FPGAs) have emerged as an attractive platform to provide hardware-like performance with software-like flexibility.

A cryptosystem is an implementation of a cryptographic algorithm, which provides security for exchanged data. All the aforementioned cryptographic algorithms provide mathematical structures which are computationally hard to break. However, the hardware implementations of these cryptographic algorithms are known to be vulnerable to attacks that do not seek to compromise the mathematical structure of the cipher, but rather to target the electrical behaviour of device implrmrntation. The measurements of signals from a physical hardware implementation (e.g. power consumption, electromagnetic emissions) can provide side-channel information that hackers can exploit. Attacks such as Differential Power Analysis [58] and Correlation Power Analysis [19] use relations between data, operations, and power consumption to derive the secret key. It should be observed that the FPGA power consumption may be orders of magnitude greater than that of a device equivalent ASIC. This makes the FPGA-mapped cryptosystems highly vulnerable to side-channel attacks based on power consumption [70, 90, 109, 110].

Power dissipation in CMOS circuits have two components: **dynamic**, which is further subdivided into switching and short-circuit, and **static**, which is also referred to as leakage. A secured hardware implementation should be robust to both and all power component attacks. Securing the chip requires the elimination of the relation between processed data and consumed power. Two main techniques can be used to achieve this [74]: (i) **hiding (or concealing)**, which balances the power consumption into a constant value or introduces a random component into the power consumption, and (ii) **masking**, which randomizes the power consumption through scrambling the input data with a random mask. Both techniques can be applied at the algorithm level (e.g., through re-writing the code in order to use operations with equal latency and power consumption), at the architecture level (e.g, through the insertion of dummy arithmetic instructions to level the power consumption), or

at the circuit level (through designing a new type of logic family). The first two approaches tend to be (i) power demanding, due to the massive replication of coarse-grained operations and, arithmetic-logic units, incur a large silicon area overhead penalty, or (ii) require a significant programming effort [30,58,115]. Hence, this dissertation focuses on circuit-level countermeasures.

Dual-Rail Logic (DRL) [74] is a circuit-level hiding countermeasure against attacks based on switching power consumption. DRL balances the switching power to create a constant value through signal differential encoding, $\mathbf{S^d} = (S, S\backslash)$, where one wire ($S$) carries the direct signal and the other ($S\backslash$) carries the complementary signal. DRL operates in two alternating phases: (i) *precharge*, during which both the direct and complementary signals are set to a common '0' value, and (ii) *evaluation*, during which either the direct signal or the complementary signal will perform a transition to '1' becoming valid, enforcing 100% activity. Many countermeasures derived from DRL have been proposed for cryptosystems mapped onto FPGAs. There is, however, a question as to whether mapping DRL circuits onto commercial FPGAs can provide sufficient robustness since such devices are natively built with single-rail logic. It has been shown that due to FPGA routing constraints it is difficult to balance the complementary loads of DRL, which is essential for the effectiveness of this type of protection logic [124]. In addition, it should be mentioned that DRL increases the robustness against switching power attacks but ignores static power consumption and the difference in the propagation delays of the dual-rail signals. As a result, attacks based on static power have been successfully mounted on DRL circuit [4, 5, 68]. Glitches and early evaluation can also leak valuable side-channel information due to different propagation delays [39, 43, 48, 61, 69, 88, 117, 118]. For example, a successful attack on a DES cryptoprocessor secured with dual-rail logic has been reported [103]. FPGA flexibility is attractive to digital designers, but FPGA's weakness to power attacks limits its use in implementing cryptosystems. Securing FPGA hardware implementations

against only one flavour of leaked side-channel information attacks does not provide sufficient robustness. Full robustness against all power component attacks is needed and constitutes the main objective and contributions of this dissertation.

1.2 Research Objectives

The objective of the described research work is to offer cryptosystem developers a reconfigurable FPGA based hardware platform that is intrinsically secured. This platform aims to maintain FPGA flexibility in implementing digital circuits while eliminating the threats of known power related attacks. Specifically, the goal is to have a secured platform that exhibits quadruple robustness to attacks based on dynamic power, static power, glitches, and early evaluation. The power consumption is to made independent of both processed data and FPGA operations while retaining the commercial FPGAs architecture, i.e. the logic design style remains in line with the commercial architectures. This feature facilitates the mapping of cryptosystems onto reconfigurable hardware. To summarize, the objectives are:

- Removed data and function dependencies of dynamic power consumption.
- Removed data and function dependencies of static power consumption.
- Achieve glitch-free (monotonic) implementations.
- Remove early evaluation effects.
- Retain the reconfiguration logics of commercial FPGAs.

1.3 Contributions

The main contributions of this research are to have a secure-by-design reconfigurable FPGA hardware that preserves the main architectural features of commercial FPGAs. The specific technical contributions are summarized below.

1. **Robustness to switching power attacks** by applying dual-rail logic in the context of FPGA Look-Up Tables (LUTs) and using the SRAMs' complementary outputs to reduce the hardware overhead.

2. **Robustness to static power attacks** of a LUT with 2 pass-transistors per branch (2TB). This eliminates the relationship between the static power consumption and the processed data – a security feature obtained by replicating the LUT multiplexer and cyclic permutation of the SRAM configurations.

3. **Robustness to static power attacks** of a 2TB LUT with eight branches (the Original four branches and the additional four Stub branches) to equalize the Hamming weight for all possible functions. This eliminates the relationship between the static power consumption and the processed data and device functions.

4. **Robustness to static power attacks with reduced area overhead** by an additional circuit that drives the Stub branches so as to ensure the symmetry of the circuitry. We showed that by using this additional circuit, heavy replication is no longer required, significantly saving area.

5. **Robustness to attacks based on glitches and intra-LUT early evaluations** through a precharge strategy and circuit synchronization technique with reduced hardware overhead, that delays the evaluation of the LUT until all its inputs arrive.

6. **Robustness to attacks based on inter-LUT early evaluation** through extending the

synchronization to multiple LUTs so that a complex circuit will not evaluate before all its global inputs turn valid.

7. **Balanced routing dynamic and static power** by securing the switch box to build a complex circuit with a group of LUTs.

Each of these presents a novel contribution to the FPGA security literature.

1.4 Dissertation Outline

The organization of the dissertation is as follows:

Chapter 2 reviews the FPGA structure and standard circuitry to better understand the proposed techniques and their impact on the area overhead of the FPGA.

Chapter 3 reviews the power consumption of digital circuits. Moreover, all attacks based on power consumption are explained. As well, it presents multiple power models that are used by the attackers to facilitate the Power Attacks.

Chapter 4 presents a literature review of the countermeasures against power analysis attacks at these levels of abstraction of protocol, algorithm, architecture, and circuit. Moreover, it presents existing FPGA based countermeasures that are robust against switching power as well as early evaluation attacks.

Chapter 5 proposes a secure-by-design look-up table and switch box to protect against side-channel attacks based on power consumption. A technique based on replication is used to conceal the dynamic and the static power. Robustness to power analysis attacks is achieved at the expense of a required area; that is in line with prior art.

Chapter 6 proposes a novel method to achieve a LUT with constant leakage. The prior replication technique is replaced with symmetrical multiplexer implementation.

Ensuring symmetrical responses not only conceals the static and dynamic power, but it also reduces the area overhead. A DES S-box is implemented with the proposed LUT to provide a proof-of-concept.

Chapter 7 proposes circuit techniques to prevent attacks based on glitches and early evaluations. Monotonic behaviour is guaranteed to eliminate glitches. Moreover, a synchronization circuit technique is proposed to delay the evaluation until all valid input arrived to prevent intra-LUT early evaluation. Furthermore, a methodology to extend the prevention of inter-LUT early evaluation to multiple LUTs is retained.

Chapter 8 concludes the dissertation and presents possible areas of future works.

Chapter 2

Field Programmable Gate Array

A Field-Programmable Gate Array (FPGA) is an integrated circuit that provides the digital designer a configurable platform on which customized computing units can be implemented. Since the introduction of FPGA around 1985, numerous applications (e.g. cryptosystems, bioinformatics, and digital signal processing) have been enhanced by using FPGAs as hardware accelerators. This great success has challenged engineers to improve FPGA's performance, flexibility, and security. As a result, designers can implement digital systems quickly and without the long time frames needed to manufacture an Application-Specific Integrated Circuits (ASIC).

This chapter covers FPGA configuration memory technology and provides a detailed description of FPGA architectures with a special focus on the Configurable Logic Blocks (CLBs) and routing [11, 16, 37, 50, 63, 101]. The last section of the chapter describes the vulnerability of FPGA-mapped cryptosystems to several types of side-channel attacks.

2.1 Overview

FPGAs consist of an array of Configurable Logic Blocks (CLBs), which can be used to implement arbitrary logic functions, and programmable Connection Boxes (CB) and

Switch Boxes (SB), which connect the CLBs to form the desired design. The FPGA's I/O pins allow communication with the outside world. Over the last two decades, different types of FPGAs have been introduced. The most prominent architecture is the so-called island-style FPGA [16].

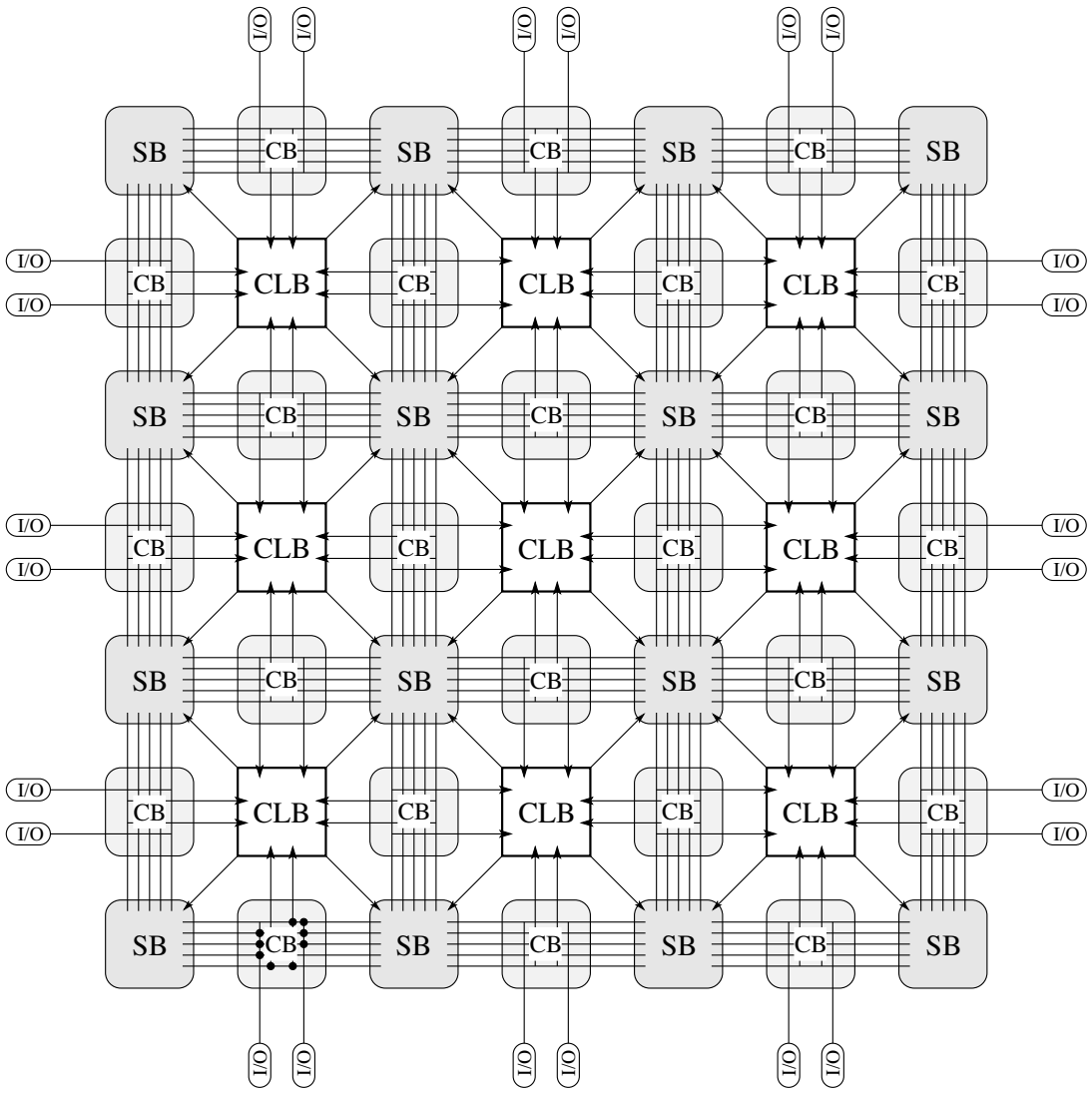


Figure 2.1: Island style FPGA.

The Island-Style (Mesh) FPGA architecture consists of a matrix of configurable logic blocks which are surrounded by a rich routing interconnection network, as shown in Figure 2.1. Routing occupies a significant proportion of the FPGA area, typically in the

range of 80-90%, leaving the configurable logic blocks to span only 10-20% of the chip area [47]. Configurable logic blocks are comprised of a number of Look-Up Tables (LUTs) which can be programmed to implement different arbitrary logic functions. These CLBs are surrounded by Switch Boxes (SB) diagonally and Connection Boxes (CB) on the four sides, as shown in Figure 2.1. The connection boxes route the CLBs' input signals whereas the switch boxes are deployed at track intersections to route CLBs' outputs along the horizontal or the vertical tracks. These interconnection boxes can be programmed to connect the CLBs through mixture wires of different lengths. The programmability feature that attracts digital designers is the ability to generate customized functional units without the need to resort to the manufacture of coustom ASIC designs. Different programming technologies can be used to store the FPGA configuration [20], are reviewed in the next section.

2.2 FPGA Memory Technologies

FPGAs rely on a specific programming technologies to configure their function and control their routing switches. SRAM [75], EPROM [40], EEPROM [29, 104], flash memory [44], and anti-fuses [18, 45] can be used to store the FPGA configuration, as reviewed below.

2.2.1 Static RAM Technology

Static RAM (SRAM) FPGAs store their configuration data into memory cells distributed throughout the device. An SRAM cell consists of two cross-coupled inverters (providing complementary outputs) and two access transistors, as shown in Figure 2.2.

As depicted in Figure 2.6 the SRAM cells are connected to the sources of the LUT pass-transistors. Further, they are used to control the CLBs internal routing (which is also referred to as local routing), as shown in Figures 2.9, and global routing, which connects the CLBs through connection and switch boxes. Series-7 from Xilinx [131] and Stratix-5 from

Altera [9] are examples of traditional SRAM-based FPGAs. Since the SRAM is volatile, the FPGA must reload its configuration each time the chip is powered ON.

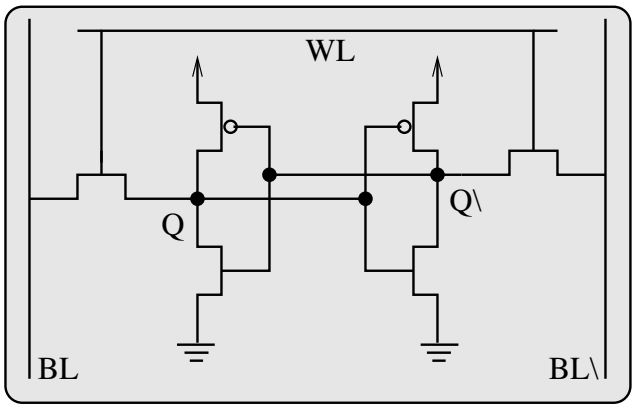


Figure 2.2: Static memory cell.

The popularity of FPGAs based on SRAM technology relies on its re-programmability and use of standard CMOS process technology. However, SRAMs occupy a significant portion of the FPGA area since they need at least six MOS transistors per bit of information, which has led to more efficient approaches.

2.2.2 Flash-based/EEPROM Technology

Flash and EEPROM memories are alternative FPGA memory technologies. The memory cell of a flash memory is a field effect transistor (FET) with two gates, namely a control gate and a floating gate. The floating gate is located under the control gate, which is insulated from the drain, source, and control gate electrodes with an oxide layer, as depicted in Figure 2.3. The floating gate forms a capacitor which is free of charge in the normal state (unprogrammed). The transistor is programmed by causing a large current to flow between the source and the drain. As a result, charge is trapped within the oxide layer under the floating gate. Flash-based FPGAs can be erased by freeing the trapped charge in the floating gate.

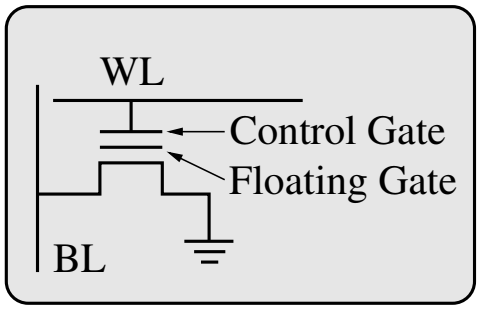


Figure 2.3: Flash memory cell.

Flash, EPROM, and EEPROM programmable logic devices have emerged in different commercial devices, such as Complex Programmable Logic Devices (CPLDs) [34], the Microsemi SmartFusion devices [2], and Lattice's XP2 programmable devices [105].

Flash-based programming technology is a non-volatile device; eliminating the need for loading the configurations from off-chip memory during the power-up. The availability of a reprogrammable flash inside the FPGA protects the configuration from being copied during a serial configuration process of SRAM-based FPGAs. It gives manufacturers the ability to build applications that retain information through power cycles, which can be useful in cryptographic applications such as tamper logging and key revocation [127]. It should be mentioned that the Flash memory cell is single-ended. This can have implications in the hardware overhead needed for implementing differential logic, are discussed in the next chapters.

2.2.3 Anti-fuse Technology

Anti-fuse is an alternative to flash programming technology that uses a one-time programmable structure to form a non-volatile link between two wires [20, 45]. High voltage is placed across the anti-fuse terminals to program the fuse. Hence, when current flows through the device, it generates enough heat to melt the dielectric layer and to form a permanent conductive link between the polycrystalline silicon (Poly-Si) and the

n+ diffusion layers as shown in Figure 2.4. The FPGA programming is performed at the time of manufacturing. Once the chip has been programmed at the manufacturer, the anti-fuse-based FPGA can not be changed or reprogrammed.

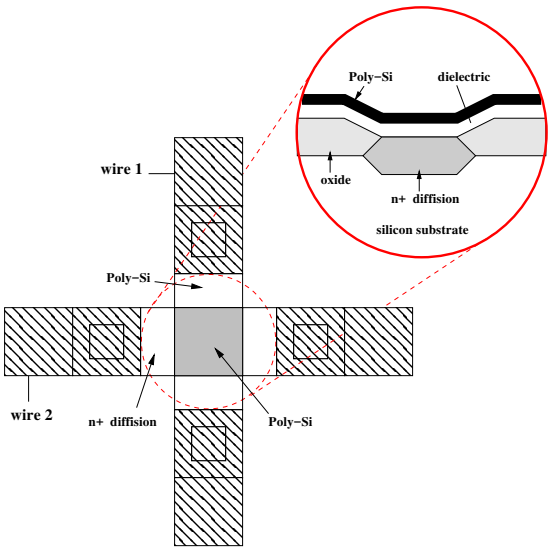


Figure 2.4: Actel antifuse programming technology.

Anti-fuse FPGAs have the advantage of being nonvolatile and are considered to be the most robust FPGAs in terms of information retention. Confidentiality and the authenticity of configuration data are preserved as there is no need for external configuration storage [127]. However, this feature comes at the price in term of flexibility. Moreover, anti-fuse device is a single-ended logic, as is flash memory which may constitute a limitation when differential logic is to be implemented.

In practice, the most commonly used technology is FPGA based on SRAM, since it can be fabricated in CMOS and it is also reprogrammable. In addition, we have chosen the SRAM-based programming technology in our work because the SRAM cell intrinsically provides the complementary output which is an important feature for the proposed techniques. It is worth mentioning that our contributions can be extrapolated to the other types of programming memories, as discussed in the next chapters.

2.3 FPGA Architecture and Circuit Implementation

This section reviews the FPGA architectural and circuit implementation elements which are relevant in securing the FPGA against attacks based on power consumption.

2.3.1 Configurable Logic Block (CLB)

A configurable logic block (CLB), a fundamental FPGA programmable unit, is comprised of a combinational part and a sequential part (the flip-flop). Over the last two decades, the combinational part has ranged between the extremes of fine and coarse-grain logic blocks. A very fine-grain logic block is built with a simple gate (NOR or NAND). This approach results in an FPGA that suffers from area-inefficiency, low performance, and high power consumption because it requires a very complex interconnection network [47]. At the other extreme, coarse-grain logic is capable of implementing complex operations. Between these two extremes, different architectures such as those based on blocks of logic gates made of transistors pairs and RAM [75], NAND gates [106], interconnected multiplexers [35], look-up tables (LUTs) [75], and PAL-style wide-input gates [130] have been proposed to provide various trade-offs. Notably, the LUT-based CLBs, like the ones used in Xilinx' and Altera's FPGAs [16], use SRAM cells to store configurations.

A CLB comprises a cluster of Basic Logic Elements (BLEs) and Intra-CLB interconnections (local routing) that allow communication between these BLEs, as shown in Figure 2.5. In addition, it contains a wide-function multiplexers to extended the LUT functionality. Each BLE consists of one Look-Up Table (LUT) and one D-type Flip-Flop (DFF). In principle, any cluster input can be connected to any BLE input. In special cases, however, commercial FPGAs may reduce this flexibility for saving area [64, 67].

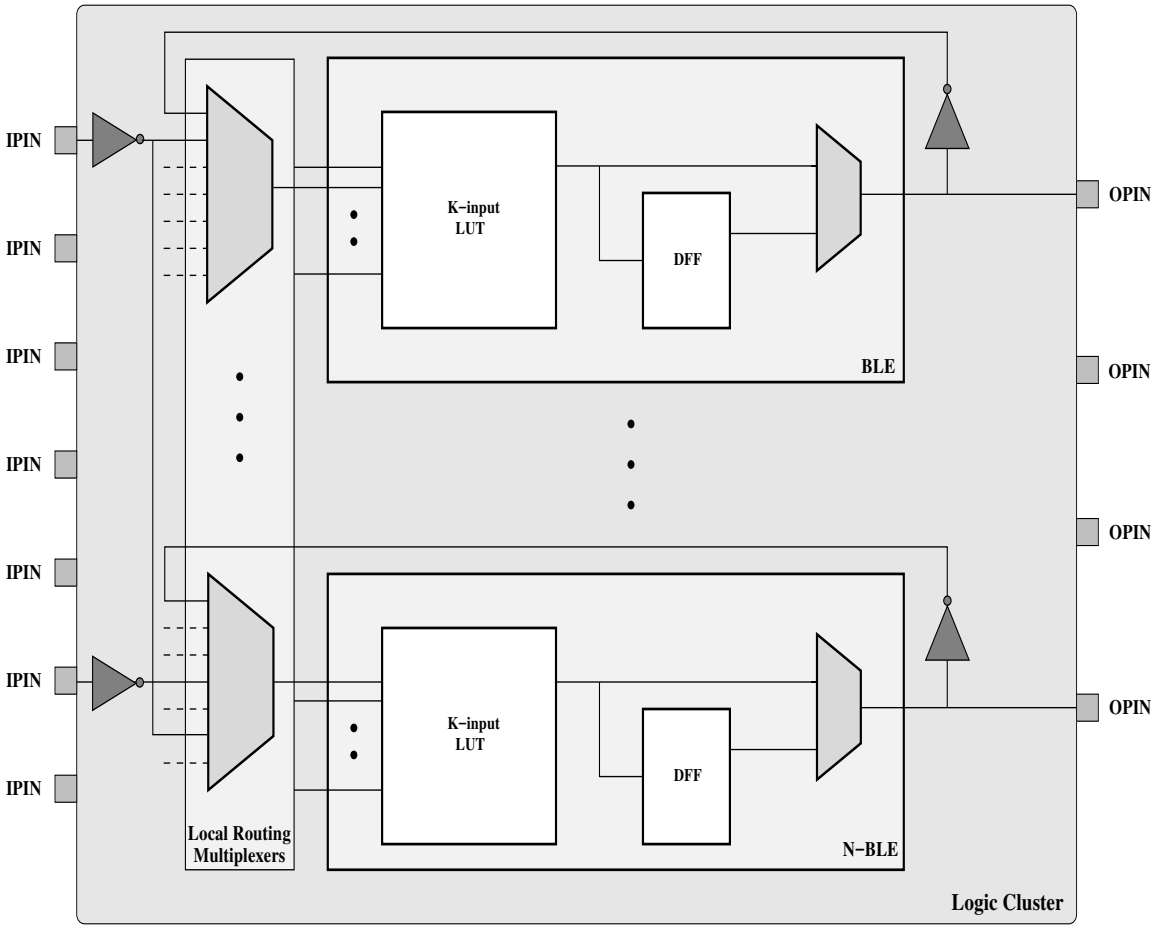


Figure 2.5: Configurable logic block.

Look-Up Table

Look-Up Tables (LUTs) are the core of the configurable logic block in FPGAs, as they can be configured to implement any combinational logic functions. An I-input LUT consists of input buffers, nMOS-based multiplexers in a tree topology, level-restoring buffers inserted between every two stages of the pass-transistors MUX as in [50], and 2^I SRAM cells to hold the LUT's configuration allowing implementation of 2^{2^I} different logic functions. A more detailed review of the circuitry of each component follows. Currently, LUTs in commercial FPGAs are implemented with 6 inputs [108]. For simplicity, we present a 4-input LUT structure (Figure 2.6).

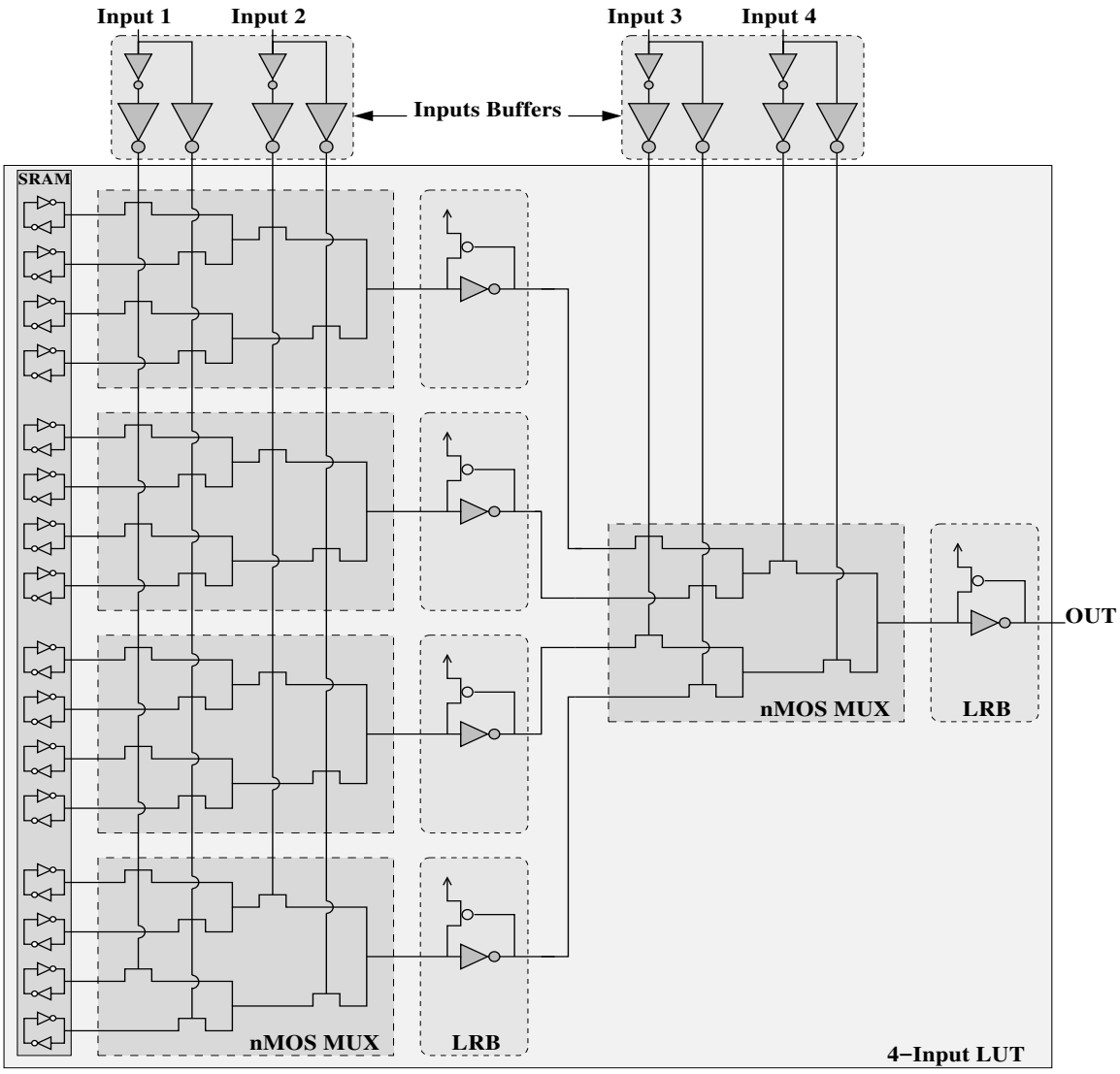


Figure 2.6: 4-input look-up table.

LUTs and Switches Buffers

When large loads are being driven, 2 or 3 concatenated buffers are recommended to optomize the propagation delay [51]. In context of FPGAs, two-stage buffers are commonly used [50]. The two-stage buffer consists of a minimum size inverter followed by an optimally sized second stage, as exemplified in Figure 2.7. The inverter pMOS/nMOS ratio is set to 2.5 to achieve equal rising and falling times.

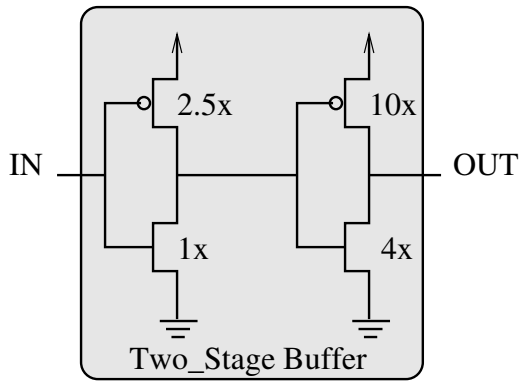


Figure 2.7: Example of two-stage buffer: the first stage is minimally sized whereas the second stage is optimally sized.

A two-stage buffer drives the BLEs' inputs and strengthens the BLEs' outputs to drive the large load of the interconnection wire. It is used in multiple places in CLBs, as shown in Figure 2.6, and switch boxes, as in Figure 2.9. These two-stage buffers are used in CLBs to generate the complementary inputs. In the interconnection network, they are interleaved with bus wire segments in order to break the quadratic dependence of the propagations delay on the wire length. It should be noticed that the FPGA interconnection network is difficult to optimize (and so are the buffers), as it is fabricated before any applications are designed and mapped onto FPGA. In fact, this increases the high power nature of FPGAs.

LUTs and Switches nMOS Multiplexer

In FPGAs, the multiplexers used in LUTs and routing boxes are implemented in pass-transistor logic. In LUTs, the inputs drive the transistors' gates, whereas, the SRAM outputs drive the transistors' sources, as depicted in Figure 2.6. In contrast, the switch box's inputs signals drive the transistors' sources and the SRAM outputs drive the transistors' gates, as shown in Figure 2.9. Full-transmission gates instead of pass-transistors can be used [26, 92], but at the expense of large Silicon area. Transmission gates are not currently used in commercial devices.

Level-Restoring Buffer

In the 4-input LUT, a four-level multiplexer route the SRAMs' configurations to the output of the BLE, as shown in Figure 2.6. This transistor chain quadratically increases the delay of the LUT. Moreover, there is a threshold voltage drop across the nMOS transistor when passing high voltage. The resulting weak '1' raises the leakage current in the downstream buffer. To break the quadratic dependence of the delay and restore the strong '1', a level-restoring buffer is inserted at every two stages. A level-restorer buffer consists of a skewed inverter [64] and a pMOS transistor, called a Keeper, to provide active feedback to the inverter, as depicted in Figure 2.8 [51].

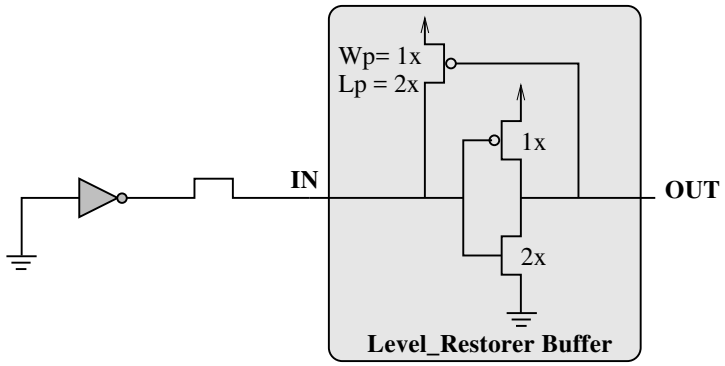


Figure 2.8: Level-restoring buffer.

The inverter is skewed to modify its switching point to half of the reduced voltage swing: $V_{DD} - V_{th}/2$, where V_{th} is the transistor's threshold voltage [50, 132]. Introducing the level-restoring buffer solves the weak '1' related-issues at the expense of the circuit complexity [84]. Hence, a strong line driver is required to overdrive the keeper. An alternative to a strong line driver is a weak keeper. The standard way is to make the keeper transistor long; this though may increase the load of the level-restoring buffer. To circumvent this issue, the keeper can be replaced with a series connection of a keeper and a bleeder to weaken the loading effect of the keeper. The gate of the bleeder transistor is connected to the ground while the keeper gate is controlled by the level-restorer's output.

It should be emphasized that the circuit is ratioed. During the 1-0 transition, the keeper transistor will turn OFF only after the signal has propagated through the skewed inverter resulting in a large short-circuit power consumption. Further discussion on the security issues of ratioed circuits is provided in Chapter 7.

2.3.2 Routing Architecture

The FPGA routing is comprised of programmable switches and wires. Programmable switches connect the I/O within and between CLBs. Wires are connected through configurable routing boxes.

Programmable switches

In FPGA, there are two types of routing resources: local and global. Local routing handles the communication within the same CLB. Global routing includes the connection and switch boxes as well as the wire segments. Both local and the global interconnection boxes are built with 1- or 2-level multiplexers followed by level-restoring buffers. Both multiplexer's levels have equal fan-in to minimize the propagation delay. These multiplexers route the signals to their destinations.

Figure 2.9 depicts the circuit of a local routing multiplexer. Each multiplexer has $I+N$ inputs, where I represents the number of signals coming from the global routing, while N is the number of BLEs outputs. As is apparent, the first level includes four $5:1$ MUXes while the second stage includes one $4:1$ MUX. To minimize the propagation delay, the fan-in for both levels should be approximately equal. A similar structure is used for the switch and connection boxes even with different topologies.

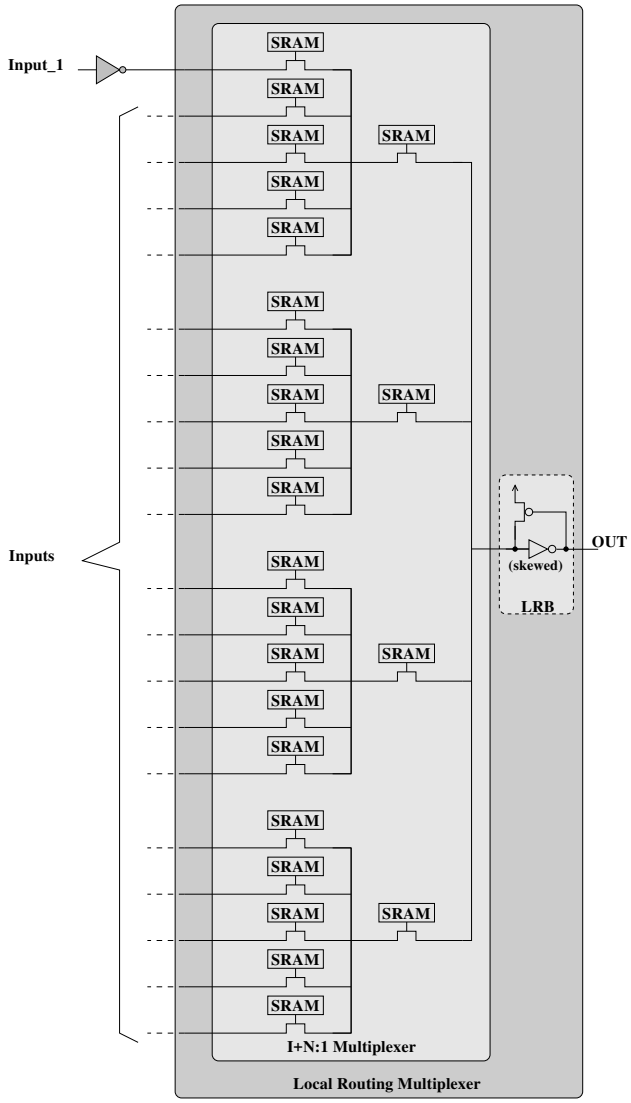


Figure 2.9: Local routing multiplexer.

Wires

As is apparent in Figure 2.1, each configurable logic block is surrounded by Switch Boxes (SBs) diagonally and Connection Boxes (CBs) on all four sides. Multiple wires can be connected through these boxes. Originally, the FPGA wires that connect the logic cluster to its neighbours were of fixed length. Then, El-Gamal [6] introduced the idea of a mixture of segmented lengths to improve the overall speed of FPGA-mapped circuitry. The segmented lengths vary from short (stretching one or two logic cells), to medium (stretching four to

eight logic cells), and long (stretching half to full length of the die) [16], as depicted in Figure 2.10. In FPGAs, routing typically occupies 80-90% of a chip's area. Due to their very complex interconnection networks, FPGAs consume significantly more power than ASICs.

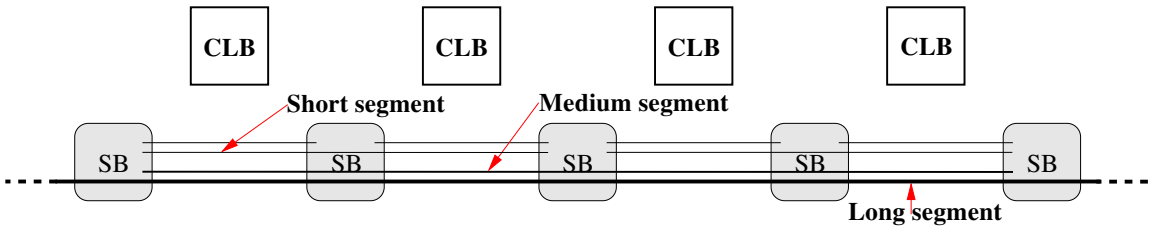


Figure 2.10: Different wire length.

2.4 Why FPGA?

Field Programmable Gate Arrays (FPGAs) play a significant role in the electronics industry. Emerging application requirements and the FPGA's features have increases FPGAs prominence. The FPGA re-programmability provides the designers with the ability to improve their implementation at any design stage in contrast with the application-specific integrated circuits (ASICs), where their expenses for faprication are high. Moreover, the non-recurrent engineering (NRE) cost of an ASIC far exceeds that of an FPGA. In addition, the higher flexibility and short time-to-market will continue to ensure FPGA's prominence. Since modern FPGAs can in general meet many of the performance requirements of ASICs, FPGAs are increasingly being used in their place [65]. There is a great potential to reduce the cost and enhance the security level of the design by using FPGAs.

2.5 FPGA Vulnerabilities

FPGAs are considered to be an attractive platform for implementing cryptographic applications due to their reconfigurability. However, cryptosystem designers should be aware of FPGA vulnerabilities that translate into degraded security levels. In this section, we explore the vulnerability of FPGAs to several types of side-channel attacks.

2.5.1 Side-Channel Attacks

Side-Channel Attacks are generally non-invasive and are based on the additional information that leaks from the implementation imperfections of cryptosystems [58]. The cryptosystem devices can leak valuable information through their physical characteristics of: energy consumption, execution time, and electromagnetic fields. A short review is presented next.

Power analysis attacks

These attacks are based on analyzing the cryptosystem power consumption during encryption or decryption operations [58]. As we mentioned, FPGAs generally consume more power than custom circuits. Furthermore, the Look-Up Tables (LUTs) in FPGAs are built with ratioed circuits, which are known to exhibit large short-circuit power consumption [129]. These features make the FPGAs highly vulnerable to power attacks. In addition, the routing limitation and non-symmetrical structure limit the designers' ability to eliminate the threat of leaked information at both the algorithm and architecture levels, as discussed in Chapter 4.

FPGAs are implemented in CMOS technology, in which the power consumption consists of switching, static, and short-circuit components. Each component may show direct dependency on the logic function and/or the process data. Attackers collect power

consumption information to exploit such dependency, and retrieve valuable information such as the secret key. For example, to acquire the power consumption signal, the attackers can connect a small resistor in series with the power supply pin and record the voltage drop accross the resistor [136]. These types of attacks are of concern because they are quick to mount and inexpensive to perform. According to [58], a successful power analysis attack on smartcards may take between a few second and a few hours. The power attacks are the strong attacks reported in the literature; therefore, they are the main focus of our research. More details about these kinds of attacks are provided in the next chapter.

Attack model

Attack setup consists of a chip under attack, a current sensor (e.g., a tiny resistor in series with the power or GND pin), a high-resolution oscilloscope to acquire the power signal, and a PC to do the statical analysis as shown in Figure 2.11. It is important to mention that I/O signals and the current drawn from the power supply are all accessible to the attacker.

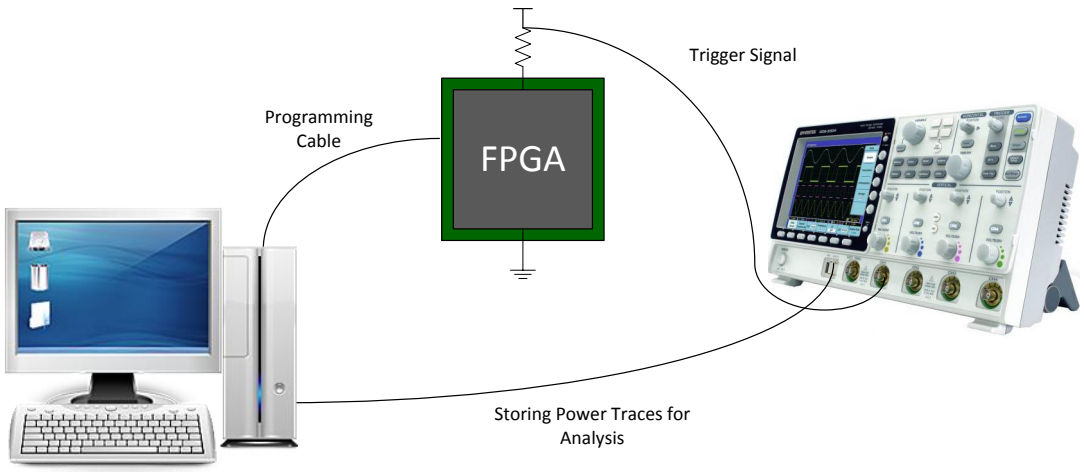


Figure 2.11: Power analysis attack setup.

To mount the attack, the attacker needs to record multiple current traces of different predicted processed data. Then, the collected current traces are classified based on the value of the MSB bit either 0 or 1. The attacker calculates the mean of each classified traces. After that, the adversary measures the difference between the means in case of the "differential power analysis." By comparing the actual measurement of the chip with the predicted, the attacker will be able to determine the key bit value.

Timing analysis attacks

Attackers have been able to extract the signature of the cryptosystems based on their execution time [59]. For example, based on the secret key, the scalar multiplication on Elliptic Curve Cryptosystems performs different point operations (which have different latencies) [8]. As a result, it is possible to determine the key bit by measuring execution time.

Electromagnetic emanation analysis attacks

In this case, the attackers measure the electromagnetic radiation emitted by the cryptosystem during its operation. Then, they analyze the collected information to extract valuable information about the cryptosystem. Similar to power analysis attacks, there can be simple and differential electromagnetic emission analysis. In the simple analysis, the attacker can retrieve valuable information by analyzing only a few recorded traces. The differential electromagnetic analysis is more sophisticated or it uses different statistical approaches to study those traces in more detail.

Side-channel attacks are of concern because the do not need heavy and expensive equipment to mount. Many successful side-channel attacks have been reported in the literature [8, 32, 55, 59, 90, 109, 110, 113, 120].

2.5.2 Fault Injection Attacks

Fault injection attacks are non-invasive attacks that cause the circuits of a cryptosystem to malfunction in predictable way, such that the attacker gains valuable information about the system. Methods to inject faults in the circuit include altering the supply voltage, temperature, and the external clock frequency. Moreover, the fault could be induced by exposing the device to radiation. Glitch and ionizing radiation analysis are the most common approaches in this type of attack. Glitch analysis attacks have been shown to be successful on cryptosystems implemented on microcontrollers [10]. Also, [36, 53, 66] have demonstrated that radiation-induced faults cause single-event upsets in the CMOS circuits. Since the FPGAs store their configuration information into SRAM cells, such attacks may flip the memory bits as presented in [3].

2.5.3 Physical Attacks

Physical attacks are invasive attacks that target the physical layer of an FPGA. They aim to obtain side-channel information by probing points inside the circuit. In these attacks, the attackers target the parts of the FPGA that are not accessible through the normal I/O pins. With the use of an optical Scanning Electron Microscope (SEM) or Focused Ion Beam (FIB), the attackers aim to retrieve the information stored in the memory, design, or the keys of a cryptosystem. Such attacks are hard to implement due to their complexity and use of high cost equipment, that would not normally be available to individuals and small organization.

2.6 Conclusion

In this chapter, we have reviewed the FPGA architecture and its configurable logic blocks and routing networks. We showed different memory technologies used in programming

the devices. In addition, we detailed the circuitry of every component and outlined their characteristics. Finally, we reviewed the FPGA features that attract designers' attention and emphasized the security aspects that must be addressed. Side-channel attacks are one of the main FPGA threats – especially since the FPGAs are power hungry.

Power analysis attacks are the raison d'être of this work since they are efficient and easy to mount by collecting and analyzing multiple power consumption traces. Hence, finding a secret key is only a question of time and the statistics obtained. Designers must be able to eliminate the threat of such attacks in order to maintain and increase the presence of FPGA technology in the cryptosystem market. Therefore, our work aims to provide reconfigurable hardware that exhibits robustness to the attacks based on dynamic power, static power, glitches, and early evaluation, while preserving the architecture of commercial FPGAs.

Chapter 3

Power Consumption and Analysis

This chapter outlines the type of power information leaked by FPGA devices. As mentioned previously, our research focuses on SRAM-based reconfigurable devices since they are the most popular platforms presently in use. In SRAM-FPGAs, the memory cells, the logic blocks, and the connection blocks are fabricated in CMOS technology [57]. This chapter discusses the power consumption in CMOS circuits, security issues that emerge from power consumption based attacks, and the power models used by attackers.

3.1 Power Consumption of CMOS Circuits

CMOS is a standard technology used in the fabrication of integrated circuits. In this section, general insights regarding the components of power dissipations in CMOS are discussed. CMOS technology has two distinct components of power dissipations: **Dynamic** and **Static** [28, 129], where the total power is given by:

$$P_{\text{Total}} = P_{\text{Dynamic}} + P_{\text{Static}} \tag{3.1}$$

A CMOS circuit only consumes dynamic power during switching. Whereas, static power is dissipated even in the absence of switching. Table 3.1 outlines the possible

switching combinations, with the following sections discussing each power component in more detail.

Table 3.1: Power consumption type based on the signal transition.

Transition	Type of Power Consumption
$0 \rightarrow 0$	Static
$0 \rightarrow 1$	Static + Dynamic
$1 \rightarrow 0$	Static + Dynamic
$1 \rightarrow 1$	Static

3.1.1 Dynamic Power

Dynamic power, the highest contributor to the total CMOS consumed power, can be further decomposed into switching power and short-circuit power.

$$P_{\text{Dynamic}} = P_{\text{Switching}} + P_{\text{Short-circuit}} \tag{3.2}$$

$P_{\text{Switching}}$ is attributed to the charging and discharging of every node's capacitance in a digital circuit and can be modeled,

$$P_{\text{Switching}} = \sum_{\text{all nodes}} C_{\text{y}}.V_{\text{DD}}^2.\alpha_{\text{y}}.f_{\text{clock}} \tag{3.3}$$

As such, $P_{\text{Switching}}$ is proportional to the switching activity of each node, α_{y}, the load capacitance at every node, C_{y}, the square of the supply voltage, V_{DD}, and the clock frequency, f_{clock}. It is worth mentioning that in some situations (e.g., pass-transistor logic) ${V_{\text{DD}}}^2$ needs to be replaced by $V_{\text{DD}}.V_{\text{swing}}$, where V_{swing} is not a full swing to V_{DD} at every node.

Activity factor α is the switching probability of a logic circuit. Therefore, if the circuit is in sleep mode, its activity factor is zero as is its dynamic power. In FPGAs, the activity factor depends on the SRAMs configuration. The activity factor of a logic circuit can be

determined as the probability of being in '0' state multiplied by the probability of being in '1' state:

$$\alpha = P_0 P_1 \tag{3.4}$$

As shown in Figure 3.1, the probability of having logic 1 at the output of the first stage AND gate is 1/4 while the probability of having logic 0 is 3/4. As a result, the activity factor for the AND gate is 3/16. The global output of this circuit has a probability of 1/16 of having logic 1. Therefore, the activity factor of this circuit output is $\alpha = 15/256$. It is apparent that the activity factor and switching are in a direct relationship. This is a kind of information that can be used by attackers to reveal the secret key.

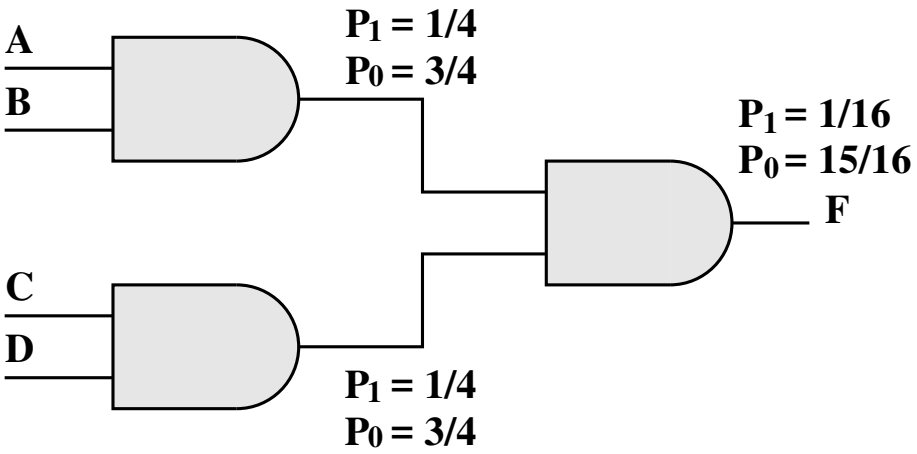


Figure 3.1: Circuit activity factor.

In digital circuits with hazard, spurious transitions called **glitches** increase the switching activity [51]. It has been reported that hazard can increase the dynamic switching power in CMOS circuits by 20% to 70% [71]. Glitches pose a serious threat to FPGA-mapped cryptosystems since they strongly depend on the processed data and FPGA configuration [69]. In addition, a difference in the input arrival times may cause the output of digital logic to switch to its final value even before all the inputs are presented. This type of effect is

known as **early evaluation** [61, 117]. Like the glitches, early evaluation may leak valuable information about cryptosystem's activity and thus the secret key.

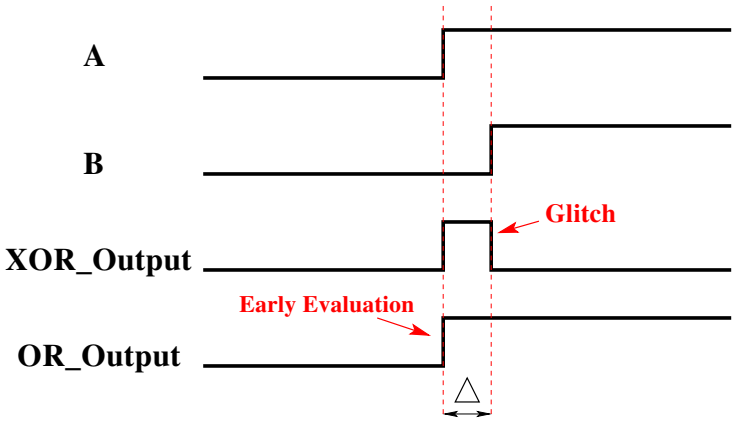


Figure 3.2: Time diagram glitches and early evaluation on 2-Input LUT.

To show the difference between the glitch and early evaluation phenomena, Figure 3.2 presents the time diagram of 2-input XOR and OR gates, where the same values are applied to both inputs, but with a relative delay, Δ. In the case of the XOR gate, the output experiences a glitch before it settles to its final value. In the case of the OR gate, the output switches to its final value even before all the inputs have arrived. In both cases, an indication about the processed data is available and can be used for attacking the cryptosystem.

The **load capacitance** $\sum C_y$ is the sum of the intrinsic node capacitances and the wire capacitances. FPGA interconnections are usually longer and more difficult to control than in ASICs [62]. This is a major limitation in implementing dual-rail logic (which aims to equalize the switching activity) as will be described in the following chapters. Switch boxes are interleaved with interconnection wires to provide the reconfigurability. Therefore, it is also important to investigate how switch boxes conceal their switching and short-circuit power.

Short-circuit power, which occurs when there is a direct conduction path between the supply rail and ground, can manifest in two situations. First, it occurs when both pull-up and pull-down networks in standard CMOS gates are partially ON for a short time during

switching from one state to the other. It also occurs in ratioed logic circuits. Due to the physical characteristics of CMOS, both short-circuit and static power consumption are always present [51].

To illustrate the dynamic power consumption in the context of FPGAs, consider the 2-input look-up table presented in Figure 3.3. As mentioned previously, the look-up table is built with three two-input multiplexers in a tree topology of pass transistor followed by the level-restoring buffer. In this example, we analyze the switching and the short-circuit power of the 2-input LUT with two different SRAM configurations shown in Figures 3.3 and 3.4.

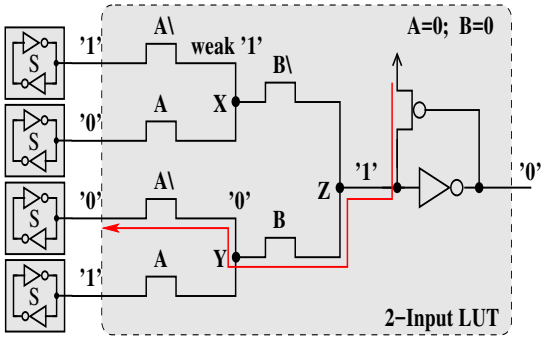


Figure 3.3: A 2-input XOR gate switching activity and short-circuit current.

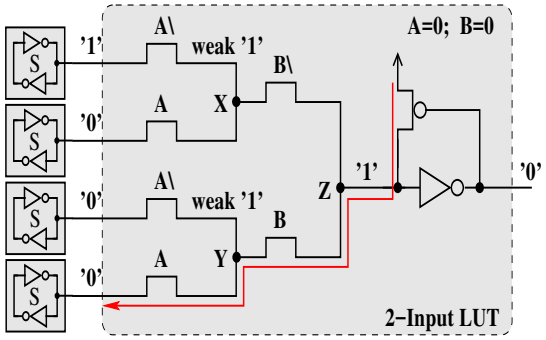


Figure 3.4: A 2-input OR gate switching activity and short-circuit current.

Figures 3.3 and 3.4 present two 2-input LUTs that implement an XOR gate and an OR gate, respectively. We illustrate the LUT switching activity under the processed data A='0' and B='0'. If every node is at '0' before evaluation, the XOR gate undergoes two toggling transitions at nodes X and Z. As a result, all nodes (X, Y, Z) make a transition to one. It is important to notice that the middle nodes X and Y do not experience full swing toggling ($0-(V_{DD}-V_{th})$) while the output node (Z) has full swing because of the level-restoring buffer.

In terms of short-circuit power, LUTs in FPGAs are built with ratioed circuits, which are known to exhibit a significant short-circuit power consumption. If the inputs (A, B) transition from (0, 0) to (0, 1) in the XOR gate, a short-circuit current is established between

V_{DD} and *GND*. This short-circuit current will flow until the keeper turns *OFF*. In the case of the OR gate, a short-circuit current (shown with a red arrow) flows between V_{DD} and *GND* when the input pair (A, B) switches from (0, 0) to (1, 1) or (any other pair value).

Table 3.2: Different toggles (↑↓) and short-circuit (S/C) occurrences under all possible input transitions for different gates.

Input		Function											
		AND		NAND		OR		NOR		XOR		XNOR	
A	B	↑↓	S/C	↑↓	S/C	↑↓	S/C	↑↓	S/C	↑↓	S/C	↑↓	S/C
$0 \rightarrow 0$	$0 \rightarrow 0$	0	0	0	0	0	0	0	0	0	0	0	0
$0 \rightarrow 0$	$0 \rightarrow 1$	0	0	0	0	1	1	1	0	1	1	1	0
$0 \rightarrow 0$	$1 \rightarrow 0$	0	0	0	0	1	0	1	1	1	0	1	1
$0 \rightarrow 0$	$1 \rightarrow 1$	0	0	0	0	0	0	0	0	0	0	0	0
$0 \rightarrow 1$	$0 \rightarrow 0$	1	0	1	0	2	1	2	0	3	1	3	0
$0 \rightarrow 1$	$0 \rightarrow 1$	2	1	2	0	2	1	2	0	2	0	2	0
$0 \rightarrow 1$	$1 \rightarrow 0$	1	0	1	0	1	0	1	0	2	0	2	0
$0 \rightarrow 1$	$1 \rightarrow 1$	2	1	2	1	1	0	1	0	3	0	3	1
$1 \rightarrow 0$	$0 \rightarrow 0$	1	0	1	0	2	0	2	1	3	0	3	1
$1 \rightarrow 0$	$0 \rightarrow 1$	1	0	1	0	1	0	1	0	2	0	2	0
$1 \rightarrow 0$	$1 \rightarrow 0$	2	0	2	1	2	0	2	1	2	0	2	0
$1 \rightarrow 0$	$1 \rightarrow 1$	2	0	2	1	1	0	1	0	3	1	3	0
$1 \rightarrow 1$	$0 \rightarrow 0$	0	0	0	0	0	0	0	0	0	0	0	0
$1 \rightarrow 1$	$0 \rightarrow 1$	1	1	1	0	0	0	0	0	1	0	1	1
$1 \rightarrow 1$	$1 \rightarrow 0$	1	0	1	1	0	0	0	0	1	1	1	0
$1 \rightarrow 1$	$1 \rightarrow 1$	0	0	0	0	0	0	0	0	0	0	0	0

(↑↓ represent the number of transitions per gate; and S/C represent the occurrence of short current power per gate)

Table 3.2 shows all possible input transitions of a 2-Input LUT and their corresponding switching activities and short-circuit currents. It is clear that each input has four possible transitions ($0 \rightarrow 0$, $0 \rightarrow 1$, $1 \rightarrow 0$, and $1 \rightarrow 1$). Hence, the input pair can have 16 possible transitions. Each input transition will generate a different number of toggles (↑↓) at the X, Y, and Z nodes. The number of toggles depends on the current and the previous state of the inputs, as well as the SRAM configuration. It should be mentioned that the short-circuit

current only occurs whenever the Z node undergoes a transition $1 \rightarrow 0$. It is apparent that by measuring the power consumption, it is possible to obtain information about the transition, that occur during the operation of a logic circuit.

3.1.2 Static Power

With the scaling down of the CMOS technology below 90nm, static power becomes a significant fraction of the overall power dissipation [1, 23, 87]. An understanding of the relationship between static power and the processed data is necessary for circuit designers to comprehand the sources and the impact of static power. Since the static power consumption exists even when there are no transitions, it is possible to measure it by simply stopping the clock and performing a DC measurement (which requires simple hardware in general). Not only that, but, the leakage current doubles with every $8-10$ °C increase in temperature which further increase the vulnerability of the cryptosystems to static power attacks [49]. There are three sources of static power as listed below and shown in Eq.(3.5):

- sub-threshold leakage between the source and the drain of a transistor,
- gate leakage from the gate to the body of a transistor, and
- junction leakage from the source and the drain to the body of a transistor.

Total static power is therefore given by,

$$P_{\text{Static}} = (I_{\text{subthreshold}} + I_{\text{gate}} + I_{\text{junction}})V_{\text{DD}} = I_{\text{leak}}V_{\text{DD}} \tag{3.5}$$

In this work, the focus is on sub-threshold leakage since it is at least two orders of magnitude larger than other types of leakage in current CMOS technology. It is therefore important to investigate under what conditions the circuit has sub-threshold leakage.

Sub-threshold leakage

Scaling down the CMOS technology requires reducing the threshold voltage. As a consequence, sub-threshold leakage increases because a transistor which is meant to be *OFF*, in reality is not entirely *OFF*. It is well-known that the subthreshold leakage depends exponentially on the drain-source voltage [51, 129] and as such, it can be a major source of side-channel information.

Gate leakage

Scaling down the technology entails shortening the length of a transistor channel. To maintain a good transistor aspect ratio, the thickness of the transistor channel needs to be comparably reduced [129]. As a result of reducing the gate oxide thickness, there is an increase in the gate leakage current through the gates of ON transistors. In modern technology nodes gate leakage is orders of magnitude smaller than the subthreshold leakage, hence it can be ignored in our analysis.

Junction leakage

Junction leakage appears from the source or drain to the substrate through the reverse-biased diodes isolation. Similar to the gate leakage, the junction leakage is much smaller than the subthreshold leakage. It will be neglected in our analysis.

Figure 3.5 presents all nMOS transistor biasing cases in a pass-transistor logic network as well as all possible leakage currents. Panel (a) shows that there is no leakage current if the transistor is ON. Cases in panel (b) present an *OFF* transistor, where the first top two cases exhibit no or extremely small leakage. The leakage in the third transistor is ignored because it is orders of magnitude smaller than the leakage current in the bottom cases. The bottom cases exhibit the highest leakage. The bottom-right case, noted as 'T', has a voltage

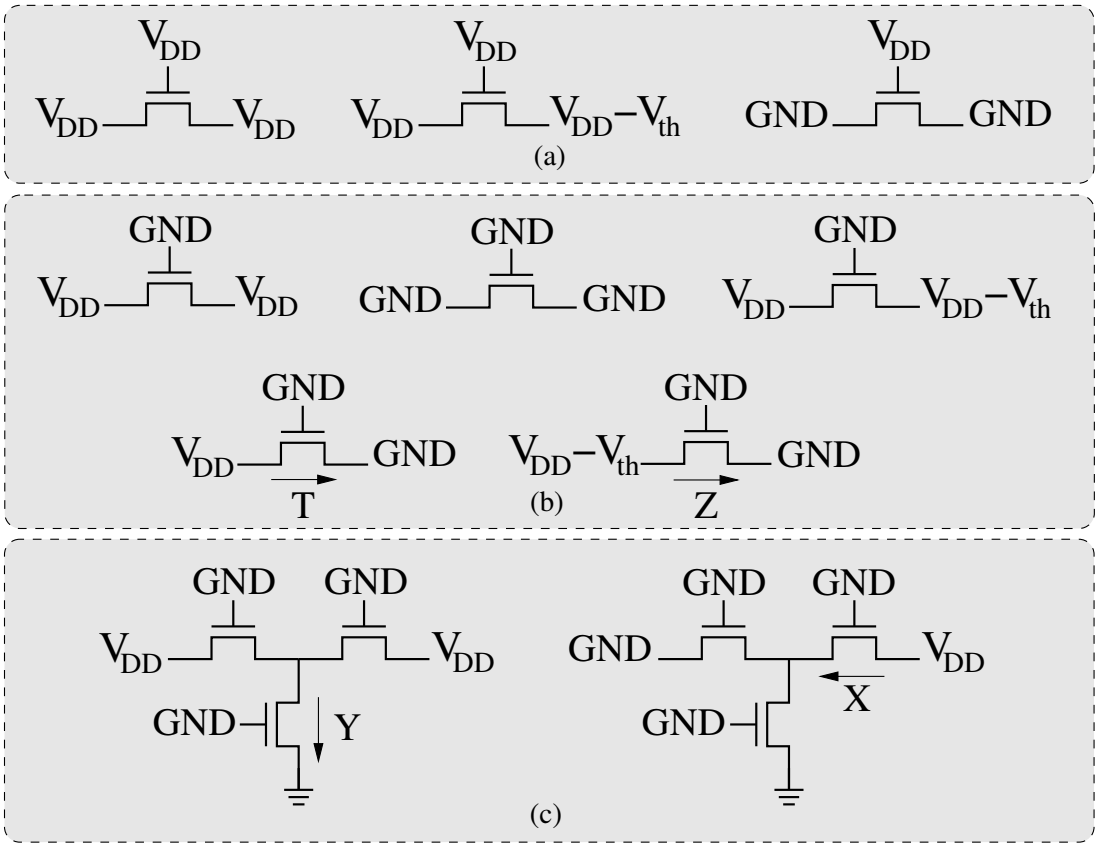


Figure 3.5: nMOS leakage behaviour.

drop of V_{dd} between the source and the drain. The bottom left case is indicated as 'Z' type leakage, where the voltage drop across the *OFF* transistor is equal to $V_{dd} - V_{th}$. Figure 3.5-c illustrates special cases of the proposed circuit techniques. The left case is noted as a 'Y' type leakage while the right one is recognized as 'X' type leakage.

In 130 nm technology, the leakage current of each nMOS transistor carries side-channel information. Leakage of type 'T' has the highest value of 283.89 pA because of the high voltage drop between the source and drain. The type 'Z' leakage (212.34 pA) is approximately 25 % lower leakage than type 'T' leakage. In the special leakage cases, type 'X' and 'Y' exhibit lower leakage currents of 72.76 pA and 45.33 pA respectively. Similarly, pMOS shows differences based on the leakage type; however, pMOS leakage is 3-10x smaller than nMOS leakage.

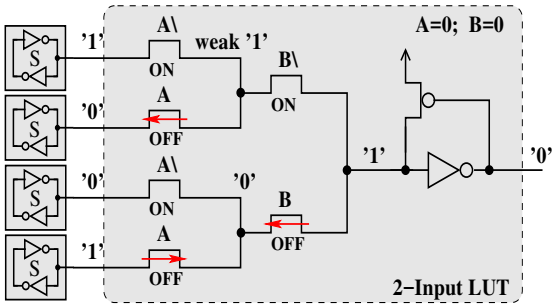


Figure 3.6: A 2-input XOR gate static leakage behavior.

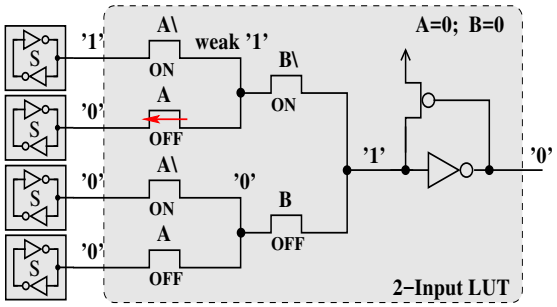


Figure 3.7: A 2-input OR gate static leakage behavior.

The static power consumption of a 2-Input LUT is important to study. Figures 3.6 and 3.7 present the leakage behaviour of an XOR gate and an OR gate, respectively. As it is apparent the two gates exhibit different leakage figures even when they process the same input data. The XOR gate has 3 components of different types of leakage (2T+Z), while the OR gate has one component (Z) of leakage. This observation points out the dependency between static power and the SRAM configuration, which can be used by attackers. Moreover, the multiplexer exhibits a current flow through the pass transistors driven by complementary inputs (e.g. (A, A\)) when their SRAM configurations have different polarities.

Static power also depends on the process data as seen in Table 3.3. Unlike dynamic power, static power is consumed even in the absence of switching (during a steady-state condition). Therefore, the 2-Input LUT has four static states compared to 16 dynamic states as in Table 3.2. Table 3.3 shows the static leakage of a number of different gates and processed data. For example, the OR gate consumes one unit of 'Z' type leakage when processing (A=0, B=0). In contrast, OR and XOR gates consume T+Z and 2T+Z units of leakage respectively. The OR gate exhibits significantly different values depending on the processed data. Overall, the 2-input LUT exhibits different leakage consumptions, depending on both processed data and the SRAM configuration. Dynamic power also exhibits an explicit dependency on the inputs value and processed functions.

Table 3.3: Static leakage for different gates and processed inputs.

Input		Function					
A	B	AND	NAND	OR	NOR	XOR	XNOR
0	0	Z	T	T+Z	T+Z	2T+Z	T+2Z
0	1	Z	T	2T	2Z	T+2Z	2T+Z
1	0	2T	2Z	Z	T	T+2Z	2T+Z
1	1	T+Z	T+Z	Z	T	2T+Z	T+2Z

3.2 Power Analysis Attacks

Attacks on cryptosystems based on power consumption analysis were introduced by Kosher et al. [58] in 1999. Since then, designers have developed defence measures to secure cryptographic ASIC devices (e.g. smart cards), which play such a major role in the modern society. Attacks can be classified as either active or passive. It is reminded that revealing the secret key is the goal of these attacks. Active attacks manipulate the cryptographic device's inputs and environment to induce abnormal behaviours in the device under attack, whereas passive attacks (such as Side-Channel Attacks (SCA)) extract the secret key by monitoring the physical properties of the cryptographic device. Countermeasure circuit-level techniques are proposed in this work with the goal of increasing the robustness of FPGAs to all known power related attacks: dynamic, static, glitches and early evaluation.

3.2.1 Simple Power Attacks

Simple Power Analysis (SPA) and Differential Power Analysis (DPA) were both introduced by Kosher et al. [58] in 1999. In SPA, the secret key is revealed by observing a single or a few traces of the power consumption. These traces can provide sufficient information to uncover the secret information mainly because the cryptographic algorithms have conditional branches which depend on single key bits.

For example, Figure 3.8 shows the power consumption of an Elliptic Curve scalar multiplication while using an unknown key. By observing the power consumption trace,

the key can be easily obtained. Scalar multiplication performs different point operations namely addition and doubling based on the scanned key bit [56, 85]. Each point operation consists of a different number of field arithmetic operations. As a result, their power consumption pattern is easily distinguishable.

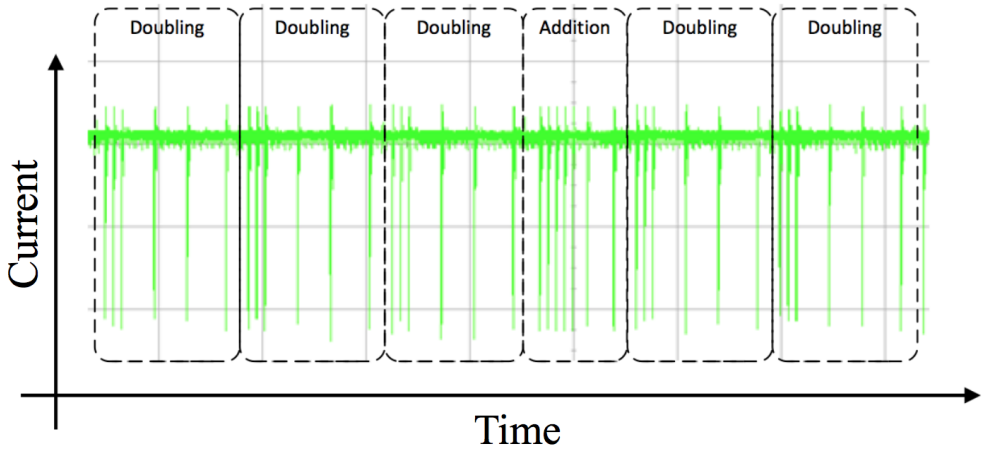


Figure 3.8: A window of power consumption trace of scalar multiplication.

Figure 3.8 illustrates that the first pattern corresponds to a point doubling operation, since it is the first executed instruction in the algorithm [8]. This pattern is followed by two identical patterns. The third doubling operation is followed by an addition. Then, two doubling patterns follow the addition pattern. As result of this pattern (DDDADD), the key performed during this window is ”00100”. This method can be applied to extract the entire key.

3.2.2 Differential Power Attacks

Differential power analysis (DPA) is an extended version of SPA, where the attackers acquire multiple power traces to be able to filter the noise and thus obtain valuable information about the secret key. The attacker uses a number of different power models

and statistical analysis to quantify the relation between power consumption and processed data an functions. In differential power analysis (DPA), no knowledge of the cryptographic device implementation is generally needed as long as the cryptographic algorithm is well-known [74].

3.2.3 Correlation Power Attacks

Correlation Power Attacks (CPAs), which are considered more poweful than the DPAs, were introduced by E Brier [19]. A CPA is based on the correlation between the actual power consumption and the predicted output of a cryptographic circuit. Therefore, the attacker looks at the correlation between the actual power values and the predicted power of the guessed key. A high correlation would mean that the attacker has been able to reduce the search in the key space.

3.2.4 Static Power Attacks

Attacks based on static power have been introduced by Lin and Burleson [68], where the authors used the dependency between the input data pattern and the static power consumption. The strength of the attack increases with newer technologies as the static power comprises a large portion of total power consumption. Moreover, unlike dynamic power, static power only depends on the current state and inputs of the circuit. As such in an FPGA environment, static power depends on the SRAMs' configurations and the input data. Authors of [4, 5, 68] show that even DPA-resistant logic styles suffer from lack of resistance to static power attacks. From looking at the observations in [4, 5, 68], it is clear that studying and providing solutions remains an open problem.

3.3 Power Models and Statistical Analyses for Attackers

3.3.1 Hamming Weight Model

The Hamming Weight is a simple power model which was initially proposed by Kosher and Brier [19, 58]. The Hamming weight model is presented in Equation (3.6):

$$Y = aH(X) + b \tag{3.6}$$

Where Y is the power consumption, H(X) is the Hamming weight of the manipulated data X, a is the proportionality factor between Y and H(X), and b includes the acquisition noise and the power variation from one clock cycle to another. By observing the bus power of a microprocessor-based cryptosystem, it is simple to determine H(X) where its value is either zero (no power consumed) or one (consumed power). As most the information being processed by the chip is transferred across the bus, this relationship can expose the secret information when it is sent across the bus.

Unlike Hamming distance, Hamming weight is simpler to use in the sense that the attacker does not need any knowledge about the netlist. This model considers the Hamming weight of the data being manipulated at given points in time. This model has been successfully applied to microprocessors with precharge buses, since they consume an enormous amount of power compared to any other single feature on the chip.

3.3.2 Hamming Distance Model

The Hamming distance is a power model that helps attackers to analyze the power consumption of the circuit under attack [19]. The Hamming distance equals the number of transitions on the bus in clock cycle. In this modelling, the simulation time of the circuit is divided into intervals. Then, the number of transitions from one interval to the following

one are counted. This gives an estimate of power consumed per interval. This model is perfectly suited to the part of the device that does not experience glitches such as data buses and registers. The Hamming distance leakage model assumes that the transitions 0 $\rightarrow$ 0 and 1 $\rightarrow$ 0, do not consume any power, whereas the transitions 0 $\rightarrow$ 1 and 1 $\rightarrow$ 1, do consume power. By using this model, multiple successful attacks have been mounted against ASIC and FPGA implementations [19, 82, 89, 109].

3.3.3 Switching Distance Model

In this model, the previous models are enhanced to distinguish the difference between charging and discharging the load capacitances. For this purpose, the normalized difference of the transition dynamic power is defined as $\delta = P_{0\rightarrow1} - P_{1\rightarrow0}/P_{0\rightarrow1}$. By using this model, it is clear that it is feasible to distinguish between 0 $\rightarrow$ 1 and 1 $\rightarrow$ 0 bit transitions, which will ultimately lead to successful attack [91].

3.3.4 Correlation Coefficient

As we have mentioned, DPA is based on a statistical analysis that determines the dependency between power consumed and processed data and functions. Hence, the Pearson correlation coefficient is an excellent choice as a statistical analysis tool when it comes to performing DPA attacks assuming the linear dependence between the processed data and power consumption. The correlation coefficient takes into account the average value and variance which leads to more accurate result.

3.3.5 Difference of Means

Difference of Means is also used to mount DPA attacks. This method models the relation between the real power consumption values and the predicted power consumption. If the

expected values of both real and hypothetical power consumptions are almost equal, then the predicted key is very likely to be correct. Unlike the correlation coefficient and the distance, the Difference of Means needs a larger number of power traces as it only considers differences of values and not the corresponding variances.

3.3.6 Distance of Means

Distance of Mean is another statistical analysis method used to determine whether two distributions have an equal mean or not. It has an advantage over the Difference of Means approach as it takes into account the variance between the two means and is not based only on subtracting them from each other. In this method, the correlation between the mean of the real measurements and the mean of hypothetical power consumption is calculated using Eq 3.7:

$$r(X,Y) = \frac{m_X - m_Y}{s_{X,Y}} \tag{3.7}$$

$S_{X,Y}$ denotes the standard deviation of the different distributions of the two sets.

3.4 Conclusion

This chapter shows that the power consumption of the device carries valuable information. During charging and discharging of the circuit load capacitances, the switching power depends on the processed data and the function. Glitches and early evaluations increase the dynamic power of the device, which in turn increases its vulnerability to power attacks. In the absence of switching, the device only consumes static power which depends only on the inputs and the SRAM configuration. It is emphasized that each component of the power consumption shows explicit dependency on the processed data and function. Therefore, a secured hardware implementation should be robust against each and every

power component attack. In the following chapter, a literature review of different countermeasures targeting the dynamic power is presented.

Chapter 4

Prior Art

Chapter 3 outlined the components of the power consumption in CMOS circuits. It has also shown that it is possible to assess the relation between the processed data, function, and each component of the power consumption. Moreover, it outlined different types of attacks which exploit these relationships.

This chapter will review the prior art in countermeasures for making circuits robust to power consumption attacks. Securing the chip is based on the elimination of exploitable relationships between processed data, function, and power consumption. There are two main approaches to acheive this goal [74]:

(i) **Hiding (or concealing)**, which either balances the power consumption into a constant value or randomizes the total system's power by processing extra operations and/or data, and

(ii) **Masking**, in which the input data is scrambled with a random mask such that the intermediate values and, therefore, the system's power consumption will exhibit a reduced (ideally zero) dependence on the unmasked input data.

These two approaches can be applied at different abstraction levels of: the protocol, the algorithm, the architecture, or the circuit. Figure 4.1 summarizes the prior art in

countermeasures and the level at which they can be applied. The countermeasures applied at the first three levels are either power demanding, due to the replication of coarse-grained arithmetic and logic units, produce a large silicon area overhead, and require a significant programming effort [30, 58, 116]. In contrast, the circuit level countermeasures should provide better results since as they directly tackle the CMOS power consumption problem.

		PAAs Countermeasures	
		Hiding	**Masking**
Level of Abstraction	**Protocol Level**	Key Update [60]	
	Algorithmic Level	− Shuffling Independet Operations[42] − Eliminating Conditional Branches[46] − Dummy Operation Insertion[27]	− Masking Intermediate Value[15] − Random Initial Point[73]
	Architecture Level	− Using same order Instruction[25] − Secure VLIW Architecture	− Arithmetic Masking
	Circuit Level	− SABL[121] − RCDDL[98] − LBDL[135] − WDDL[124] − AWDDL[86] − BCDL[88] − iWDDL[78] − DWDDL[134] − STTL[99]	− MDPL[124] − DRSL+[102] − RSL[119] − DRSL[24]

WDDL: Wave Dynamic Differential Logic
BCDL: Balanced Cell-based Differential Logic
MDPL: Masked Dual-rail Precharge Logic
DRSL: Dual-rail Random Switching Logic
RCDDL: Reduced Complementary Dynamic and Differential Logic
SABL: Sense Amplifier Based Logic
LBDL: LUT-Based Differential Logic
STTL: Secure Triple Trail Logic
AWDDL: Asynchronous WDDL
iWDDL: isolated WDDL
DWDDL: Double WDDL
DRSL+: positive DRSL
RSL: Random Switching Logic

Figure 4.1: Various power attack countermeasures at different abstraction levels.

This chapter gives an overview of the existing countermeasures to power analysis attacks focusing on circuit level solutions. The countermeasures are grouped based on their level of abstraction (protocol, algorithm, architecture, and circuit).

4.1 Protocol Level Countermeasures

In order to counteract power analysis attacks, Kocher et al. proposed a mechanism to periodically update the secret key [60]. According to this mechanism, the user updates the

key for every execution session of a cryptographic algorithm, so that the power consumption pattern will change. The key update mechanism increases the robustness of all types of side channel attacks (power consumption, electromagnetic emission, timing attacks). The disadvantage of updating the key is that it is appropriate for short sessions. Recent cryptosystems rely on Physically Unclonable Functions (PUF) as random number generation hardware to creat secret keys [112]. Since PUF cannot be changed after manufacturing, updating the key is impractical in such cryptosystems. A solution to this problem was proposed by Medwed et al. [78,79]. The basic idea is to generate a master key $\boldsymbol{k}$ to derive a new session key $\boldsymbol{k*}$ using a modular multiplication for each execution of an algorithm. However, in [33], Dobraunig et al. presented a simple key-recovery attack on the re-keying scheme proposed in [78, 79]. The major disadvantage of countermeasures based on key updating is the large hardware and software overhead. Protocol level countermeasures are very rare because they are difficult to apply, especially when the design engineer does not have the freedom to change the crypto protocols.

4.2 Algorithm Level Countermeasures

Algorithm level countermeasures aim to make the power consumption independent of the processed data through applying different techniques such as inserting dummy operations, shuffling concurrent operations, etc. Every algorithm has a major weakness to power analysis attacks, which mainly come from conditional branch operations that depend on the secret key bit or sequence of operations of different latencies. As a consequence, simple power attacks can in general derive the secret key.

To better illustrate the algorithmic countermeasures, we can consider the binary scalar multiplication algorithm on an Elliptic Curve as an example (Algorithm 1). In this algorithm, the secret key $\boldsymbol{k}$ is multiplied by a point on the elliptic curve [56, 85]. In

ECC, the k sizes range between 160 to 521 bits as recommended by the National Institute of Standards and Technology (NIST) [38]. As is apparent in Algorithm 1, the scalar multiplication executes different point operations based on the secret key bit. If the key bit is '0', only point doubling (D) will be executed. Point doubling is followed by point addition (A) when the key bit is '1'. Since each point operation consists of a different number of field arithmetic operations, the attackers can determine the pattern of the point operation (and thus the secret key) by observing a single power trace.

Algorithm 1 Binary scalar multiplication algorithm

Input: $P \in E(\mathbb{F}_P),\ k = (k_{l-1}, ..., k_0)$,

Output: $Q = kP$

$Q \leftarrow P$

for $l = ((l-1)$ downto $0)$ **do**

$Q \leftarrow 2Q$; point doubling operation (shorter latency)

if $k_l = 1$ then ; branch depends on the key bit

$Q \leftarrow P + Q$; point addition operation (longer latency)

end for

return Q

Figure 4.2 exemplifies the power consumption waveform of the scalar multiplication. It is apparent that the operations executed are Doubling, Doubling, Doubling, Addition, Doubling, and Doubling, which correspond to a key-bit sequence of "00100". This method can be applied to extract the entire key.

As mentioned, hiding and masking techniques can be applied at the algorithm level to eliminate the relation between the secret key and the power consumption.

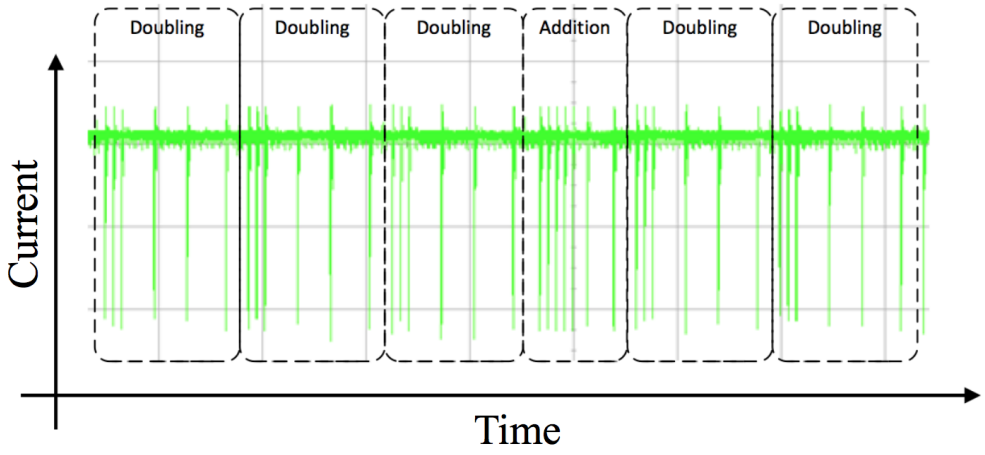


Figure 4.2: A window of power consumption trace of scalar multiplication.

4.2.1 Hiding countermeasures

Algorithm-level hiding algorithmic countermeasures are based on concealing or randomizing the power consumption while processing the same intermediate value as the unprotected implementation. This can be achieved through operation shuffling or dummy operation insertion.

Shuffling Independent Operations

Shuffling is a hiding technique that randomly permutes independent operations to change the shape of the power consumption traces. The practicability of these techniques depends on the number of independent operations that exists in the cryptographic algorithm [42,76].

Dummy Operations Insertion

Another way to hide the relationship through randomizing the power consumption trace is to randomly insert dummy operations during the execution of the cryptographic algorithm. As the type, launching time, and location of the dummy operation are random, the total power consumption will also be random. These techniques do not make the attack infeasable

but rather increase the workload of acquiring more power traces. Such countermeasure have been successfully attacked by Clavier et al. in [27].

Unifying EC Point Operations

As previously mentioned, the points operations in ECC have a different number of arithmetic operations. By forcing each of the point operations to run the same number and type of arithmetic operations, the power consumption is concealed through balancing it into a constant value.

Eliminating Conditional Branches

By eliminating the conditional branch, the conventional scalar multiplication algorithm runs both the point doubling and point addition operations, which have different power consumption patterns, without leaking any information about the key-bits. For instance, the Always Double-and-Add algorithm [46] executes a point doubling (D) operation followed by a point addition (A) operation for each bit of the key independent of its value (Algorithm 2).

Algorithm 2 Always Double-and-Add algorithm

Input: $P \in E(\mathbb{F}_P),\ k = (k_{l-1}, ..., k_0),$

Output: $Q = kP$

$Q \leftarrow P$

for $l = (k-2 \text{ downto } 0)$ **do**

$Q_0 \leftarrow 2Q$

$Q_1 \leftarrow P + Q_0$

$Q \leftarrow Q_{kl}$

end for

return Q

In [25], Chevallier-Mames et al. generalized the idea behind the Always Double-and -Add algorithm by introducing a so-called side-channel atomicity. Point doubling and point addition are represented by a number of atomic blocks, each having the same set of modular operations. This technique leads to a further reduction in the dependence of the power consumption on the processed data.

Many other countermeasures can be adapted to hide the relation between the consumed power and the processed data, such as: (1) using a constant execution path code; (2) choosing operations that leak less information in their power consumption; (3) balancing Hamming weights and state transitions. A detailed discussion on this subject is beyond the dissertation scope and will not be provided.

4.2.2 Masking countermeasures

The idea behind masking is to remove the relation between the power consumption and processed data through randomizing the intermediate data values. This approach is widely used at this level of abstraction when the device's power consumption characteristics cannot be easily changed without a major reimplementation effort.

Masking Intermediate Values

One way to randomize the power consumption signal is to replace the intermediate variable V, which is in direct relation to the secret key, with a masked version V_m. The mask is generated using either a Boolean operation (XOR) or arithmetic operations (modular addition) of the variable V with a random mask m:

$$V_m = V \oplus m$$

$$V_m = V + m \bmod n$$

As result, the power consumption needed to process the randomized intermediate V_m is largely independent of the actual intermediate value V.

To increase the adversary's uncertainty, the intermediate variable V can be randomly split into multiple shares $V_1, ..., V_d$. Then, the mask operation can be performed separately on each component. The attacker will then need to attack each component separately in order to extract the information [15, 21, 41, 96, 97].

Random Initial Point

Random Initial Point (RIP) is another masking countermeasure used against power analysis attacks [72]. In this technique, the eliptic curve intial point (P) in the scalar multiplication (Algorithm 1) is replaced with the point ($P+R$), where R is randomly chosen. The actual value kP can be obtained by substracting kR from the output $k(P+R)$. Similar techniques were proposed by C. K. Kim et al. for RSA [54]. However, these countermeasures are vulnerable to power analysis by exploiting specially chosen input messages [133].

Both algorithm-level hiding and masking countermeasures are power demanding due to the replication of coarse-grained arithmetic-logic units, required a large silicon area overhead, and can require a significant programming effort [30, 58, 116].

4.3 Architecture Level Countermeasures

Architecture-level countermeasures are based on the modification of the instruction set to eliminate the relation between the power consumption of the computing engine and the processed data. Examples of such countermeasures are outlined below.

First-order Operation Only

In this class, designers are restricted to using only operations that have equal (typically unitary) power and latency. This can be achieved through subdividing a cryptosystem's operations into equal parts or by building atomic blocks that execute the same set of

operations. Atomicity is a well-known technique in securing the elliptic curve cryptosystem. In [25], Chevallier-Mames et al. generalized the Always Double-and-Add algorithm by introducing the side-channel atomicity. Point doubling and point addition are implemented by a set of atomic blocks that each has the same set of modular operations.

Secure VLIW Architecture

The Very Long Instruction Word (VLIW) architecture allows a concurrent execution of a set of instructions. It can be used as a countermeasure by scheduling operations of different order in parallel, ensuring that the secure implementation has a constant total operation order every clock cycle.

Secure Data Encoding

Secure Data Encoding is a technique that hides information by enforcing an equal Hamming distance or Hamming weight. For instance, the real circuit processes the binary value '101' whereas the mirror circuit processes the complementary number '010' at the same time [128]. As another example, in [81] the authors use leakage-resistant arithmetic [14], which is based on the Residue Number System (RNS) algorithm. The RNS algorithm provides randomization, both at the circuit level (spatial randomization) and the data level (arithmetic masking). This technique is effective in providing robustness to attacks based on Hamming distance or weight. Although this technique is able to conceal the switching power, other power components can still leak information as mentioned in Chapter 3.

The advantage of the secure data encoding countermeasure is that the user is not concerned about how the code is compiled into assembly. Unfortunately, such techniques may increase the concealment level based on the processed function but not on the processed data, which can now become the attackers' new target. An architecture which has power independent of both data and operation is difficult to design.

4.4 Circuit Level Countermeasures

Circuit level countermeasures target the origin of the problem rather than change the power consumption pattern at higher levels of abstraction. Circuit level countermeasures can also be classified into two classes: **Hiding (or concealing)** and **Masking**. Both classes can use the Dual-Rail Logic (DRL) family as the base of their techniques. DRL [74] balances the dynamic power consumption into a constant value through signal differential encoding, $\mathbf{S^d} = (S, S\backslash)$, where one wire ($S$) carries the direct signal and the other ($S\backslash$) carries the complementary signal. DRL operates in two alternating phases: (i) *precharge*, during which both the direct and complementary signals are set to a common '0' value, and (ii) *evaluation*, during which either the direct signal or the complementary signal will perform a transition to '1' becoming valid, and thereby enforcing 100% activity. *Hiding* aims to increase the symmetry between the dual rails to conceal the power consumption. *Masking*, on the other hand, randomizes the intermediate data in order to further increase the level of security against power attacks.

4.4.1 Masking countermeasures

One way to implement *masking* is to randomize the intermediate values by scrambling the circuit inputs with a random mask. This will also randomize the power consumption and thus break the relation between the actual input data and the power consumed.

Masked Dual-rail Precharge Logic (MDPL)

Wave Dynamic Differential Logic (WDDL) [124] is a *hiding* technique that employs dual -rail circuit with a special place-and-route method to ensure equal complementary loads. The MDPL has proposed to remove the WDDL place-and-route constraint by combining the dual-rail circuit with a random mask [94]. In MDPL, each signal value is represented

by its masked value $(data \oplus mask)$ where the mask is randomly updated every clock cycle. MDPL employs the precharge method of WDDL [124] where all signals, including the complementary outputs and the mask signal, are precharged to a common value (e.g. '0') before the evaluation phase. In the evaluation phase, MDPL calculates complementary outputs using Majority (MAJ) gates. A MAJ gate output is high when more than 50% of its inputs are high. For instance, the MDPL AND gate is implemented using two MAJ gates. The true value $Q_m = MAJ(a_m, b_m, m)$ is produced by one MAJ gate, and the complemented output $\overline{Q_m} = MAJ(\overline{a_m}, \overline{b_m}, \overline{m})$ is produced by the other MAJ gate. All other gates can be built with MDPL AND gates. As shown in Suzuki and Saeki [117], MDPL leaks information due to early evaluation effect as also confirmed by MDPL inventor later [93]. To solve this problem, an improved MDPL (iMDPL) [93] synchronizes all the inputs by using an evaluation-precharge detection unit (EPDU). The EPDU unit delays the evaluation phase until all valid inputs arrive to eliminate the early evaluation problem. However, the complexity of basic gates is significantly increased, resulting in a 200% area overhead.

Dual-rail Random Switching Logic (DRSL)

RSL [119] is a countermeasure technique based on equalizing the transition probability by means of a single random mask bit. This technique operates in two phases based on an enable signal to calculate the output or force it to '0'. To avoid early evaluation, the enable signal should remain '0' until all inputs arrive. Such scheme has been shown to be still vulnerable to power attacks [122].

DRSL [24] is a countermeasure technique that combines the idea of MDPL [94] and Random Switching Logic (RSL) [119]. In this technique, the enable signal remains a global precharge signal, while a local precharge signal is generated to synchronize the input signals. DRSL is robust to early evaluation attacks. However, DRSL only ensures synchronized arrival times for the evaluation phase, but not for the precharge phase. Thus,

glitches may occur during the return to the precharge phase. As well, the difference between the load capacitances and/or propagation delays open a backdoor for the attacker [102]. DRSL+ [31] is another scheme that solves the problem of DRSL by limiting the design to positive logic. This solution, however, comes at the expense of at least 2.5× larger area.

4.4.2 Hiding countermeasures

The *hiding* technique attempts to achieve a constant power consumption to hinder the relation between the consumed power and the processed inputs data and function of the cryptosystem circuit. Another method of hiding is to randomize the power consumption through injecting noise [73].

Since the signal-to-noise ratio (SNR) is reduced, the correlation between the processed data and power consumption is also reduced. Such an approach is susceptible to power attacks if more power consumption traces are acquired. Randomized clock frequency is another countermeasure that can be combined with algorithmic level countermeasures (inserting dummy operation). In this technique, the algorithm executes dummy operations using randomized internal clock frequencies [22].

A number of different techniques based on the DRL logic have been proposed in the literature. These countermeasures have either been developed for ASICs, resulting in a non-configurable implementation, or proposed for mapping onto commercial FPGA architectures. In the following subsections, we will review these techniques highlighting their advantages and drawbacks.

4.4.3 Countermeasures developed for ASICs

Sense Amplifier Based Logic

Sense Amplifier Based Logic (SABL) [121] combines the idea of differential and dynamic logic. As previously mentioned, the differential logic ensures a power consumption which is independent of the polarity of the transition. That is, the transitions $(0,1) \rightarrow (1,0)$ and $(1,0) \rightarrow (0,1)$ exhibit an equal dynamic power consumption. In the lack of a transition $(0,1) \rightarrow (0,1)$ or $(1,0) \rightarrow (1,0)$, the dynamic power is zero (thus independent of the state). This logic is not sufficient to conceal power consuption as it partitions the power into two groups: dynamic power consumed $(0,1) \rightarrow (1,0)$) or no dynamic power consumed $(0,1) \rightarrow (0,1)$ [51]. This observation allows the attackers to classify the power consumption based on the processed inputs and functions. On the other hand, the dynamic logic partitions the input sequence into two groups – either dynamic power consumed $(0,1) \rightarrow (1,1)$ or no dynamic power consumed $(1,0) \rightarrow (0,0)$. Therefore, the authors of [121] recognized the need to combine differential logic and dynamic logic in order to ensure single transition per clock cycle that is independent of input value. It should be mentioned that balancing the outputs load capacitance, which is required in any dual-rail logic, is not a sufficient measure to conceal power consumption, since some degree of asymmetry still remains in the circuit. Therefore, SABL solves this problem by ensuring that for every input sequence, equal capacitances (load and intrinsic) are charged after being discharged during the precharge phase.

Wave Dynamic Differential Logic (WDDL)

Wave Dynamic Differential Logic (WDDL) [124] follows the principle of SABL while it can be adapted for ASIC, using a regular standard cell library with Static Complementary CMOS gates; and for FPGA, using Look-Up Tables (LUTs), and registers. WDDL operates in two phases: precharge and evaluation. In WDDL, the precharge value propagates

through the combinatorial logic instead of using a global precharge to reset the logic. While in the evaluation phase, the combinatorial cell gets its true input values and uses De_Morgen Laws to determine the value of the true signal and the complementary one. Since it operates differentially, only one of the two outputs will encounter a transition. WDDL is a dynamic and differential logic and therefore it has a single transition per clock cycle independent of processed input thereby increasing disclosure time and reducing the correlation between the consumed power and the processed data. A major drawback of WDDL is that only AND and OR operators can be used to implement a secure design. Thus, the area overhead of the WDDL is three times the single-ended logic and reaches up to six times in the FPGA context [124]. Also, in order to preserve the precharge wave generation, an inverter should not be used because it will stop the precharge wave. Furthermore, mapping WDDL onto FPGAs will exhibit the problem of unbalanced load capacitances since FPGA designers have no control its routing. This unbalanced load capacitances limit the security of WDDL against power analysis attacks in reconfigurable hardware. In addition, the authors in [117, 118] introduced another vulnerability known as an early evaluation (EE) which leaks valuable information.

Secure Triple Trail Logic (STTL)

STTL is another family of differential and dynamic logic [99]. It aims at further reducing the relation between processed data and the computing time of other dual-rail techniques. Instead of using dual-rail encoding, STTL is represented by triple-rail encoding. The extra rail is used to ensure the validity of the signals' inputs and outputs while it does not convey any information about the processed data bit. The third rail uses low switching current gates to ensure that the validity signals are delayed to prevent any activity before valid inputs arrive. Robustness of STTL on FPGAs against DPA has been shown in [107]. However, it also shows a significant area overhead; for example, an AND gate with two inputs was

achieved by using 11 LUT (6 LUTs for the logic and 5 LUTs for the delay of the valid signal) instead of a single LUT.

Reduced Complementary Dynamic and Differential Logic (RCDDL)

RCDDL is a dynamic and differential logic that generates its complimentary output using the true data path [98]. RCDDL consists of four so-called segments. The product-term segment and the summation segment generates the direct path. The force gate segment generates the complementary output while the fourth segment is the precharge generator. This implementation is composed of a sum of products terms rather than single-depth logic gates such as AND and OR gates. RCDDL allows the use of negative logic without stopping the precharge propagation wave, but it imposes a special sizing constraint on the transistors of the gate force segment to ensure the equal arrival time at the two complimentary outputs. Hence, it can not be made out of a standard cell. This complication along with the FPGA structure where all gates performed using the same LUT units restricts the use of RCDDL in FPGA. However, due to the reuse of the true data path to generate the complementary one, RCDDL has less area overhead than WDDL and MDPL. For example, the RCDDL XOR gate consists of 26 transistors in comparison to 36 transistors and 72 transistors for the WDDL XOR gate and the MDPL XOR gate respectively [115]. For the same reason, the overall power consumption is less than with other schemes. Moreover, RCDDL is more secure due to its lower power variance as shown in [114]. However, early evaluation is still a problem in RCDDL.

LUT-Based Differential Logic (LBDL)

LUT-Based Differential Logic (LBDL) [135] is a countermeasure that takes advantage of the LUT structure, where the LUT propagation delay is independent of the implemented function. In LBDL, the SRAM cells are removed, and the drains of the first level

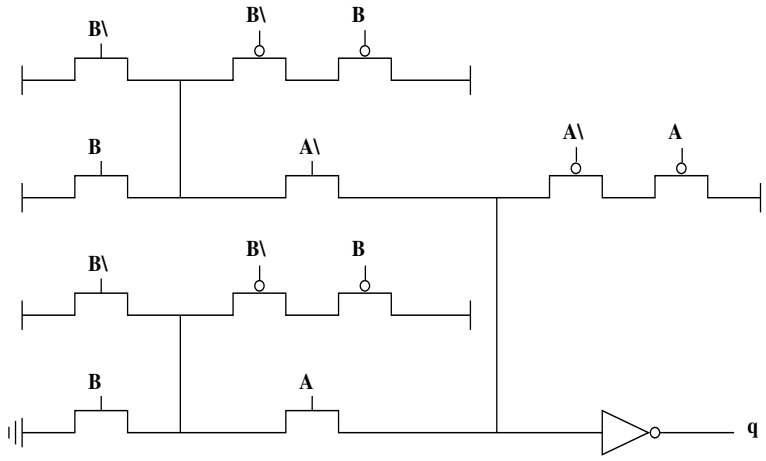


Figure 4.3: LUT based 2-input AND gate

multiplexer's transistors are connected to either V_{DD} or GND. The memory effect is removed by connecting all intermediate nodes to logic '1' through pMOS transistors during the precharge phase. To conceal the dynamic power, two LBDL LUTs are used to implement the differential logic. Hence, LBDL increases the robustness over WDDL by eliminating the effect of the floating nodes. However, the circuit is no longer configurable, as the function is defined by hardwired connections to VDD or ground. Besides, the 2-input LBDL circuit requires $6\times$ larger area than the unsecure circuit, as it uses six pMOS transistors to implement the precharge feature as presented in Figure 4.3. LBDL works for the 2-input LUT; however, extrapolating LBDL to design a 6-input LUT as on commercial FPGA results in some issues that need to be addressed. First, the impact of early evaluation degrades the robustness of the circuit. Second, the static power effect has not been addressed.

All of the aforementioned circuit level countermeasures were developed for ASICs. Although these countermeasures provide higher computing speed, the resulting implementation is non-configurable. They increase the robustness to attacks based on dynamic power consumption, but the static power can also reveal valuable side-channel information.

4.4.4 Countermeasures Applied to Commercial FPGA-mapped Circuit

Wave Dynamic Differential Logic (WDDL) and its variants

WDDL is a countermeasure that is suitable for FPGA platforms as well as for ASICs. However, mapping WDDL directly into FPGAs will face the problem of unbalanced load capacitances since generally the designers have no control over the routing in FPGAs. Such routing constraints reduce the security of WDDL against power analysis attacks. Furthermore, the authors in [117, 118] introduced another vulnerability created by different inputs arrival known as an early evaluation (EE) effects, which can leaks valuable information.

In order to map WDDL into FPGA, various techniques were introduced to overcome the mentioned drawbacks. First, double WDDL (DWDDL) was introduced in [134]. It solves the problem of unbalanced outputs loads by copying and doubling the whole circuit (including routing). The doubled part of the circuit operates as a compliment to the original WDDL. Hence, it is maintaining the exact number of charged and discharged capacitors in the circuit. This approach increases the area overhead more than $10\times$. Another approach is isolated WDDL (iWDDL) [77]. iWDDL isolates the two complimentary paths into different regions in the FPGA fabric. iWDDL tackles the leakage based on the hamming-distance model caused by the unprecharged register between the output and the memory latched. Moreover, R. P. McEvoy et al. [77] solve the precharging problem by using back-to-back registers acting as a two-phase flip-flop. Therefore, an extra register is required. Double backend WDDL (DBWDDL) [12] is similar to the iWDDL approach. Path switching is a technique used to reduce the effect of the unbalanced routing in WDDL [13]. The idea is to randomly switch the complementary signals by using an XOR gate or a multiplexer at both source and destination cells in the circuits. This hardens the DPA by randomizing the power as result of path switching. In this approach, XOR gates

are utilized to work as inverter instead of the registers in iWDDL. In our work, we assume that we have an equal complementary capacitive loads.

To cure the early evaluation problem, multiple countermeasures have been proposed in the literature. In the following sections, we presented different countermeasures in detail.

Balanced Cell-based Differential Logic (BCDL)

BCDL is an instance of dual-rail logic, which was proposed for FPGA-mapped circuits [88]. It prevents signal early propagation in the evaluation phase by using a synchronization logic, which guarantees that the evaluation does not commence before all inputs turn valid. Early evaluation during the precharge phase is prevented by means of a global precharge signal.

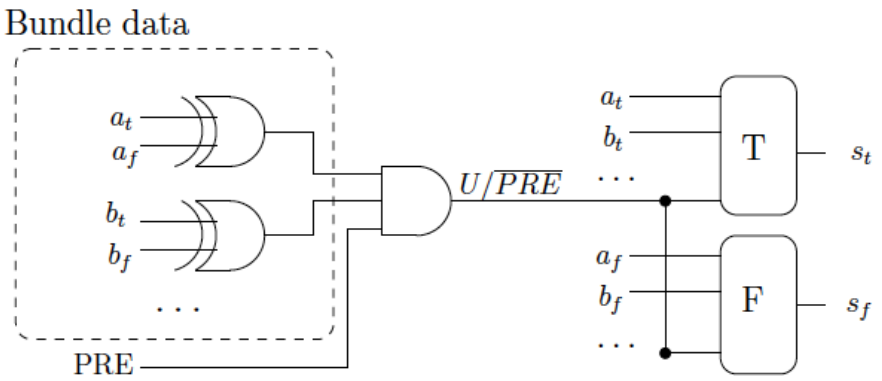


Figure 4.4: Bundle data precharge circuit [88].

Figure 4.4 shows the bundle data circuit used for the LUT inputs synchronization. The LUT precharge signal is controlled by global precharge through 3-input AND gate where the other two inputs are left to the outputs of the Bundle data circuit. The output of the synchronization logic must drive the left-most input of the LUT, which translates into significant routing constraints and more than 50% overhead in FPGA logic utilization (including the corresponding SRAM cells) as shown in Figure 4.5.

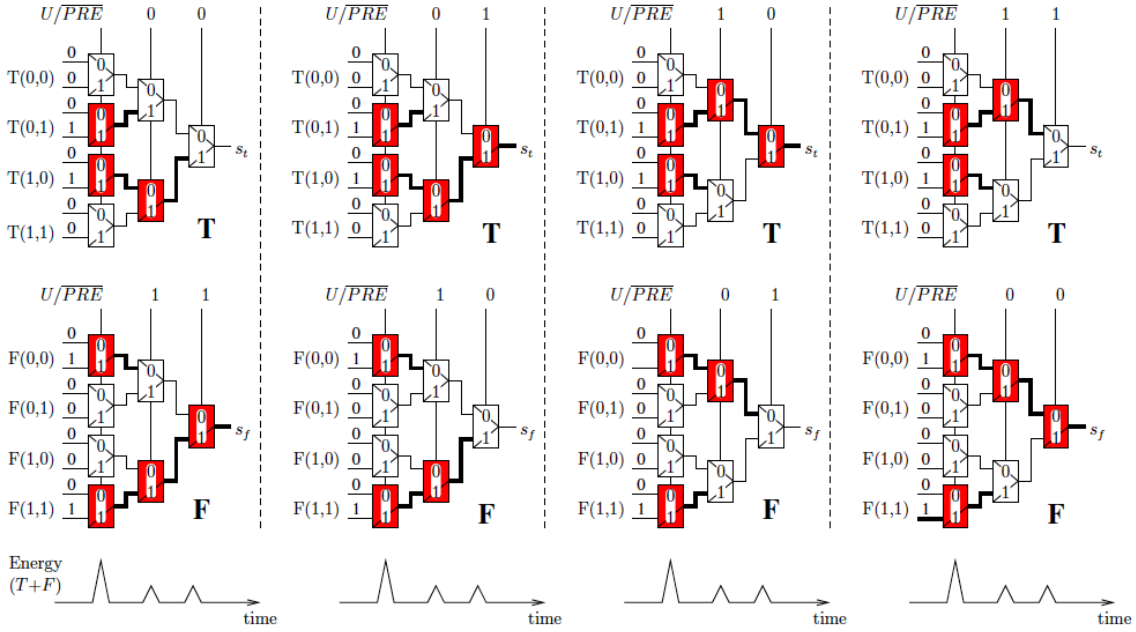


Figure 4.5: 2-input XOR using BCDL [88]

Figure 4.5 presents 2-input XOR in BCDL. It is apparent that at least 3-input LUT is required. The left-most input of the LUT is preserved for the output of the synchronization logic (U/$\overline{PRE}$). When the U/$\overline{PRE}$ signal is LOW, it enables the precharge value downstream of the SRAMs. Consequently, independent from the processed date, every middle node is precharged to '0'. When U/$\overline{PRE}$ is high, it allows the 2-input XOR configuration to propagate through. BCDL robustness against early evaluation attacks is claimed only for implementing 2-input logic functions. This is a major limitation since modern FPGAs are built with 6-input logic LUTs. In addition, the static power consumption has not been addressed, which may lead to weakness against attacks based on static power consumption.

Dual Rail Precharge Logic without Early Evaluation

Dual Rail Precharge Logic without Early Evaluation (DPL-noEE) abandons the global precharge signal used in BCDL. Instead, the SRAM configuration is modified to prevent the evaluation when at least one input is invalid as seen in Table 4.1 [17].

Table 4.1: The SRAM configuration of DPL-noEE AND/NAND gates.

DPL-noEE				AND_T		AND_F		Input state in the DPL protocol
a_T	a_F	b_T	b_F	FC80		FAE0		
0	0	0	0	0	0	0	0	All NULL0
0	0	0	1	0		0		Transitional from NULL0
0	0	1	0	0		0		Transitional from NULL0
0	0	1	1	0		0		Faulty
0	1	0	0	0	8	0	E	Transitional from NULL0
0	1	0	1	0		1		All VALID: (a, b) = (0, 0)
0	1	1	0	0		1		All VALID: (a, b) = (0, 1)
0	1	1	1	1		1		Transitional from NULL1
1	0	0	0	0	C	0	A	Transitional from NULL0
1	0	0	1	0		1		All VALID: (a, b) = (1, 0)
1	0	1	0	1		0		All VALID: (a, b) = (1, 1)
1	0	1	1	1		1		Transitional from NULL1
1	1	0	0	1	F	1	F	Faulty
1	1	0	1	1		1		Transitional from NULL1
1	1	1	0	1		1		Transitional from NULL1
1	1	1	1	1		1		All NULL1

Although this strategy provides implicit robustness against early evaluation attacks during the evaluation phase, it allows early evaluation attacks during the precharge phase [86]. Its overhead in FPGA logic utilization is significant, reaching 75% for 2-input logic functions. Robustness against static power attacks has not been assessed. Compared to BCDL, the intra-LUT dynamic activity is not constant.

Precharge-Absorbed Dual-rail Precharge Logic

Precharge-Absorbed Dual-rail Precharge Logic (PA-DPL) is another approach to counteract early evaluation attacks [48]. Since PA-DPL uses two global signals to control the precharge and evaluation phases, the overhead in FPGA logic utilization is 75%. Due to the need to implement synchronization through two global signals rather than a single one (as was achieved in BCDL), the computing speed of this logic is reduced to half compared to

BCDL. The intra-LUT dynamic and static power consumptions are not assessed raising concerns about circuit robustness against power attacks.

Asynchronous WDDL

Asynchronous WDDL (AWDDL) eliminates the early evaluation during both the precharge and evaluation phases of a dual-rail logic through a latch formed by connecting the LUT output back to one of its inputs as shown in Figure 4.6 [86]. While the LUT output does not switch before all inputs turn valid, the activity inside the LUT may carry side-channel information.

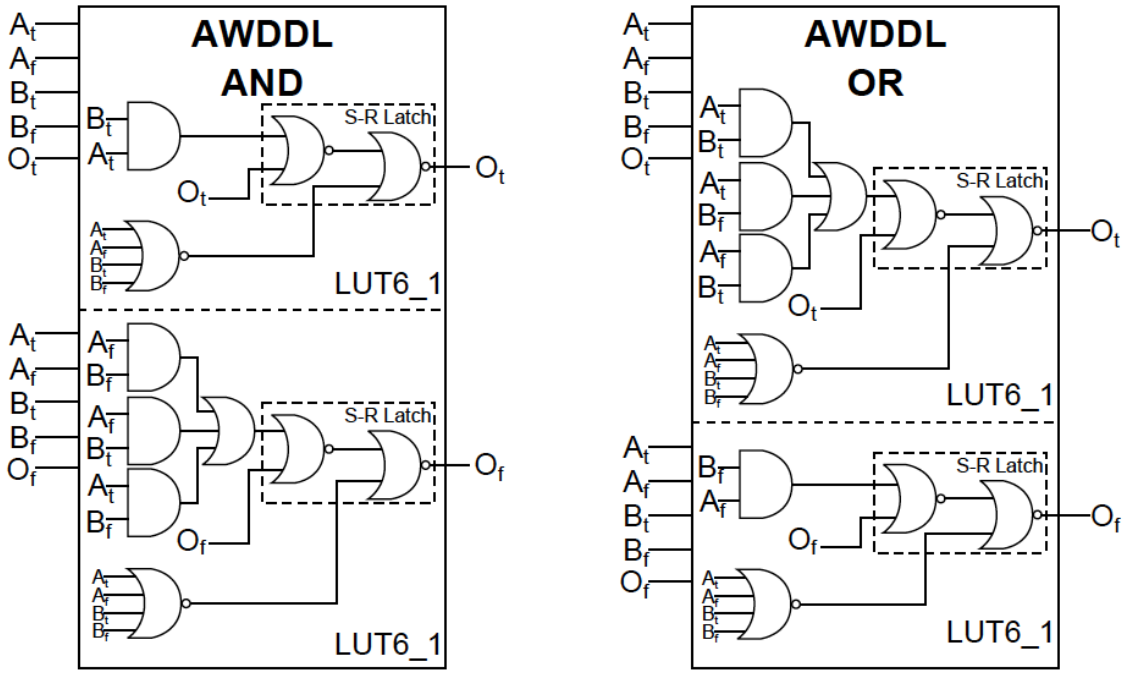


Figure 4.6: AWDDL AND-OR gates [86].

This is a significant weakness as LUTs in FPGAs are built with ratioed circuits, which are known to exhibit a large short-circuit power consumption as discussed in Chapter 2. Robustness against static power attacks has not been assessed. AWDDL has 75% overhead in FPGA logic utilization.

4.5 Conclusion

This chapter presents various countermeasures against power analysis attacks at different levels of abstractions. As demonstrated, logic families and techniques provide robustness to the switching power analysis attack. Only a few countermeasures address the attacks based on static power and early evaluation. However, none of those families provide simultaneous robustness against all types of power attacks, namely dynamic, static, glitches, and early evaluation. Therefore, the goal of this work is to provide reconfigurable hardware that is robust against all power analysis attacks.

Chapter 5

Secured-by-Design Look-Up Tables and Switch-Boxes

5.1 Introduction

Engineers are attracted by the architectural flexibility offered by Field Programmable Gate Arrays (FPGAs) in implementing their cryptographic systems, since they can test and modify their designs after chip manufacturing process. Behavioural implementations of cryptographic algorithms, which are robust to theoretical cryptanalysis, are known to be vulnerable to attacks based on the side-channel information leaked by the physical system. Circuit level techniques that reduce the leaked side-channel information would require skills in analog design, where the vast majority of cryptographic designers have skills in digital design. It is our goal to offer digital designers intrinsically secure reconfigurable hardware, relieving them from the task of eliminating the side-channel information threats through advanced analog design.

In general, FPGAs have a significantly larger power consumption than Application-Specific Integrated Circuits (ASIC) [62], making them vulnerable, if not very vulnerable,

to side-channel attacks based on power consumption. Dual-Rail Logic (DRL), which is a countermeasure against switching power consumption attacks, was proposed for ASIC, but can also be conceptually mapped onto FPGAs, as presented in Chapter 4. Since DRL is based on duplication, it would require one set of Look-Up Tables (LUTs) to perform the direct function and another set of LUTs to perform the complementary function. As a result, since a single transition is performed per unit time, the switching power consumption is concealed. However, switching power is only one component of the total power consumption. Addressing all power components is essential in the design of truly secure reconfigurable hardware. A number of features that should be incorporated into a secured-by-design logic family follows:

- **Differential encoding**, where each signal is represented in dual-rail format. Since the data is encoded as $(1,0)$ or $(0,1)$, both logic values being therefore present, it would not be possible, in general, to relate the power consumption on the processed data.
- **Symmetrical logic**, such that the dual rails see equal load capacitances. This ensures equal rail switching times and, therefore, equal switching and short-circuit power consumption per signal transition.
- **Single transition per cycle** for each signal to ensure a unity activity factor and, therefore, constant dynamic power consumption.
- **Dynamic logic** that operates in alternate phases (namely precharge and evaluation), to guarantee that every transition originates from a known state. This removes the power dependence on the previous valid state (the so-called *memory effect*).
- **Single-polarity** precharge of the dual-rail logic (e.g., $(0,0)$), to guarantee monotonic operation during the evaluation phase. This will prevent attacks based on glitches.

- **Smart replication** of the logic to eliminate the relation between the static power consumption and the processed data and operations.
- **User-programmable syncronization ability**, which allows elimination of the early evaluation effects and the associated attacks.

In this chapter, we present a Look-Up Table (LUT) and Switch Box (SB), which have been designed in order to incorporate all the above mentioned features. As such, they will be robust to all known power analysis attacks. This chapter is organized as follows. First, we introduce a two-Transistor Branch (2TB) LUT and show its advantages over the standard (commercial) LUT. This comparison is followed by disclosing dynamic and static power concealment techniques to remove the dependency of the 2-input 2TB-LUT power consumption on the processed data. The limitations of those techniques are also discussed. Next, we modify the structure of the 2-input LUT in order to achieve operation independent power consumption, a step essential in building a secured 6-input LUT. Finally, we present a circuit technique to conceal the standard routing box power consumption.

5.2 LUT with Two-Transistors Branches

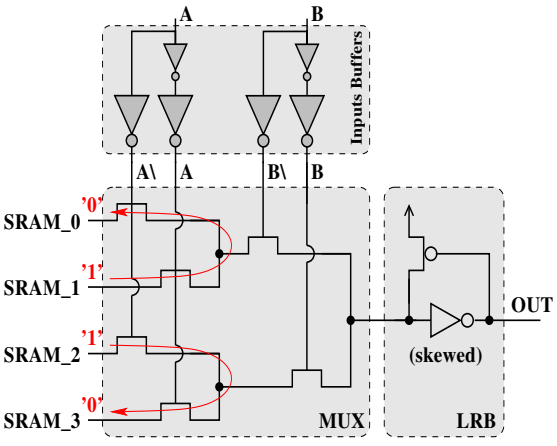


Figure 5.1: 2-input standard LUT.

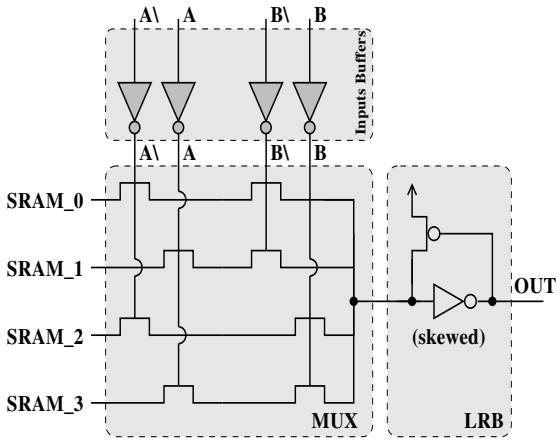


Figure 5.2: 2-input 2TB LUT.

As discussed in Chapter 2, LUTs in commercial FPGAs are built as a tree of 2:1 multiplexers implemented in pass-transistor logic. Figure 5.1 shows a 2-input LUT, which consists of three 2:1 multiplexers implementing a 4:1 multiplexer and a level-restoring buffer. In the case of a 4-input LUT, five 2-input LUTs are used to propagate the desired SRAM value to the output. A 6-input LUT requires twenty-one 2-input LUTs. Therefore, securing the 2-input LUT will eventually offer the opportunity to secure the multiple-input LUTs.

In the standard 4:1 multiplexer built with three 2:1 multiplexers (two for the first level and one for the second level), the transistor gates of each level are driven by complementary input pairs (A, A\and B, B\). If the SRAM configurations of the 2:1 MUX have different polarities, the 2:1 MUX exhibits a ”sneak” path leakage for any input values are shown in Figure 5.1 (red arrow). Hence, the leakage of the first level depends on the SRAM configuration. XOR/XNOR gates are a good case study to show the dependency on the SRAM configurations, since they exhibit maximum leakage for any input combination, as shown in Table 5.1. This fact makes the concealment of the static power difficult to achieve, as it is caused by the asymmetric structure of the standard LUT.

Moreover, there is a delay difference between the complementary inputs, which is caused by the inverter between them, as shown in Figure 5.1. As a result, when the SRAM value in the adjacent branch is opposite, a short-circuit current will flow during transitions. Since the complementary input is generated through a hardwired inverter, the only way to encode the input in a dual-rail format is to use two different LUT inputs, one for the direct input and one for the complementary input. As consequence, the 4-input LUT can implement only two-input logic functions resulting in 50% of the LUT resources being wasted. In addition, if the inputs are in the precharge phase, then half of the pass transistors are *ON* to propagate the precharge value. Consequently, one of the complementary circuits exhibits a short-circuit current that depends on the input arrival. This fact explains why

implementing DRL-based countermeasures on FPGAs suffers from early evaluation effects. Since the standard LUT has multiple leakage figures, it makes data-independent power consumption difficult to achieve.

To address these issues, a symmetrical LUT is built with two-transistor branches (2TB), as depicted in Figure 5.2. It is expected that a 2TB-LUT will generate a symmetrical response in power consumption, a feature which will simplify the effort to conceal the power consumption. The 2TB-LUT structure has additional advantages. First, this structure reduces the snake path leakage which depends on SRAM configuration. It also separates the complementary inputs to eliminate the early evaluation effect. Furthermore, separating the complementary inputs eliminates the flow of a short-circuit current during transitions.

The 2TB LUT is a one-level 4:1 multiplexer. It uses two transistors in each branch to select the right LUT output value. In both standard and 2TB LUTs, the middle nodes have been connected to the ground through a transistor. The transistor gate is driven by a global precharge signal to pull the node to logic '0', thus removing the memory effect. According to our simulation performed with Cadence tool, 2TB LUT configuration has, on average, a higher leakage power consumption of 5%. However, it has a beneficial effect as it reduces the overall hardware replication overhead, as discussed in the following sections. 2TB LUT also allows each input driver to see an equal number of nMOS gates, thus balancing the complementary load seen by the previous LUTs and/or buffers.

By comparing the classical 2-input LUT and 2TB LUT in terms of the static leakage behaviour, their dependency on the processed data and functions is shown. As mentioned earlier, the static power can be measured easily by stopping the clock during a steady state of the LUT. For 2-input LUT there are four possible input combinations and 16 possible functions, as shown in Table 5.1. Examples of calculating the circuit leakage in the evaluation phase are provided in Figures 5.3 and 5.4. As a case study, it is assumed that the configurations are SRAM_0 = '1', SRAM_1 = '1', SRAM_2 = '1', and SRAM_3 = '0'

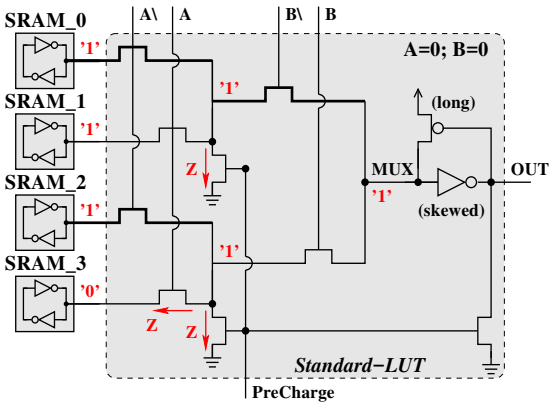


Figure 5.3: Leakage calculation of 2-input standard LUT.

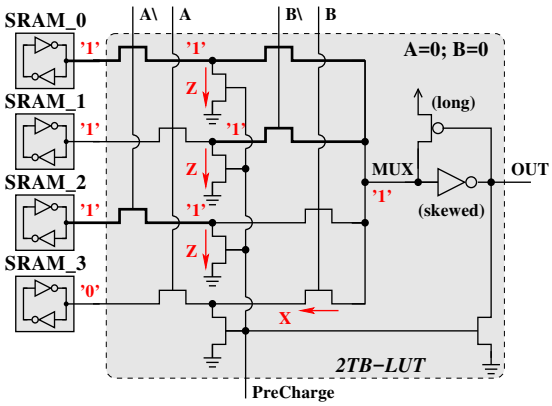


Figure 5.4: Leakage calculation of 2-inputs 2TB LUT.

for both standard and 2TB LUTs (this corresponds to an AND function). For an input combination A = '0' and B = '0', the SRAM_0 branches are *ON*; therefore, the node MUX = '1'. A voltage drop across a single *OFF* transistor generates a leakage current of 'T' or 'Z', depending on the node value (weak '1' or '1', respectively). A voltage drop across a series connection of two *OFF* transistors generates a leakage current equal to a fraction X (as in the SRAM_3 branch of the 2TB LUT). It is apparent that the standard LUT leakage is equal to $3Z$, while the 2TB LUT generates a leakage of $3Z+X$. By changing the input combination to A = '1' and B = '1', the SRAM_3 branches are *ON*. Therefore, the node MUX = '0'. Hence, the leakage figure of both the standard and the 2TB LUT change to $T+2Z$ and $T+2Z+X$, respectively. The complete leakage figures for all possible inputs and functions are provided in Table 5.1.

It must be recalled that T, Z, X, and Y represent the leakage components presented in Chapter 3. It is apparent that the static power depends on the input values and SRAM configurations. Table 5.1 clearly indicates that even if we add the leakage figures of the complementary functions, the leakage still depends on the processed data and functions. For example, by adding the leakages of a NAND and AND gate for A = '1' and B = '1',

Table 5.1: All possible static leakage of different 2-input gates and processed inputs.

Function	Circuit Element	Std-LUT				2TB-LUT			
	Inputs (A B)	0 0	0 1	1 0	1 1	0 0	0 1	1 0	1 1
F1=0	SRAM_0 = 0 SRAM_1 = 0 SRAM_2 = 0 SRAM_3 = 0	0	0	0	0	0	0	0	0
F1\= 1	SRAM_0 = 1 SRAM_1 = 1 SRAM_2 = 1 SRAM_3 = 1	2Z	2Z	2Z	2Z	3Z + Y	3Z + Y	3Z + Y	3Z + Y
F1 + F1\	Total	2Z	2Z	2Z	2Z	3Z + Y	3Z + Y	3Z + Y	3Z + Y
F2= NAND	SRAM_0 = 0 SRAM_1 = 0 SRAM_2 = 0 SRAM_3 = 1	T	T	3Z	T+2Z	X	T	2Z	T+3Z+ X
F2\=AND	SRAM_0 = 1 SRAM_1 = 1 SRAM_2 = 1 SRAM_3 = 0	3Z	3Z	2T+ Z	T+2Z	3Z+ X	4Z + Y	T+2Z + Y	T+2Z+ X
NAND + AND	Total	T+3Z	T+3Z	2T+4Z	2T+4Z	3Z+2X	T+4Z + Y	T+4Z + Y	2T+5Z+2X
F3	SRAM_0 = 0 SRAM_1 = 0 SRAM_2 = 1 SRAM_3 = 0	3Z	T+2Z	T	T	2Z	T+3Z+ X	X	T
F3\	SRAM_0 = 1 SRAM_1 = 1 SRAM_2 = 0 SRAM_3 = 1	2T+ Z	T+2Z	3Z	3Z	T+2Z + Y	T+2Z+ X	3Z+ X	4Z + Y
F3+F3\	Total	2T+4Z	2T+4Z	T+3Z	T+3Z	T+4Z + Y	2T+5Z+2X	3Z+2X	T+4Z + Y
F4	SRAM_0 = 0 SRAM_1 = 0 SRAM_2 = 1 SRAM_3 = 1	2Z	T+ Z	2Z	T+ Z	2Z+ X	T+2Z+ X	2Z+ X	T+2Z+ X
F4\	SRAM_0 = 1 SRAM_1 = 1 SRAM_2 = 0 SRAM_3 = 0	T+ Z	2Z	T+ Z	2Z	T+2Z+ X	2Z+ X	T+2Z+ X	2Z+ X
F4+F4\	Total	T+3Z	T+3Z	T+3Z	T+3Z	T+4Z+2X	T+4Z+2X	T+4Z+2X	T+4Z+2X
F5	SRAM_0 = 0 SRAM_1 = 1 SRAM_2 = 0 SRAM_3 = 0	T	T	T+2Z	3Z	T	X	T+3Z+ X	2Z
F5\	SRAM_0 = 1 SRAM_1 = 0 SRAM_2 = 1 SRAM_3 = 1	3Z	3Z	T+2Z	2T+ Z	4Z + Y	3Z+ X	T+2Z+ X	T+2Z + Y
F5+F5\	Total	T+3Z	T+3Z	2T+4Z	2T+4Z	T+4Z + Y	3Z+2X	2T+5Z+2X	T+4Z + Y
F6	SRAM_0 = 0 SRAM_1 = 1 SRAM_2 = 0 SRAM_3 = 1	2T	2T	4Z	4Z	T + X	T + X	4Z+ X	4Z+ X
F6\	SRAM_0 = 1 SRAM_1 = 0 SRAM_2 = 1 SRAM_3 = 0	4Z	4Z	2T	2T	4Z+ X	4Z+ X	T + X	T + X
F6+F6\	Total	2T+4Z	2T+4Z	2T+4Z	2T+4Z	T+4Z+2X	T+4Z+2X	T+4Z+2X	T+4Z+2X
F7 = XNOR	SRAM_0 = 0 SRAM_1 = 1 SRAM_2 = 1 SRAM_3 = 0	T+3Z	2T+2Z	2T+2Z	T+3Z	T+3Z + Y	T+2Z	T+2Z	T+3Z + Y
F7\= XOR	SRAM_0 = 1 SRAM_1 = 0 SRAM_2 = 0 SRAM_3 = 1	2T+2Z	T+3Z	T+3Z	2T+2Z	T+2Z	T+3Z + Y	T+3Z + Y	T+2Z
XNOR + XOR	Total	3T+5Z	3T+5Z	3T+5Z	3T+5Z	2T+5Z + Y	2T+5Z + Y	2T+5Z + Y	2T+5Z + Y
F8 = NOR	SRAM_0 = 0 SRAM_1 = 1 SRAM_2 = 1 SRAM_3 = 1	T+2Z	2T+ Z	3Z	3Z	T+3Z+ X	2Z	T	X
F8\= OR	SRAM_0 = 1 SRAM_1 = 0 SRAM_2 = 0 SRAM_3 = 0	T+2Z	3Z	T	T	T+2Z+ X	T+2Z + Y	4Z + Y	3Z+ X
NOR + OR	Total	2T+4Z	2T+4Z	T+3Z	T+3Z	2T+5Z+2X	T+4Z + Y	T+4Z + Y	3Z+2X

the total leakage is equal to $2T + 4Z$ whereas by changing the input combination to A = '0' and B = '0', the total leakage is equal to $T + 3Z$.

In conclusion, applying DRL as the only countermeasure is not enough to conceal the total power consumption of the LUT as it only conceals the dynamic power. It is important to mention that the leakage sum of the complementary functions which have equal numbers of 0s and 1s exhibits no data dependency such as XOR and XNOR gates.

5.3 Techniques to Conceal the Power Consumption

Chapter 4 showed that DRL is a duplication-based circuit technique that conceals the dynamic power consumption. However, other power components can still pose the threat of leaking information. In this section, we are going to present the implementation of the DRL family in the framework of reconfigurable hardware (FPGA), along with the design details of LUTs and switches that are robust to all four of the major power threats mentioned previously. In the design, we have incorporated the features stated earlier, while taking into consideration the increase in area and power consumption.

5.3.1 Data Independent Power Consumption

In the secured logic element proposed, the nMOS Multiplexer is replicated four times with the configuration SRAM cells in a cyclic permutation, as shown in Figure 5.5. The idea behind this replication strategy is to have each of the four SRAM configuration bits propagated to the output of a distinct 2TB-LUT. Thus, since the four configurations (SRAM_0, SRAM_1, SRAM_2, and SRAM_3) are all active at any given time, the 2TB-LUT will exhibit a leakage which is independent of the input data. The dependency of the dynamic power consumption on the processed data is removed by a dual-rail technique, in which the complementary SRAM outputs provide the configuration bits as shown in

Figure 5.5. The circuit is also symmetrical at its inputs so that each of the input data drivers (A, A\, B, and B\) sees an identical load (as the number of driven OFF transistors is the same in each driver of a complementary pair as are the ON transistors with a source at 0 and 1, respectively). As a result, the dynamic power consumption of the proposed logic element is independent of the input data (complete simulation figures will be presented in the next section). Finally, as the numbers of SRAM outputs in 0 and 1 are equal, their leakage is also independent of the data as well as the SRAM configuration. As shown in Figure 5.5, precharge transistors are provided, such that all the internal nodes of the LUT are connected to ground during the precharge phase. This full connectivity technique ensures that the precharge state is well defined, a feature which eliminates the memory effect and ensures that every single node in the circuit comes from a known state. This is similar to [126], although the full connectivity technique was proposed in a different context.

To review, four circuit transformations are proposed to build a secure-by-design FPGA LUT that is robust against both dynamic and static power analysis attacks:

- **2-transistor branches**- to provide matched loads to the LUT input buffers,
- **Replication + cyclically permuted SRAM cells** - to conceal the static power consumption,
- **Dual-rail logic** - to conceal the dynamic power,
- **Precharge transistors** - to remove memory effects.

5.3.2 Results and Discussion

The proposed method of replication ensures that where the output is selected based on the original input, all possible input combinations are applied in parallel. Along with the use

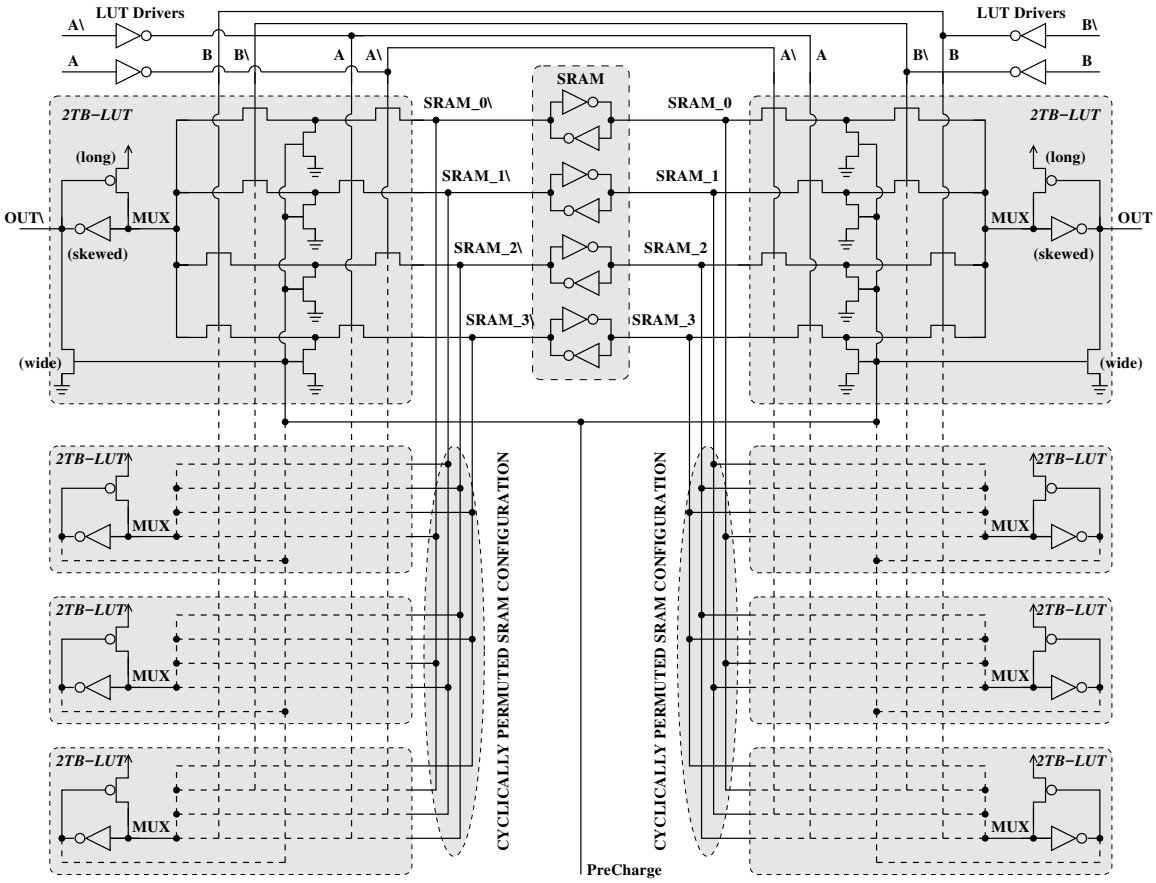


Figure 5.5: Logic element with replication and dual-output.

of DRL, the Hamming-weight and Hamming-distance of the secured LUT will show no dependency on the processed data.

An example of calculating the circuit leakage in the evaluation phase is provided in Figure 5.6. It is assumed that the configuration is SRAM_0 = '1', SRAM_1 = '0', SRAM_2 = '0', and SRAM_3 = '0' corresponds to an OR function. For an input combination A = '0' and B = '0', the SRAM_0 branch is *ON*; therefore, the node MUX = '1'. The leakage components (T, X, Y, Z) were presented in Chapter 3. A voltage drop across a single *OFF* transistor generates a leakage current of T and Z depending on the node value. If the voltage drops from weak '1' (like in the SRAM_1 branch), the leakage current is Z. If the voltage drops from strong '1' (like the one comes from the node MUX), the leakage current is T. A voltage drop across a series connection of two *OFF* transistors generates

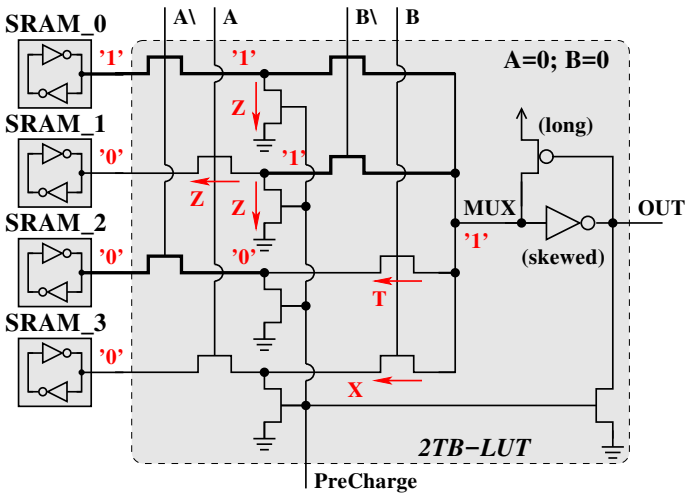


Figure 5.6: 2TB LUT. An example of leakage calculation (A = '0', A\ = '1', B = '0', B\ = '1', SRAM_0 = '1', SRAM_1 = '0', SRAM_2 = '0', SRAM_3 = '0'; thick transistors are *ON*, thin transistors are *OFF*).

a leakage current equal to a fraction X (as in branches SRAM_3 and SRAM_3\), if two ends are connected to ground while the third end is connected to V_{DD}. In the case of two ends being connected to V_{DD} and one end being connected to ground, the generated current leakage is a fraction Y. It is apparent that the leakage of the Original leakage is equal to $T+3Z+X$.

The complete leakage figures are provided in Table 5.2 which clearly shows that the leakage of the LUT and its replica is equal to $4T+16Z+4X+2Y$ for any input combination of an AND gate. It is thus confirmed that our proposed 2TB LUT exhibits a leakage that is independent of data, but not of configuration (for example, AND + NAND has a leakage equal to $4T+16Z+4X+2Y$, whereas XOR+XNOR has a leakage equal to $6T+18Z+4X+2Y$).

To confirm our leakage calculation in simulation, a benchmark including five secured 2-input LUTs was configured to implement all relevant 2-input logic. The schematics were edited with Cadences Virtuoso, and simulations were performed with Cadences Spectre.

The simulation power consumption figures for the 90nm and 65nm design kits from ST Microelectronics are presented in Table 5.3.

Table 5.2: Average power consumption figures for the secured LUT.

Normalized Leakage					
Logic Gates	Circuit Element	Inputs (A B)			
		0 0	0 1	1 0	1 1
AND	LUT	3Z+ X	4Z + Y	T+ 2Z + Y	T+ 2Z+ X
	Stub	2T+ 9Z+ X+2Y	T+ 7Z+2X	2T+ 5Z+2X	T+ 6Z+ X+2Y
NAND	LUT	X	T	2Z	T+ 3Z+ X
	Stub	2T+ 4Z+ X	2T+ 5Z+2X+ Y	T+ 7Z+2X+ Y	T+ 5Z+ X
AND + NAND	LUTs + Stubs	4T+16Z+4X+2Y	4T+16Z+4X+2Y	4T+16Z+4X+2Y	4T+16Z+4X+2Y
OR	LUT	T+ 3Z+ X	2Z	4Z + Y	3Z+ X
	Stub	T+ 2Z+ X	2T+ 3Z+ X	T+ 5Z+2X	2T+ 5Z+ X
NOR	LUT	T+ 2Z+ X	T+ 2Z + Y	T	X
	Stub	T+ 9Z+ X+2Y	T+ 9Z+2X+ Y	2T+ 7Z+2X+ Y	2T+ 8Z+ X+2Y
OR + NOR	LUTs + Stubs	4T+16Z+4X+2Y	4T+16Z+4X+2Y	4T+16Z+4X+2Y	4T+16Z+4X+2Y
XOR	LUT	T+ 2Z	T+ 3Z + Y	T+ 3Z + Y	T+ 2Z
	Stub	2T+ 7Z+2X+ Y	2T+ 6Z+2X	2T+ 6Z+2X	2T+ 7Z+2X+ Y
XNOR	LUT	T+ 3Z + Y	T+ 2Z	T+ 2Z	T+ 3Z + Y
	Stub	2T+ 6Z+2X	2T+ 7Z+2X+ Y	2T+ 7Z+2X+ Y	2T+ 6Z+2X
XOR + XNOR	LUTs + Stubs	6T+18Z+4X+2Y	6T+18Z+4X+2Y	6T+18Z+4X+2Y	6T+18Z+4X+2Y
F=A	LUT	T + X	T + X	4Z+ X	4Z+ X
	Stub	2T+ 4Z+3X	2T+ 4Z+3X	T+ 8Z+3X	T+ 8Z+3X
F= A\	LUT	4Z+ X	4Z+ X	T + X	T + X
	Stub	T+ 8Z+3X	T+ 8Z+3X	2T+ 4Z+3X	2T+ 4Z+3X
(F=A) + (F=A\)	LUTs + Stubs	4T+16Z+8X	4T+16Z+8X	4T+16Z+8X	4T+16Z+8X
F=1	LUT	3Z + Y	3Z + Y	3Z + Y	3Z + Y
	Stub	9Z +3Y	9Z +3Y	9Z +3Y	9Z +3Y
F= 0	LUT	0	0	0	0
	Stub	0	0	0	0
(F= 1) + (F= 0)	LUTs + Stubs	12Z +4Y	12Z +4Y	12Z +4Y	12Z +4Y

Table 5.3 shows that for all functions and in both technologies, simulation figures show no dependency on processed data. Therefore, we can state that the static and dynamic power are both concealed. However, the power consumption still depends on SRAM configuration. In the 65nm technology node, for example, the leakage and the total power consumptions are 1.11 μW and 5.52 μW for the AND/NAND gate, respectively, whereas the power figures are equal to 1.14 μW and 5.53 μW for XOR/XNOR gate; this means a 3% systematic difference in leakage and 0.25% in total power.

Table 5.3: Average power consumption figure for secured LUT.

Node	Test Circuit	Inputs (A B)	Leakage Power	Total Power
90nm	AND/NAND (1 LUT)	1 1	314 nW	7.156 μW
		0 1		
		1 0		
		0 0		
	OR/NOR (1 LUT)	1 1	314 nW	7.156 μW
		0 1		
		1 0		
		0 0		
	XOR/XNOR (1 LUT)	1 1	320.9 nW	7.136 μW
		0 1		
		1 0		
		0 0		
	A/A\ (1 LUT)	1 1	326.2 nW	7.019 μW
		0 1		
		1 0		
		0 0		
	0s/1s (1 LUT)	1 1	294.9 nW	7.00 μW
		0 1		
		1 0		
		0 0		
65nm	AND/NAND (1 LUT)	1 1	1.11 μW	5.52 μW
		0 1		
		1 0		
		0 0		
	OR/NOR (1 LUT)	1 1	1.11 μW	5.52 μW
		0 1		
		1 0		
		0 0		
	XOR/XNOR (1 LUT)	1 1	1.14 μW	5.53 μW
		0 1		
		1 0		
		0 0		
	A/A\ (1 LUT)	1 1	1.12 μW	5.53 μW
		0 1		
		1 0		
		0 0		
	0s/1s (1 LUT)	1 1	1.06 μW	5.49 μW
		0 1		
		1 0		
		0 0		

It is also necessary to state that, since the LUT transistors are all OFF during the precharge phase, the evaluation is triggered only when both A and B signals become active. Unlike the classical LUT, the complementary inputs are driven by separate drivers; therefore, no early evaluation can occur during the precharge or evaluation phases. Moreover, since we use a global precharge signal and the logic is monotonic, the proposed circuits are glitch free.

Table 5.4: Estimated silicon area for LUT.

Component	Area (× minimum size)			Area Ratio
	Standard	Overhead	Total	
LUT				
Pass Transistors	6	58	64	-
SRAM	32	32	64	-
Precharge middle	0	8	8	-
Precharge out	0	16	16	-
Drivers	16	0	16	-
Input Inverters	8	-8	0	-
Keeper (L=5)	5	35	40	-
LRB (skewed)	4	28	32	-
TOTAL LUT	71	169	240	3.4×

With respect to hardware overhead, the dual-rail logic conceptually doubles the silicon area. However, it was reported that the DRL mapped onto Virtex-II FPGA requires at least 6× more area than standard logic [124]. A further refinement of the dual-rail logic, the WDDL, showed an 11× area overhead versus the standard logic. In the circuit we propose, since the SRAM latch already provides the complementary output, the SRAM is shared between the complementary (left-right) LUT parts, (Figure 5.5). The SRAM is also shared between the replicas – although its size needs to be slightly increased to support the additional loading. In addition, the inverters needed to generate the complementary inputs for LUTs and switches in standard FPGAs are no longer needed, as dual-rail logic intrinsically provides complementary signals. This is an indication that a large hardware overhead is not needed in our secured circuitry.

Table 5.4 shows the estimated silicon areas normalized to the technological minimum size for the secured 2-input LUT. The standard MUX requires six transistors [50, 92], whereas eight transistors are needed for each MUX replica which translates into 88 = 64 transistors per secured LUT. The SRAM area is increased from 32 to 64 to support the additional loading. The other sizes are in line with commercial FPGAs [50]. Therefore, we can estimate that the secured LUT is 3.4 larger than the standard LUT. Since the LUTs occupy only a low fraction of the total FPGA silicon area [47], it can be concluded that this area overhead is minimal compared to the prior art. As a result, the proposed circuit techniques preserve reconfigurability while achieving quadrable robustness against dynamic power, static power, early evaluation, and glitch-based attacks. The overhead silicon area that secures the circuit is a good trade-off for the achieved robustness.

It must be mentioned that the robust circuit is only a 2-input LUT, while the commercial FPGAs nowadays have at least 6-input LUTs. As presented in Chapter 2, multiple input LUT consists of a number of 2-input LUTs that are connected in a tree fashion. Therefore, it is difficult to build a secure 6-input LUT without making the dynamic and static power consumption of a 2-input LUT independent of both the input data and SRAM configurations. As can be observed, only the first stage multiplexers of the 6-input LUT have a static configuration through SRAMs. The later stage multiplexers have dynamic configuration that depends on the processed inputs of the previous stage. This aspect is discussed in the next section.

5.4 Data and Function Independent Power Consumption

The achieved level of security against power analysis attacks for 2-input LUTs shown in the previous section is not sufficient to build secured multiple-input LUTs. To illustrate, let us configure 4-input LUT to implement a 4-input AND gate. The 4-input LUT has four 2-input

LUTs in the first stage and one 2-input LUT in the second stage. The first stage LUTs are directly connected to the SRAM cells meaning that they have a static configuration. From top to bottom, the first three 2-input LUT's SRAM cells are configured with '0s', while the SRAM configurations of the bottom 2-input LUT are SRAM_0 = '0', SRAM_1 = '0', SRAM_2 = '0', and SRAM_3 = '1'. Using Table 5.2, the first stage of the 4-input LUT has a total leakage equal to $3 \times (12Z + 4Y) + (4T + 16Z + 4X + 2Y)$. The second stage is a dynamic configuration dependent on the processed data of the first stage. If the input combination are A = '1' and B = '1', the second stage LUT adds to the total leakage a value of $(4T + 16Z + 4X + 2Y)$. Hence, the total leakage is $3 \times (12Z + 4Y) + 2 \times (4T + 16Z + 4X + 2Y)$. In the case of other input combination, the total leakage is $4 \times (12Z + 4Y) + (4T + 16Z + 4X + 2Y)$. Implementing other arbitrary functions leads to a strong correlation between the leakage figures and the processed data.

To build a secure multiple-input LUT, it is essential to modify the 2-input LUT to remove the power consumption dependency on the SRAM configuration. It is difficult to satisfy a double leakage independency of data and function using Standard LUT for two reasons. First, the structure of the classical LUT is not symmetrical. Second, there is a leakage current flowing through the transistors in the first level of the multiplexer that are driven by two mutually exclusive inputs (A, A\). Such a leakage depends directly on the SRAM configuration –a feature that hardens the concealment of the power consumption.

By looking at Table 5.1, each differential SRAM configurations has three possible Hamming-weight classes. In the first class, the direct SRAM configuration has zero Hamming-weight, and the complementary one has a Hamming-weight equal to 4 and vica-versa. In the second class, the direct and the complementary LUTs' have Hamming-weights equal to 1 and 3 and vica-versa. In the third class, where both complementary LUTs Hamming-weight are equal to '2', the classical LUT still depends on the specific SRAM configuration. Hence, it confirms the difficulty of having data and

function independent of power consumption with the standard LUT. Unlike the classical LUT, the 2TB exhibits symmetry. Henceforth, the 2TB LUT will be used in the subsequent design. In addition, the function which has an equal Hamming-weight shows an increase in the symmetry that reduces the need for replicas.

To achieve both data and configuration-independent leakage, an extra 4-branch Stub is proposed in which the avoidance of any collision with the original LUT is achieved by all Stub branches being *OFF* for any input combination and configuration as shown in Figure 5.7. Specifically, two pass transistors have their gates hardwired to ground, three pass transistors have their gates driven by the input signal A, and the remaining three pass transistors have their gates driven by the complementary input signal A\. Noticeably, the pass transistors selection pattern in the stub resembles that of the original LUT, a feature that provides symmetry in the leakage consumption. The stub branches are driven by the complementary SRAM outputs, so that every LUT–Stub pair sees exactly four '0' and four '1' values; that is, all possible SRAM configuration classes have an equal Hamming weight of 4.

An example of calculating the circuit leakage in the evaluation phase is provided in Figure 5.7. It is assumed that the configuration is SRAM_0 = '1', SRAM_1 = '0', SRAM_2 = '0', and SRAM_3 = '0'. For an input combination A = '0' and B = '0', the SRAM_0 branch is *ON*; therefore, the node MUX = '1'. As previously mentioned, a voltage drop across a single *OFF* transistor generates a leakage current of T and Z depending on the node value. If the voltage drops from a weak '1' (as in the SRAM_1 branch), the leakage current is Z. If it drops from a strong '1' (as provided by the node MUX), the leakage current is T. A voltage drop across a series connection of two *OFF* transistors generates a leakage current equal to a fraction X (as in branches SRAM_3 and SRAM_3\) if the two ends are connected to ground, while the other end is connected to V_{DD}. In the case of two ends connected to V_{DD} where one is connected to ground, the

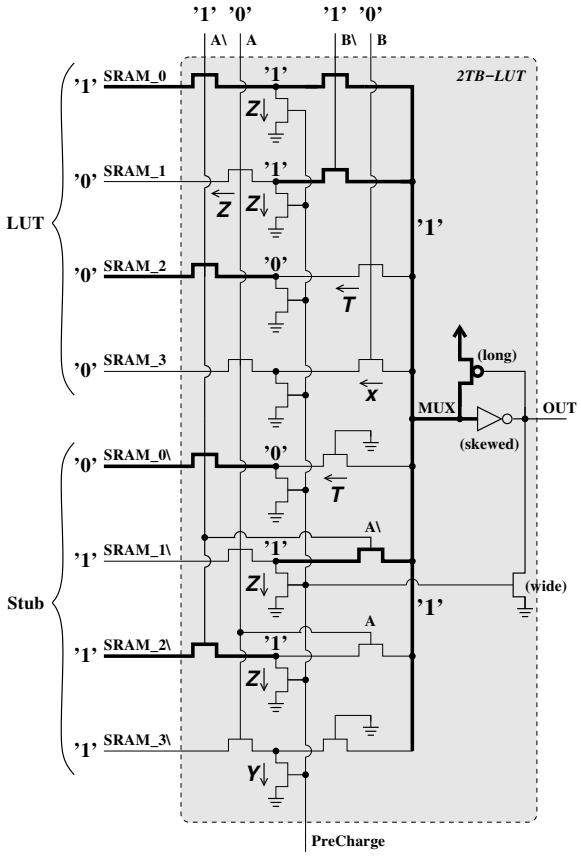


Figure 5.7: New LUT. An example of leakage calculation (A = '0', A\ = '1', B = '0', B\ = '1', SRAM_0 = '1', SRAM_1 = '0', SRAM_2 = '0', SRAM_3 = '0'; thick transistors are *ON*, thin transistors are *OFF*).

generated current leakage is a fraction Y. It is clear that the LUT and Stub leakages are equal to $T+3Z+X$ and $T+2Z+Y$, respectively. The complete leakage figures of the eight branched LUT and its replica are provided in Table 5.5.

5.4.1 Result and Discussion

As shown in Table 5.5, the total leakage is equal to $12T+36Z+8X+4Y$ for any input combination and configuration (or function). Thus, the LUT configuration dependency presented in Table 5.2 is eliminated.

The eight-branched LUT is not only leakage independent of data and function, but it is also the kernel of building multiple-input LUT. As mentioned in Chapter 2, the 4-input

Table 5.5: Average power consumption figures for the secured LUT.

Normalized Leakage					
Logic Gates	Circuit Element	Inputs (A B)			
		0 0	0 1	1 0	1 1
AND	LUT	2T+ 5Z+ X+ Y	2T+ 6Z +2Y	2T+ 5Z+ X+ Y	T+ 4Z+ X
	Stub	5T+14Z+3X+2Y	5T+13Z+4X+ Y	5T+14Z+3X+2Y	6T+15Z+3X+3Y
NAND	LUT	T+ 4Z+ X	2T+ 4Z	T+ 4Z+ X	2T+ 5Z+ X+ Y
	Stub	4T+13Z+3X+ Y	3T+13Z+4X+ Y	4T+13Z+3X+ Y	3T+12Z+3X
AND + NAND	LUTs + Stubs	12T+36Z+8X+4Y	12T+36Z+8X+4Y	12T+36Z+8X+4Y	12T+36Z+8X+4Y
OR	LUT	2T+ 5Z+ X+ Y	T+ 4Z+ X	2T+ 4Z	T+ 4Z+ X
	Stub	3T+12Z+3X	4T+13Z+3X+ Y	3T+13Z+4X+ Y	4T+13Z+3X+ Y
NOR	LUT	T+ 4Z+ X	2T+ 5Z+ X+ Y	2T+ 6Z +2Y	2T+ 5Z+ X+ Y
	Stub	6T+15Z+3X+3Y	5T+14Z+3X+2Y	5T+13Z+4X+ Y	5T+14Z+3X+2Y
OR + NOR	LUTs + Stubs	12T+36Z+8X+4Y	12T+36Z+8X+4Y	12T+36Z+8X+4Y	12T+36Z+8X+4Y
XOR	LUT	2T+ 5Z+ X+ Y	2T+ 4Z	2T+ 4Z	2T+ 5Z+ X+ Y
	Stub	4T+13Z+3X+ Y	4T+14Z+4X+2Y	4T+14Z+4X+2Y	4T+13Z+3X+ Y
XNOR	LUT	T+ 4Z+ X	2T+ 6Z +2Y	2T+ 6Z +2Y	T+ 4Z+ X
	Stub	5T+14Z+3X+2Y	4T+12Z+4X	4T+12Z+4X	5T+14Z+3X+2Y
XOR + XNOR	LUTs + Stubs	12T+36Z+8X+4Y	12T+36Z+8X+4Y	12T+36Z+8X+4Y	12T+36Z+8X+4Y
F=A	LUT	T+ 4Z+ X	T+ 4Z+ X	2T+ 5Z+ X+ Y	2T+ 5Z+ X+ Y
	Stub	5T+14Z+3X+2Y	5T+14Z+3X+2Y	4T+13Z+3X+ Y	4T+13Z+3X+ Y
F= A\	LUT	2T+ 5Z+ X+ Y	2T+ 5Z+ X+ Y	T+ 4Z+ X	T+ 4Z+ X
	Stub	4T+13Z+3X+ Y	4T+13Z+3X+ Y	5T+14Z+3X+2Y	5T+14Z+3X+2Y
(F=A) + (F=A\)	LUTs + Stubs	12T+36Z+8X+4Y	12T+36Z+8X+4Y	12T+36Z+8X+4Y	12T+36Z+8X+4Y
F=1	LUT	T+ 4Z+ X	T+ 4Z+ X	T+ 4Z+ X	T+ 4Z+ X
	Stub	3T+12Z+3X	3T+12Z+3X	3T+12Z+3X	3T+12Z+3X
F= 0	LUT	2T+ 5Z+ X+ Y	2T+ 5Z+ X+ Y	2T+ 5Z+ X+ Y	2T+ 5Z+ X+ Y
	Stub	6T+15Z+3X+3Y	2T+ 5Z+ X+ Y	2T+ 5Z+ X+ Y	2T+ 5Z+ X+ Y
(F= 1) + (F= 0)	LUTs + Stubs	12T+36Z+8X+4Y	12T+36Z+8X+4Y	12T+36Z+8X+4Y	12T+36Z+8X+4Y

LUT is built with five 2-input LUTs connected in a tree topology. In this technique, a dual-rail logic has been used; therefore, a ten 2-input 8-branch LUTs were required to build a 4-input LUT. Hence, the total leakage is $10 \times (12T + 36Z + 8X + 4)$ independent of the data and function being processed. The total leakage depends only on the number of LUT inputs.

Due to the extended symmetry introduced by the stub, the dynamic power consumption is also independent of the configuration. All these assertions have been validated by simulation. The schematics were edited with Cadence's Virtuoso, and simulations were performed with Cadence's Spectre. The determination was made that, for example, the

4-input LUT has a leakage of 868.5 nW and a total (dynamic + static) power of 36.25 μW in the IBM 130nm technology.

Conceptually, the dual-rail logic doubles the silicon area. Since the SRAM latch already provides the complementary output, the SRAM is shared between the complementary LUT parts. The SRAM is also shared between the cyclically permuted replicas. The inverters needed to generate the complementary LUT inputs in standard FPGAs are no longer needed as dual-rail logic provides the complementary signals. Table 7.2 shows the estimated silicon areas normalized to the technological minimum size for the secured 4-input and 6-input LUTs. The standard 4:2:1 MUX requires six transistors [50], whereas 16 transistors (8 for the basic LUT and 8 for the stub) are needed for each MUX replica; this translates into $16 \times 8 = 128$ transistors for each secured 2-input LUT, $128 \times (4+1) = 640$ transistors for each secured 4-input LUT, and $128 \times (5 \times 4 + 1) = 2688$ transistors for each secured 6-input LUT. The SRAM area is doubled in order to support the additional loading. The other sizes are in line with commercial FPGAs [50].

The circuit techniques for dynamic power concealment incur a silicon area overhead of 7.0$\times$ and 7.9$\times$, respectively. It was reported in [124] that the dual-rail logic mapped onto Virtex-II FPGA required at least 6$\times$ more area than standard logic while it remains vulnerable to power attacks based on static, glitches and early evaluation. Further refinements of the dual-rail logic showed an 11$\times$ area overhead versus the standard logic. It is apparent that our area increase is in line with the prior art, whereas a quadruple robustness against dynamic power, static power, early evaluation, and/or glitch-based attacks is provided. Finally, we would like to mention that LUTs occupy only a low fraction of the total FPGA silicon area [47]. Thus, we can consider this area overhead to be a good trade-off for the achieved security level.

Table 5.6: Estimated silicon areas for LUTs.

LUT type	Component	Area (× minimum size)			Area Ratio
		Standard	Overhead	Total	
4-input LUT	Pass Transistors	30	610	640	-
	SRAM	128	128	256	-
	Precharge middle	0	320	320	-
	Precharge out	0	80	80	-
	Drivers	32	32	64	-
	Input Inverters	16	-16	0	-
	Keeper (L=3)	10	110	120	-
	LRB (skewed)	20	140	160	-
	TOTAL LUT	236	1404	1650	7.0×
6-input LUT	Pass Transistors	126	2562	2688	-
	SRAM	512	512	1024	-
	Precharge middle	0	1344	1344	-
	Precharge out	0	336	336	-
	Drivers	48	48	96	-
	Input Inverters	24	-24	0	-
	Keeper (L=3)	42	462	504	-
	LRB (skewed)	84	588	672	-
	TOTAL LUT	836	5828	6664	7.9×

5.5 Switch Box

Switch Boxes are essential components in forming larger circuits. They surround the CLBs to connect their inputs and outputs as well as establish connections between different wire segments, as shown in Figure 2.1. Programmable routing boxes including switch boxes, connection boxes, local routing MUXes are built in the same way. They use only 2-level multiplexers followed by the level-restoring buffer, see Figure 2.9. They are built from the same components as the LUT, but they are structured differently. Unlike the LUT, the switch box's inputs come through the drain terminals in each Multiplexer branch. Moreover, SRAMs control the switch box's transistors gates. As shown earlier, the LUT output does not only depend on its inputs, but also depends on the SRAM configuration. Hence, the proposed secure LUT techniques will not work for the switch box. Therefore,

a new 4:1 switch box that can overcome the difficulties of balancing the LUT's output and difficulties of balancing the routing is proposed in Figure5.8.

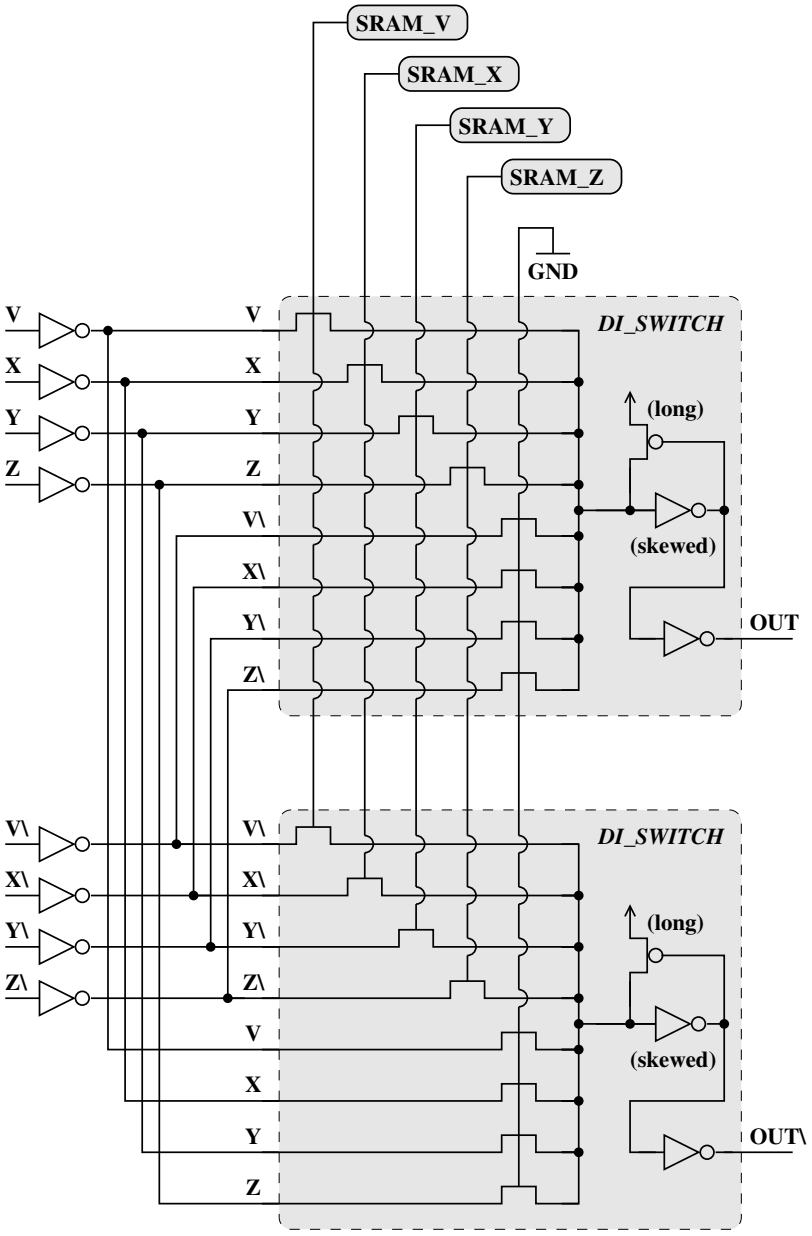


Figure 5.8: Secured switch box.

The leakage dependence of input data is reduced by duplicating each input where each duplicate is permanently OFF, and is driven with the complementary signal. Thus, when logic 1 is being passed, the total leakage equals four nMOS leakage units, whereas

when logic 0 is being passed, the total leakage equals four nMOS leakage units and one pMOS leakage unit due to the Keeper. The dynamic power consumption is concealed by a dual-rail technique. This also has the beneficial effect of completely removing the leakage dependence on the input data as the total leakage is always equal to eight nMOS and one pMOS leakage units. Similar to the secured LUT, each input signal driver sees a data-independent load equal to one source of an ON transistor plus one source of an OFF transistor if the signal passes through, or two OFF transistor sources if the signal does not pass through.

5.5.1 Result and Discussion

The proposed switch box shows a perfect power consumption concealment. In addition, it boosts the LUT to balance its outputs in order to level the power consumption. The proposed switch box was edited with Cadences Virtuoso, and simulations were performed with Cadences Spectre. The simulation power consumption figures for the 90nm and 65nm design kits from ST Microelectronics are presented in Table 5.7. Clearly, passing '1' or '0' exhibits no difference in the static power or in the total power.

Table 5.7: Average power consumption figures for secured switch.

Node	Test Circuit	Inputs (A B)	Leakage Power	Total Power
90nm	Switch (4 inputs)	Passing '1'	170.3 nW	4.41 μW
		Passing '0'	170.3 nW	4.41 μW
65nm	Switch (4 inputs)	Passing '1'	975.9 nW	4.44 μW
		Passing '0'	975.9 nW	4.44 μW

By sharing the SRAM in the secured switch box, the area overhead has been reduced. In Table 5.8, the secured switch is 1.5 larger than its standard counterpart. This overhead area is considered to be a good trade off in terms of the achieved robustness against power attacks.

Table 5.8: Estimated silicon area for switch.

Component	Area (× minimum size)			Area Ratio
	Standard	Overhead	Total	
Pass Transistors	4	12	16	-
SRAM	16	0	16	-
Drivers	16	0	16	-
Keeper (L=5)	5	5	10	-
LRB (skewed)	4	4	8	-
TOTAL Switch	45	21	66	1.5×

5.6 Conclusion

To summarize, we have proposed a 4-branch 2-input LUT which has been replicated four times with the configuration SRAM cells organized in a cyclic permutation. These techniques ensure robustness to all power analysis attacks namely dynamic, static, glitches, and early evaluation. This secure 2-input LUT is robust with low area overhead. However, it is limited to 2-input LUTs that means building multiple-input LUTs from the proposed 2-input LUT will leak information. To overcome this limitation, a change was made to the 4-branch 2-input LUT to 8 branches, which equalizes the Hamming-weight of the processed function; the same method was applied in replicating the MUX and organizing the SRAM cells. This method maintains robustness to all power attacks. Moreover, multiple-input LUTs can be built with 8-branch LUT in tree toplogy as discribed in 5.4 [7]. 8-branch LUTs with 4× replication techniques introduced significant area overhead, but that area overhead is in line with prior-art. Moreover, it may become susceptible to attacks based on early evaluation. In the next chapter, the goal is to significantly reduce the cost in term of area and consequently the power consumption. In Chapter 7, glitches and early evaluation attacks will be addressed.

Chapter 6

Technique to Secure LUTs with Reduced Hardware Overhead

The secured-by-design LUT and switch box described in the previous chapter are based on replication. These circuits are robust to attacks based on dynamic and static power consumption. However, this robustness comes at the expense of a large area overhead. Achieving the same level of robustness with a lower area overhead is highly desirable, and the method for enabling that result is explained in this chapter.

6.1 Introduction

In the commercial LUTs, the complementary inputs are generated through local hardwired inverters. Since this feature does not allow the implementation of a precharge phase for each separate input (where both the direct and complementary signals would have equal logic values), it also makes the use of dual-rail encoding difficult. Although it is possible to use two LUT inputs to emulate a single dual-rail encoded signal, that will double the silicon area, as mentioned in the previous chapters. In our initial secured LUT [6], we eliminate these local inverters and encode each and every signal in dual-rail. With the penalty of

using two wires and two drivers per signal, the intrinsic one-inverter delay between pair signals and, therefore, all short-circuit currents within the MUX are eliminated. In this way, attacks based on the early evaluation of signals are much more difficult to mount.

As described in Chapter 5, in order to conceal the leakage dependence on processed data, we proposed to implement the nMOS multiplexer with two-transistor branches and replicate the resulting multiplexer four times with the configuration SRAM cells organized in a cyclic permutation. Furthermore, we also proposed to add a stub of four extra branches to each nMOS multiplexer replica in order to conceal the leakage dependence on the configured function [7]. This enables building secured LUTs with more than two inputs. The hardware overhead of this resulting secured LUT is significant (on the order of 8x). In this chapter, design techniques are proposed to eliminate the replication needed for concealing the dynamic and static power consumption.

6.2 Technique to reduce the area overhead

It should be recalled that in our data- and configuration-secured 2-input LUT [7], two pass transistors in the Stub have their gates configured to ground, whereas the other pass transistors have their gates driven by the paired inputs (A, A\). To avoid the $3\times$ replication of this initial work, the right transistor in each branch (including that with a 'grounded' gate) is to be driven by a signal originating in the additional circuit as shown in Figure 6.1. This circuit generates four Stub control signals (IN_B5, IN_B6, IN_B7 and IN_B8), which resemble the right-gate signals in the Original multiplexer in the sense that the Stub branches are always *OFF* in order to prevent any collision between the Original multiplexer and the Stub. Since the selection pattern of the pass transistors in the Stub resembles that of the Original multiplexer, the symmetry of the leakage power consumption is ensured. As portrayed in the initial work, the Stub branches are still driven by the complementary

SRAM outputs, so that every LUT–Stub pair sees exactly four ’0’ and four ’1’ values at its source electrodes. Therefore, all functions will have an equal Hamming-weight to 4. It is important to notice that the new circuit shown in Figure 6.1 still uses only LUT inputs, thus the LUT architecture is maintained.

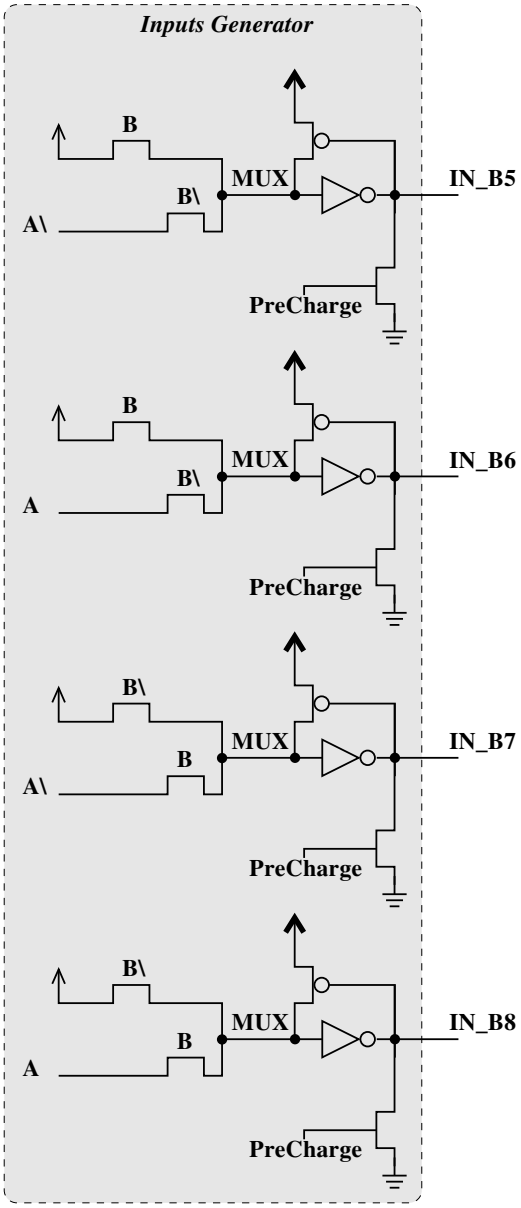


Figure 6.1: Stub control signal generator.

To provide more details on the operation of the Stub Control Signal Generator we have introduced (Fig. 6.1), we mention that the implemented logic functions are:

$$\text{IN\_B5} = \overline{\text{A}\backslash + \text{B}}$$

$$\text{IN\_B6} = \overline{\text{A} + \text{B}}$$

$$\text{IN\_B7} = \overline{\text{A}\backslash + \text{B}\backslash}$$

$$\text{IN\_B8} = \overline{\text{A} + \text{B}\backslash}$$

These gates are implemented in pass-transistor logic ensuring data-independent leakage and low area overhead. This is a worthwhile trade-off in terms of silicon area, as both the $3\times$ replication and the need to oversize the SRAM to support this replication in the initial secured LUT are eliminated. The Stub control signals IN_B5, IN_B6, IN_B7 and IN_B8 are connected to the STub as shown in Figure 6.2.

6.2.1 Result and Discussion

An example of calculating the leakage of the improved circuit during the evaluation phase is provided in Figure 6.3. For the input combination A = '0' and B = '0', the SRAM_0 branch is *ON*; therefore, the node MUX in the NAND gate is equal to '1', and in the AND gate is equal to '0'. As in the cases discussed in the previous chapters, a voltage drop across a single *OFF* transistor generates a leakage current of T or Z depending on the node value. If the voltage drops from weak '1' (as in the SRAM_1 branch), the leakage current is Z. If it drops from strong '1', the leakage current is T. A voltage drop across a series connection of two *OFF* transistors generates a leakage current equal to a fraction X (as in branches SRAM_3 and SRAM_3\) if two ends are connected to ground while the other end is connected to V_{DD}. In the case of two ends being connected to V_{DD} and one end being connected to ground, the current leakage is a fraction Y. The total leakage generated in the AND gate by the Original MUX and the Stub is $2T + 5Z + X + Y$, is apparent in Figure 6.3a.

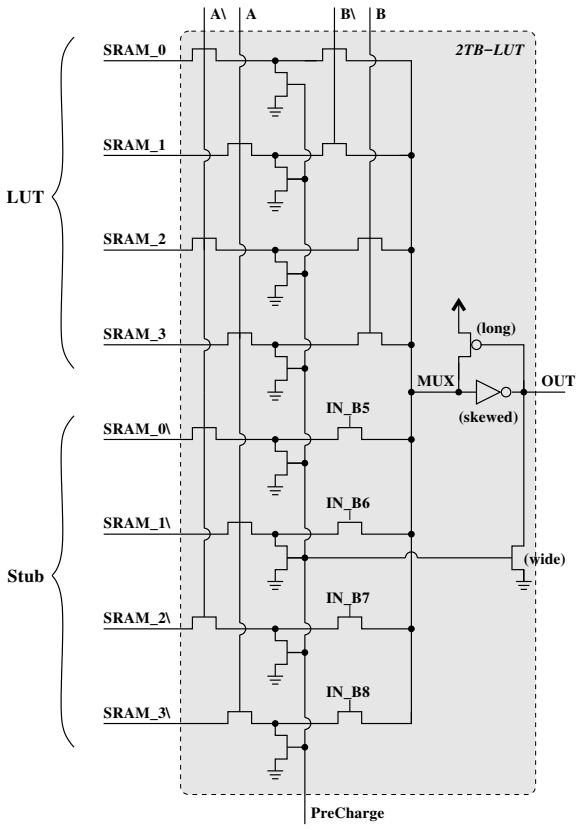


Figure 6.2: New secured 2-input LUT.

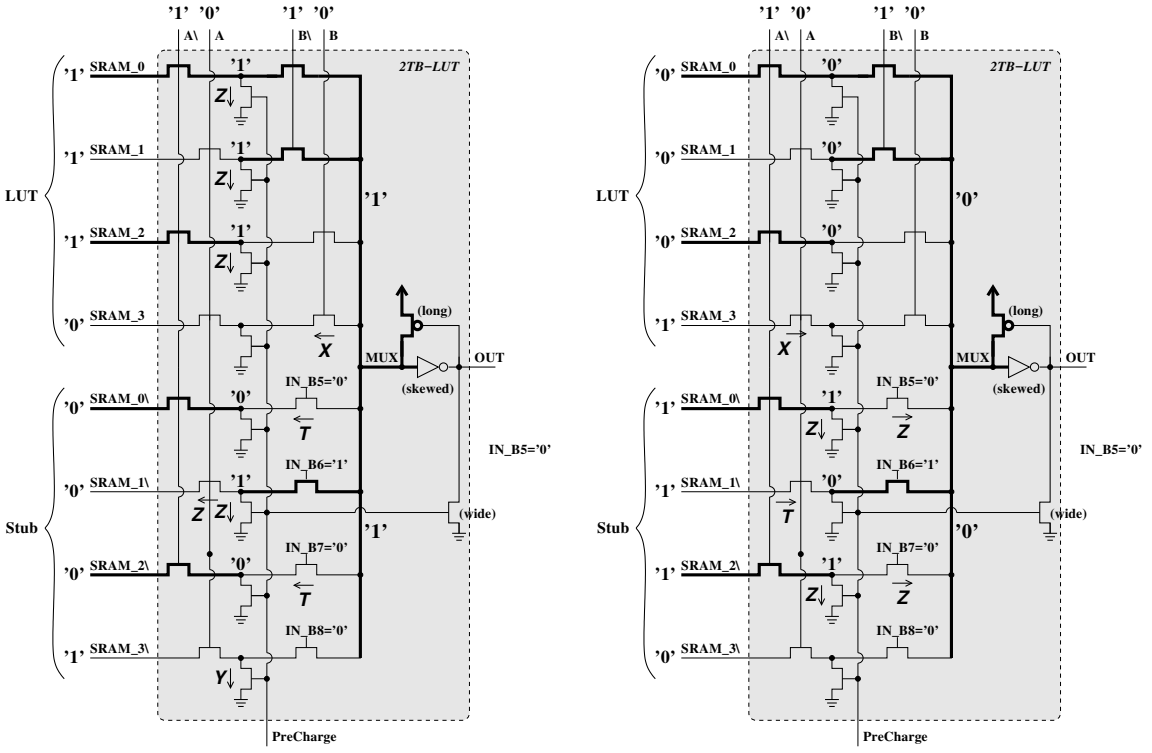
In Figure 6.3b, a leakage of $T + 4Z + X$ is produced by the NAND gate. The grand total of the leakage of the AND and NAND gates is $3T + 9Z + 2X + Y$. This last figure is function independent, as indicated in Table 6.1.

Table 6.1 shows that the total leakage of the dual-rail LUTs is equal to $3T + 9Z + 2X + Y$ for any input combination and configuration (or function). This constitues a 75% reduction in leakage compared to the technique based on replication described in the previous chapter, which was equal to $12T + 36Z + 8X + 4Y$. The penalty of such a reduction in leakage is the need to have synchronized dual-rail signals (A, A\) and (B, B\), to prevent an unsymmetrical leakage of power. Synchronizing the dual-rail LUT inputs is also needed to prevent early evaluation attacks, as will be discussed in the Chapter 7.

It should be emphasized that the Stub Control Signal Generator is also secure. We have

Table 6.1: All possible static leakage of different 2-input gates and processed inputs.

Function	Circuit Element	Eight Branch 2TB-LUT			
	Inputs (A B)	0 0	0 1	1 0	1 1
F1=0	SRAM_0 = 0 SRAM_1 = 0 SRAM_2 = 0 SRAM_3 = 0	T+4Z+ X	T+4Z+ X	T+4Z+ X	T+4Z+ X
F1\= 1	SRAM_0 = 1 SRAM_1 = 1 SRAM_2 = 1 SRAM_3 = 1	2T+5Z+ X+Y	2T+5Z+ X+Y	2T+5Z+ X+Y	2T+5Z+ X+Y
F1 + F1\	Total	3T+9Z+2X+Y	3T+9Z+2X+Y	3T+9Z+2X+Y	3T+9Z+2X+Y
F2= NAND	SRAM_0 = 0 SRAM_1 = 0 SRAM_2 = 0 SRAM_3 = 1	T+4Z+ X	2T+2Z+ X	T+4Z+ X	2T+5Z+ X+Y
F2\=AND	SRAM_0 = 1 SRAM_1 = 1 SRAM_2 = 1 SRAM_3 = 0	2T+5Z+ X+Y	T+7Z+ X+Y	2T+5Z+ X+Y	T+4Z+ X
NAND + AND	Total	3T+9Z+2X+Y	3T+9Z+2X+Y	3T+9Z+2X+Y	3T+9Z+2X+Y
F3	SRAM_0 = 0 SRAM_1 = 0 SRAM_2 = 1 SRAM_3 = 0	T+4Z+ X	T+7Z+ X+Y	T+4Z+ X	T+4Z+ X
F3\	SRAM_0 = 1 SRAM_1 = 1 SRAM_2 = 0 SRAM_3 = 1	2T+5Z+ X+Y	2T+2Z+ X	2T+5Z+ X+Y	2T+5Z+ X+Y
F3+F3\	Total	3T+9Z+2X+Y	3T+9Z+2X+Y	3T+9Z+2X+Y	3T+9Z+2X+Y
F4	SRAM_0 = 0 SRAM_1 = 0 SRAM_2 = 1 SRAM_3 = 1	T+4Z+ X	2T+5Z+ X+Y	T+4Z+ X	2T+5Z+ X+Y
F4\	SRAM_0 = 1 SRAM_1 = 1 SRAM_2 = 0 SRAM_3 = 0	2T+5Z+ X+Y	T+4Z+ X	2T+5Z+ X+Y	T+4Z+ X
F4+F4\	Total	3T+9Z+2X+Y	3T+9Z+2X+Y	3T+9Z+2X+Y	3T+9Z+2X+Y
F5	SRAM_0 = 0 SRAM_1 = 1 SRAM_2 = 0 SRAM_3 = 0	T+4Z+ X	T+4Z+ X	2T+5Z+ X+Y	T+4Z+ X
F5\	SRAM_0 = 1 SRAM_1 = 0 SRAM_2 = 1 SRAM_3 = 1	2T+5Z+ X+Y	2T+5Z+ X+Y	T+4Z+ X	2T+5Z+ X+Y
F5+F5\	Total	3T+9Z+2X+Y	3T+9Z+2X+Y	3T+9Z+2X+Y	3T+9Z+2X+Y
F6	SRAM_0 = 0 SRAM_1 = 1 SRAM_2 = 0 SRAM_3 = 1	T+4Z+ X	2T+2Z+ X	2T+5Z+ X+Y	2T+5Z+ X+Y
F6\	SRAM_0 = 1 SRAM_1 = 0 SRAM_2 = 1 SRAM_3 = 0	2T+5Z+ X+Y	T+7Z+ X+Y	T+4Z+ X	T+4Z+ X
F6+F6\	Total	3T+9Z+2X+Y	3T+9Z+2X+Y	3T+9Z+2X+Y	3T+9Z+2X+Y
F7 = XNOR	SRAM_0 = 0 SRAM_1 = 1 SRAM_2 = 1 SRAM_3 = 0	T+4Z+ X	T+7Z+ X+Y	2T+5Z+ X+Y	T+4Z+ X
F7\= XOR	SRAM_0 = 1 SRAM_1 = 0 SRAM_2 = 0 SRAM_3 = 1	2T+5Z+ X+Y	2T+2Z+ X	T+4Z+ X	2T+5Z+ X+Y
XNOR + XOR	Total	3T+9Z+2X+Y	3T+9Z+2X+Y	3T+9Z+2X+Y	3T+9Z+2X+Y
F8 = NOR	SRAM_0 = 0 SRAM_1 = 1 SRAM_2 = 1 SRAM_3 = 1	T+4Z+ X	2T+5Z+ X+Y	2T+5Z+ X+Y	2T+5Z+ X+Y
F8\= OR	SRAM_0 = 1 SRAM_1 = 0 SRAM_2 = 0 SRAM_3 = 0	2T+5Z+ X+Y	T+4Z+ X	T+4Z+ X	T+4Z+ X
NOR + OR	Total	3T+9Z+2X+Y	3T+9Z+2X+Y	3T+9Z+2X+Y	3T+9Z+2X+Y



(a) An AND Gate using eight Branch 2-input LUT. (b) An NAND Gate using eight Branch 2-input LUT.

Figure 6.3: AND/NAND gates leakage calculation.

four NOR gates built with pass-transistor logic. All possible input combinations are applied in parallel. As such, the dynamic power is independent of the processed data. The total static leakage of this circuit is equal to $2T$ and is also independent of the processed data.

Due to the extended symmetry introduced by the stub, the dynamic power consumption is also independent of the configuration. All these assertions have been verified by simulation. The schematics were edited with Cadence's Virtuoso, and simulations were performed with Cadence's Spectre. We determined that, for example, the 6-input LUT has a leakage of 726.8 nW and a total (dynamic + static) power of 93.64 μW in the IBM 130nm technology.

In terms of area overhead, the dual-rail logic conceptually doubles the silicon area. Since the SRAM latch already provides the complementary output, the SRAM is shared between the complementary LUT parts. Similar to the circuits discussed in Chapter 5,

Table 6.2: Estimated silicon areas for LUTs.

LUT type	Component	Area (× minimum size)			Area Ratio
		Standard	Overhead	Total	
4-input LUT	Pass Transistors	30	130	160	-
	SRAM	128	0	128	-
	Precharge middle	0	80	80	-
	Precharge out	0	20	20	-
	Drivers	32	32	64	-
	Input Inverters	16	-16	0	-
	Keeper (L=3)	10	10	20	-
	LRB (skewed)	20	20	40	-
	IN Generator	0	72	72	-
	TOTAL LUT	236	328	584	2.47×
6-input LUT	Pass Transistors	126	546	672	-
	SRAM	512	0	512	-
	Precharge middle	0	336	336	-
	Precharge out	0	84	84	-
	Drivers	48	48	96	-
	Input Inverters	24	-24	0	-
	Keeper (L=3)	42	42	84	-
	LRB (skewed)	84	84	168	-
	IN Gen	0	108	108	-
	TOTAL LUT	836	1224	2060	2.47×

the inverters needed to generate the complementary LUT inputs in standard FPGAs are no longer needed as dual-rail logic provides the complementary signals. Table 6.2 shows the estimated silicon areas normalized to the technological minimum size for secured 4-input and 6-input LUTs. The standard 4:2:1 MUX requires six transistors [50], whereas 16 transistors (8 for the basic LUT and 8 for the stub) are needed for each MUX. This translates into $16 \times 2 = 32$ transistors per secured 2-input LUT, $32 \times (4+1) = 160$ transistors per secured 4-input LUT, and $32 \times (5 \times 4 + 1) = 672$ transistors per secured 6-input LUT. The other sizes are in line with commercial FPGAs [50].

The circuit techniques for power concealment incur the same silicon area overhead of 2.47× for both 4-input LUTs and 6-input LUTs. This means that a reduction of the area of our prior technique based on replication [7] by 35% and 43%, respectively has

been achieved. It was reported in [124] that the dual-rail logic mapped onto Virtex-II FPGA required at least 6× more area than unsecure standard logic. Further refinements of the dual-rail logic showed an 11× area overhead versus the standard logic. Our secured multiple-Input LUT area overhead shows a remarkable decrease in comparison with the prior art, whereas a quadruple robustness against dynamic power, static power, early evaluation, and/or glitch-based attacks is provided (early evaluation and glitches will be discussed in the next chapter).

6.3 Implementation of the S-Boxes

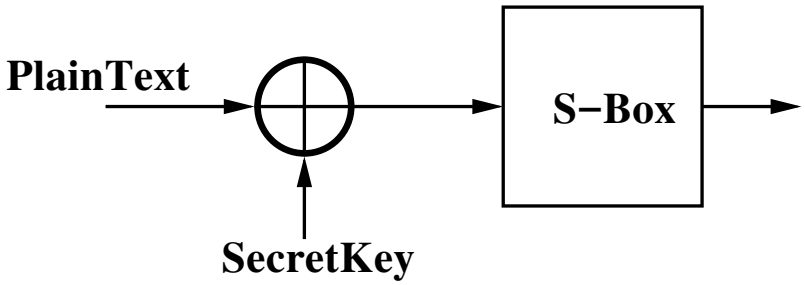


Figure 6.4: Partial of DES circuit.

As a case study, we have implemented a part of the Data Encryption Standard (DES), whose block diagram is shown in Figure 6.4. This part of DES consists of an XOR gate and an S-box which is common in other encryption standards such as AES [80]. The S-box converts its six input bits into four output bits with a non-linear function. This structure is commonly used in the security community to check the robustness against power attacks [123]. In this study, we show the concealment of both static and dynamic power consumption achieved through using our secured LUT.

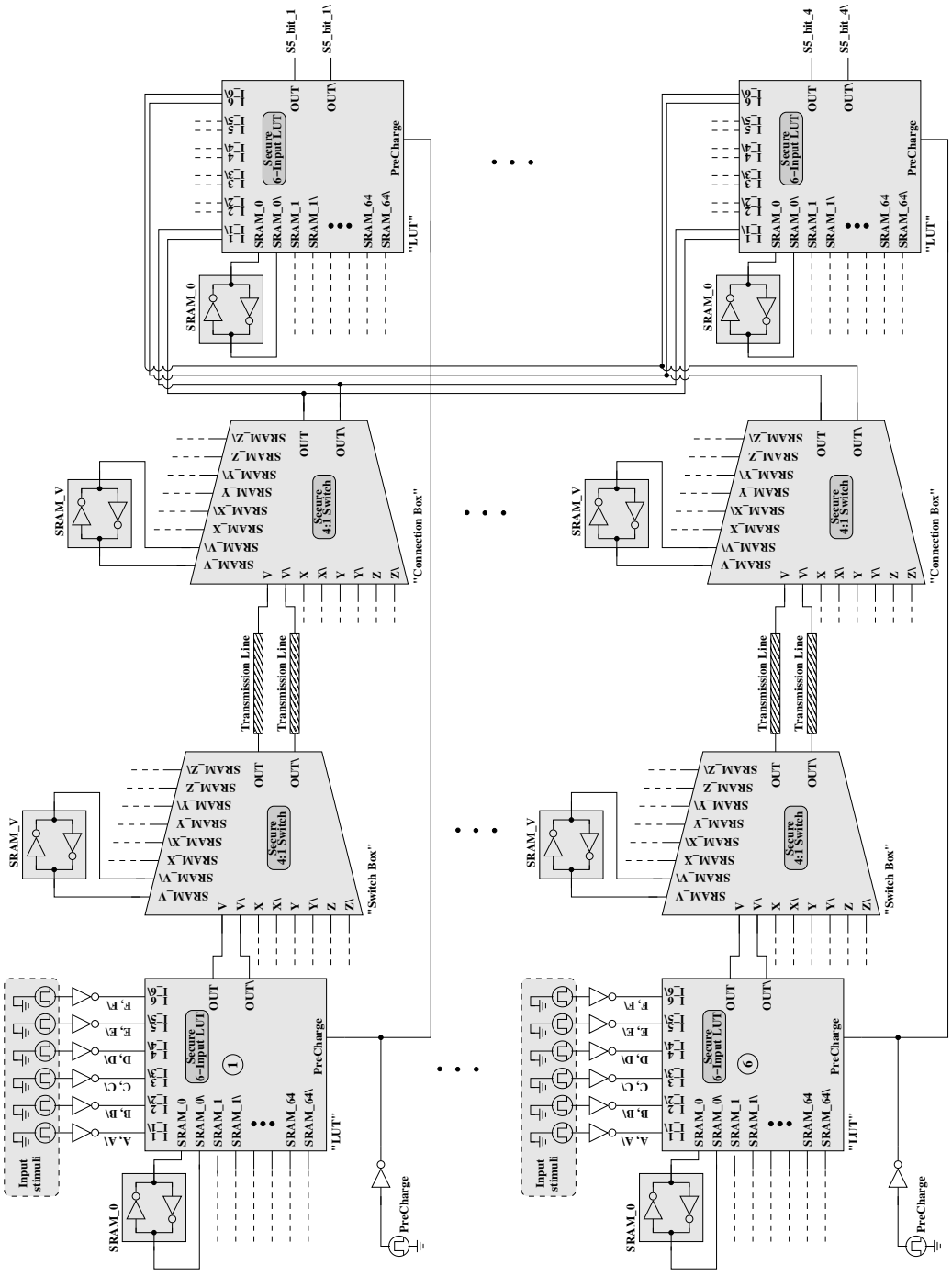


Figure 6.5: Benchmark test.

Figure 6.5 depicts the circuit of the partial DES implementation using the proposed secure LUT and the secure switch box. Four 6-input secured LUTs are needed to build the S-box. Each of the S-box LUT's input is the output of an XOR gate. Therefore, we need six LUTs to complete the implementation. To connect these LUTs, twelve secured switch boxes are required. To proof the concept, all the wires have equal length.

6.3.1 Result and Discussion

The schematics of a partial DES implementation were edited with Cadences Virtuoso, and simulations were performed with Cadences Spectre in the 130nm design kits from IBM. In this implementation, we adapt our techniques to the LUTs and switch boxes presented in Chapter 2.

As a case study, we have tried all 64 possible 6-bit keys and the plaintexts. The simulation shows that there are no change in the static and the dynamic power consumption based on the processed data. The implemented circuit has a leakage of 14.72 μW and a total (dynamic + static) power of 5.05 mW, as evidenced in Figures 6.6 and 6.7. As a result, we can state that the implementation is secure.

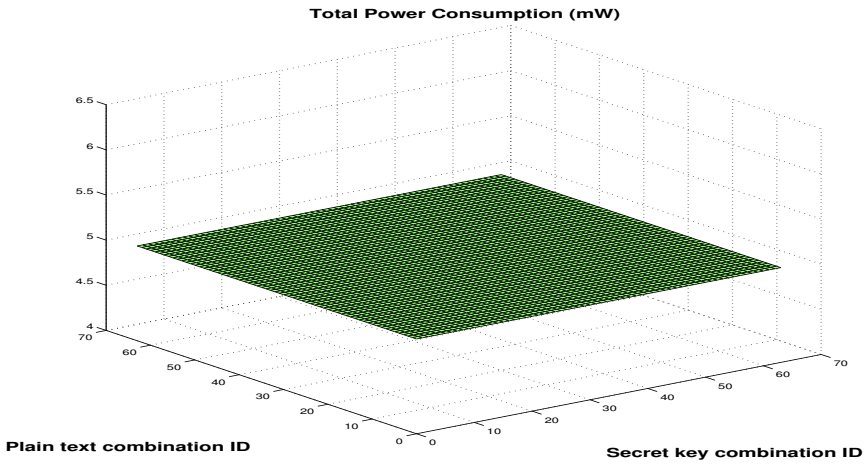


Figure 6.6: Total power consumption over all possible keys and plain texts.

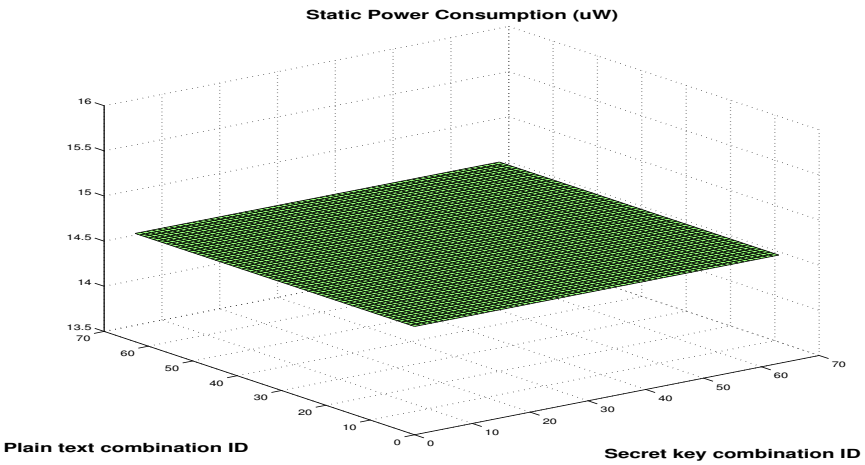


Figure 6.7: Static power consumption over all possible keys and plain texts.

Chapter 7

Eliminating Glitches and Early Evaluation Effects

Circuit-level countermeasures (such as DRL [74]) offer intrinsic security to side-channel attacks. In the previous two chapters, Look-Up Tables and switch-boxes based on Dual-Rail Logic which conceal both dynamic and static power consumption were described. However, DRL ignores any timing effects. Therefore, three possible scenarios may occur during the phase transitions from precharge to the evaluation, and then back to precharge:

- the complementary outputs may produce spurious transitions called ***glitches*** [51]
- the complementary outputs of a digital circuit may switch to its final value even before all the inputs arrive – a phenomenon called ***early evaluation*** [61, 117]
- the complementary loads may not be equal, causing mismatched power consumption and different arrival times between the complementary inputs into the next stage.

Despite the preservation of 100% switching activity in DRL, there might be a difference between complimentary load capacitances due to placement and routing limitations in FPGAs which will leak valuable information. Multiple techniques have been proposed

in the literature to balance the complementary routing, such as Fat Wire [125], divided backend duplication [12], and Isolated WDDL (IWDDL) [77]. These solutions are going to be relied upon if unequal load capacitances problem arise. In this work, we assume matched load capacitances.

Glitches and early evaluation can also leak valuable side-channel information [39, 43, 48, 61, 69, 88, 117, 118]; for example, a successful attack on a DES cryptoprocessor secured with dual-rail logic has been reported [103]. In the context of dual-rail logic, glitches can be eliminated by means of monotonic logic [7, 95], whereas preventing early evaluation effects requires synchronization logic as described later in this chapter.

Chapter 7 shows that our proposed FPGA LUT exhibits quadruple robustness to attacks based on dynamic power, static power, glitches, and early evaluation, while preserving the architecture of commercial FPGAs. The hardware overhead needed to secure the circuit is also smaller than in prior art. To summarize, our contributions to providing robustness to glitches and early evaluation attacks are as follows:

- precharging strategy that maintains monotonic behaviour
- synchronization circuit with reduced hardware overhead, which delays the evaluation of the LUT until after all its inputs turn valid
- methodology to extend the synchronization to other LUTs, so that a complex circuit will not evaluate before all inputs turn valid.

7.1 Countering Glitches

Glitches may occur during the phase transition from precharge to evaluation and back to precharge. With respect to the precharge circuitry presented in Figure7.1, it is apparent that the complementary outputs are precharged to zero to ensure monotonic behaviour. Spurious transitions to '1' may occur at the outputs when the 5^{th} and 6^{th} inputs (the (E, E\)

and (F, F\)) arrive before the other inputs. This phenomenon can leak information about the propagation delays (but not on processed data) in the cryptosystem.

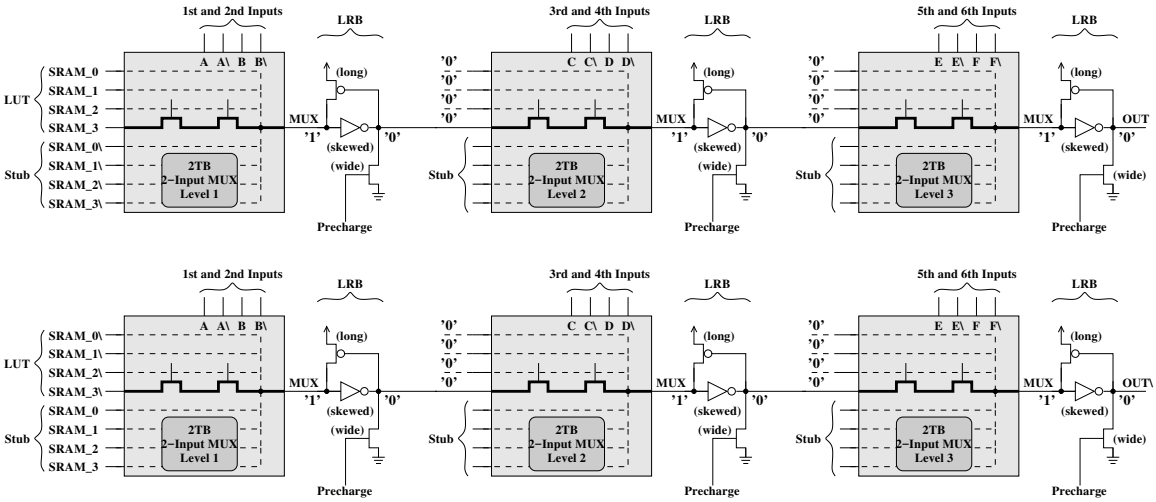


Figure 7.1: Precharging circuitry (the logic values represent the precharge states; the thick branches are *ON* during evaluation).

To elaborate, Figure 7.2 shows the timing diagram for a dual-rail LUT implementing a 6-input AND/NAND gate. In this diagram, we show the presence of a glitch due to different input arrivals. When the precharge signal goes low, the evaluation phase starts. It is assumed that the inputs (E, E\) and (F, F\) arrive before the other inputs allowing the precharge values '0' of the previous stage to propagate to the LUT outputs, switching both of them to '1'. When the other inputs arrive, one of the complementary outputs will switch back to '0' (OUT in our example). Hence, a glitch (highlighted by the red dashed line) appears at this output, indicating different input arrival times. Even though the gates have balanced loads and are thereby robust to dynamic power attacks, valid input signals can arrive at different times due to differences in the depths of the logical paths. The desired behaviour would not exhibit such a glitch.

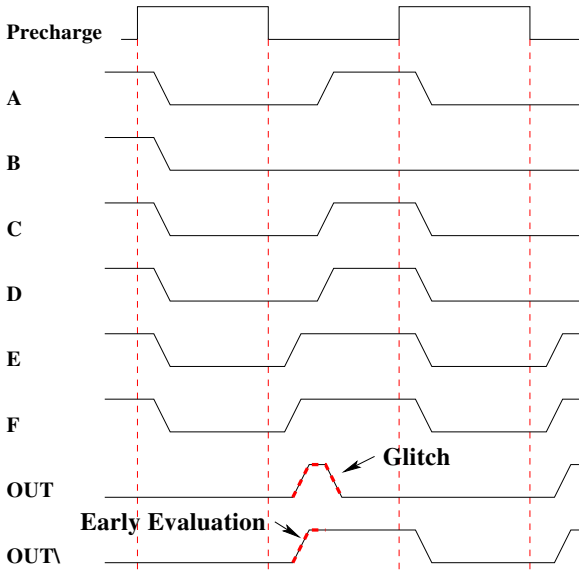


Figure 7.2: Timing diagram of AND/NAND gates.

7.1.1 Circuit technique to prevent glitches

To cure the presence of glitches at LUT complementary outputs, additional inverters (the gray-filled ones) are provided in order to change the polarity of the signal at the outputs of the first and second stages (Figure 7.3). By this method, all the left-transistor sources of the MUX'es in the second and third stages are connected to a a high logic level during precharge guaranteeing that no signal changes or glitches are encountered if any of the (C, C\), (D, D\), (E, E\), or (F, F\) input pairs arrive before (A, A\) or (B, B\) pairs. It should be mentioned that, since the 2-input LUT transistors are all *OFF* during the precharge phase, the evaluation is triggered only when both **A** and **B** signals become active. The same observation can be made for the 4-input and 6-input LUTs. As a result, the proposed circuits are intrinsically glitch free making attacks based on glitches impossible.

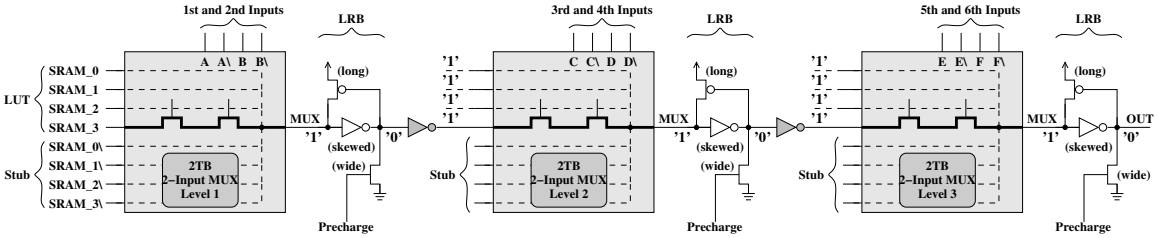


Figure 7.3: Precharging circuitry (the logic values represent the precharge states; the thick branches are *ON* during evaluation).

7.2 Countering Early Evaluation

In Section 5.3.1, we proposed a 2-input LUT which is intrinsically secured against all four types of power attacks. The robustness to early evaluation attacks comes from the pass-transistor logic with two-transistor branch (2TB) that is used to build the pass-transistor multiplexer (MUX) [6]. Extending the number of LUT inputs to four or six, as is common in commercial FPGA architectures today, is not a straightforward task. Level-restoring buffers are required between every two stages of pass transistors [50]. This is equivalent to concatenating 2-input LUTs, as shown in Figure 7.4.

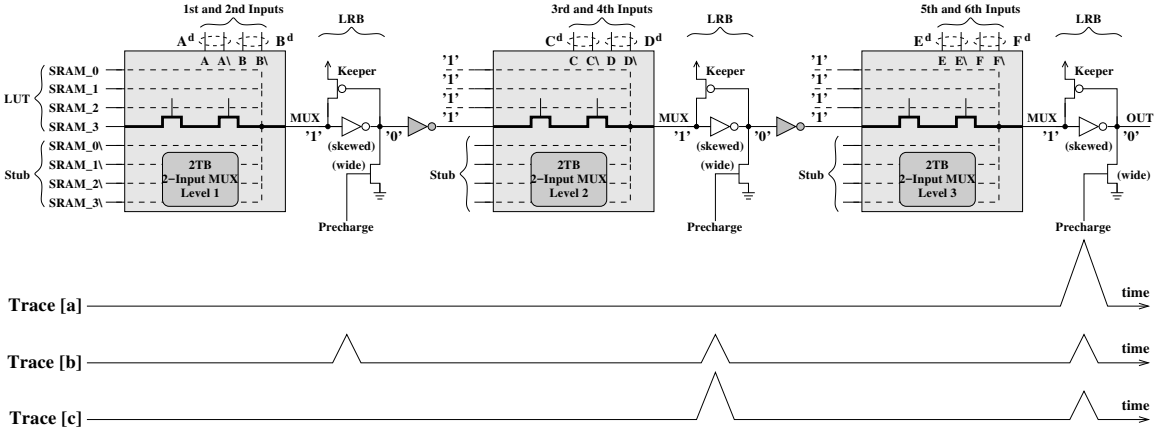


Figure 7.4: Six-input LUT built with two-input LUTs [7] (only the direct output signal is shown) and three possible power waveforms. The logic values represent the precharge states. The thick branches will be *ON* during evaluation.

Depending on the input order of arrival, different internal 2-input LUTs in a 6-input LUT (Fig. 7.4) may trigger even before the global output switches to its final value. For

example, if the order of arrival is $\mathbf{F^d} \rightarrow \mathbf{E^d} \rightarrow \mathbf{D^d} \rightarrow \mathbf{C^d} \rightarrow \mathbf{B^d} \rightarrow \mathbf{A^d}$ ($\mathbf{F^d}$ is the fastest differential pair, and $\mathbf{A^d}$ is the slowest), then there will be no change in the logic values along the propagation path before $\mathbf{A^d}$ turns valid, since the precharge logic values are preserved. Once signal $\mathbf{A^d}$ arrives, the information will propagate from SRAM to global output, and the power consumption will present a single large spike at the end of the transition as shown in Trace **[a]** of Figure 7.4. If the order of arrival is $\mathbf{A^d} \rightarrow \mathbf{B^d} \rightarrow \mathbf{C^d} \rightarrow \mathbf{D^d} \rightarrow \mathbf{E^d} \rightarrow \mathbf{F^d}$, the power consumption will present three spikes, each being of lower magnitude, since the internal 2-input LUTs are triggered at different times as the SRAM information propagates from the left-most input to output as shown in Trace **[b]** of Figure 7.4.. Combinations of these two extreme cases are possible where the power consumption presents one and two spikes as shown in Trace **[c]** of Figure 7.4.. This *intra-LUT* switching activity translates into a data-dependent power consumption of a significant magnitude as it is a short-circuit power of the ratioed logic inverter-Keeper. As such, it can be exploited by early evaluation attacks.

An activity pattern similar to the one generated at the intra-LUT level can also occur at the *inter-LUT* level. The 6-input LUTs forming a complex circuit may trigger at different times as their inputs turn valid. As a result, the inter-LUT power consumption will also be data-dependent and thus exploitable by early evaluation attacks. Both the intra-LUT and inter-LUT activities are very difficult to compensate at the algorithm level in FPGAs due to the major routing constraints involved and/or the large amount of logic required to balance the different propagation delays.

We next present a circuit technique to synchronize the operation of the 2-input LUTs, so that robustness to early evaluation attacks of the 6-input LUT is guaranteed by design. As previously described, the existing robustness to attacks based on dynamic and static powers, as well as glitches is preserved. We also show that our secured-by-design multiple-input

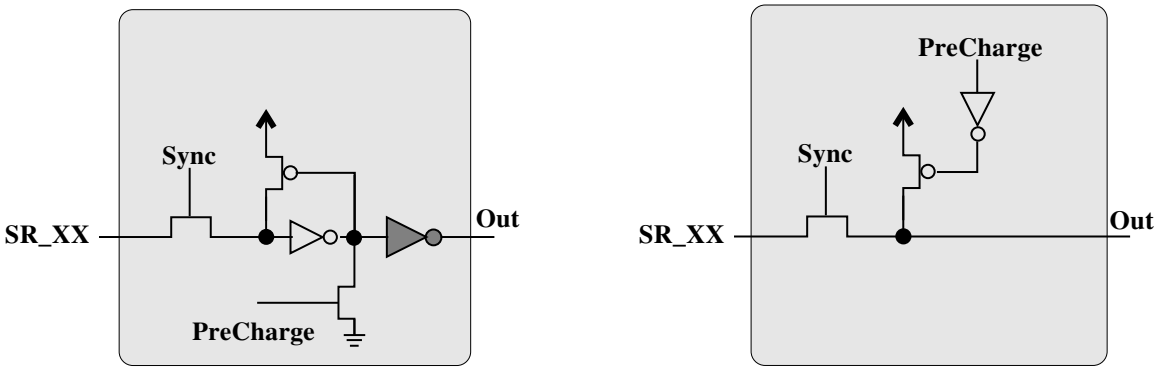
LUT provides support for synchronizing larger digital circuits, thus preventing inter-LUT early evaluation.

7.2.1 Countering Intra-LUT Early evaluation

Early evaluation activity within the 6-input LUT can be prevented as long as the delays of all six inputs can be statically determined. In this case, the slowest signal is to be routed to a Level-1 input which is the closest to SRAM, as shown in Figure 7.4. This strategy prevents any intra-LUT activity before the slowest input arrives, but it generates a significant routing constraint, which, in turn, may impair the routability of the design. This is a serious impediment to the use of dual-rail logic which requires balanced routing [124]. It is therefore important to support synchronization in hardware.

In the proposed synchronization circuitry, an nMOS switch and a level-restoring buffer are added at each SRAM cell output, as shown in Figure 7.5. This switch remains *OFF*, disabling any switching activity downstream of the SRAM cells, until all the LUT inputs arrive (that is, turn valid from their precharge state). The nMOS switch is controlled by a synchronization signal, called ***Sync***, and will turn *ON* after all LUT inputs arrive. In order to preserve the signal polarity and the monotonic behaviour of our prior-work LUT [7], an additional inverter (drawn in gray) is deployed in series with the level-restoring buffer as shown in Figure 7.5a). Our simulations, carried out on Cadence's Virtuoso and Spectre, indicate that this synchronization element increases the area of the secured 6-input LUT by approximately 19%, a figure which is far below those previously described in the prior art. A second implementation option is presented in Figure 7.5b in which the level-restoring buffer and the additional inverter have been removed. The hardware overhead of the second element reduces to approximately 6%. However, the nMOS switch will be connected in series with the two pass transistors of the Level-1 LUT. Such a topology will not only

increase the total delay of the LUT, but will also exhibit an increased delay sensitivity to mismatch and process variations. For this reason, the first option is preferred.



(a) EE resistance circuit I. (b) EE resistance circuit II.

Figure 7.5: Proposed EE resistance circuits.

The implementation of the synchronization signal, ***Sync***, is shown in Figure 7.6. An *OR* gate is built with a series connection of groups of two parallel-connected pass transistors and a level-restoring buffer with a precharge transistor. The gates of the transistors in the same group are driven by the complementary signals in a dual-rail pair. It must be recalled that each complementary signal $\mathbf{S^d}$ in dual-rail logic can be either in the precharge state $(0,0)$, or in the evaluation state $(1,0)$ or $(0,1)$, with the state $(1,1)$ never occurring. In order to pass '0' to the level-restoring buffer, all dual-rail inputs need to be in their evaluation state, so that one transistor per group is *ON*. This allows the short-circuit current to flow from Keeper to Ground until the level-restoring buffer switches to the opposite state turning the Keeper *OFF* and raising ***Sync*** to '1'. If at least one of the LUT inputs is still in its precharge state, then its corresponding group is *OFF*, no short-circuit current can flow in the circuit, and the level-restoring buffer preserves its precharge state, keeping ***Sync*** to '0'. In terms of early evaluation robustness, it is important to observe that this behaviour guarantees that short-circuit power is not generated in the synchronization circuitry nor the 6-input LUT before all inputs become active.

Due to its six pass transistors connected in series, the proposed synchronization circuit

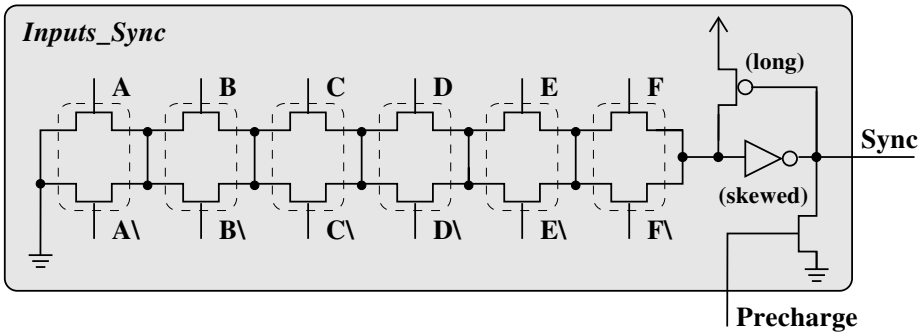


Figure 7.6: LUT inputs synchronization circuit.

increases the overall LUT latency. To avoid excessive latency levels, the sizes of these pass transistors must be adjusted. By increasing the width to $4\times$ the minimum size, the LUT delay increases by approximately 40% compared to our prior-work LUT [7]. In comparison, prior-art techniques such as BCDL would need a LUT delay to support synchronization. Moreover, the latency of the FPGA interconnection network is commonly much greater than any LUT latency. As such, the synchronization latency is acceptable. The hardware overhead is small as only a single *OR* gate for the entire 6-input LUT is needed.

The proposed secured-by-design 6-input LUT against early evaluation attacks reduces the number of power consumption spikes, which are generated by the different input arrival times, to 1. How to prevent early evaluation activity of a group of LUTs will now be demonstrated.

7.2.2 Countering Inter-LUT Early evaluation

To illustrate how to prevent early evaluation attacks against a complex circuit built with multiple LUTs, an 8-bit ripple carry adder with two input arguments $\mathbf{A^d} = A_7^d \ldots A_0^d$ and $\mathbf{B^d} = B_7^d \ldots B_0^d$ is presented. Each signal, with the exception of *Precharge*, is dual-rail encoded as indicated by the use of superscript **d**. As shown in Figure 7.7, the adder is built as a cascade of 1-bit half adders, each of which is mapped onto a secured-by-design

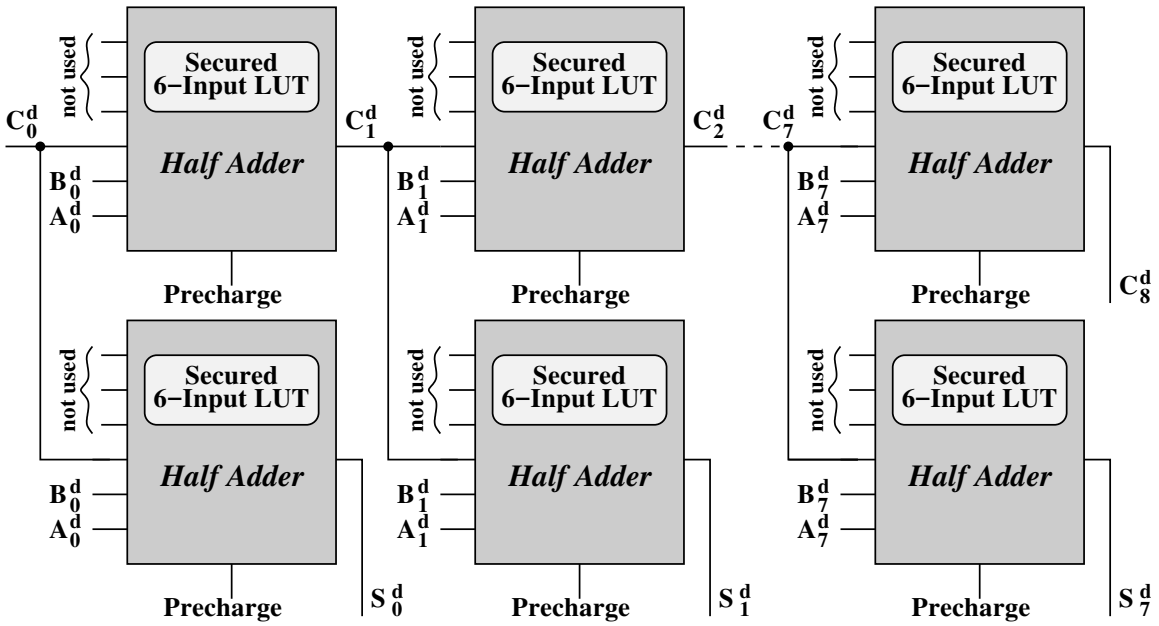


Figure 7.7: 8-bit ripple carry adder implemented with the proposed secured LUT (index **d** indicates a dual-rail signal).

6-input LUT. A total of 16 LUTs are needed to implement the ripple carry adder – eight for calculating the carry bits, $C_1^d \ldots C_8^d$, and eight for calculating the sum bits, $S_0^d \ldots S_7^d$. Since the calculation of C_i^d requires a valid C_{i-1}^d, where $i = 0 \ldots 7$, the carry signal propagation generates a chain of true dependencies.

Assume, for example, that A_1^d and/or B_1^d are still in their precharge state (thus, invalid from the point of view of early evaluation analysis) as C_1^d becomes available. This means that there will be no switching activity downstream of the first 1-bit half adder. The calculation will resume after the A_1^d and B_1^d arrive. This scenario translates into a power consumption pattern that carries information on the arrival time of the input bits, despite each LUT possessing intrinsic early-evaluation immunity. To prevent early evaluation attacks based on this inter-LUT activity profile, it is needed that the carry propagation commences only after all input bits have arrived. To achieve this, the input signals $\mathbf{A^d} = A_7^d \ldots A_0^d$, $\mathbf{B^d} = B_7^d \ldots B_0^d$, and C_0^d need to be synchronized.

Figure 7.8 presents the methodology of delaying the evaluation phase until after all

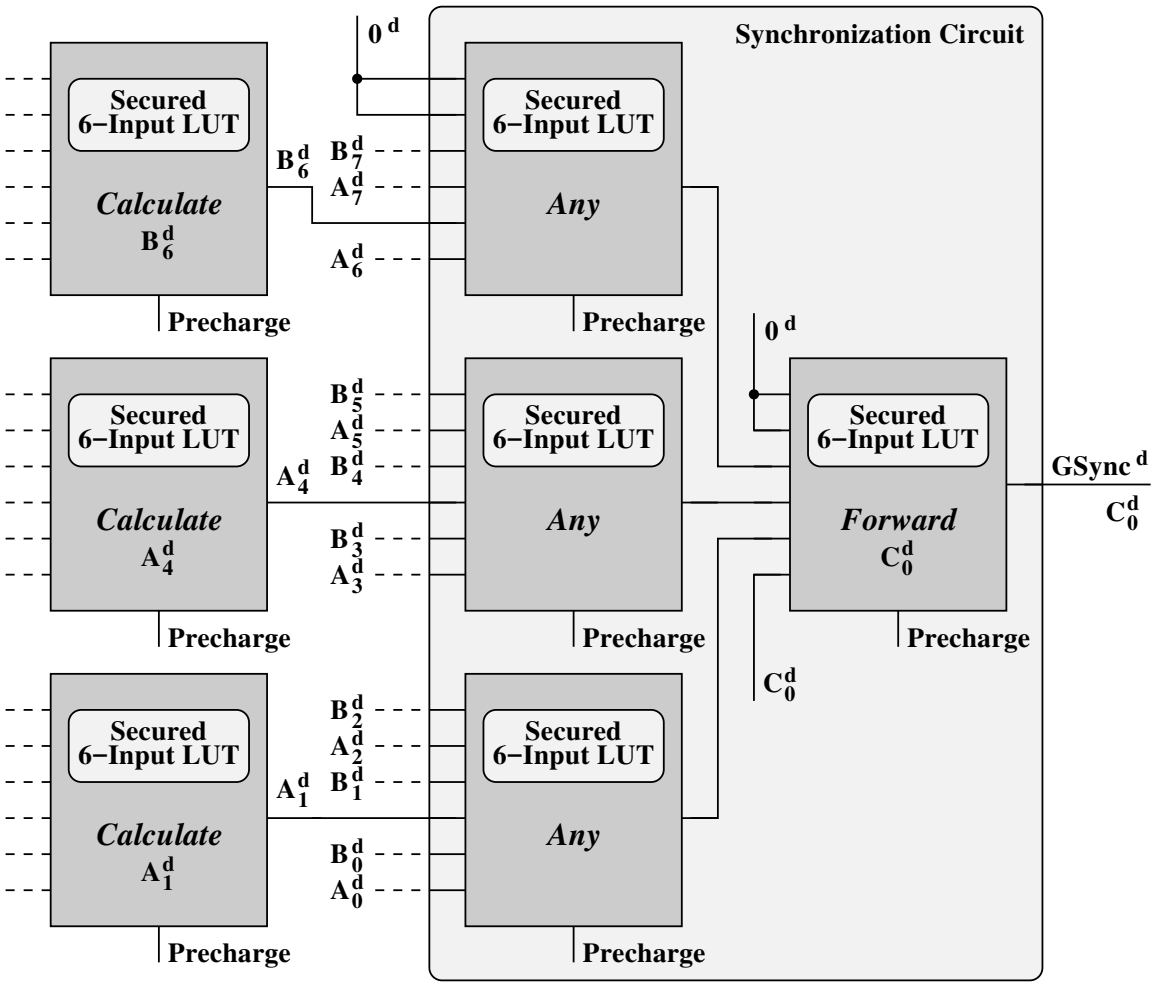


Figure 7.8: Synchronizing multiple LUTs' inputs to prevent inter-LUT early evaluation.

inputs of a multiple-LUT circuit arrive. The synchronization circuit uses four early-evaluation-secured LUTs connected in a tree structure that derives a global synchronization signal, GSync$^{\mathrm{d}}$. After all the bits of the input arguments $\mathbf{A^d}$ and $\mathbf{B^d}$ have arrived, GSync$^{\mathrm{d}}$ becomes a copy of the input carry signal, GSync$^{\mathrm{d}} = C_0^d$. The LUTs in the first stage play only a synchronization role, so *Any* logic function can be implemented in their SRAM cells. The LUT in the second stage *Forwards* C_0^d to the output, thereby implementing the identity function. The global synchronization signal, GSync$^{\mathrm{d}}$, will drive the carry input of the 8-bit ripple carry adder as shown in Figure 7.7. Thus, the addition will commence only after all input bits arrive, and the power signal will no longer carry

early evaluation information. Some of the LUTs in the first synchronization stage can trigger as their inputs arrive. However, since these activities will be syncronized with the LUTs producing the input bits (those *Calculate* LUTs in Figure 7.8), they do not carry early evaluation information in addition to what the *Calculate* LUTs already provide. The designer has the option to extend syncronization over the upstream or downstream LUTs as the security of the application requires.

To summarize our results to this point, the described synchronization circuitry ensures that the power consumption does not depend on the processed data and the configured logic function. The dynamic power does not carry information, as each input is encoded differentially in a dual-rail logic style. Based on the leakage current, it is in principle possible to determine how many inputs have arrived. However, since the leakage current will not reveal which of those inputs have arrived, the static power does not leak side-channel information. As well, the short-circuit power occurs only after all inputs arrive which means it does not carry side-channel information. The logic exhibits monotonicity, thus glitch-based attacks are not possible. As a result, the circuits mapped onto our FPGA will be robust to dynamic power, static power, early evaluation, and glitch-based attacks. For the sake of completeness, we present a diagram of the secured-by-design 6-input LUT in Figure 7.9.

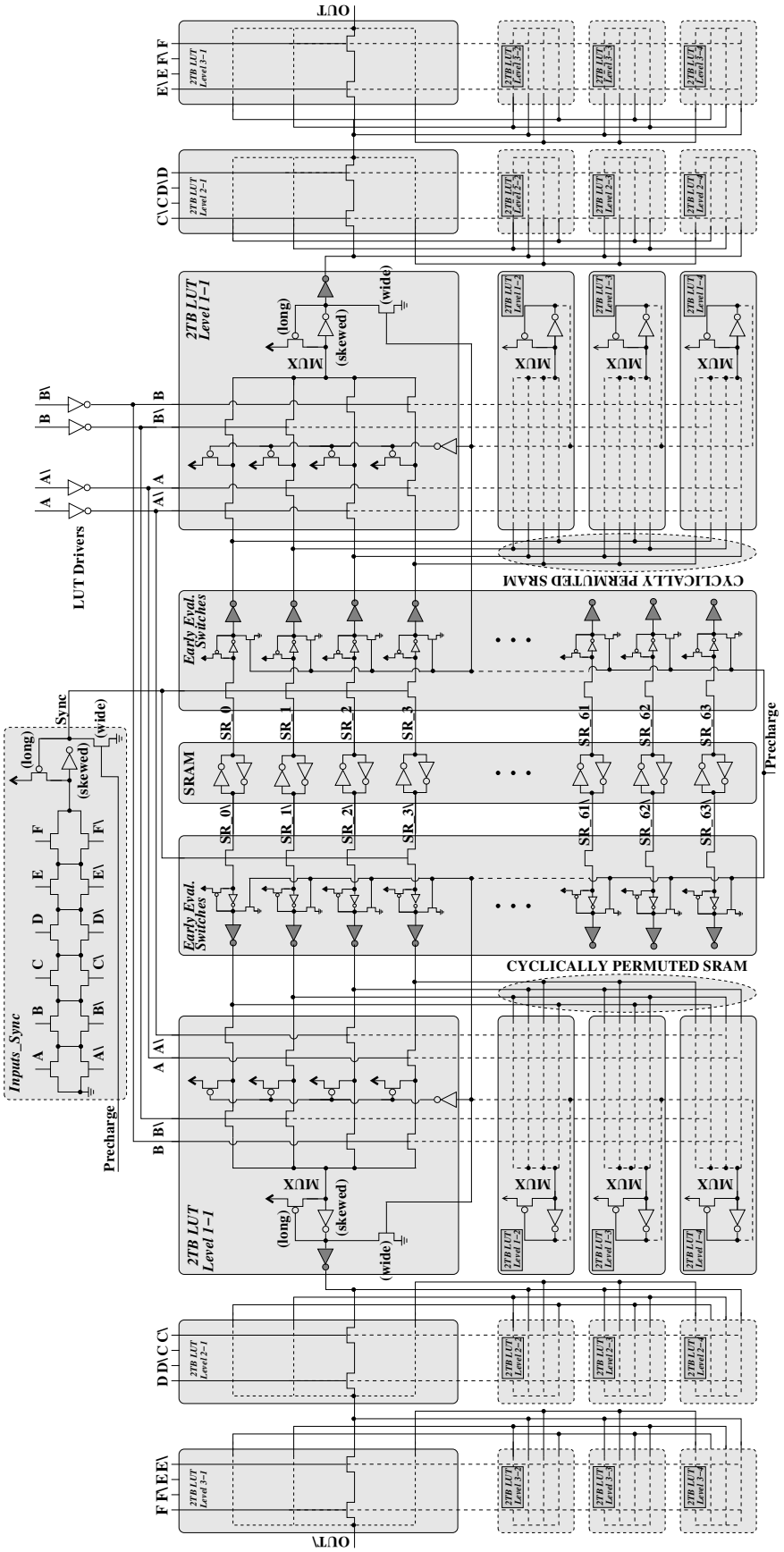


Figure 7.9: Proposed synchronization circuit and early evaluation resistance circuit for a 6-Inputs LUT.

7.2.3 Results and Discussion

Proposed methods to prevent glitches and early evaluation work with both the techniques presented in Chapter 5 and 6. The basic approach in securing the circuit is *replication* which must be performed in a way that minimizes the hardware overhead. Our LUT takes advantage of the availability of SRAM complementary outputs in implementing dual-rail logic which conceals dynamic power consumption. The precharge of the dual-rail logic is implemented with nMOS transistors at the outputs of the level-restoring buffers. Memory effects are removed by nMOS transistors connected between the supply rail and the middle nodes of the 2TB-LUTs. In Chapter 5, static power consumption is concealed through quadruple replication of the 2TB-LUT branches (see those 2TB-LUT Level 1-1, Level 1-2, Level 1-3, Level 1-4), and cyclic permutation of the configuration SRAM cells whereas in Chapter 6, an additional circuit is proposed to provide symmetrical structure with a significant reduction in area. The gray inverters adjust signal polarities to guarantee monotonic behaviour and, therefore, a glitch-free operation. Early evaluation attacks is prevented by using a synchronization circuit to delay the evaluation upon the arrival of all valid inputs.

Tables 7.2 and 7.1 provide estimates of the silicon area normalized to the technological minimum size for the secured-by-design four and six input LUT using two different techniques. The first technique is the $4\times$ replica with cyclic permutation presented in Chapter 5. The second technique is the additional circuit to ensure symmetrical LUT implementation as presented in Chapter 6. The *Basic* column shows the area figures for a non-secured (commercially similar) FPGA. The *Overhead* and *Total* columns show the area figures for the secured design. Area ratio to the the standard LUT are shown in the last column.

Because of its duplication strategy, the dual-rail logic doubles the hardware area. Since the SRAM latch already provides the complementary output, it can be shared between

Table 7.1: Estimated silicon areas for secure 4-input LUT.

Component	Area (× minimum size)						
	Basic	Chapter 5			Chapter 6		
		Overhead	Total	Ratio	Overhead	Total	Ratio
Pass Transistors	30	610	640	-	130	160	-
SRAM	128	128	256	-	0	128	-
Precharge middle	0	320	320	-	80	80	-
Precharge out	0	80	80	-	20	20	-
Drivers	32	32	64	-	32	64	-
Input Inverters	16	-16	0	-	-16	0	-
Keeper (L=3)	10	110	120	-	10	20	-
LRB (skewed)	20	140	160	-	20	40	-
IN Sync	0	16	16	-	16	16	-
Early Ev. Switches	0	384	384	-	384	384	-
IN Gen	0	-	-	-	72	72	-
TOTAL LUT	236	1418	2040	8.6 ×	752	984	4.2 ×

the dual-rail halves so that the associated hardware overhead used to conceal the dynamic power is reduced. The replication techniques mentioned in Chapter 5 that are needed to conceal static power occupy more than 36% of the total LUT area while this figure is reduced to 14% by using the techniques presented in Chapter 6. The early evaluation protection circuitry requires 1556 transistors per secured 6-input LUT – a small figure compared to prior art. Overall, the secured LUT will need about 10 times more area than the non-secured one if the techniques presented in Chapter 5 are used. Using techniques presented in Chapter 6 will require only 4.35 times increase in the silicon area.

It was reported that dual-rail logic mapped onto Virtex-II FPGA requires 6× more resources than standard logic [124]. Improvements of the WDDL showed an 11× area overhead versus standard logic [52, 134]. Implementing the circuit technique presented in Chapter 6 reduces the area overhead significantly over the prior art (4.35×), while a quadruple robustness against dynamic power, static power, early evaluation, and/or glitch-based attacks is provided. By using the replication technique, it is apparent that our area overhead is in line with the prior art. The user of the proposed circuit will not

Table 7.2: Estimated silicon areas for secure 6-input LUT.

Component	Area (× minimum size)						
	Basic	Chapter 5			Chapter 6		
		Overhead	Total	Ratio	Overhead	Total	Ratio
Pass Transistors	126	2562	2688	-	546	672	-
SRAM	512	512	1024	-	0	512	-
Precharge middle	0	1344	1344	-	336	336	-
Precharge out	0	336	336	-	84	84	-
Drivers	48	48	96	-	48	96	-
Input Inverters	24	-24	0	-	-24	0	-
Keeper (L=3)	42	462	504	-	42	84	-
LRB (skewed)	84	588	672	-	84	168	
IN Sync	0	20	20	-	20	20	-
Early Ev. Switches	0	1536	1536	-	1536	1536	-
IN Gen	-	-	-	-	132	132	-
TOTAL LUT	836	7384	8220	9.8×	2804	3640	4.35×

see any difference by using a standard FPGA. Therefore, the architecture of the proposed secured-by-design LUT remains in line with commercial FPGAs. Since the interconnection network is known to occupy more than 90% of the FPGA area [47], we can consider the area overhead incurred to be a good trade-off for the achieved security level.

7.3 Conclusion

Circuit techniques providing robustness to glitches and early evaluation attacks in addition to the existing robustness to dynamic power and static power have been described. The reconfigurability of the platform is preserved; thus, the comfort of implementing robust cryptosystems without any special design techniques is offered to cryptoysystems developers. The silicon area overhead is small compared to prior art.

Chapter 8

Conclusion and Future Work

8.1 Conclusions

Securing FPGAs against the potential threats given by power analysis attacks is essential as an increasing number of cryptosystems use FPGAs to gain hardware-like performance with software-like flexibility. Offering an FPGA-based implementation that is robust to power attacks is more challenging that in ASICs, where the designer has full control over the implementation options. In this dissertation, we offer a secure-by-design FPGA, whose circuitry is designed to provide intrinsic robustness to power attacks while preserving the reconfiguration architecture of commercial FPGAs. The power consumption of all circuits mapped to our secured FPGA is concealed, enabling FPGA-mapped cryptosystems to be immune to power attacks. As such, the digital designer gains security-by-design without needing to overlay upper level security abstractions.

In reviewing the FPGA architecture and implementation in Chapter 2, limitations are observed in the commercial FPGAs robustness to power attacks. First, FPGAs are natively built with a single-ended CMOS logic – a feature that makes mapping of balanced dual-rail logic onto FPGA difficult although critical to achieving power attack robustness. Moreover,

FPGAs are not built with electrically symmetrical multiplexers causing them to leak information about the processed data and the implemented function. Scaling down technological features the in newer technology nodes increases the threat of static power attacks. FPGAs are built with ratioed circuits, which are known to exhibit a large short-circuit power consumption. This makes the FPGA vulnerable to glitches and early evaluation attacks. Every single aspect of power consumption can leak valuable information. Therefore, secured designs must address each and every component of power consumption.

Many countermeasures at different abstraction levels have been introduced in the literature to prevent power analysis attacks spanning protocol, algorithm, architecture, and circuit. However, none of these countermeasures is design to provide simultaneous robustness against all four types of power attacks. Since the countermeasures applied at the first three levels of abstractions do not tackle the problem at its source, their implementations can: i) be power demanding, due to the massive replication of the coarse-grained arithmetic-logic units, ii) incur a large silicon area overhead penalty, or iii) require significant programming effort. We have proposed circuit level countermeasures aimed at creating a secured-by-design reconfigurable hardware to address these issue.

Chapter 5 presented the proposed symmetrical two-transistor branch LUT to replace the standard multiplexer. This 2TB LUT reduces the required replication to conceal the static power to $4\times$. Replication and cyclic permutation are sufficient to ensure robustness to all four attack types. To build secured LUTs with more than two inputs, the power consumption of the 2-input LUT must be made independent of both the processed data and implemented LUT function. This requirement is difficult to achieve with electrically asymmetrical (standard) LUTs. It was shown that by adding a Stub of four extra branches to the 2TB LUT equalizes the Hamming weight enabling the building of secured LUTs with multiple inputs. These multiple-input LUTs exhibit robustness to the both dynamic

and static power attacks, and increased robustness to early evaluation attacks. However, significant area overhead is required, as is in line with prior art.

The eight-branch LUT ensures electrical symmetry on the multiplexer implementation as well as the SRAM configuration, but unbalances the loading seen by the LUT input drivers. To eliminate the large $4\times$ area overhead due to replication and to balance the loading, we propose an additional circuit, called Stub Control Signal Generator, to drive the Stub branches. We showed that by using this additional circuit, the $4\times$ replication is no longer needed, and the loading seen by the LUT input drivers becomes balanced.

As the LUT is a ratioed circuit logic, the threat of short-circuit power information leakage is increased. A precharge strategy has been proposed to provide robustness to glitches through maintaining monotonic behaviour. Circuit techniques providing robustness to early evaluation attacks in addition to the existing dynamic power and static power robustness have been described. The resulting secured-by-design FPGA LUT prevents attacks based on early evaluation. The silicon area penalty is reasonable for the robustness achieved making the secure-by-design FPGA attractive to cryptoysystems developers.

Table 8.1: Comparison of Different Countermeasures Security Features.

Logic	Robust to Power Attack based on				Reconfigurability	Architecture Preserved	Area Overhead
	Dynamic	Static	Glitches	Early Evaluation			
WDDL [124]	✓				✓		6 ×
BCDL [88]	✓		✓		✓		11 ×
LBDL [135]	✓		✓				7 ×
DPL-noEE [17]	✓		✓		✓		10 ×
AWDDL [86]	✓		✓		✓		42 ×
Ch.5	✓	✓	✓	✓	✓	✓	9.8 ×
Ch.6	✓	✓	✓	✓	✓	✓	4.35 ×

Table 8.1 compares between different countermeasures proposed in the literatures and our proposed techniques. This comparison focus in the level of robustness to different power attacks, reconfigurability, preserving the FPGA architecture, area overhead. WDDL prevent switching power attack by equalizing the number of transition; however, it is subject to attacks based on glitches and early evaluation effects. To overcome these

weaknesses, BCDL, DPL-noEE, and AWDDL have been proposed; however, the intra LUT short-circuit current, which leaks information about the processed data, is not addressed. Hence, it is not secure against attacks based on early evaluation. Moreover, the LUT is not fully utilized by using the full LUT to implement 2-input logic function. LBDL is another technique has used the LUT structure to equalize evaluation time for any function; hence, it prevents glitches. All the above mentioned techniques subject to attacks based on static power and early evaluation. Achieving such robustness to a specific implementation at different level of abstraction may be viable. However, it will be at the expense of large silicon area and degraded performance. In this dissertation, we propose a secured reconfigurable hardware to attacks based on dynamic power, static power, glitches, and early evaluation while requiring a much smaller area overhead than prior art as shown in Table 8.1. The achieved level of power concealment has been possible by tackling the main cause at the circuit level.

To summarize, our contributions are:

1. **Robustness to switching power attacks** by applying dual-rail logic in the context of LUTs and using the SRAMs' complementary outputs.
2. **Robustness to static power attacks** by 2TB LUT with four branches to eliminate the relation between the static power consumption and the processed data – a security feature obtained by replicating the LUT multiplexer and cyclic permutation of the SRAM configurations.
3. **Robustness to static power attacks** by 2TB LUT with eight branches to eliminate the relation between the static power consumption and both the implemented function and processed data. This security feature is obtained by replicating the LUT multiplexer and cyclic permutation of the SRAM configurations. In addition, having eight branches LUT equalizes the Hamming weight for all possible functions.

4. **Robustness to static power attacks with reduced area overhead** by an additional circuit that drives in a smart way the Stub branches to ensure symmetry. We showed that by using this additional circuit, the 4× replication is no longer needed, a feature that leads to a significant area saving.
5. **Robustness to attacks based on glitches and intra-LUT early evaluation** by proposing well-defined precharge strategy and circuit synchronization technique with reduced hardware overhead, which delays the evaluation of the LUT until all its valid inputs arrive.
6. **Robustness to attacks based on inter-LUT early evaluation** by proposing methodology to extend the synchronization to multiple LUTs, so that a complex circuit will not evaluate before all its global inputs turn valid.
7. **Balance routing static power** by securing the switch box to build a complex circuit of a group of LUTs.

8.2 Future Work

In this dissertation we have focused on providing reconfigurable FPGA hardware circuitry that is robust to power consumption side-channel attacks. The general architecture of commercial FPGAs (i.e., six-input LUTs, SRAM configuration memory, and 4- or 8-input switch boxes) is preserved. The circuit robustness to power attacks is achieved through the replication of the pass-transistor logic and the reuse of the SRAM cells. If the reconfiguration feature is not needed (e.g., when an ASIC is to be manufactured), the SRAM cells can be replaced with hardwired connections to the voltage supply rail or ground. Since all the security results of this dissertation should remain valid in such a scenario, it should be possible to build secured-by-design ASICs using our secured pass-transistor logic. Validating this statement is left for future work.

In the circuits we proposed, we made no special assumption with respect to the SRAM cells (except for their current capability). This means that an FPGA in which the SRAM cells are replaced with Flash or anti-fuse memory cells should still be secure. The design of a secured non-SRAM FPGAs is also left for future work.

In a commercial FPGA context, the digital designers in general have no control over the routing. This is a feature that can cause difficulties in using balanced loads in implementing dual-rail logic in commercial FPGA architectures. Our syncronization circuit, introduced in Chapter 7, delays the evaluation of the output signal until all the inputs arrive, a feature reduces the reliance on balanced dual-rail loads. Investigating this aspect in an ASIC context, where better control over routing is available, is left for future work.

All the schematics of the circuits described in this dissertation have been edited with Cadence Virtuoso and simulated with Cadence Spectre (a Spice-like simulator). This confirms that our approach is conceptually correct at the circuit level, but confirmation at the layout level is left for future work.

Bibliography

[1] Afshin Abdollahi, Farzan Fallah, and Massoud Pedram. Leakage current reduction in CMOS VLSI circuits by input vector control. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 12(2):140–154, 2004.

[2] FPGA Actel SmartFusion. Device Family.

[3] Michel Agoyan, Jean-Max Dutertre, Amir-Pasha Mirbaha, David Naccache, Anne-Lise Ribotta, and Assia Tria. How to flip a bit? In *On-Line Testing Symposium (IOLTS), 2010 IEEE 16th International*, pages 235–239. IEEE, 2010.

[4] Massimo Alioto, Simone Bongiovanni, Milena Djukanovic, Giuseppe Scotti, and Alessandro Trifiletti. Effectiveness of leakage power analysis attacks on DPA-resistant logic styles under process variations. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 61(2):429–442, 2014.

[5] Massimo Alioto, Luca Giancane, Giuseppe Scotti, and Alessandro Trifiletti. Leakage power analysis attacks: A novel class of attacks to nanometer cryptographic circuits. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 57(2):355–367, 2010.

[6] Ziyad Almohaimeed and Mihai Sima. Secured-by-design FPGA: look-up tables and switch-boxes. In *Nordic Circuits and Systems Conference (NORCAS): NORCHIP & International Symposium on System-on-Chip (SoC), 2015*, pages 1–4. IEEE, 2015.

[7] Ziyad Almohaimeed and Mihai Sima. Look-Up tables with multiple inputs for secured-by-design FPGAs. In *Circuits and Systems (MWSCAS), 2016 IEEE 59th International Midwest Symposium on*, pages 1–4. IEEE, 2016.

[8] Ziyad Mohammed Almohaimeed. Increasing the Robustness of Point Operations in Co-Z Arithmetic against Side-Channel Attacks. Master's thesis, University of Victoria, 2013.

[9] Altera. *Stratix V Device Design Guidelines*, Dec. 2013.

[10] Ross Anderson and Markus Kuhn. Low cost attacks on tamper resistant devices. In *International Workshop on Security Protocols*, pages 125–136. Springer, 1997.

[11] Navid Azizi and Farid N Najm. Look-up table leakage reduction for FPGAs. In *Custom Integrated Circuits Conference, 2005. Proceedings of the IEEE 2005*, pages 187–190. IEEE, 2005.

[12] Karthik Baddam and Mark Zwolinski. Divided Backend Duplication Methodology for Balanced Dual Rail Routing. In *CHES*, volume 5154, pages 396–410. Springer, 2008.

[13] Karthik Baddam and Mark Zwolinski. Path switching: a technique to tolerate dual rail routing imbalances. *Design Automation for Embedded Systems*, 12(3):207–220, 2008.

[14] Jean-Claude Bajard, Laurent Imbert, Pierre-Yvan Liardet, and Yannick Teglia. Leak resistant arithmetic. In *CHES*, volume 3156, pages 62–75. Springer, 2004.

[15] Josep Balasch, Benedikt Gierlichs, Roel Verdult, Lejla Batina, and Ingrid Verbauwhede. Power analysis of Atmel CryptoMemory–recovering keys from secure EEPROMs. In *Cryptographers Track at the RSA Conference*, pages 19–34. Springer, 2012.

[16] Vaughn Betz, Jonathan Rose, and Alexander Marquardt. *Architecture and CAD for deep-submicron FPGAs*, volume 497. Springer Science & Business Media, 1999.

[17] Shivam Bhasin, Sylvain Guilley, Florent Flament, Nidhal Selmane, and Jean-Luc Danger. Countering early evaluation: an approach towards robust dual-rail precharge logic. In *Proceedings of the 5th Workshop on Embedded Systems Security*, page 6. ACM, 2010.

[18] J al Birkner, A Chan, HT Chua, A Chao, K Gordon, B Kleinman, P Kolze, and R Wong. A very-high-speed field-programmable gate array using metal-to-metal antifuse programmable elements. *Microelectronics Journal*, 23(7):561–568, 1992.

[19] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 16–29. Springer, 2004.

[20] Stephen D Brown, Robert J Francis, Jonathan Rose, and Zvonko G Vranesic. *Field-Programmable Gate Arrays*, volume 180. Springer Science & Business Media, 1992.

[21] Suresh Chari, Charanjit Jutla, Josyula Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In *Advances in CryptologyCRYPTO99*, pages 791–791. Springer, 1999.

[22] Xavier Charvet and Herve Pelletier. Improving the DPA attack using Wavelet transform. In *NIST Physical Security Testing Workshop*, volume 46, 2005.

[23] Zhanping Chen, Mark Johnson, Liqiong Wei, and Kaushik Roy. Estimation of standby leakage power in CMOS circuits considering accurate modeling of transistor stacks. In *Proceedings of the 1998 international symposium on Low power electronics and design*, pages 239–244. ACM, 1998.

[24] Zhimin Chen and Yujie Zhou. Dual-rail random switching logic: a countermeasure to reduce side channel leakage. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 242–254. Springer, 2006.

[25] Benoît Chevallier-Mames, Mathieu Ciet, and Marc Joye. Low-cost solutions for preventing simple side-channel analysis: Side-channel atomicity. *IEEE Transactions on computers*, 53(6):760–768, 2004.

[26] Charles Chiasson and Vaughn Betz. Should FPGAs abandon the pass-gate? In *Field Programmable Logic and Applications (FPL), 2013 23rd International Conference on*, pages 1–8. IEEE, 2013.

[27] Christophe Clavier, Jean-Sébastien Coron, and Nora Dabbous. Differential power analysis in the presence of hardware countermeasures. In *Cryptographic Hardware and Embedded SystemsCHES 2000*, pages 13–48. Springer, 2000.

[28] Jean-Sebastien Coron, David Naccache, and Paul Kocher. Statistics and secret leakage. *ACM Transactions on Embedded Computing Systems (TECS)*, 3(3):492–508, 2004.

[29] Roger Cuppens, Cornelis D Hartgring, Jan F Verwey, Herman L Peek, FAH Vollebragt, Elisabeth GM Devens, and IA Sens. An eeprom for microprocessors and custom logic. *IEEE Journal of Solid-State Circuits*, 20(2):603–608, 1985.

[30] J-L Danger, S Guilley, L Barthe, and P Benoit. Countermeasures against physical attacks in FPGAs. In *Security Trends for FPGAS*, pages 73–100. Springer, 2011.

[31] Jean-Luc Danger, Sylvain Guilley, Shivam Bhasin, and Maxime Nassar. Overview of dual rail with precharge logic styles to thwart implementation-level attacks on hardware cryptoprocessors. In *Signals, Circuits and Systems (SCS), 2009 3rd International Conference on*, pages 1–8. IEEE, 2009.

[32] Elke De Mulder, Pieter Buysschaert, SB Ors, Peter Delmotte, Bart Preneel, Guy Vandenbosch, and Ingrid Verbauwhede. Electromagnetic analysis attack on an fpga implementation of an elliptic curve cryptosystem. In *Computer as a Tool, 2005. EUROCON 2005. The International Conference on*, volume 2, pages 1879–1882. IEEE, 2005.

[33] Christoph Dobraunig, Maria Eichlseder, Stefan Mangard, and Florian Mendel. On the security of fresh re-keying to counteract side-channel and fault attacks. In *International Conference on Smart Card Research and Advanced Applications*, pages 233–244. Springer, 2014.

[34] Chris Dunlap and Tom Fischaber. Configuring Xilinx FPGAs Using an XC9500 CPLD and Parallel PROM, 2000.

[35] Abbas El Gamal, Jonathan Greene, Justin Reyneri, Eric Rogoyski, Khaled A El-Ayat, and Amr Mohsen. An architecture for electrically configurable gate arrays. *IEEE Journal of Solid-State Circuits*, 24(2):394–398, 1989.

[36] J Fabula, J Moore, and A Ware. Understanding neutron single-event phenomena in FPGAs. *Military Embedded Systems*, 3(2), 2007.

[37] Umer Farooq, Zied Marrakchi, and Habib Mehrez. FPGA architectures: An overview. *Tree-based Heterogeneous FPGA Architectures*, pages 7–48, 2012.

[38] PUB Fips. 186-2. digital signature standard (DSS). *National Institute of Standards and Technology (NIST)*, 20:13, 2000.

[39] Wieland Fischer and Berndt M Gammel. Masking at gate level in the presence of glitches. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 187–200. Springer, 2005.

[40] D Frohman-Bentchkowsky. A fully-decoded 2048-bit electrically-programmable MOS ROM. In *Solid-State Circuits Conference. Digest of Technical Papers. 1971 IEEE International*, volume 14, pages 80–81. IEEE, 1971.

[41] Louis Goubin and Jacques Patarin. DES and differential power analysis the duplication method. In *Cryptographic Hardware and Embedded Systems*, pages 728–728. Springer, 1999.

[42] Philipp Grabher, Johann Großschädl, and Dan Page. Non-deterministic processors: FPGA-based analysis of area, performance and security. In *Proceedings of the 4th Workshop on Embedded Systems Security*, page 1. ACM, 2009.

[43] Sylvain Guilley, Sumanta Chaudhuri, Laurent Sauvage, Tarik Graba, Jean-Luc Danger, Philippe Hoogvorst, Vinh-Nga Vong, Maxime Nassar, and Florent Flament. *Shall we trust WDDL?*, pages 208–215. Springer, 2009.

[44] Daniel C Guterman, ISAM H Rimawi, RD Halvorson, and DJ McElroy. An electrically alterable nonvolatile memory cell using a floating-gate structure. *IEEE Journal of Solid-State Circuits*, 14(2):498–508, 1979.

[45] Esmat Hamdy, John McCollum, S-O Chen, Steve Chiang, Shafy Eltoukhy, Jim Chang, Ted Speers, and Amr Mohsen. Dielectric based antifuse for logic and memory ICs. In *Electron Devices Meeting, 1988. IEDM'88. Technical Digest., International*, pages 786–789. IEEE, 1988.

[46] Darrel Hankerson, Alfred J Menezes, and Scott Vanstone. *Guide to elliptic curve cryptography*. Springer Science & Business Media, 2006.

[47] Scott Hauck and Andre DeHon. *Reconfigurable computing: the theory and practice of FPGA-based computation*, volume 1. Morgan Kaufmann, 2010.

[48] Wei He, Eduardo de la Torre, and Teresa Riesgo. A precharge-absorbed DPL logic for reducing early propagation effects on FPGA implementations. In *Reconfigurable Computing and FPGAs (ReConFig), 2011 International Conference on*, pages 217–222. IEEE, 2011.

[49] B Hoefflinger. High-Dynamic-Range (HDR) Vision: Microelectronics, Image Processing. *Computer Graphics (Springer Series in Advanced Microelectronics), Springer-Verlag New York, Inc., Secaucus, NJ*, 2007.

[50] Eddie Hung, Steven JE Wilton, Haile Yu, Thomas CP Chau, and Philip HW Leong. A detailed delay path model for FPGAs. In *Field-Programmable Technology, 2009. FPT 2009. International Conference on*, pages 96–103. IEEE, 2009.

[51] M Rabaey Jan, Chandrakasan Anantha, and Nikolic Borivoje. Digital integrated circuits: a design perspective, 2003.

[52] Jens-Peter Kaps and Rajesh Velegalati. DPA resistant AES on FPGA using partial DDL. In *Field-Programmable Custom Computing Machines (FCCM), 2010 18th IEEE Annual International Symposium on*, pages 273–280. IEEE, 2010.

[53] Tanay Karnik and Peter Hazucha. Characterization of soft errors caused by single event upsets in CMOS processes. *IEEE Transactions on Dependable and Secure Computing*, 1(2):128–143, 2004.

[54] ChangKyun Kim, JaeCheol Ha, Sung-Hyun Kim, Seokyu Kim, Sung-Ming Yen, and SangJae Moon. A secure and practical CRT-based RSA to resist side channel attacks. In *International Conference on Computational Science and Its Applications*, pages 150–158. Springer, 2004.

[55] ChangKyun Kim, Martin Schläffer, and SangJae Moon. Differential side channel analysis attacks on FPGA implementations of ARIA. *ETRI journal*, 30(2):315–325, 2008.

[56] Neal Koblitz. Elliptic Curve Cryptosystems. *Mathematics of computation*, 48(177):203–209, 1987.

[57] Cetin Kaya Koc. *Cryptographic Engineering*. Springer Science & Business Media, 2008.

[58] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *Advances in cryptologyCRYPTO99*, pages 789–789. Springer, 1999.

[59] Paul C Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *Annual International Cryptology Conference*, pages 104–113. Springer, 1996.

[60] Paul C Kocher. Leak-resistant cryptographic indexed key update, March 25 2003. US Patent 6,539,092.

[61] Konrad J Kulikowski, Mark G Karpovsky, and Alexander Taubin. Power attacks on secure hardware based on early propagation of data. In *On-Line Testing Symposium, 2006. IOLTS 2006. 12th IEEE International*, pages 6–pp. IEEE, 2006.

[62] Ian Kuon and Jonathan Rose. Measuring the gap between FPGAs and ASICs. *IEEE Transactions on computer-aided design of integrated circuits and systems*, 26(2):203–215, 2007.

[63] Ian Kuon, Russell Tessier, and Jonathan Rose. FPGA architecture: Survey and challenges. *Foundations and Trends in Electronic Design Automation*, 2(2):135–253, 2008.

[64] Guy Lemieux and David Lewis. *Design of interconnection networks for programmable logic*, volume 22. Springer, 2004.

[65] Philip HW Leong. Recent trends in FPGA architectures and applications. In *Electronic Design, Test and Applications, 2008. DELTA 2008. 4th IEEE International Symposium on*, pages 137–141. IEEE, 2008.

[66] Austin Lesea, Saar Drimer, Joseph J Fabula, Carl Carmichael, and Peter Alfke. The rosetta experiment: atmospheric soft error rate testing in differing technology FPGAs. *IEEE Transactions on Device and Materials Reliability*, 5(3):317–328, 2005.

[67] David Lewis, Elias Ahmed, Gregg Baeckler, Vaughn Betz, Mark Bourgeault, David Cashman, David Galloway, Mike Hutton, Chris Lane, Andy Lee, et al. The Stratix II logic and routing architecture. In *Proceedings of the 2005 ACM/SIGDA 13th international symposium on Field-programmable gate arrays*, pages 14–20. ACM, 2005.

[68] Lang Lin and Wayne Burleson. Leakage-based differential power analysis (LDPA) on sub-90nm CMOS cryptosystems. In *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*, pages 252–255. IEEE, 2008.

[69] Hongying Liu, Guoyu Qian, Satoshi Goto, and Yukiyasu Tsunoo. Correlation power analysis based on Switching Glitch model. In *International Workshop on Information Security Applications*, pages 191–205. Springer, 2010.

[70] Yingxi Lu, Maire P O'Neill, and John V McCanny. Differential Power Analysis of a SHACAL-2 hardware implementation. In *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*, pages 2933–2936. IEEE, 2008.

[71] Yuanlin Lu and Vishwani D Agrawal. CMOS leakage and glitch minimization for power-performance tradeoff. *Journal of Low Power Electronics*, 2(3):378–387, 2006.

[72] Hideyo Mamiya, Atsuko Miyaji, and Hiroaki Morimoto. Efficient countermeasures against RPA, DPA, and SPA. In *CHES*, volume 3156, pages 343–356. Springer, 2004.

[73] Stefan Mangard. Hardware countermeasures against DPA – a statistical analysis of their effectiveness. In *CT-RSA*, volume 2964, pages 222–235. Springer, 2004.

[74] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power analysis attacks: Revealing the secrets of smart cards*, volume 31. Springer Science & Business Media, 2008.

[75] David Marple and Larry Cooke. An MPGA compatible FPGA architecture. In *Custom Integrated Circuits Conference, 1992., Proceedings of the IEEE 1992*, pages 4–2. IEEE, 1992.

[76] David May, Henk L Muller, and Nigel P Smart. Non-deterministic processors. In *ACISP*, volume 1, pages 115–129. Springer, 2001.

[77] Robert P McEvoy, Colin C Murphy, William P Marnane, and Michael Tunstall. Isolated WDDL: a hiding countermeasure for differential power analysis on FPGAs. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, 2(1):3, 2009.

[78] Marcel Medwed, Christophe Petit, Francesco Regazzoni, Mathieu Renauld, and François-Xavier Standaert. Fresh Re-keying II: Securing Multiple Parties against Side-Channel and Fault Attacks. In *CARDIS*, volume 7079, pages 115–132. Springer, 2011.

[79] Marcel Medwed, François-Xavier Standaert, Johann Großschädl, and Francesco Regazzoni. Fresh Re-keying: Security against Side-Channel and Fault Attacks for Low-Cost Devices. *AFRICACRYPT*, 6055:279–296, 2010.

[80] Kevin Meritt. Differential Power Analysis attacks on AES. *Cryptography II, VCSG-706*, pages 1–17, 2012.

[81] Daniel Mesquita, Benoît Badrignans, Lionel Torres, Gilles Sassatelli, Michel Robert, and Fernando Moraes. A cryptographic coarse grain reconfigurable architecture robust against DPA. In *Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International*, pages 1–8. IEEE, 2007.

[82] Thomas S Messerges, Ezzat A Dabbish, and Robert H Sloan. Examining smart-card security under the threat of power analysis attacks. *IEEE transactions on computers*, 51(5):541–552, 2002.

[83] Frederic P Miller, Agnes F Vandome, and John McBrewster. Advanced encryption standard. 2009.

[84] Scott Miller, SIMA Mihai, and Michael McGUIRE. Alternatives in designing level-restoring buffers for interconnection networks in field-programmable gate arrays. In *Digital System Design Architectures, Methods and Tools, 2007. DSD 2007. 10th Euromicro Conference on*, pages 138–146. IEEE, 2007.

[85] Victor S Miller. Use of elliptic curves in cryptography. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 417–426. Springer, 1985.

[86] Amir Moradi and Vincent Immler. Early propagation and imbalanced routing, how to diminish in FPGAs. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 598–615. Springer, 2014.

[87] Siva Narendra, Vivek De, Dimitri Antoniadis, Anantha Chandrakasan, and Shekhar Borkar. Scaling of stack effect and its application for leakage reduction. In *Proceedings of the 2001 international symposium on Low power electronics and design*, pages 195–200. ACM, 2001.

[88] Maxime Nassar, Shivam Bhasin, Jean-Luc Danger, Guillaume Duc, and Sylvain Guilley. BCDL: a high speed balanced DPL for FPGA with global precharge and no early evaluation. In *Proceedings of the Conference on Design, Automation and Test in Europe*, pages 849–854. European Design and Automation Association, 2010.

[89] Siddika Berna Ors, Frank Gurkaynak, Elisabeth Oswald, and Bart Preneel. Power-Analysis Attack on an ASIC AES implementation. In *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*, volume 2, pages 546–552. IEEE, 2004.

[90] Siddika Berna Örs, Elisabeth Oswald, and Bart Preneel. Power-analysis attacks on an FPGA-first experimental results. In *CHES*, volume 2779, pages 35–50. Springer, 2003.

[91] Eric Peeters, François-Xavier Standaert, and Jean-Jacques Quisquater. Power and electromagnetic analysis: Improved model, consequences and comparisons. *Integration, the VLSI journal*, 40(1):52–60, 2007.

[92] Tao Pi and Patrick J Crotty. FPGA lookup table with transmission gate structure for reliable low-voltage operation, October 26 2004. US Patent 6,809,552.

[93] Thomas Popp, Mario Kirschbaum, Thomas Zefferer, and Stefan Mangard. Evaluation of the masked logic style MDPL on a prototype chip. *Cryptographic Hardware and Embedded Systems-CHES 2007*, pages 81–94, 2007.

[94] Thomas Popp and Stefan Mangard. Masked dual-rail pre-charge logic: DPA-resistance without routing constraints. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 172–186. Springer, 2005.

[95] Thomas Popp and Stefan Mangard. *Masked dual-rail pre-charge logic: DPA-resistance without routing constraints*, pages 172–186. 2005.

[96] Emmanuel Prouff and Matthieu Rivain. Masking against side-channel attacks: A formal security proof. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 142–159. Springer, 2013.

[97] Emmanuel Prouff, Matthieu Rivain, and Régis Bevan. Statistical analysis of second order differential power analysis. *IEEE Transactions on computers*, 58(6):799–811, 2009.

[98] Srividhya Rammohan, Vijay Sundaresan, and Ranga Vemuri. Reduced complementary dynamic and differential logic: a CMOS logic style for DPA-resistant secure IC design. In *VLSI Design, 2008. VLSID 2008. 21st International Conference on*, pages 699–705. IEEE, 2008.

[99] Alin Razafindraibe, Michel Robert, and Philippe Maurine. Improvement of dual rail logic as a countermeasure against DPA. In *Very Large Scale Integration, 2007. VLSI-SoC 2007. IFIP International Conference on*, pages 270–275. IEEE, 2007.

[100] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[101] Jonathan Rose, Abbas El Gamal, and Alberto Sangiovanni-Vincentelli. Architecture of field-programmable gate arrays. *Proceedings of the IEEE*, 81(7):1013–1029, 1993.

[102] Minoru Saeki and Daisuke Suzuki. Security Evaluations of MRSL and DRSL Considering Signal Delays. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 91(1):176–183, 2008.

[103] Laurent Sauvage, Sylvain Guilley, Jean-Luc Danger, Yves Mathieu, and Maxime Nassar. Successful attack on an FPGA-based WDDL DES cryptoprocessor without place and route constraints. In *Proceedings of the Conference on Design, Automation and Test in Europe*, pages 640–645. European Design and Automation Association, 2009.

[104] A Scheibe and W Krauss. A two-transistor SIMOS EAROM cell. *IEEE Journal of Solid-State Circuits*, 15(3):353–357, 1980.

[105] Lattice Semiconductor. Third Generation Non-Volatile FPGAs Enable System on Chip Functionality. *Lattice Web Site: www. latticesemi. com*, 2007.

[106] Plessey Semiconductors. ERA 60100 Preliminary Data Sheet. *Swindon, England*, 1989.

[107] Rafael Soares, Ney Calazans, Victor Lomné, Philippe Maurine, Lionel Torres, and Michel Robert. Evaluating the robustness of secure triple track logic through prototyping. In *Proceedings of the 21st annual symposium on Integrated circuits and system design*, pages 193–198. ACM, 2008.

[108] Product Specification. Virtex-5 Family Overview. 2006.

[109] François-Xavier Standaert, Sıddıka Berna Örs, and Bart Preneel. Power Analysis of an FPGA. In *Proceedings of the workshop on Cryptographic Hardware and Embedded Systems–CHES*, volume 4, pages 30–44. Springer, 2004.

[110] François-Xavier Standaert, Sıddıka Berna Ors, Jean-Jacques Quisquater, and Bart Preneel. Power analysis attacks against FPGA implementations of the DES. In *FPL*, volume 3203, pages 84–94. Springer, 2004.

[111] Data Encryption Standard et al. Data encryption standard. *National Bureau of Standards, US Department of Commerce*, 1977.

[112] G Edward Suh and Srinivas Devadas. Physical unclonable functions for device authentication and secret key generation. In *Proceedings of the 44th annual design automation conference*, pages 9–14. ACM, 2007.

[113] Song Sun, Zijun Yan, and Joseph Zambreno. Experiments in attacking FPGA-based embedded systems using differential power analysis. In *Electro/Information Technology, 2008. EIT 2008. IEEE International Conference on*, pages 7–12. IEEE, 2008.

[114] Vijay Sundaresan, Srividhya Rammohan, and Ranga Vemuri. Power invariant secure IC design methodology using reduced complementary dynamic and differential logic. In *Very Large Scale Integration, 2007. VLSI-SoC 2007. IFIP International Conference on*, pages 1–6. IEEE, 2007.

[115] Vijay Sundaresan, Srividhya Rammohan, and Ranga Vemuri. Defense against side-channel power analysis attacks on microelectronic systems. In *Aerospace and Electronics Conference, 2008. NAECON 2008. IEEE National*, pages 144–150. IEEE, 2008.

[116] Vijay Sundaresan, Srividhya Rammohan, and Ranga Vemuri. Defense against side-channel power analysis attacks on microelectronic systems. In *Aerospace and Electronics Conference, 2008. NAECON 2008. IEEE National*, pages 144–150. IEEE, 2008.

[117] Daisuke Suzuki and Minoru Saeki. Security evaluation of DPA countermeasures using dual-rail pre-charge logic style. In *CHES*, volume 4249, pages 255–269. Springer, 2006.

[118] Daisuke Suzuki and Minoru Saeki. An analysis of leakage factors for dual-rail pre-charge logic style. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 91(1):184–192, 2008.

[119] Daisuke Suzuki, Minoru Saeki, and Tetsuya Ichikawa. Random Switching Logic: A Countermeasure against DPA based on Transition Probability, 2004. dice@iss.isl.melco.co.jp 12755 received 3 Dec 2004.

[120] Shaohua Tang, Weijian Li, Jianhao Wu, Zheng Gong, and Ming Tang. Power analysis attacks against FPGA implementation of KLEIN. *Security and Communication Networks*, 9(18):5849–5857, 2016.

[121] Kris Tiri, Moonmoon Akmal, and Ingrid 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. IEEE, 2002.

[122] Kris Tiri and Patrick Schaumont. Changing the odds against masked logic. In *Selected Areas in Cryptography*, volume 4356, pages 134–146. Springer, 2006.

[123] Kris Tiri and Ingrid Verbauwhede. Securing encryption algorithms against DPA at the logic level: Next generation smart card technology. In *CHES*, volume 2779, pages 125–136. Springer, 2003.

[124] Kris Tiri and Ingrid Verbauwhede. A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation. In *Proceedings of the conference*

on Design, automation and test in Europe-Volume 1, page 10246. IEEE Computer Society, 2004.

[125] Kris Tiri and Ingrid Verbauwhede. Synthesis of Secure FPGA Implementations. *IACR Cryptology ePrint Archive*, 2004:68, 2004.

[126] Kris Tiri and Ingrid Verbauwhede. Design method for constant power consumption of differential logic circuits. In *Design, Automation and Test in Europe, 2005. Proceedings*, pages 628–633. IEEE, 2005.

[127] Stephen M Trimberger and Jason J Moore. FPGA security: Motivations, features, and applications. *Proceedings of the IEEE*, 102(8):1248–1265, 2014.

[128] Colin D Walter. Sliding windows succumbs to Big Mac attack. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 286–299. Springer, 2001.

[129] Neil Weste and David Harris. *CMOS VLSI Design: A Circuits and Systems Perspective*. Addison-Wesley Publishing Company, USA, 4th edition, 2010.

[130] Sau C Wong, HC So, Jung H Ou, and John Costello. A 5000-gate CMOS EPLD with multiple logic and interconnect arrays. In *Custom Integrated Circuits Conference, 1989., Proceedings of the IEEE 1989*, pages 5–8. IEEE, 1989.

[131] Xilinx. *7 Series FPGAs Data Sheet: Overview*.

[132] Michitarou Yabuuchi and Kazutoshi Kobayashi. NBTI-induced delay degradation analysis of FPGA routing structures. *Information and Media Technologies*, 7(4):1346–1352, 2012.

[133] Sung-Ming Yen, Wei-Chih Lien, Sang-Jae Moon, JaeCheol Ha, et al. Power analysis by exploiting chosen message and internal collisions-vulnerability of checking

mechanism for RSA-decryption. In *Mycrypt*, volume 3715, pages 183–1956. Springer, 2005.

[134] Pengyuan Yu and Patrick Schaumont. Secure FPGA circuits using controlled placement and routing. In *Proceedings of the 5th IEEE/ACM international conference on Hardware/software codesign and system synthesis*, pages 45–50. ACM, 2007.

[135] Daheng Yue, Yan Sun, Minxuan Zhang, Shaoqing Li, and Yutong Dai. A look-up-table based differential logic to counteract DPA attacks. In *ASIC, 2009. ASICON'09. IEEE 8th International Conference on*, pages 855–858. IEEE, 2009.

[136] Babak Zakeri. On studying Whitenoise Stream-Cipher against Power Analysis Attacks. Master's thesis, University of Victoria, 2012.