Université catholique de Louvain
ICTEAM Institute
Ecole Polytechnique de Louvain
Electrical Engineering Department

TRANSISTOR-LEVEL DESIGN OF LOW-POWER NANOSCALE DIGITAL CIRCUITS FOR SECURE APPLICATIONS

Dina Kamel

Thesis submitted in partial fulfillment
of the requirements for the degree of
Docteur en Sciences d'Ingénieur

Dissertation committee:
Prof. D. Flandre (ICTEAM, UCLouvain, Belgium), adviser
Prof. J-D. Legat (ICTEAM, UCLouvain, Belgium), member
Prof. F-X. Standaert (ICTEAM, UCLouvain, Belgium), member
Dr. D. Bol (ICTEAM, UCLouvain, Belgium), member
Prof. C. Piguet (CSEM / EPFL, Switzerland), member
Prof. I. Verbauwhede (K.U.Leuven, Belgium), member
Prof. C. De Vleeschouwer (ICTEAM, UCLouvain, Belgium), President

TRANSISTOR-LEVEL DESIGN OF LOW-POWER NANOSCALE DIGITAL CIRCUITS FOR SECURE APPLICATIONS

Dina Kamel
ICTEAM Institute
Ecole Polytechnique de Louvain
Electrical Engineering Department

Université catholique de Louvain
Louvain-la-Neuve (Belgium)

To my parents,
to my husband,
and to my daughter

CONTENTS

ACKNOWLEDGMENTS

All praise is due to ALLAH, the almighty, for guiding me through my whole life and for blessing me with the strength to start and finish this work.

If it were not for my parent, I would not be here today. They are always there for me with their love, care, support, constant encouragement and guidance through all the years of my life. They are my role models in all aspects of life and they are always proud of me. My sweet mother with her dedication encouraged me to pursue my masters degree and she was delighted and provided me with all the love and help in her power during the years of this PhD. My loving father, who is not with us today, guided me with all the love and care in the world. During the sleepless studying nights, he would stay up late just to see if I needed any help and this is just a single example. They taught me perseverance, hard work, contentment, patience, how to love and care for others and all the values I cherish. Words are incapable to express my gratitude to them.

I would not have started this thesis if it were not for my husband, Mostafa. After arriving in Belgium to begin his PhD, he searched for a PhD position for me as well and he contacted Prof. Flandre, my adviser who kindly provided me with the opportunity to work with him. During the years of the PhD, he always stood by me and encouraged me to continue. With his professional view, he advised me in all possible ways. All the technical discussions we had enriched my knowledge about the devices, which are his field of expertise. I know I can always rely on him and I could not have finished without him.

My sweetheart, Mona entered my life in the middle of the PhD and I could not have been happier. Her beautiful smile, cute giggles, loving hugs made all difficulties melt away. Returning home after a hard day of work just to start playing with her provided me with unlimited happiness.

I would like to sincerely thank Prof. Denis Flandre, my adviser, for giving me the honor to work with him on this PhD. When I first arrived in Belgium, I had not finished my masters yet. Prof. Flandre kindly provided me with all the means to continue my masters to be able to start the PhD. He offered me an internship opportunity through which I had access to all the necessary software tools in the laboratory that enabled me to finish the work of my masters. During the internship year and the PhD years, he was always there when I needed advice. His deep scientific knowledge, detailed explanations and advice, both

scientifically and administratively helped me achieve this distinguished degree.

Although not officially an adviser, I have always considered Prof. François-Xavier Standaert as a co-adviser of this thesis. He helped me understand all the cryptographic aspects I needed to finish this PhD. We had long fruitful discussions through which he would explain to me all the difficult details and provide me with the necessary materials. Thanks to him, I learned a lot about cryptography which I really enjoy and would like to continue working on.

I would like to sincerely thank Dr. David Bol, committee member and my colleague in the office during the PhD for all the support he gave me. I specially appreciate the long brainstorming discussions that we had and all the advice he gave me. I learnt a lot from his expertise in the digital field. During tape-outs, measurements, writing papers, I knew I could count on his help.

I would also like to thank my committee members, Prof. Jean-Didier Legat, Prof. Christian Piguet, Prof. Ingrid Verbauwhede and Prof. Christophe De Vleeschouwer for accepting to be members of this committee, for the time they gave to read this thesis and for their valuable advice, corrections and recommendations that they gave me during the private defense. Their guidance is of great value to this thesis.

My family members, specially my brother Abdelfattah and my aunt Salwa have always been there for me and constantly encouraged me. Abdelfattah always reminded me of how my father used to be proud of me and this made me really happy. He never misses an opportunity to support me through every achievement. My aunt Salwa used to always call and raise my spirits specially during the hard writing period of the thesis. I will never forget the support and encouragement of my uncles, Sherief, Mostafa and Ahmed, my aunts, Amal and Nahla, my grandmother Malak and my cousins, Ola and Cleopatra.

I cannot forget Prof. Mohamed Dessouky, my masters adviser. I am truly grateful to him for all the support he has given me. I learnt a lot from him, not only during the masters, but also during the last year of college and the graduation project. He gave me the basis of analog circuit design. It was also a great honor to work with him in the former IPD team at Mentor Graphics Egypt.

I would like to thank my parents in law for encouraging me after taking my masters degree to continue and pursue a PhD degree.

I also would like to thank all my former and current colleagues in the laboratory and the CRYPTO group. My office colleagues, Cédric Hocquet, Julian de Vos, Angelo Kuti, François Botman and François Durvaux were always of great help, we shared many scientific discussions and brainstorming. Special thanks goes to my colleagues at the CRYPTO group for all the fruitful

seminars they presented which improved my cryptographic knowledge. I would also like to thank my former CRYPTO colleagues, François Macé and François Gosset for providing me with all the technical materials that helped me a lot start my work in this PhD and for all the scientific discussions we had.

I would like to express my gratitude to all my friends in Belgium that helped me cope with the new environment when I first arrived. They were very supportive and they encouraged me a lot during my PhD. Also, my friends in Egypt provided me with moral and sentimental support. Special thanks goes to my friend Amal Gaafar for her encouragement and the many brainstorming scientific discussions we had.

I have special thanks to Sylvie Baudine, the secretary of the CRYPTO group. The time, patience, administrative and personal advice she gave helped me a lot during my PhD.

I would like to thank the Walloon Region for the financial support of this work through the E-USER project (WIST program) and the SKYWIN S@T project.

Dina Kamel

July 03, 2012

ABSTRACT

Highly power-constrained secure applications such as radio frequency identification (*RFID*) and *contactelss smart cards*, present a great challenge for circuit designers who try to determine the best tradeoff between security and performance. They feature tight power consumption and area budgets while relaxing the timing constraint. Circuit designers are directly involved since "computationally" secure applications are not secure anymore due to the physical information leakage of the hardware implementations that use static CMOS (e.g. the power consumption, the electromagnetic radiation or the timing information). These types of physical information leakage are exploited by side-channel attacks. Consequently, it is the responsibility of circuit designers to develop new techniques that maximize the security of the implementation with minimum cost in terms of power consumption and area in light of the growing concern of variability associated to nano-CMOS technologies as well. Hardware-level solutions, specially dynamic differential logic (*DDL*) styles are seemingly attractive as they tackle the problem directly where it lies and they potentially allow for hardware optimizations to reduce the power consumption and the area. Unfortunately, most of the proposed DDL styles in the literature consume a minimum of 2× the power consumption of static CMOS and also occupy a minimum of 2× its area.

In this context, the work of this PhD thesis aims at optimizing the design of the previously introduced dynamic differential swing-limited logic (*DDSLL*) style for low power, small area and adequate security enhancement compared to static CMOS. The use of complex differential pull down networks (DPDN) to implement n-input functions, sharing the common blocks of DDSLL among the different functions and lowering the voltage swing are the major design guidelines to reduce the power consumption and the circuit footprint. Implementing these techniques on a LP 65 nm CMOS technology, DDSLL consumes $\sim$ power consumption to static CMOS and occupies $\sim$ area. Another important study is the impact of process variations on the power closure of highly power-constrained applications. The dynamic power consumption (P_{dyn}) of complex static CMOS combinatorial circuits suffers from glitch-induced activity factor (α_F) variability that leads to a 75 % increase in P_{dyn} using the 3σ metric for high yield. However, DDSLL features a low impact of process variations on its power consumption (only 3 %) thanks to its deterministic α_F. As for the resistance against side-channel attacks, DDSLL is 10× harder to attack (using worst-case templates) than static CMOS. Even though the security gain of DDSLL is not sufficient compared to other types of countermeasures (e.g. masking - exponential gain and noise addition - linear gain), its low power consumption and area overheads give more room to combine countermeasures.

ACRONYMS

3sDDL	3-state differential dynamic Logic
ABB	Adaptive body biasing
AES	Advanced encryption standard
BCDL	Balanced cell-based dual-rail logic
BDD	Binary decision diagram
CAD	Computer aided design
CD	Critical dimension
CMOS	Complementary metal-oxide semiconductor
CMP	Chemical mechanical polishing
CPA	Correlation power analysis
CRSABL	Charge recycling sense amplifier based logic
D2D	Die-to-die
DDL	Dynamic differential logic
DDSLL	Dynamic differential swing-limited logic
DES	Data encryption standard
DIBL	Drain-induced barrier lowering
DPA	Differential power analysis
DPDN	Differential pull-down network
DRP	Dual-rail precharge
DRSL	Dual-rail random switching logic
DyCML	Dynamic current mode logic
ECC	Elliptic curve cryptography
EM	Electromigration
ESD	Electrostatic discharge
FF	fast NMOSFET, fast PMOSFET
FO	Fan-out
GND	Ground signal
GP	General purpose

HVT	High V_t
I/O	Input/output
IC	Integrated circuit
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
L2L	Lot-to-lot
LER	Line edge roughness
LP	Low power
LVT	Low V_t
MDPL	Masked dual-rail precharge logic
MTD	Measurements to disclosure
NBTI	Negative bias temperature instability
NED	Normalized energy deviation
NMOS	N-channel MOSFET
NSD	Normalized standard deviation
OTV	Oxide thickness variation
PCB	Printed circuit board
PD	Partially depleted
PI	Perceived information
PMOS	P-channel MOSFET
RDF	Random dopant fluctuation
RFID	Radio frequency identification
RO	Ring oscillator
ROBDD	Reduced order binary decision diagram
S-box	Substitution box
SABL	Sense amplifier based logic
SNR	Signal-to-noise ratio
SOI	Silicon-on-Insulator
SPA	Simple power analysis
SRAM	Static random-access memory
SS	Slow NMOSFET, Slow PMOSFET
STTL	Secure triple track logic
SVT	Standard V_t
TT	Typical conditions for NMOSFET and PMOSFET

W2W	Wafer-to-wafer
WDDL	Wave dynamic differential logic
WID	Within die
ZIF	Zero insertion force

LIST OF SYMBOLS

α_F	Activity factor	
β	Coefficient for the linear model of the stochastic attack	
C_c	Cipher text	
$C_{eff\ L}$	Effective load capacitance ($C_L \times \alpha_F$)	F
C_i	Gate input capacitance	F
C_L	Load capacitance	F
clk	Clock signal	
C_{ox}	Oxide capacitance	F/m^2
C_o	Gate output capacitance	F
C_{rout}	Routing capacitance	F
ΔV	Output voltage swing	V
D_{in}	Input signal to the S-box	
D_{out}	Output signal from the S-box	
E_{dyn}	Dynamic energy energy	J
E_{SC}	Short circuit energy	J
E_{SW}	Switching energy	J
f	Frequency	Hz
f_{gen}	Generator frequency	Hz
g	Basis for the linear model of the stochastic attack	
H	Random variable representing either the Hamming weigh or the Hamming distance models	
I_{on}	MOSFET on current	A
I_{off}	MOSFET off current	A
K_s	Secret key	
L	Random variable representing the power traces	

L_g	Gate length	m
μ	Mean value	
$P_{el.\,noise}$	Electronic noise power consumption	W
P_{exp}	Exploitable power consumption	W
P_{dyn}	Dynamic power consumption	W
$P_{sw.\,noise}$	Switching noise power consumption	W
P_{leak}	Leakage power consumpiton	W
P_{tot}	Total power consumption	W
ρ	Correlation coefficient	
σ	Standard deviation value	
T	Temperature	$^{\circ}C$
T_0	Room temperature	$^{\circ}C$
t_d	Propagation delay	s
T_{ox}	Oxide thickness	m
V_{DD}	Supply voltage	V
V_{GS}	MOSFET gate-source voltage	V
V_t	MOSFET threshold voltage	V
V_{tn}	NMOSFET threshold voltage	V
V_{tp}	PMOSFET threshold voltage	V
W	Gate width	m
X	Random variable representing the processed data	

AUTHOR'S PUBLICATION LIST

Journal papers

[1] **Dina Kamel**, M. Renauld, D. Bol, F.-X. Standaert, and D. Flandre, "Analysis of dynamic differential swing limited logic for low-power secure applications," *Journal of Low Power Electronics and Applications, JLPEA*, vol. 1, no. 2, pp. 98–126, 2012.

[2] C. Hocquet, **Dina Kamel**, F. Regazzoni, J.-D. Legat, D. Flandre, D. Bol, and F.-X. Standaert, "Harvesting the potential of nano-CMOS for lightweight cryptography: an ultra-low-voltage 65 nm AES coprocessor for passive RFID tags," *Journal of Cryptographic Engineering*, vol. 1, no. 1, pp. 79–86, Apr. 2011.

Conference proceedings

[3] **Dina Kamel**, F.-X. Standaert, and D. Flandre, "Scaling trends of the AES S-box low power consumption in 130 and 65 nm CMOS technology nodes," in *IEEE International Symposium on Circuits and Systems, ISCAS*, May 2009, pp. 1385 –1388.

[4] **Dina Kamel**, C. Hocquet, F.-X. Standaert, D. Flandre, and D. Bol, "Glitch-induced within-die variations of dynamic energy in voltage-scaled nano-CMOS circuits," in *Proceedings of the European Solid-State Circuits Conference, ESSCIRC*, Sept. 2010, pp. 518 –521.

[5] M. Renauld, F.-X. Standaert, N. Veyrat-Charvillon, **Dina Kamel**, and D. Flandre, "A formal study of power variability issues and side-channel attacks for nanoscale devices," in *EUROCRYPT*, 2011, pp. 109–128.

[6] M. Renauld, **Dina Kamel**, F.-X. Standaert, and D. Flandre, "Information theoretic and security analysis of a 65-nanometer DDSLL AES S-Box," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES*, 2011, pp. 223–239.

[7] **Dina Kamel**, D. Bol, and D. Flandre, "Impact of layout style and parasitic capacitances in full adder," in *IEEE International SOI Conference, SOI*, Oct. 2008, pp. 97 –98.

[8] **Dina Kamel**, D. Bol, F.-X. Standaert, and D. Flandre, "Comparison of ultra-low-power and static CMOS full adders in 0.15 μm FD SOI CMOS," in *IEEE International SOI Conference, SOI*, Oct. 2009, pp. 1 –2.

[9] **Dina Kamel**, M. Dessouky, and D. Flandre, "Enhanced performance of SERDES current-mode output driver using 0.13 μm PD SOI CMOS," in *IEEE International SOI Conference, SOI*, Oct. 2009, pp. 1 –2.

[10] D. Bol, **Dina Kamel**, D. Flandre, and J.-D. Legat, "Nanometer MOSFET effects on the minimum-energy point of 45nm subthreshold logic," in *Proceedings of the International Symposium on Low Power Electronics and Design, ISLPED*, 2009, pp. 3–8.

[11] C. Hocquet, D. Bol, **Dina Kamel**, and J.-D. Legat, "Assessment of 65nm subthreshold logic for smart RFID applications," in *Proceedings of the Workshop Faible Tension, Faible Consommation (FTFC)*, 2009, p. 4.

CHAPTER 1

INTRODUCTION

Low-power (LP) applications have been gaining a lot of attention since their appearance about a decade ago. Among them are radio frequency identifier (RFID) tags [1] which, as the name implies, use radio frequency to identify different types of subjects. Smart cards [2] represent another important category that shares a significant amount of the LP applications market, specially the contactless type. They provide various services such as identification, authentication, data storage and application processing. These low-power applications share the fact that they are highly constrained in terms of power consumption / energy and chip area while featuring loose speed constraints. Certainly, they are power- / energy-constrained because they either harvest power from the environment or operate on small batteries. Also, they require small chip area in order to fit in miniature tags / cards / nodes and to reduce the overall cost of the application.

To be able to lower the power consumption, designers rely on several techniques. Operating at low frequencies is one of the straightforward methods due to the linear relation between the dynamic power consumption (P_{dyn}) and the frequency of operation. For example, in passive RFID tags, the operating frequency of the digital blocks can be as low as 100 kHz [3]. For smart cards, the operating frequency is in the MHz range. For example, in contactless smart cards, such as [2], the clock frequency is 13.56 MHz (compliant with the ISO/IEC 14443 standard [4]). Furthermore, designers developed several power-management tech-

niques such as voltage scaling that aims at meeting tight power constraints. It benefits from the quadratic relation between the dynamic power consumption and the supply voltage (V_{DD}) at the expense of excess delay. Voltage scaling can either be static with a fixed V_{DD} reduction at design time or dynamic with on-demand V_{DD} lowering in low-power modes [5]. Also, the use of advanced technologies further reduces the dynamic power consumption thanks to the shrinking devices, thus gate capacitances. In addition, the availability of multiple process flavors and different threshold voltages (V_t) for transistors allows to minimize leakage power consumption as well.

However, when reaching nanometer-scale geometries for MOSFET devices, variability becomes a serious concern [6] such that circuit designers face a challenging task to meet the market's constrained specifications in presence of variability. Indeed, the delay and the leakage power consumption of nano-CMOS circuits are inevitably influenced to a great extent. Consequently, it is important that the designers take these effects into account during early design stages. For example, cycle time margins are required to accommodate worst-case delay which has been traditionally dealt with at design time by carrying out global process corner simulations. However, the recent increase of uncorrelated variability sources in nanometer CMOS technologies such as random dopant fluctuations (RDF) and line edge roughness (LER) has motivated the development of statistical static timing analysis tools [7]. They help to avoid overestimating margins, by taking averaging effect and parametric yield into consideration. As a result, CAD-tool providers raced to integrate statistical simulation tools that incorporate variability in order to properly model its impact on the different design parameters which allows designers to efficiently characterize the performance of the circuits and minimize the effect of variability at design time.

Another important aspect of such applications is the security level of the implementation. There are many methods to protect an implementation and at different abstraction levels (protocol, e.g. [8], algorithm, e.g. [9] and hardware, e.g. [10]). Previously, the algorithm was considered to be "computationally" secure against classical cryptanalysis if breaking it requires impractical computing power [11]. However, new sources of information leakage emerged at the hardware-level that jeopardize the security of these algorithms. In the late 1990's, side-channel attacks proved to be extremely powerful because they rely on observing and analyzing the inherent physical information leakages of implementations, regardless the strength of the protocol or the algorithm [11]. These information leakages can be the power consumption of the device [12], electromagnetic radiation [13] or timing information [14]. During the same period, fault attacks [15] also became popular. They try to induce errors in the computation to manipulate the functionality of the implementation and recover the secret information.

In this study we are mainly interested in side-channel attacks, especially the ones that involve power analysis [12] because they are easily performed without requiring sophisticated equipment or methods to extract the secret information (basically they need an oscilloscope and a PC to monitor, capture and process

the power information). Therefore, they pose a real threat to the applications that demand certain levels of security. As a result, countless countermeasures arose to mitigate the physical information leakage and they are grouped in two main categories; namely, hiding and masking. Hiding is mainly concerned with eliminating the power consumption dependency on the processed data. On the other hand, masking aims at decorrelating the power consumption and the intermediate values of an implementation by randomizing these intermediate values. Both categories of countermeasures exist at the algorithm-level, the architecture-level and the gate-level. However, these countermeasures cause significant performance degradations, such as large area, high power consumption overheads and low throughput.

Nevertheless, hiding techniques at the gate-level are particularly appealing, more specifically the ones that involve the usage of dedicated logics such as the dynamic differential logic (DDL) styles. They are seemingly attractive as they tackle the problem directly where it lies and they offer a substantial opportunity for optimizing the power consumption and the footprint of the logic. These techniques aim at hiding the secret information by minimizing the signal-to-noise ratio (SNR) through the reduction of the signal [11]. Let us specify here that the signal refers to the variance of the exploitable power consumption and the noise denotes the noise power consumption variance. This hiding technique can be achieved by equalizing the instantaneous power consumption of the implementation during each clock cycle which is the challenging task of the DDL styles.

In the following section we elaborate more on our motivations behind the choice of logic, design methodology, type of analysis and assessment techniques in the next section. Then, we detail our contributions in each aspect; namely, design and performance, variability study and security assessment in Section 1.2. Finally, we describe the thesis outline in Section 1.3.

1.1 MOTIVATION

The work in this dissertation is mainly motivated by the three following concerns:

1. For highly power-constrained secure applications, the need for implementing a low-power, small size, side-channel resistant security module is crucial. Several DDL styles are proposed in the literature, for example, wave dynamic differential logic (WDDL) [16], sense amplifier based logic (SABL) [10], dynamic current mode logic (DyCML) [17], ... etc. However, these styles suffer from high power consumption and area overheads compared to the traditional static CMOS (a minimum factor of 2 for both the power consumption and area). Moreover, they do not completely fulfill the primary requirement which they were employed for and that is, balancing the instantaneous power consumption during each clock cycle, rendering the power consumption independent on the data being processed. Consequently, an important research goal is to determine the best tradeoff

between security and performance, specially for highly power-constrained secure applications.

Therefore, we took the work that has been done by Hassoune *et al.* in [18] as a starting point. They proposed a new DDL family that is based on low-swing operation in order to reduce the power consumption of the implementation. Also, based on preliminary metrics (that later on proved to be inaccurate), the new DDL family features a security gain compared to other DDL styles. Based on these givens, we decided to adopt the dynamic differential swing-limited logic (DDSLL) that belongs to this DDL family and further improve its design and performance in order to achieve the best security - performance tradeoff.

2. With the rising concern of process variations in nano-scale technologies, thorough studies have been made on the effect of variability on delay, which is important for timing closure [19] and functionality [20], and on the leakage power. However, dynamic power has been considered as weakly sensitive to variability [21, 22], until recently we reported in [23] that dynamic power consumption of combinatorial circuits is susceptible to process variations as well due to glitch-induced activity factor (α_F) fluctuations, especially at low voltages. This is particularly important because large deviations in the dynamic power threatens the power closure of highly power-constrained applications.

   Also to our knowledge, there has not been any variability studies for any of the proposed DDL styles in the literature. Since the power consumption of DDSLL is dominated by dynamic power even at low frequencies (as will be explained later in Chapter 2), it is interesting to study how its dynamic power behaves in presence of variability, knowing that the dynamic power of static CMOS suffers from glitch-induced α_F variations.

3. Several techniques are proposed to counteract the growing threat of side-channel attacks at different abstraction levels. Special attention is given to the hardware-level countermeasures that aim to solve the problem directly where it lies. Masking and hiding are the main two categories of the countermeasures against side-channel attacks. Furthermore, hiding that targets the power consumption directly, is further subdivided into two classes whose goal is to reduce the signal-to-noise (SNR) ratio. First, noise addition that is concerned with increasing the noise level. Second, DDL styles that try to equalize the power consumption in each clock cycle. Unfortunately, the use of these countermeasures lead to significant performance degradations, such as high power consumption overheads and low throughput as well as larger area penalty. Since DDSLL shows a significant reduction of the power consumption and area overheads compared to other DDL styles, it may be considered as a potential *efficient* countermeasure against side-channel attacks for highly power-constrained secure applications.

In this dissertation we try to answer the following question: "What is the numerical amount of security gain brought by the use of DDL styles in general and more specifically, DDSLL?" This question is brought up in the context of comparing DDLs to masking and noise addition countermeasures. Masking is known to increase the security gain exponentially with the order of the mask (if the amount of noise present is large enough) [24]. On the other hand, noise addition shows a linear increase of security gain. Meanwhile, DDL styles increases the security gain by adding a constant compared to an unprotected implementation.

Until recently, no unified method was used to quantify the security gain of these DDL styles, each one or group are characterized individually. This makes it extremely difficult to point out countermeasure that is superior to all others. As a result, a unified framework is introduced in [25] to solve this problem and we employ it in our security assessment. Briefly, it separates two important, usually confused, issues, namely the evaluation of the implementation ("How good is the implementation ?") and the evaluation of the adversary ("How good is the adversary ?"). Using an information theoretical analysis, a worst-case adversary (e.g. profiled template attacks [26]) is considered in order to fairly evaluate different countermeasures. A security analysis is then used to numerically evaluate the strength of a particular adversary to mount a successful attack using the measurement to disclosure (MTD) metric.

Another pressing issue that needs to be addressed, especially in nano-scale technologies, is the impact of process variations on security. To the best of our knowledge, the simulated experiments in [27] are the only available reference. Using their own developed security metric, the authors in [27] showed that process variability deteriorates the security of both static CMOS and SABL (as a representative of DDL styles) in the same way. However, these conclusions do not reflect the effect of variability on actual attack scenarios. Therefore, in this dissertation we try to qualitatively and numerically evaluate the impact of process variations on the security of the static CMOS S-box implemented in LP 65 nm CMOS technology using the unified framework introduced in [25].

1.2 CONTRIBUTIONS

In view of the current need of implementing secure modules at the gate-level in highly resource constrained applications, mainly in terms of power consumption and chip area, this dissertation focuses on the following:

1. Developing a compact DDL style that consumes less power than static CMOS, that is the dynamic differential swing limited logic (DDSLL) style.

   The DDSLL style was first realized by Hassoune *et al* [18]. It is mainly a low-swing DDL style that features a feed-back circuitry to cut off the current

source once the evaluation is done, thus limiting the static current. It was designed to consume low power, yet it was never compared to static CMOS. Additionally, the performances of DDSLL were evaluated using simulation results and still needed to be validated with real measurements.

In this dissertation, we take additional steps in the design of the DDSLL style in order to further reduce its power consumption below that of static CMOS using some well known and other new proposed techniques [28] summarized as follows:

- Using an advanced technology node in order to reduce the dynamic power by lowering the gate capacitance and also the routing capacitance, since as the design shrinks the connecting routes are shorter, hence lower routing capacitance. As a result, good layout techniques can benefit from the shrinking of the routing lengths and lead to reduced routing capacitance. In this dissertation, the 65 nm CMOS technology is used as recommended by [29].
- Using the low-power (LP) flavor provided by the 65 nm CMOS technology in order to reduce the static power as this provides longer channel length transistors with larger oxide thickness and higher V_t to lower the subthreshold and the gate leakage currents [29].
- Using complex differential pull-down networks (DPDN) in the design of n-input functions, thus limiting the need of using several gates to implement complex n-input functions which results in lower power consumption.
- Sharing some resources among the DDSLL blocks, mainly the feedback circuitry, the dynamic current source and the self-timing buffer, hence, saving more power consumption.
- Optimizing the voltage swing to lower the power consumption of DDSLL without jeopardizing its functionality.

In this context a design methodology is proposed to design a compact low-power DDSLL AES S-box using the LP 65 nm CMOS technology [28]. The performance of the DDSLL S-box is compared to SABL and DyCML S-boxes as well as two version of static CMOS S-boxes; namely a 2-input static CMOS S-box that limits the gates to simple 2-input gates and a 4-input static CMOS S-box that uses 4-input functions as needed (similar to the DDL styles). Simulation results at typical conditions (1.2 V, 25°C and TT conditions) without adding the parasitic routing capacitances - only gate capacitances are accounted for, are plotted in Fig. 1.1. It shows that the power consumptions of DDSLL and DyCML S-boxes are similar and they are about 1.9× lower than that of the SABL S-box across the frequency range. Also, when compared to static CMOS, they feature a 2× and 1.5× reduction in power consumption at 100 kHz, and 1.45× and 12 % reduction in power consumption above 1 MHz compared to the 2-input and 4-input versions of static CMOS, respectively. It is worth noting

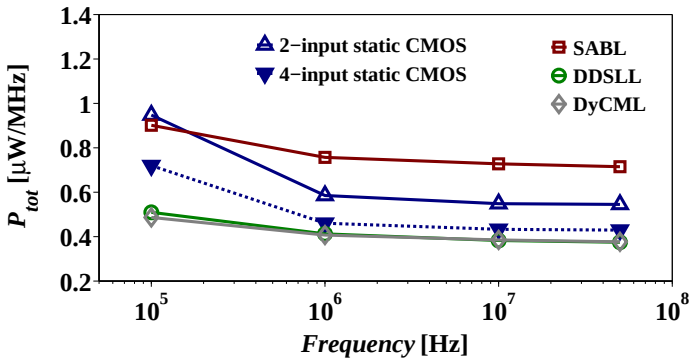


Fig. 1.1.: SPICE simulations of total power consumption (without adding the extracted routing parasitics) *versus* frequency of 2-input and 4-input static CMOS, SABL, DyCML and DDSLL S-boxes (average over 10 input transitions).

that the SABL S-box consumes similar power consumption to the 2-input static CMOS at 100 kHz and about 31 % higher power compared to the 4-input version. We further validate these results by performing measurements using the DDSLL and 2-input static CMOS S-boxes. Under the same conditions as simulations, the measured power reduction brought by the DDSLL S-box is found to be 35 % which is fairly predicted by post-layout simulation (37.8 %), compared to 2-input static CMOS S-box at 100 kHz.

Another interesting point that DDSLL achieve is the reduction of the leakage power as seen by the reduced slope below 1 MHz. Rough calculations show that 2-input and 4-input static CMOS, SABL, DyCML and DDSLL S-boxes consume 40 nW, 28 nW, 16 nW, 11 nW and 9 nW leakage power, respectively. This advantage is particularly useful at high temperatures as we show that the power consumption reduction brought by DDSLL over static CMOS increases even more with temperature thanks to its lower leakage power (the ratio $(P_{tot}|_{staticCMOS}/P_{tot}|_{DDSLL})$ increases from 1.5× to 4× as the temperature rises from $25°C$ to $125°C$).

On the other hand, the SABL, DDSLL and DyCML S-boxes have a delay penalty of 70 %, 2.5× and 6.7×, respectively when compared to either versions of the static CMOS S-boxes which have comparable delays. However, the delay overhead of DDSLL is considered moderate and still respects the loose timing constraints of the low-power applications. As for the area, due to the compact implementation of the DDSLL S-box (using the sharing principle and implementing complex functions with 4-input DPDNs), its area is 1125 μm^2 which is only 12 % larger than its counterpart in static CMOS (2-input version). Also, an important fact about DDSLL is that it is designed in a full-custom fashion which means a longer time-to-market and higher design costs compared to semi-custom designed logic styles.

Therefore, we demonstrate that reducing the power consumption of DDL styles that use the DPDN concept in general can be achieved by applying three main steps; share the common blocks, implement the functions as complex DPDNs when necessary and reduce the voltage swing [28]. The first two steps rendered the SABL S-box more power efficient than the original version [10] ($\sim 2\times$). The final step led to further reduction in the power consumption such that DDSLL and DyCML that are both low-swing logic styles feature $1.85\times$ less power than SABL. As a result, we prove that DDSLL can achieve similar or even less power consumption than static CMOS if properly designed using these steps. Despite the higher delay penalty and design costs (due to being full-custom), DDSLL features low power and occupies a small footprint relative to other DDL styles, which renders it a potential candidate for low-power applications.

2. Study the impact of glitch-induced within-die variations on the dynamic (energy) power consumption of voltage-scaled nano-CMOS circuits as an extension to the work that we did in [23]. We also investigate the effect of process variations on the DDSLL style, with respect to static CMOS. To our knowledge, this is the first variability study performed on DDSLL as a representative of DDL styles.

   In the first part of the variability analysis, we thoroughly study the dynamic energy (power) variations in voltage-scaled logic circuits; namely ring oscillators and static CMOS S-box implemented in LP 65 nm CMOS technology which represent two different types of logic circuits. The dynamic power consumption of an n-gate static CMOS circuit is given by:

$$P_{dyn} = V_{DD}^2/2 \times f_{clk} \times \sum_{j=1}^{n} (\alpha_{F,j}.C_{L,j}) \tag{1.1}$$

   where V_{DD} is the supply voltage, f_{clk} denotes the clock frequency, $\alpha_{F,j}$ represents the activity factor and $C_{L,j}$ is the load capacitance of the j^{th} gate.

   We report that uncorrelated WID delay variability significantly magnifies the dynamic energy (power) variations of combinatorial circuits at low voltages via the activity factor (α_F) because of the generated random glitches [23]. We measured relative E_{dyn} standard deviation up to 8 % for the static CMOS S-box using input patterns with 256 transitions at 0.4 V. However, worst-case input transitions also exist that feature higher E_{dyn} variations. Due to the random nature of the glitch-induced α_F variability, it is subject to averaging over the number of input transitions. Therefore, averaging over a large number of input transitions can lead to a false sense of variability tolerance. Moreover, the glitch-induced α_F phenomenon is not restricted to low-voltage operation. As technology scales further, E_{dyn} (P_{dyn}) is expected to gain more sensitivity to the glitch-induced α_F variability even while the operation is at nominal supply voltage. From this study,

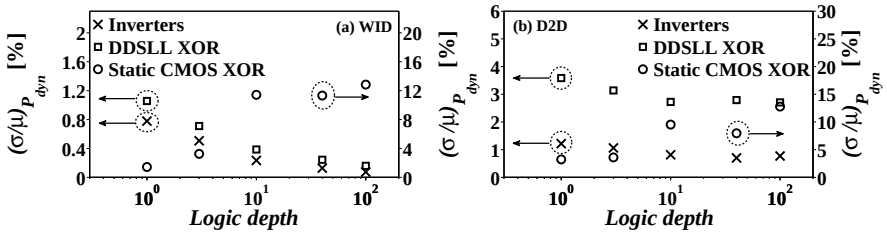


Fig. 1.2.: SPICE simulations of dynamic power (a) WID and (b) D2D variability of inverter chain, static CMOS XOR chain and DDSLL XOR chain with different logic depths 25 °C, 1.2 V supply (100 Monte-Carlo runs with local process variations).

we show that dynamic energy (power) variability cannot be neglected in combinatorial circuits and that statistical simulations are required for properly capturing worst-case energy (power) consumption.

In the second part of the variability study, we assess the impact of process variations on the power consumption [28], delay and functionality at low supply voltages of the DDSLL style with respect to static CMOS. We first demonstrate in Fig. 1.2 (a) the averaging out of the WID P_{dyn} variability of DDSLL with the logic depth (of course true for circuit size as well) similar to the chain of inverters, thanks to the deterministic α_F of the DDSLL style. Also in Fig. 1.2 (b), DDSLL features nearly constant D2D P_{dyn} variations, independent on the logic depth, again in accordance with the chain of inverters due to its deterministic α_F. On the other hand, static CMOS suffers from an accelerating WID and D2D dynamic power variabilities for increasing logic depth because of the glitch-induced α_F variations. The importance of taking the P_{dyn} variability into account is to meet the power specification set by the highly power-constrained applications. Considering the 3σ metric for high yield, the dynamic power of the static CMOS S-box rises 75 %, whereas the percentage increase is only 3 % in the case of the DDSLL S-box. Consequently, in case of tight power budgets, DDSLL is expected to meet power closure with minimum power deviations, whereas static CMOS can potentially jeopardize it due to its high dynamic power variability, not to mention the leakage power variability as well.

Second, we show that the delay is also affected by the glitch-induced α_F variability. The WID delay variability of static CMOS circuits was expected to average out with the logic depth because of the random nature of the C_L and I_{on} WID variabilities. Also, the D2D delay variability was expected to be independent on the logic depth since both C_L and I_{on} D2D variations are correlated. However, in the case of combinatorial static CMOS circuits, glitch-induced α_F variability causes the WID and D2D

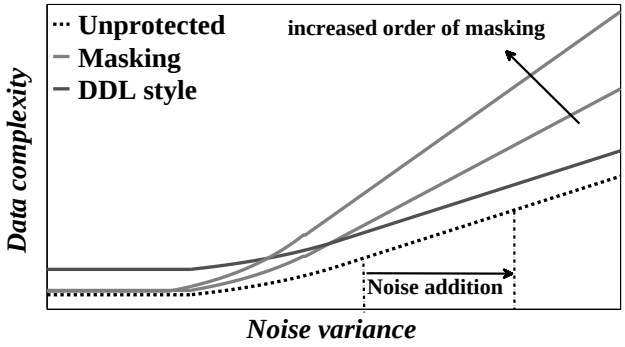


Fig. 1.3.: Data complexity in function of the noise for different types of countermeasures; namely, masking with different orders, noise addition and using DDL styles.

delay variabilities to increase with the logic depth, similar to its effect on dynamic power.

Finally, we prove that the functionality of the DDSLL at low voltage is lost due to WID process variations. This is compliant with the claim in [30] that dynamic circuits are vulnerable to variability because the information is saved as charge on the output node capacitor. Moreover, DDSLL suffers from a diminishing voltage swing at low supply voltages that also contribute to its sensitivity to WID process variations. Therefore, we propose several techniques to overcome this problem that represent an interesting research topic for the future.

3. Assess the security of the DDSLL S-box implemented in LP 65 nm CMOS technology [31], using the methodology proposed by Standaert *et al* in [25] in order to answer the raised question in Section 1.1. That is, "What is the numerical amount of security gain brought by the use of DDL styles in general and more specifically, DDSLL?". We also analyze the impact of variability on the security of the static CMOS S-box implemented in LP 65 nm CMOS technology [32].

   In a worst-case scenario, we show that the security enhancement brought by the use of DDSLL over static CMOS against the most powerful adversary who uses profiled template attacks [26], is one order of magnitude in terms of MTD [31]. This value is rather modest compared to masking that features an exponential gain as the order of mask increases and noise addition that provides a linear increase of security gain as shown in Fig. 4.1. Nevertheless, the low power and area overheads of DDSLL give room for adding another countermeasure, for example, masking.

In non-worst-case scenarios, actual adversaries that use non-profiled distinguishers (they do not have the full capability to profile the leakage functions of the implementation in an off-line phase) face new challenges. More specifically, the non-linearity of the leakage functions and the growing concern of process variations in sub-100 nm technologies.

First, we tackle the non-linearity of the leakage function of the DDSLL style [31]. Although it does not affect the security analysis performed using the optimal adversary in a worst-case scenario, actual adversaries that use non-profiled distinguishers (they do not have the full capability to profile the leakage function of the implementation in an off-line phase) encounter great challenges. Indeed, they rely on the fact that the leakage functions of the attacked implementations are linear [33], thus lowering the chance of a successful attack. The stochastic approach [34] is a powerful tool to analyze the linearity nature of the leakage functions because it is a linear, regression-based approach that approximates the leakage function with a linear function. In this study, we consider the "on-the-fly" variant of the stochastic approach proposed in [35]. Simulation results show that the leakages of the DDSLL S-box are completely non-linear. Contrarily, actual leakages are rather linear except of a few samples that feature some non-linearity. This is attributed to the measurement artifacts that are not taken into consideration during simulations. As a result, practical attacks are feasible against DDSLL even if only a few linear leakage samples are available. Therefore, we raise an open question whether it is possible that designers consider non-linearity as a design guideline while developing secure logic styles.

Second, process variations associated with the scaling of CMOS technology impacts both types of distinguishers, profiled and non-profiled. In the case of profiled distinguishers, we report that extending template attacks to infer process variations causes a significant loss of information as a result of profiling more than one die, so the template is not accurate for any die. Subsequently, actual adversaries using profiled attacks face a challenging task trying to build accurate templates, due to technology scaling. Nevertheless, the use of profiled template attacks is still important in the worst-case scenario because they provide the lower limit which the designer should consider while evaluating an implementation. As for the non-profiled distinguishers, we show that variability causes the correlation power analysis (CPA) attacks [36] using the Hamming weight model to fail due to the deficiency of the model to accurately estimate the leakage function of the static CMOS S-box at 65 nm CMOS technology and clearly beyond. Contrarily, "on-the-fly" stochastic attacks are completely successful since in presence of process variations, the leakage functions of each die are still linear and can be separately estimated by a linear function. Therefore, we conclude that process variations necessitate the use of new distinguishers that are capable of modeling the leakage function of the target die "on-the-fly".

1.3 THESIS OUTLINE

Chapter 2. The concept of dynamic differential logic styles is briefly explained to manifest the challenges the designers experience while developing such logic styles and the pitfalls some logic styles feature. State of the art DDL styles are shown with the focus on SABL and DyCML as we use them as references to compare the performance of DDSLL. We then describe the original circuit topology of DDSLL proposed by Hassoune *et a.* in [18] and its functional operation followed by the methodology we adopted to design compact low-power DDSLL style circuits. The design phase is divided into the following stages barring in mind the two design goals that are minimizing the power consumption and the area:

- Decide upon technologically related and transistor sizing selections.
- Design the DPDNs of the complex functions based on the use of binary decision diagrams (BDD)s [37] with special care taken to symmetrize the DPDNs and the differential routes.
- Identify the blocks that can share the available resources.
- Design the interface circuit with static CMOS.

During the design of the DDSLL S-box, we implemented two versions, one with compact DPDNs that employs some of the balancing techniques in order to limit the power consumption and the area overheads, the other one with balanced DPDNs that uses all the balancing techniques described in this dissertation. We show that the latter still suffers from data-dependent power consumption (slightly lower than the compact version) due to imbalances in the internal node capacitances of some of the DPDNs with a 7.15 % and 12.6 % power consumption and delay penalties, respectively. Therefore, we decided to focus on the compact version of the DDSLL S-box. The area of the DDSLL S-box is 1125 μm^2 which is only 12 % larger than its counterpart in static CMOS. We then start the performance analysis phase by describing the power consumption and delay test-benches that we used, in addition to the measurement setup. We show the importance of the sharing principle where the DDSLL that uses it features 2.7× less power than that without sharing. We then detail the performance results at the simulation level (pre- and post-layout) and at the measurement level. Measurement results prove to correspond well to simulations, they respect the ratio of the power consumption and the delay between both S-boxes despite somehow deviating from the absolute numerical values. The power consumption results are briefly discussed in Section 1.2. As for the delay, the DDSLL S-box suffers from 2.5× higher delay that static CMOS. We also analyze the effect of temperature on the power consumption and the delay of the S-boxes. We report that the power consumption reduction brought by DDSLL over static CMOS increases even more with temperature thanks to its lower leakage power (1.5× to 4× as the temperature rises from 25°C to 125°C). On the other hand, the delay

of both S-boxes is more insensitive to temperature. We also perform a PVT analysis where we show that the power consumption of DDSLL features less sensitivity to process corners as the temperature is raised compared to static CMOS.

Chapter 3. The sources of variability are first identified to give a clear insight of the importance of studying its effects in nano-scale technologies. Then we divide the variability study into two parts. In the first part we address the impact of variability on the dynamic power and energy of voltage-scaled nano-CMOS circuits. We demonstrate how the ring oscillators represent a perfect test-case to study the effect of variability on the load capacitance (C_L) as it is the only factor that plays a role in the variation of the dynamic power / energy. We reinforce the usual belief that the with-in-die (WID) C_L variations are averaged out over the number of stages in the ring oscillator, whereas the correlated die-to-die (D2D) C_L variability is constant, independent on the number of stages in the ring oscillator and independent on the supply voltage. We then investigate the impact of the glitch induced α_F variability on the dynamic power consumption of the AES S-box that exemplifies complex combinatorial circuits. The main conclusion of this study is briefly discussed in Section 1.2. In the second part of this chapter we compare the power consumption and delay performances as well as the functionality at low supply voltages of DDSLL to static CMOS in presence of variability. We report that due to its deterministic α_F, WID variations of the dynamic power of DDSLL averages out over logic depth and circuit size and D2D variations remain constant and independent on logic depth and circuit size. These phenomena are illustrated in Section 1.2 along with a short description of their impact on DDSLL and static CMOS S-boxes. As for the delay, we reveal the impact of glitch-induced α_F variations on static CMOS circuits, where due to WID and D2D process variations, the delay deviations increase rapidly with the logic depth. However, DDSLL maintains an averaged out and constant WID and D2D delay variabilities with the logic depth, respectively. Finally explain the reason behind the failure of the DDSLL S-box to operate at low supply voltages and provide some propositions to solve this problem.

Chapter 4. The preliminaries and definitions related to the security assessment are first detailed to set a basic foundation for circuit designers who are not familiar with the security aspects. We start by describing the attack environment related to side-channel analysis, following a brief discussion of the models that are normally used by actual attackers. We then list the different types of distinguishers that we employ in our study grouped as profiled and non-profiled. The main classes of countermeasures are described; namely, masking and hiding with its subclasses, noise addition and DDL styles. We note that masking brings an exponential increase of security as the order of the mask increases and the noise addition linearly enhances the security, while the DDL styles add a constant to the security of the unprotected implementation. As a result, we investigate how significant is this constant brought by the use of DDSLL with respect to the

other categories of countermeasures in light of the low performance overheads it achieves [31]. To achieve this goal, we define the security tools and metrics that we employed. We use three tools that have their dedicated metrics; namely, observation of the power traces and compute the standard deviation over the different inputs as the metric, application of the information theoretic analysis [25] using a worst-case scenario and its metric is the perceived information [32] and finally using the security analysis to numerically assess the different types of distinguishers with the MTD metric [25]. Using the profiled template attacks in a worst-case scenario, the DDSLL is found to improve the security by one order of magnitude (MTD) compared to static CMOS and it features similar trends as SABL and DyCML styles using the perceived information metric. We take the security assessment one step further and investigate the challenges encountered by actual (suboptimal) adversaries. That is, the effect of the non-linearity nature of the leakage functions and the growing concern of process variations in sub-100 nm technologies. The impact of these two aspects are briefly demonstrated in Section 1.2.

Bibliography

[1] R. Weinstein, "RFID: A technical overview and its application to the enterprise," *IT Professional*, vol. 7, no. 3, pp. 27–33, 2005.

[2] Sony. (2010) Near field communication (NFC) dynamic tag: RC S801. [Online]. Available: http://www.sony.net/Products/felica/business/tech-support/data/fp_rcs801_1.02.pdf

[3] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, "Strong authentication for RFID systems using the AES algorithm," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES*, 2004, pp. 357–370.

[4] ISO. [Online]. Available: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=50942

[5] T. Burd, T. Pering, A. Stratakos, and R. Brodersen, "A dynamic voltage scaled microprocessor system," *IEEE Journal of Solid-State Circuits, JSSC*, vol. 35, no. 11, pp. 1571 –1580, Nov. 2000.

[6] A. Asenov, A. Brown, J. Davies, S. Kaya, and G. Slavcheva, "Simulation of intrinsic parameter fluctuations in decananometer and nanometer-scale MOSFETs," *in IEEE Transactions on Electron Devices, TED*, vol. 50, no. 9, pp. 1837 – 1852, Sept. 2003.

[7] S. Nassif, K. Bernstein, D. Frank, A. Gattiker, W. Haensch, B. Ji, E. Nowak, D. Pearson, and N. Rohrer, "High performance CMOS variability in the 65nm regime and beyond," in *IEEE International Electron Devices Meeting, IEDM*, Dec. 2007, pp. 569 –571.

[8] S. Dziembowski and K. Pietrzak, "Leakage-resilient cryptography," in *FOCS*, 2008, pp. 293–302.

[9] L. Goubin and J. Patarin, "DES and differential power analysis (the "duplication" method)," in *Proceedings of the First International Workshop on Cryptographic Hardware and Embedded Systems, CHES*. London, UK: Springer-Verlag, 1999, pp. 158–172. [Online]. Available: http://portal.acm.org/citation.cfm?id=648252.752372

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

[11] S. Mangard, E. Oswald, and T. Popp, *Power analysis attacks - revealing the secrets of smart cards*. Springer, 2007.

[12] P. C. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *CRYPTO*, 1999, pp. 388–397.

[13] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, "The EM side-channel(s)," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES*, 2002, pp. 29–45.

[14] P. C. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," in *CRYPTO*, 1996, pp. 104–113.

[15] D. Boneh, R. A. DeMillo, and R. J. Lipton, "On the importance of checking cryptographic protocols for faults (extended abstract)," in *EUROCRYPT*, 1997, pp. 37–51.

[16] K. Tiri and I. Verbauwhede, "A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation," in *Proceedings of Design, automation and test in Europe, DATE*, 2004, pp. 246–251.

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

[18] I. Hassoune, F. Macé, D. Flandre, and J.-D. Legat, "Dynamic differential self-timed logic families for robust and low-power security ICs," *Integration, the VLSI Journal*, vol. 40, no. 3, pp. 355–364, 2007.

[19] N. Verma, J. Kwong, and A. Chandrakasan, "Nanometer MOSFET variation in minimum energy subthreshold circuits," *IEEE Transactions on Electron Devices, TED*, vol. 55, no. 1, pp. 163 –174, Jan. 2008.

[20] J. Kwong and A. Chandrakasan, "Variation-driven device sizing for minimum energy sub-threshold circuits," in *Proceedings of the International Symposium on Low Power Electronics and Design, ISLPED*, Oct. 2006, pp. 8 –13.

[21] D. Blaauw, K. Chopra, A. Srivastava, and L. Scheffer, "Statistical timing analysis: From basic principles to state of the art," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, TCAD*, vol. 27, no. 4, pp. 589 –607, Apr. 2008.

[22] A. Srivastava, D. Sylvester, and D. Blaauw, *Statistical analysis and optimization for VLSI: timing and power*, ser. Series on integrated circuits and systems. Springer, 2005. [Online]. Available: http://books.google.be/books?id=WqsQTyOu5jwC

[23] D. Kamel, C. Hocquet, F.-X. Standaert, D. Flandre, and D. Bol, "Glitch-induced within-die variations of dynamic energy in voltage-scaled nano-CMOS circuits," in *Proceedings of the European Solid-State Circuits Conference, ESSCIRC*, Sept. 2010, pp. 518 –521.

[24] F.-X. Standaert, N. Veyrat-Charvillon, E. Oswald, B. Gierlichs, M. Medwed, M. Kasper, and S. Mangard, "The world is not enough: Another look on second-order DPA," in *ASIACRYPT*, 2010, pp. 112–129.

[25] F.-X. Standaert, T. G. Malkin, and M. Yung, "A unified framework for the analysis of side-channel key recovery attacks," in *Proceedings of the 28th Annual International Conference on Advances in Cryptology: the Theory and Applications of Cryptographic Techniques*, ser. EUROCRYPT. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 443–461. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-01001-9_26

[26] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES*, 2002, pp. 13–28.

[27] L. Lin and W. Burleson, "Analysis and mitigation of process variation impacts on power-attack tolerance," in *Proceedings of the 46th ACM/IEEE Design Automation Conference, DAC*, Jul. 2009, pp. 238 –243.

[28] D. Kamel, M. Renauld, D. Bol, F.-X. Standaert, and D. Flandre, "Analysis of dynamic differential swing limited logic for low-power secure applications," *Journal of Low Power Electronics and Applications, JLPEA*, vol. 1, no. 2, pp. 98–126, 2012. [Online]. Available: http://www.mdpi.com/2079-9268/2/1/98/

[29] D. Kamel, F.-X. Standaert, and D. Flandre, "Scaling trends of the AES S-box low power consumption in 130 and 65 nm CMOS technology nodes," in *Proceedings of IEEE International symposium of circuits and systems, ISCAS*, May 2009, pp. 1385 –1388.

[30] S. Ghosh and K. Roy, "Parameter variation tolerance and error resiliency: New design paradigm for the nanoscale era," *Proceedings of the IEEE*, vol. 98, no. 10, pp. 1718 –1751, Oct. 2010.

[31] M. Renauld, D. Kamel, F.-X. Standaert, and D. Flandre, "Information theoretic and security analysis of a 65-nanometer DDSLL AES S-Box," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES*, 2011, pp. 223–239.

[32] M. Renauld, F.-X. Standaert, N. Veyrat-Charvillon, D. Kamel, and D. Flandre, "A formal study of power variability issues and side-channel attacks for nanoscale devices," in *EUROCRYPT*, 2011, pp. 109–128.

[33] N. Veyrat-Charvillon and F.-X. Standaert, "Generic side-channel distinguishers: Improvements and limitations," in *CRYPTO*, 2011, pp. 354–372.

[34] W. Schindler, K. Lemke, and C. Paar, "A stochastic model for differential side channel cryptanalysis," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES, Springer, LNCS 3659*. Springer, 2005, pp. 30–46.

[35] K. Lemke-rust, "Models and algorithms for physical cryptanalysis," Ph.D. dissertation, University of Bochum, Bochum, 2007. [Online]. Available: http://www.emsec.rub.de/media/crypto/attachments/files/2010/04/thesis_lemke_rust.pdf

[36] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES*, 2004, pp. 16–29.

[37] S. Akers, "Binary decision diagrams," *IEEE Transactions on Computers*, vol. C-27, no. 6, pp. 509 –516, Jun. 1978.

CHAPTER 2

DYNAMIC DIFFERENTIAL SWING LIMITED LOGIC (DDSLL): DESIGN AND PERFORMANCE

2.1 INTRODUCTION

Low-power applications that require a certain amount of security such as passive Radio Frequency IDentification (RFID)[1–3] and smart cards [4, 5] feature loose constraints in terms of speed performance, but are highly challenging in terms of power consumption and chip area. Hence, to reduce the power consumption, designers either reduce the power supply or the operating frequency or both. For example, in passive RFID tags, the operating frequency of the digital blocks can be as low as 100 kHz [6]. For smart cards, the operating frequency is in the MHz range, for example in contactless smart cards such as [4, 7], the clock frequency is 13.56 MHz compliant with the ISO/IEC 14443 standard [8]. Another important aspect of such applications is the security level of the implementation [6, 9, 10]. To obtain a secure chip, a combination of light-weight low-power secure protocol, algorithm and hardware should be implemented.

Currently, static CMOS logic is widely used in digital circuit design for low-power (LP) applications due to its inherent low power, small size (compared to dynamic differential logic styles (DDL)), robustness to voltage scaling and scalability with technology. It is also highly automated in CAD tools which makes it an ideal choice for logic functions. However for secure applications, static CMOS logic is not suitable, since its dynamic power consumption is highly correlated to the processed data as will be explained in Chapter 4. Such a deficiency leads to leaking information about the secured data through analyzing the instantaneous

power consumption. Resulting deciphering is called power-analysis (PA) attacks [11]. There exist many solutions (protocol-level, e.g., [12], algorithm-level, e.g., [13], hardware-level, e.g., [14]) which allows improving security against such attacks. However, these solutions in general cause significant performance degradations (power consumption, chip area and latency overheads).

As a result, an important research goal is to determine the best security *versus* performance tradeoff. We consider hardware-level solutions, in particular DDL styles that are intuitively attractive as they tackle the problem directly where it lies. More specifically, we study the dynamic differential swing-limited logic (DDSLL) that was developed in [15]. As the name implies, DDSLL is a low-swing logic which suggests a potential reduction in power consumption. Of course, several logic styles are proposed in the literature, such as sense amplifier based logic (SABL) [14], wave dynamic differential logic (WDDL) [16], charge-recycling SABL [17], dynamic current mode logic (DyCML) [18], . . . etc. However, all these styles prove to be much less efficient than static CMOS because of their high power consumption, area and/or latency overheads, making them unattractive for low-cost, low-power applications. Indeed, one exception to the above mentioned logic styles is the DyCML that is actually low-swing, hence similar to DDSLL, it is expected to reduce the power consumption overhead. In fact, simulation results of a Khazad S-box [19] using 0.13 μm 1.2 V CMOS PD SOI technology show that DDSLL consumes comparable power to that of DyCML while having a reduction in delay ranging from a factor of 2.2 - 5.2, depending on the output swing of the DyCML [15]. These results imply that DDSLL can be an interesting solution if it proves to enhance the security as well. Anyway, we study the security aspect of DDSLL based on detailed assessments using security tools and metrics in Chapter 4.

Consequently, in this chapter we focus on the design of DDSLL with the aim to achieve similar or even better performances compared to static CMOS, keeping in mind the main goal of DDSLL as a DDL style which is, eliminating the data dependency of the power consumption. For this purpose, we use the advanced encryption standard (AES) [20] S-box as a case study. We start by introducing our design methodology to achieve these goals in the first part of this chapter as an extension to the work presented in [21]. The initial step is the circuit design where we first select an appropriate technology. Here, we choose the 65 nm CMOS technology with the low-power (LP) flavor as suggested by [22]. Then, we discuss how the different transistors of a DDSLL gate are sized in order to optimize the power consumption - delay tradeoff. The second step in our methodology is the design of the differential pull-down network (DPDN). This is considered a crucial step as it involves proper balancing of the differential nodes in order to reduce the power consumption data dependency. We employ the binary decision diagram (BDD) [23] to create and optimize the DPDN of the different functions of the DDSLL S-box. However, achieving fully balanced DPDNs is not just a tedious job, but also it is power and area consuming. We performed preliminary simulation on two versions of the DDSLL S-box, one with balanced DPDNs and the other with more compact DPDNs. By visually

inspecting the current traces and calculating the standard deviation over the inputs, we see that the balanced DDSLL S-box brings only a slight reduction in the power consumption sensitivity to the input pattern at the expense of 7.15 % increase in total power compared to the compact DDSLL S-box. Consequently, an interesting research topic would be to investigate other balancing techniques with the goal of reducing the power consumption dependency on the data to a minimum level without increasing the performance overheads. Anyway, our focus for the remainder of the chapter will be on the compact DDSLL S-box.

The third step is to maximize the efficiency of the DDSLL by sharing the common blocks. By this we mean that several gates operating at the same time can share the self-timing, feed-back and parts of the precharge circuitry of the DDSLL. Hence, sharing contributes to the reduction of both the area and the power consumption of the DDSLL S-box. Simulation results show that the DDSLL S-box that uses the sharing principle features 2.7× less power than that without sharing. The last step of the methodology discusses how to interface DDSLL gates with static CMOS. The output of DDSLL needs a special buffer to connect to static CMOS. To be able to do that, we suggest to not only transform the low-swing signal to a full-swing signal, but also to eliminate the precharge phase of the signal. That is, preserve the evaluated output signals during precharge.

The second part of this chapter investigates the performance of the DDSLL style with respect to static CMOS and also SABL and DyCML styles [21]. It is worth mentioning that the static CMOS S-box was designed in this work in two versions; a 2-input static CMOS S-box that limits the gates to simple 2-input gates and a 4-input static CMOS S-box that uses 4-input functions as needed (similar to the DDL styles). Simulation results at typical conditions (without adding the parasitic routing capacitances - only gate and junction capacitances are accounted for) show that the power consumptions of DDSLL and DyCML S-boxes are similar and they are about 1.9× lower than that of the SABL S-box across the frequency range. Also, when compared to static CMOS, they feature a 2× and 1.5× reduction in power consumption at 100 kHz compared to the 2-input and 4-input versions of static CMOS, respectively. As for the delay, the SABL, DDSLL and DyCML S-boxes have a delay penalty of 70 %, 2.5× and 6.7×, respectively when compared to either versions of the static CMOS S-boxes which have comparable delays. These observations confirm the results obtained in [15] which compare the performance of DDSLL to DyCML using 0.13 μm 1.2 V CMOS PD SOI technology.

In addition, measurement results demonstrate the reduction of power consumption brought by DDSLL. When compared to the 2-input static CMOS S-box, the DDSLL S-box has 1.5× less power which is fairly predicted by the post-layout simulation results. Also, the measured delay shows that DDSLL is 2.5× slower than 2-input static CMOS. From these results, we confirm that sharing the common blocks, using complex DPDNs when necessary and reducing the voltage swing are three important guidelines in designing a low-power DDL style. As a result, DDSLL is considered a good compromise relative to the other DDL

styles because it reduces the power consumption with a relatively moderate delay penalty compared to static CMOS.

We further demonstrate the impact of temperature on the power consumption of the 2-input static CMOS and the DDSLL S-boxes. We illustrate the exponential increase in the total power consumption of both S-boxes at high temperatures due to the escalating contribution of the leakage power. Moreover, the gap between the power consumption of DDSLL and static CMOS S-boxes widens as the temperature rises from $25°C$ to $125°C$; the ratio increases from $1.5\times$ to $4\times$ in favor of the DDSLL S-box. This is the result of the reduced leakage power of DDSLL thanks to its large stack of transistors, its dynamic current source that cuts off the current when the gates evaluate their functions and the low-swing operation. However, these ratios are expected to be somehow less if we use the 4-input static CMOS S-box instead of the 2-input version because it features large stack of transistors similar to DDSLL which decreases the leakage power by 40 %. Nevertheless, judging from the leakage power of both versions of the static CMOS and DDSLL S-boxes, it is still expected that the latter experiences less power consumption at high temperatures.

2.2 CONCEPT OF THE DYNAMIC DIFFERENTIAL LOGIC (DDL) STYLE

Due to the inefficiency of the static CMOS logic to implement secure cryptographic circuits, the concept of DDL is proposed to overcome this problem. The main goal of DDL is to hide the secret information directly associated to the power consumption of the circuit, by having a constant power in each clock cycle. The dynamic differential logic mainly combines two concepts; the differential logic concept and the precharge, also known as the dynamic logic concept [24]. The differential logic equalizes the power consumption of the two input transitions (0 - 1) and (1 - 0), where there is a charging/discharging event and also those of (0 - 0) and (1 - 1), where there is no power consumed. However, one can still differentiate between these two categories. On the other hand, the dynamic logic requires a precharge event every clock cycle. Thus, the power consumption is equal both input transitions (1 - 1) and (0 - 1), where there is a charging event and it is also equal for the (0 - 0) and (1 - 0) input transitions, where there is a discharging event. Nevertheless, we can still discriminate between the charging and discharging events. Clearly, neither the differential logic nor the dynamic logic equalize the instantaneous power consumption by itself. As a result, merging the two concepts to achieve a dynamic differential logic is mandatory.

Even though the main goal of the DDL style is to hide the secret information by equalizing the contribution of all input transitions to the overall instantaneous power consumption, it remains theoretically ideal. Proposed state of the art DDL styles try to achieve this ideal goal, however, it is not an easy task. Also, applying the concept of DDL inherently means that in each clock cycle, the circuit always consumes the maximum amount of power [11]. As a result,

designers face several challenges at both the security and performance levels. Mitigating glitches, overcoming the problem of different arrival times of the signals to an arbitrary gate (known as the early propagation effect) and balancing the capacitances of the differential signals are the three main categories that the designer must address to ensure the security goal of the DDL concept. Also, reducing the power consumption, decreasing the footprint and increasing the throughput of the DDL circuit are the general performance requirements. Consequently, designers should optimize the best security - performance tradeoff. Nevertheless, depending on the application, some of the above mentioned constraints are not necessary, thus, lightening the load of requirements the designer must fulfill. For example, low-power secure applications such as, RFIDs, contactless smart-cards and wireless sensor networks are highly constrained applications in terms of power consumption and die size while featuring a loose constraint on throughput.

2.3 STATE OF THE ART

Several logic styles are proposed in the literature, such as sense amplifier based logic (SABL) [14], wave dynamic differential logic (WDDL) [16], charge-recycling SABL [17], dynamic current mode logic (DyCML) [18], 3-state differential dynamic Logic (3sDDL) [25], balanced cell-based dual-rail logic (BCDL) [26], secure triple track logic (STTL) [27], dual-rail random switching logic (DRSL) [28], masked dual-rail precharge logic (MDPL) [29], dynamic differential self-timed logic families [15], ... etc. In this section we focus on two particular state of the art DDL styles, namely SABL and DyCML mainly because of their ability to solve security related issues associated with other DDL styles. More specifically, in their design they avoid glitches, early propagation effects and they either try to eliminate or reduce the power consumption sensitivity to unbalanced differential signals. Another reason for choosing these DDL styles is their potential performance overhead reduction as will be shown later in our analysis. Of course, other types of DDL styles feature the same design guidelines. However to avoid redundancy, we limit our comparison with DDSLL to only SABL and DyCML styles.

2.3.1 Sense amplifier based logic (SABL)

SABL [14] is a dynamic differential logic whose functions are built using differential pull down networks (DPDN)s. In order to have data-independent power consumption, SABL is designed to guarantee that in every clock cycle it charges a total capacitance with a constant value despite the fact that during each clock cycle different capacitances are switched. This goal is based on the fact that designing a logic that is dynamic and differential is a crucial step, but not sufficient because of the existence of unbalanced capacitances of the differential signals. The total capacitance comprises, the gate output capacitance C_o, the interconnect capacitance to the subsequent connected gates C_{rout} and the input ca-

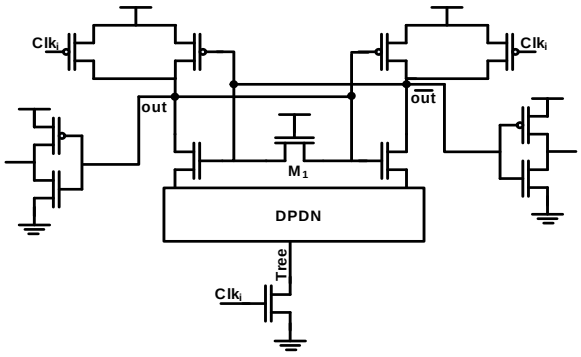


Fig. 2.1.: The schematic of a generic SABL gate.

pacitances of these gates C_i. To balance the gate differential output capacitances, the complementary outputs need to be driven by an equal number of transistors (having the same parameters). As for the differential input capacitances of the subsequent gates, they can be balanced by connecting the differential outputs to the same number of gate terminals. Despite these considerations, the differential interconnect capacitances and the internal DPDN nodes still need to be equalized which is a tedious job. Therefore, SABL aims to solve this problem by applying two principles:

1. It is a DDL logic, hence it switches the output once every clock cycle independent on the input value and sequence.

2. During each clock cycle, it charges / discharges the same capacitance value: one of the balanced output capacitances $(C_o + C_{rout} + C_i)$ + the sum of all the internal node capacitances of the gate DPDN.

SABL achieves the second principle by introducing a means to overcome the problem of unbalanced internal node capacitances without having to balance them. This is simply realized by connecting an always ON transistor M_1 between both outputs of the DPDN as shown in the generic SABL gate shown in Fig. 2.1 where it is responsible for:

– providing a path for subthreshold currents, thus preventing floating nodes.

– discharging all internal nodes during the evaluation phase.

During the evaluation phase, the clock signal Clk_i is high forming a path to ground and discharging one of the output nodes through the DPDN. Thanks to transistor M_1, all internal nodes and their capacitances discharge along with the discharging output node and its capacitance disregarding the fact that the internal node capacitances are most probably unbalanced. The same concept apply during the precharge phase where the Clk_i signal is low and both out

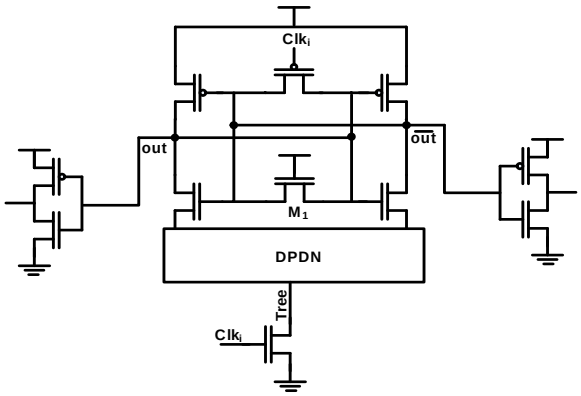


Fig. 2.2.: The schematic of a generic CRSABL gate.

and $\overline{out}$ are precharged to V_{DD}. Here, the value of the precharged capacitance is always constant regardless of the input pattern or the input sequence because all the discharged nodes and their capacitances will be precharged. Also, It is worth mentioning that SABL gates are connected to each other in a domino fashion using an inverter at each output node as seen in Fig. 2.1.

The authors in [14] implemented a Kasumi [30] substitution box (S9-box) using a 0.18 μm, 1.8 V technology node. It comprises 5 inverters, 86 NANDs and 92 XORs. Simulations are done using Hspice with random input sequence of length 500 clock cycles and they show that the energy of the SABL S9-box is slightly less than 2× that of static CMOS. Also, the area of the SABL S9-box exhibits the same increase factor compared to static CMOS. Not only the average power consumption of SABL is considered high (compared to static CMOS), but also the peak precharge power consumption is relatively high because the precharge event takes place at the same time for all gates.

This drawback motivated the authors of [14] to design a low power variant of SABL which is the charge recycling sense amplifier based logic (CRSABL) [17]. Its main goal is to reduce both the average power consumption and the peak precharge current of SABL using two key features:

1. To use the charge recycling principle in order to reduce the amount of needed charge from the supply by reusing the charge of one of the output nodes to charge the remaining nodes.

2. To charge the output and the internal nodes during the precharge phase to intermediate values instead of V_{DD}.

Fig. 2.2 shows the generic CRSABL gate. The main difference between CRSABL and SABL gates is the substitution of the two precharge clocked transistors by a single clocked transistor between the output nodes.

At the beginning of the precharge phase, the clock signal Clk_i goes low and the output nodes are shorted through the clocked transistor as the charged output from the previous evaluation phase drops and the discharged output node rises to $V_{DD} - |V_{tp}|$, where V_{tp} is the threshold voltage of the clocked PMOS transistor. Now, both the PMOS cross coupled transistors are turned off and both the NMOS ones are turned on charging the internal nodes X and Y to $V_{DD} - |V_{tp}| - V_{tn}$, where V_{tn} is the threshold voltage of the cross coupled NMOS transistors. Hence, the charge at the high output node is recycled and used to charge the low output node and the internal nodes. During the evaluation phase, the Clk_i signal is high forming a path to ground and discharging one of the output nodes through the DPDN to GND. At the same time the other output node is charged from $V_{DD} - |V_{tp}|$ to V_{DD} through the cross coupled transistors.

Simulation results using the same conditions as in [14] show that CRSABL saves 20 % energy per cycle compared to SABL and also reduces the peak precharge current by 63 %. However, the delay increases by 34 %. Also, the power consumption of CRSABL is reduced by 40 % compared to SABL.

2.3.2 Dynamic current-mode logic (DyCML)

DyCML [18] is designed as an enhancement to the MOS current-mode logic (MCML) [31] as it cancels the associated leakage power dissipation. Meanwhile, it maintains the main benefits of current-mode logic which are reducing the dynamic power by decreasing the output voltage swing and enhancing the performance. DyCML is also a self-timed logic that generates an internal clock for each block indicating the end of operation (either evaluation or precharge).

Fig. 2.3 shows a schematic diagram of the generic DyCML gate. The main concept of DyCML lies in the dynamic current source represented by transistors M_1, M_2 and the capacitor C_1. During precharge the input clock signal $Clki$ is low discharging transistor $C1$, which is used as a capacitor, through transistor M_2, thus transistor $C1$ serves as a virtual ground. At the same time, both out and $\overline{out}$ are precharged to V_{DD}. During the evaluation phase, the clock signal Clk_i is high turning on the dynamic current source through transistor M_1 allowing a path to the virtual ground formed by transistor C_1 and discharging one of the output nodes through the DPDN.

The design of this capacitor depends on two factors according to Equation 2.1 [18], the required voltage swing and the output load capacitance which includes drain capacitances of the DPDNs, gate capacitances of the associated latch, input gate capacitances of the following connected stages, and the routing parasitic capacitances. As a result, the size of this capacitor is different from one gate to another, which makes the design of the DyCML logic difficult.

$$V_{swing} \,.\, C_L = W_{C1} \,.\, L_{C1} \,.\, C_{ox} \,.\, (V_{DD} - V_{swing}) \tag{2.1}$$

where V_{swing}, C_L, W_{C1}, L_{C1} and C_{ox} are the required voltage swing, output load capacitance, width of transistor C_1, length of transistor C_1 and the gate

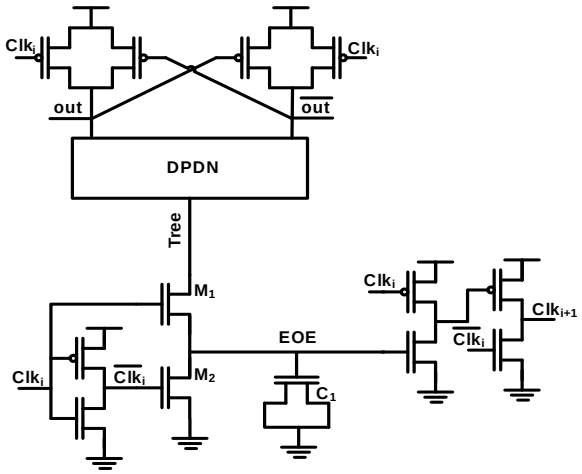


Fig. 2.3.: The schematic of a generic DyCML gate.

oxide capacitance, respectively. Moreover, Fig. 2.3 also shows the self-timing buffer used to generate a completion signal for the following stage to start its subsequent phase. It depends on the low-swing signal of the capacitor to generate the output clock Clk_{i+1}.

Simulation results of a Khazad S-box [19] using 0.13 μm partially depleted (PD) SOI CMOS technology are used to compare the performance of DyCML to SABL [32]. The Khazad S-box designed with both logic styles adopts the use of complex functions through complex DPDNs. However, the design of the DyCML gates of the Khazad S-box required special attention to the sizing of the gates because of the variations of fan-out, which was not needed for SABL. DyCML with output swings of 0.8 V and 0.4 V has a power reduction of 33 % and 50 %, respectively compared to SABL at 100 MHz. As for the area, the DyCML Khazad S-box is 9 % smaller than that of SABL.

2.4 DDSLL IN A NUTSHELL

In order to design a digital circuit using the DDSLL style, several features and design guidelines have to be first understood. First, we introduce the general topology and describe the functionality of the DDSLL style as described in [15]. In the next section, we provide the implementation details that we employed to implement the case study; an AES S-box, using the DDSLL style.

2.4.1 Circuit topology

The DDSLL style is one of several DDL self-timed logic styles developed by [15] for secure low-power applications. It features a precharge phase where all

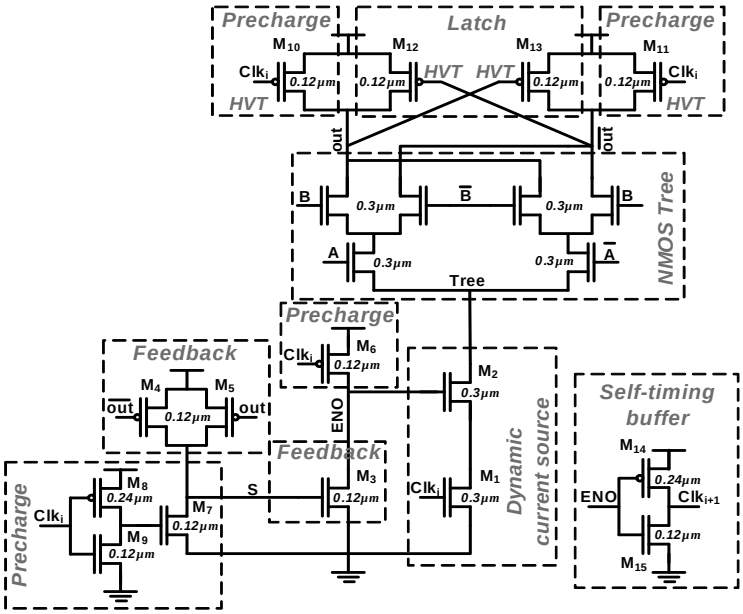


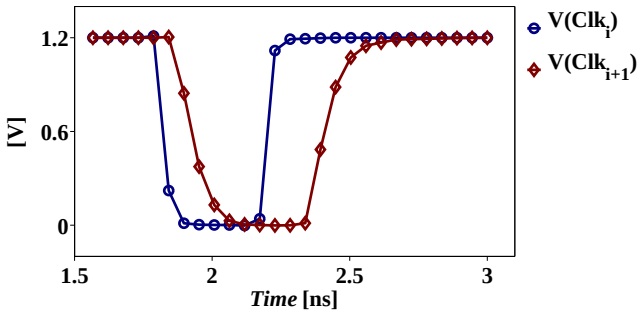
Fig. 2.4.: DDSLL XOR schematic

differential outputs are charged to V_{DD}. Similar to the DyCML logic, DDSLL operates in a self-timing scheme. It employs a dynamic current source to eliminate the leakage power consumption associated with regular current-mode logic styles. The cut off of this current source is performed when a feed-back loop detects the completion of the evaluation. If operated with a self-time scheme, the completion signal denoted by ENO (i.e ENd of Operation) is propagated to the next logic stage to start its evaluation phase.

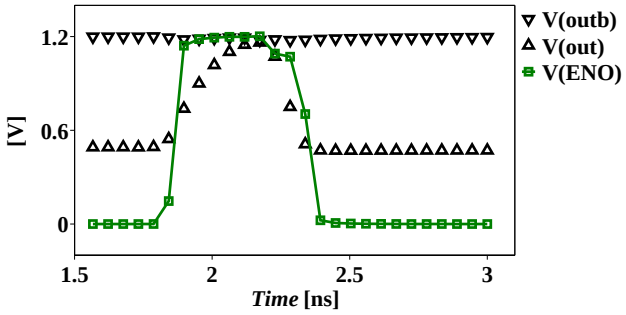
Fig. 2.4 shows the schematics of a simple XOR gate implemented using DDSLL. It uses the same DPDN as DyCML to evaluate the function at hand. Transistors M_1 and M_2 operate as the dynamic current source, while transistors M_3 - M_5 perform the feed-back operation necessary for cutting off the current after evaluation. Transistors M_6 - M_{11} form the precharge circuit, and transistors M_{12} and M_{13} function as a latch. The self-timing buffer used for the DDSLL style is a simple inverter as shown in Fig. 2.4.

2.4.2 Functional operation

The operation of the DDSLL style is quite simple. It consists of two phases: precharge and evaluation. Fig. 2.5 shows how the precharge and evaluation phases function. During the precharge phase, the input clock signal Clk_i is low, discharging node S to GND and charging node ENO to V_{DD}. However, there



(a) DDSLL XOR input/output clocks.



(b) DDSLL XOR output/internal nodes.

Fig. 2.5.: DDSLL XOR functional operation.

is no current path from V_{DD} to GND as M_1 is switched off. Meanwhile, both output nodes *out* and $\overline{out}$ are precharged to V_{DD} via transistors M_{10} and M_{11}.

During the evaluation phase, the input clock signal Clk_i is high, allowing current to flow through the dynamic current source ($M1$, $M2$), as node ENO was previously charged to V_{DD}, in order to provide a discharge path for one of the precharged output nodes. Depending on the logic function and the inputs, there will be a single low impedance path from one of the output nodes to GND. As soon as one of the output nodes falls below $V_{DD} - V_{tP|FB}$, where $V_{tP|FB}$ is the threshold voltage of the feed-back PMOS transistors, node S will be charged to V_{DD} turning on the feedback transistor M_3, which in turn discharges node ENO to GND switching off the dynamic current source. Meanwhile, one of the latch transistors (M_{12}, M_{13}) turns on as the output node connected to its gate falls below $V_{DD} - V_{tP|L}$, where $V_{tP|L}$ is the threshold voltage of the latch PMOS transistors, preserving the voltage of the other output node at V_{DD}.

The self-timing buffer acts as the interface between cascaded stages of the DDSLL style, as it delivers a slightly shifted version of the input clock indicating the termination of the evaluation phase of the current block.

2.5 DDSLL IMPLEMENTATION DETAILS

In this section, we detail the design guidelines we proposed in [21] to optimize the performance of the DDSLL[1] compared to static CMOS and other DDL styles. We first provide the design methodology of DDSLL followed by the detailed steps that lead to a full picture of how to implement complex circuits such as the AES S-box using DDSLL.

2.5.1 DDSLL design methodology

The initial step in the circuit design is to choose an appropriate technology. For low-power applications, it is important to have low dynamic and leakage power consumptions. As suggested in [22], the choice of the 65 nm CMOS technology node with low-power (LP) flavor reduces both the dynamic power and the two major contributors to leakage power; namely, the subthreshold and gate leakages compared to 0.13 μm CMOS technology. An advanced encryption standard (AES) [20] S-box is implemented using the LP 65 nm CMOS technology with standard V_t devices. It features 1.8× and 5.6× reduced dynamic and leakage power consumptions, respectively when compared to the same S-box implemented using the 0.13 μm CMOS technology with high V_t devices. Then we select the transistor sizes with the aim of reducing the power consumption without jeopardizing the functionality at different operating conditions. For example, the use of minimum feature size transistors only may lead to insufficient voltage swing if process variations or different operating temperatures are taken into consideration. Further details are provided in Section 2.5.2.

The second step is the design of the DPDN which is a rather challenging task due to their excess sensitivity to the data patterns. To achieve optimum security, the differential output nodes and internal nodes of the DPDN have to be completely balanced. Of course, this is the ideal case. Realistically, partial balancing of the differential nodes is achievable using certain techniques that we detail in Section 2.5.3. Here, we use the BDD [23] to create the DPDNs and further optimize them. The capacitance counting technique used by the tool that is proposed in [33] is a good start to try and balance the DPDN. Furthermore, special care has to be taken at the layout level to preserve the symmetry of the DPDNs and balance the capacitances of the differential routes as well.

The third step involves the reduction of the redundant operations by sharing the common blocks of the gates that function at the same time. The goal is to decrease the overall power consumption of the DDSLL S-box and also minimize the area. This can be achieved by sharing the self-timing buffer, the feed-back transistors and parts of the precharge circuitry as explained in Section 2.5.4. The last step in the methodology discusses how to interface DDSLL gates with static CMOS as illustrated in Section 2.5.5. Normally, a low-swing full-swing buffer

[1]In this study we consider only combinatorial circuits. Implementing storage elements and flipflops using DDSLL would be interesting for future research.

is proposed in the literature, but this does not take into account the fact that the differential outputs are precharged to V_{DD} every clock cycle. As a result, we propose a special buffer that does not only transform the low-swing signal to a full-swing signal, but also eliminates the precharge phase of the signal. That is, preserve the evaluated output signals during precharge. This buffer is only needed at the output of DDSLL circuits to connect to inputs of static CMOS gates. However, the input of DDSLL gates can be directly connected to the output of static CMOS.

2.5.2 Circuit design

As was previously shown in [22], the choice of the technology node with the appropriate flavor and device type greatly affects the circuit delay and power consumption performance. In general, high-performance applications benefit from scaling while low-power applications suffer from increased leakage [34]. This is the main reason for developing both the General Purpose (GP) and the Low Power (LP) flavors in advanced technology nodes such as the 65 nm to serve high-performance and low-power applications, respectively.

2.5.2.1 Technology selection

As discussed in [22], simulation results of the AES S-box using a 0.13 μm CMOS technology interestingly showed that at 100 kHz, the power consumption is dominated by leakage power (96.4 % in case of low V_t devices and 54 % in case of high V_t devices). A thorough investigation was consequently conducted to select the most appropriate technology in order to minimize the power consumption of the S-box [22]. Despite the reduction in dynamic power, a major disadvantage of advanced technologies is their high gate leakage. However, it can be mitigated by employing the LP flavor introduced in sub 100 nm technology nodes. Another drawback of these advanced technologies is the relatively high subthreshold current. Nevertheless, the use of LP flavor coupled with high V_t devices reduces the subthreshold current substantially. Indeed, the dynamic power of the S-box implemented using LP 65 nm CMOS technology with standard V_t (SVT) devices, is reduced by a factor of 1.8 compared to the one implemented using 0.13 μm CMOS technology with high V_t devices. As for the leakage power, the reduction factor is 5.6×. These observations lead to favoring the use of the 65 nm CMOS technology with LP flavor and SVT devices. Further details are provided in Appendix C. On the other hand, the delay of the S-box implemented using this technology with SVT devices was found to be slightly higher than that of a 0.13 μm CMOS technology with low V_t devices thanks to the significant reduction of the gate capacitance. As a result, we choose to use the 65 nm technology node with LP flavor in order to reduce both the dynamic power and leakage power without sacrificing the performance.

2.5.2.2 DDSLL circuit design using LP 65 nm technology node

With respect to the sizing of the transistors, all are designed with minimum gate length ($L_g = 0.06\mu m$) to minimize the gate capacitances; thus, reducing the power consumption. As for the transistors width:

- The precharge, latch, feed-back circuits and the self-timing buffer are designed using minimum feature size transistors ($W = 0.12\mu m$) to reduce the power consumption except for the PMOS transistors in the inverters of the precharge circuit and self-timing buffer ($W = 0.24\mu m$) to maintain the duty cycle of the input clock.
- The dynamic current source uses wider transistors ($W = 0.3\mu m$) in order to drive sufficient current from the DPDN during the evaluation period providing the desired output voltage swing.
- The DPDN is also designed with wider transistors ($W = 0.3\mu m$) for two reasons, the first is to increase the output voltage swing and the second is to reduce the effect of within-die (WID) variability on the output voltage swing.

Regarding the choice of the devices, this technology offers three different V_t devices, low-V_t (LVT), standard-V_t (SVT) and high-V_t (HVT). Most DDSLL devices are chosen to be SVT devices to reduce the leakage power without great loss of performance as shown in [22]. Contrarily, the latch and transistors M_{10}-M_{11} of the precharge circuit use HVT devices in order to reduce the leakage that charges the low-voltage output nodes during the evaluation phase especially at higher temperatures. The choice of HVT for these transistors was favored after performing preliminary simulations on the S-box using all SVT devices at high temperature ($125°C$) which lead to the failure of the S-box due to the charging of the low-voltage output nodes during evaluation.

2.5.3 Differential pull down network (DPDN) creation: Design and Layout

The DDSLL style is developed as a countermeasure against power analysis attacks [15, 35]. As a result, special care has to be taken while designing the DPDNs, especially since large DPDNs can be used to implement complex functions with even three or four inputs in order to save area. Hence, the capacitances have to be balanced between both output nodes and for all input sequences. The gate load capacitance has three main contributors: (1) The gate intrinsic output capacitance C_o, (2) the parasitic routing capacitance C_{rout} and (3) the intrinsic input capacitance of the load gate C_i. According to [25], the following items guarantee a balanced DPDN network:

1. Both out and $\overline{out}$ of each DPDN must be connected to the same number of parallel branches in order to have the same load.
2. The number of series connected transistors in each $out/\overline{out}$ branch should not depend on the input of the implemented function.

3. The total number of connected capacitances for each input sequence should be the same.

4. The layout of the DPDN should preserve the symmetry between *out* and $\overline{out}$ branches and also balance the routes in order to match the interconnect capacitances.

The DPDN representing a certain function can be easily created from the binary decision diagram (BDD) [23] used to define this function [33, 36].

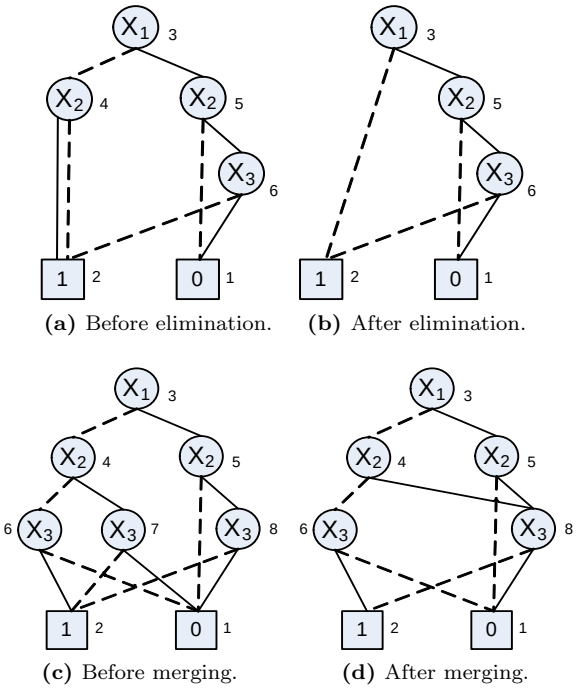
2.5.3.1 Binary decision diagram (BDD)

BDD is used as a tool to optimize the design of the DPDNs representing large complex functions [36]. These diagrams can be easily created using the Shannon expansion theorem, where a Boolean function can be recursively represented by the sum of two sub-functions of the original to form a graph structure.

The basic representation of Boolean functions using BDDs is defined according to [37]:

- A function graph is a rooted, directed graph with vertex set V containing two types of vertices.
  - A *nonterminal* vertex v whose attributes are: an input x_i ($i \in$ 0,.....,m-1) of the function (f:$0, 1^m \in 0,1$), an index $index(v) \in 1,.....,n$ and two children vertices $low(v)$ and $high(v)$.
  - A *terminal* vertex v whose attributes are: an index $index(v)$ and a value $value(v) \in 0,1$.
- A function graph can be reduced by:
  - Eliminating the redundant nodes: If the two children vertices of a vertex v are identical ($low(v) = high(v)$), then vertex v is said to be redundant and can be removed such that its parent vertex can connect directly to its child vertex. For example in Fig. 2.6 (a), vertex with index 4 is redundant.
  - Merging the duplicate nodes: A vertex is said to be duplicated if other vertices that define the same function exist. Therefore, all duplicate vertices can be removed and replaced by only one vertex. For example in Fig. 2.6 (c), vertex with index 7 is a duplicated version of vertex with index 8.

  and thus transforming the BDD into a reduced ordered binary decision diagram (ROBDD).



(a) Before elimination. **(b)** After elimination.

(c) Before merging. **(d)** After merging.

Fig. 2.6.: Reduction of binary decision diagram: (a), (b) Eliminating the redundant nodes and (c), (d) merging the duplicate nodes [33].

2.5.3.2 DPDN design using BDD

The use of BDDs is extended from optimizing the Boolean functions, in terms of minimizing their representations, to mapping the BDDs into differential gates exploiting the isomorphism between the DPDN and the BDD implementing the Boolean function [36].

The mapping is done as follows:

- Each nonterminal vertex of the BDD is substituted by a net of the DPDN.
- Each edge connecting a nonterminal vertex to its children vertices is substituted by an NMOS transistor whose source and drain are the parent nonterminal vertex and the child vertex, respectively.
- The terminal vertices whose values are 0 and 1 are mapped to f and f', respectively.

Fig. 2.7 shows an example of mapping both an AND and XOR functions from BDD to their equivalent DPDN counterparts.

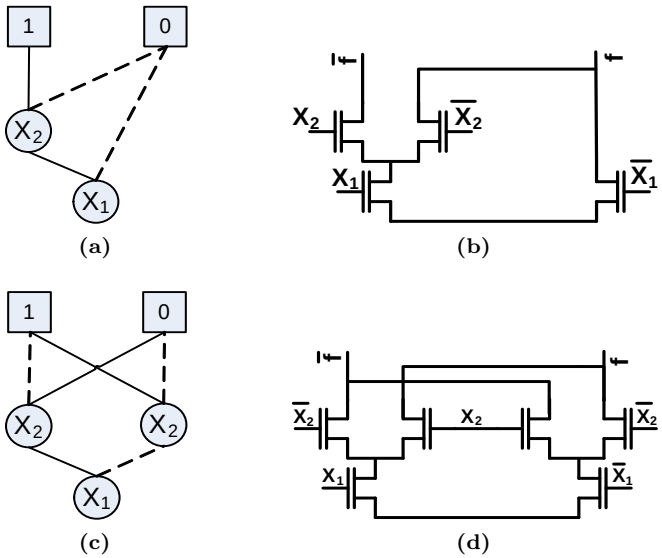


Fig. 2.7.: Mapping BDD to DPDN for AND (a,b) and XOR (c,d) functions.

2.5.3.3 Balancing the capacitances of DPDN using BDD

As mentioned in Section 2.5.3, balancing the capacitances between both output nodes of a DPDN and for all input sequences is extremely important to counteract power analysis attacks. Accordingly, a tool based on BDD is proposed in [33] to explore different implementations of the targeted DPDN in order to predict the most secure structure. It is based on modeling the power consumption related to a single input sequence using the following assumptions:

- Variations in the power consumption with different input sequences are a result of different number of internally connected capacitances in the DPDN that are charged / discharged during the evaluation phase [14].
- The diffusion capacitances of the transistors are only considered.
- The drain-gate and source-gate capacitances are considered equal and they are the only ones to play a role.
- To predict the power consumption associated to an input sequence, only the number of normalized capacitances that are connected to the discharging output node are counted.
- The number of normalized connected capacitances are: two source-drain capacitances connected to each vertex, a drain-gate capacitance as a contribution of each sub-vertex. Fig. 2.8 shows an AND gate as an example of how the tool counts the capacitances connected to the discharging output

node and Table 2.1 provides the number of normalized connected capacitances according to the input sequence.

Table 2.1.: Predicted number of capacitances connected to the discharging node based on the input sequence of an AND gate [33].

		f	discharged node	connected capacitances
0	0	0	out	8
0	1	0	out	8
1	1	0	out	5
1	1	1	out	7

The function of the tool proposed in [33] is summarized as follows:

- It investigates all the possible structures (input ordering) of a certain function by building the BDDs and applying the reduction rules. For example a 4-input function can have 4! i.e. 24 different structures.
- For each structure, it builds a model by counting the number of capacitances connected to the discharging output node of the DPDN for all the input sequences.
- For each structure, it creates a SPICE netlist relative to the transistor implementation of the BDD. The netlist is then simulated and the power consumption related to each input sequence is extracted.
- For each structure, the variance of the connected capacitance related to the input sequences is calculated and the structure that has the minimum capacitance variance is chosen as it indicates a minimum difference in the power consumption between different input sequence making it more difficult to attack.

The use of this tool helps to find the most information leakage efficient structure among all possible n! different structures of n-input functions. This will be further investigated in details in Section 2.6.

2.5.3.4 Balancing the capacitances of DPDN from the layout perspective

As the technology scales down, the parasitic routing capacitance becomes the dominating contributor of the total gate load capacitance [34, 38]. Therefore, special care has to be taken while designing the routes at the layout stage. As a result, a technique proposed in [39, 40] employs standard place and route tools to route the differential signals guaranteeing the matching of the routing capacitances within a few percent. This technique is based on routing the output of any differential pair as one "fat" wire, which has the width of two differential parallel wires plus the spacing between them, then transforming it into differential parallel wires as shown in Fig. 2.9. It guarantees that the differential

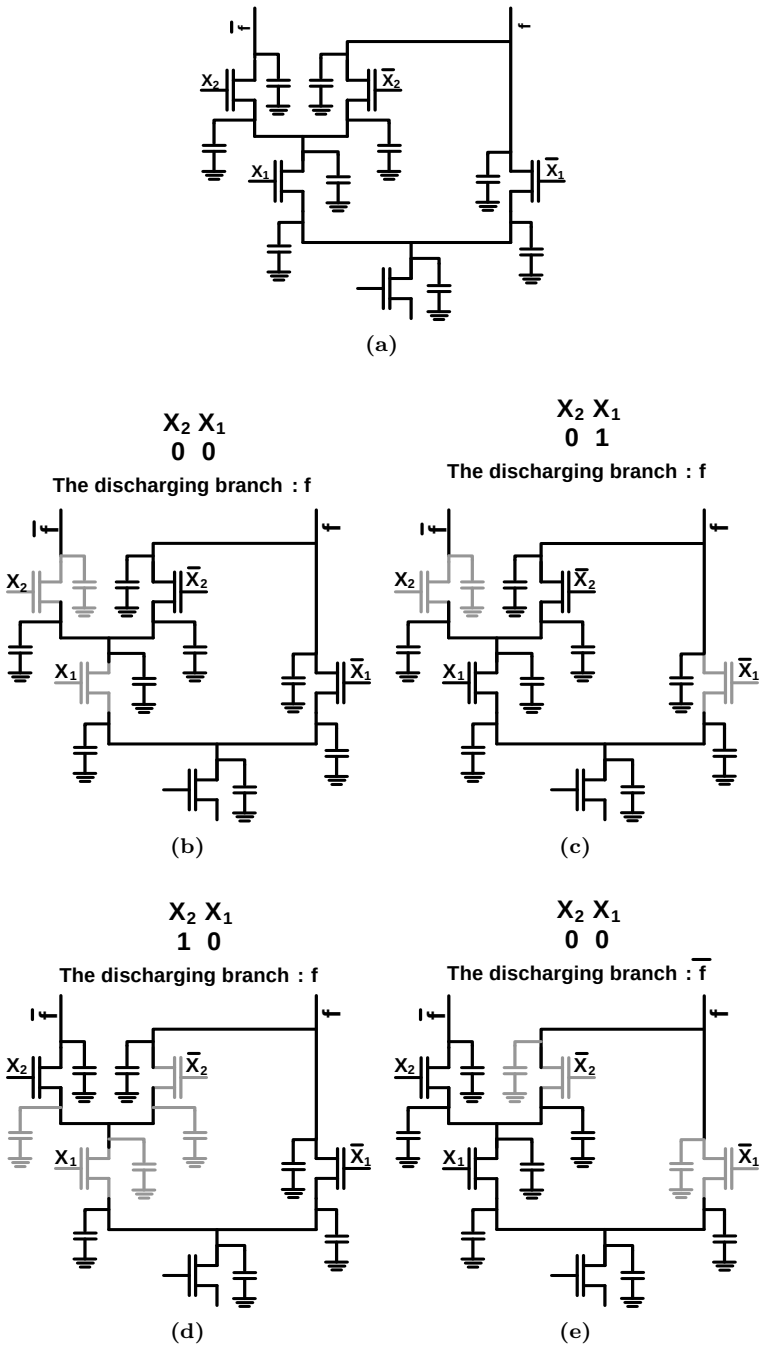


Fig. 2.8.: AND gate capacitances that are taken into account by the tool (a) [33], capacitances that are connected to the discharging node when the inputs X_2X_1=00 (b), X_2X_1=01 (c), X_2X_1=10 (d), X_2X_1=11 (e).

Fig. 2.9.: Differential pair routing methodology: (a) Fat wire representation and (b) Transformation of the fat wire into differential routes [40].

wires are always routed in adjacent tracks [39] ensuring the matching between the differential routes. Also, the DPDNs themselves need to be balanced at the layout level. That is, special care has to be taken while placing the differential transistors in order to preserve the symmetry and thus, maintain the balance of between the capacitances of the differential nodes. The basic layout guidelines that we employed are discussed in Appendix D along with illustrative examples of the layouts of the Inversion in $GF(2^4)$ function and the whole AES S-box.

2.5.4 Sharing principle

The sharing of common blocks is generally used to reduce the power consumption overhead of these blocks and also to reduce the die size. It is applicable to differential current-mode logic styles which are either dynamic such as DyCML or static as in MCML [41–43]. For example, in subthreshold MCML [41] the feedback bias circuit, which defines the gate voltage of the PMOS load devices can be shared among several logic gates as shown in Fig. 2.10. Also in [43] the replica bias in the subthreshold source couple logic (STSCL), which biases both the PMOS load devices and the tail current source, can be shared. However, sharing in these cases causes the design to be vulnerable to within-die variability as the feedback bias / replica bias should be well matched to the MCML/SCL gates in order to minimize the deviation of the output voltage swing. As a result, a deviation of 40 mV in STCL for minimum size devices using 0.18 μm technology is reported by [43].

On the other hand, the principle of sharing in DDSLL is quite different. Fig. 2.11 depicts the technique we introduce so that several DPDNs can share the same feed-back, dynamic current source, self-timing buffer, and part of the precharge circuit. However, the latch transistors connected to the output nodes cannot be shared as they are needed for each output of the DPDNs. Also the output precharge transistors are not shared, although they can be, because of the excess complexity that would be added at the layout phase. Accordingly,

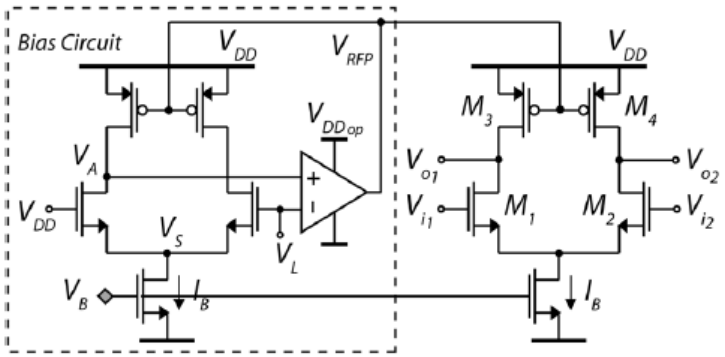


Fig. 2.10.: Schematic of an MCML inverter and its feed-back bias circuit that can be shared among several gates [41].

sharing is useful among gates whose operations can be performed at the same time, since in this case these functions will terminate their evaluation process at the same time. Therefore, they need a single dynamic source, a single feed-back circuit and a single self-timing buffer.

Finally, the $out/\overline{out}$ of the DPDN that derive the shared feed-back circuit should be chosen to be the most loaded outputs (which means the slowest signals). This way, the output clock Clk_{i+1} of these shared functions is generated after the slowest output is evaluated which guarantees successful operation and avoids early propagation (not taking WID variations into account which pose a potential threat to the success of DDSLL)[2]. Moreover, choosing the functions that will share the common blocks and the most loaded output signals (by calculating the fan-out of each gate) are done manually. Of course, this indicates that thorough testing is needed not only at nominal conditions, but also in presence of process variations. Also, maintaining the same output voltage swing of the gates that employ the sharing principle as the other simple gates is not an easy task and due to the manual design of these gates, their output voltage swings experience some deviations. Nevertheless, automating this procedure to avoid the tedious manual job and to increase the efficiency and reliability of the sharing principle would be interesting for future research. In addition, this will increase the output voltage swing of the shared blocks as the slowest output will cut off the current of the dynamic current source after a longer period.

[2]However SPICE simulation results while enabling WID process variations, using 100 Monte-Carlo runs show that the adopted design guidelines helped avoid the failure of DDSLL due to WID process variations at 1.2 V supply and there were not any early propagation problems visible.

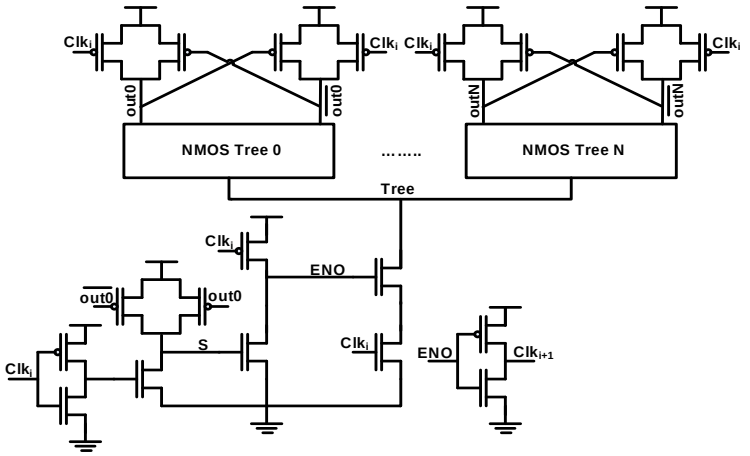


Fig. 2.11.: Schematic of DDSLL gates using the sharing principle.

2.5.5 Interface with static CMOS logic

To interface the output of the DDSLL style with the input of the static CMOS logic, a special buffer is needed to convert the DDSLL low-swing clocked signal to a full-swing non-clocked signal. Low-swing to full-swing buffers exist in literature, for example in [18]. However, it does not take into account the fact that the differential outputs are both precharged to V_{DD} every clock cycle, while in static CMOS logic this is not the case, unless the use of flip-flops is intended to sample the correct output state. Therefore, the buffer interface needs not only to convert the low-swing signal to a full-swing signal, but also to preserve the evaluated output during the precharge phase of the clock.

Fig. 2.12 shows the schematic of the output buffer we propose to interface the DDSLL style with the static CMOS logic. All transistors are SVT devices with minimum feature size. The input signals in and $\overline{in}$ are the differential low-swing outputs of the DDSLL style, while the out and $\overline{out}$ signals are the input signals to the static CMOS logic. In order to explain the functionality of the output buffer let us assume that during evaluation in is the low signal and $\overline{in}$ is the high signal. During the evaluation phase, transistor M_6 is on, charging node $\overline{out}$ to V_{DD} which in turn will switch on transistor M_1. Now the two series NMOS transistors M_1 and M_3 are on, discharging node out to GND. During the precharge phase, both in and $\overline{in}$ signals are high, turning on transistors M_3 and M_4. As a consequence, the output buffer acts as a latch preserving the voltages of the differential full-swing previously evaluated outputs. The operation of the output buffer is shown in Fig. 2.13.

On another hand, the interface between the output of the static CMOS logic and the input of the DDSLL style does not require special buffers, it is sufficient to directly connect the output of the static CMOS logic to the input of the DDSLL style similar to [18].

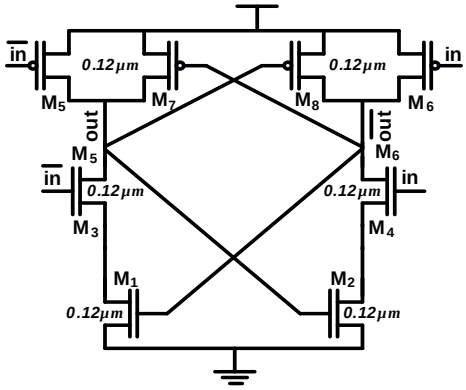


Fig. 2.12.: DDSLL output buffer interface with static CMOS logic

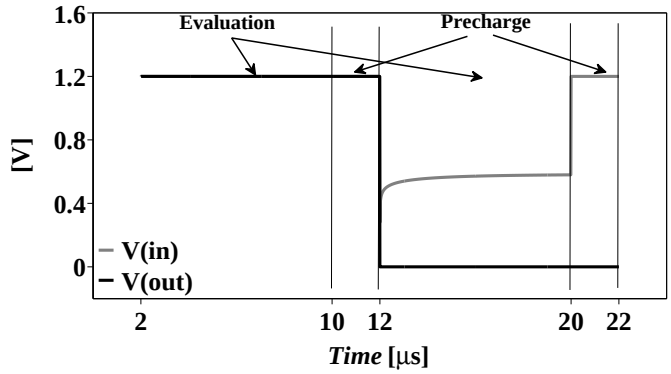


Fig. 2.13.: Operation of the DDSLL output buffer interface with static CMOS logic

2.5.6 Current implementation summary

In the current manufactured implementation of the DDSLL S-box the following design items are considered:

- In the design of the DPDNs of the six 4-input functions of the DDSLL S-box care has been given to items 1 and 4 only in Section 2.5.3 namely: Both *out* and $\overline{out}$ of each DPDN must be connected to the same number of parallel branches in order to have the same load and the layout of the DPDN should preserve the symmetry between *out* and $\overline{out}$ branches in order to match their interconnect capacitances, respectively.

- BDDs are designed using both reduction techniques (eliminating the redundant nodes and merging the duplicate nodes) in order to create the most compact implementation of each function. As a result, while designing the

BDDs for the manufactured implementation of the DDSLL S-box care was not given to obtain the most balanced implementations. However, a second version of the DDSLL S-box that features balanced implementations of the six 4-input functions is designed and discussed in the following section.

– DPDN creation and layout are done in a full-custom manner. Accordingly, the differential wires are manually routed in adjacent tracks ensuring the matching between them in the same manner proposed in [39, 40] only without the use of standard place and route tools.

2.6 COMPARISON BETWEEN COMPACT AND BALANCED DPDNS

In order to utilize the concept of the tool proposed in [33] and briefly described in Section 2.5.3.3, the complex functions in the AES S-box have to be first identified. Then, we apply the same technique as the tool to find the most balanced structure for each of the complex functions. Notably, there are 6 functions that can be implemented in a complex fashion; 4 functions are in the Inversion $GF(2)^4$ block and 2 functions are in the Multiplication $GF(2)^2$ block.

In the Inversion $GF(2)^4$ block:

$$X_3^{-1} = \overline{x_3}x_2\overline{x_1}\overline{x_0} + \overline{x_3}x_2\overline{x_1}x_0 + \overline{x_3}x_2x_1\overline{x_0} + \overline{x_3}x_2x_1x_0 + x_3\overline{x_2x_1x_0} + x_3\overline{x_2}x_1\overline{x_0} + x_3x_2\overline{x_1}x_0 + x_3x_2x_1\overline{x_0} \tag{2.2}$$

$$X_2^{-1} = x_3\overline{x_2x_1}x_0 + x_3\overline{x_2}x_1x_0 + x_3x_2\overline{x_1}x_0 + x_3x_2x_1\overline{x_0} + x_3x_2x_1x_0 + x_3x_2\overline{x_1x_0} + \overline{x_3}x_2\overline{x_1x_0} + \overline{x_3}x_2\overline{x_1}x_0 \tag{2.3}$$

$$X_1^{-1} = \overline{x_3x_2}x_1\overline{x_0} + \overline{x_3x_2}x_1x_0 + \overline{x_3}x_2\overline{x_1x_0} + \overline{x_3}x_2x_1x_0 + x_3\overline{x_2x_1x_0} + x_3\overline{x_2x_1}x_0 + x_3\overline{x_2}x_1x_0 + x_3x_2\overline{x_1}x_0 \tag{2.4}$$

$$X_0^{-1} = \overline{x_3x_2x_1}x_0 + \overline{x_3x_2}x_1\overline{x_0} + \overline{x_3}x_2\overline{x_1x_0} + \overline{x_3}x_2x_1\overline{x_0} + \overline{x_3}x_2x_1x_0 + x_3\overline{x_2}x_1x_0 + x_3x_2\overline{x_1x_0} + x_3x_2x_1\overline{x_0} \tag{2.5}$$

In the Multiplication $GF(2)^2$ block:

$$k_0 = \overline{q_1w_1}q_0w_0 + \overline{q_1}w_1q_0w_0 + q_1\overline{w_1}q_0w_0 + q_1w_1\overline{q_0w_0} + q_1w_1\overline{q_0}w_0 + q_1w_1q_0\overline{w_0} \tag{2.6}$$

$$k_1 = \overline{q_1}w_1q_0\overline{w_0} + \overline{q_1}w_1q_0w_0 + q_1\overline{w_1q_0}w_0 + q_1\overline{w_1}q_0w_0 + q_1w_1\overline{q_0w_0} + q_1w_1q_0w_0 \tag{2.7}$$

Fig. 2.14 (a) and (b) show the BDDs of two different implementations of the function k_3^{-1} in the Inversion $GF(2)^4$ block as an example. In Fig. 2.14 (a), the compact structure is developed (6 vertices) and in Fig. 2.14 (b), the balanced structure (11 vertices), which is allegedly information leakage efficient, is established. It is worth noting that the only reduction rule applied for the balanced structure is: merging the duplicate nodes, since eliminating the redundant node causes imbalance in the structure. Also, the transistor level of the structures of both implementations are shown in Fig. 2.14 (c) and (d). The number of connected capacitances to the discharging node according to each input sequence (input) is detailed in Table 2.2. The total number of input sequences is 16 (2^4 since the functions are implemented using 4-input gates). As a result of balancing the structure of the function k_3^{-1} in the Inversion $GF(2)^4$ block, the variance of the capacitance over the different input sequences is 0, while for the compact structure, it is equal to 8.25. However, balancing the structure leads to an increase of almost 2× in the number of transistors used. The BDDs and DPDNs of the remaining 5 functions are detailed in Appendix B, whereas the capacitance count of each structure is given in Table 2.2.

Table 2.2.: Difference between the compact and balanced structures of all functions in the inversion $GF(2)^4$ and multiplication $GF(2)^2$ blocks in terms of the number of capacitances connected to the discharging output node based on the input sequence, and the standard deviations of these capacitances.

Function	Parameter	compact	balanced
K_3^{-1}	# input	2 2 4 8	16
	# capacitances	16 15 13 9	17
	std	2.87	0
K_2^{-1}	# input	8 8	12 4
	# capacitances	12 9	17 16
	std	1.55	0.45
K_1^{-1}	# input	6 2 8	12 4
	# capacitances	18 16 13	18 16
	std	2.41	0.89
K_0^{-1}	# input	4 4 4 4	4 10 2
	# capacitances	17 16 14 12	18 17 16
	std	1.98	0.62
k_1	# input	1 2 1 2 4 2 4	12 4
	# capacitances	17 16 15 14 13 12 10	18 16
	std	2.28	0.89
k_0	# input	2 2 6 2 4	12 4
	# capacitances	16 15 13 12 10	18 16
	std	2.05	0.89

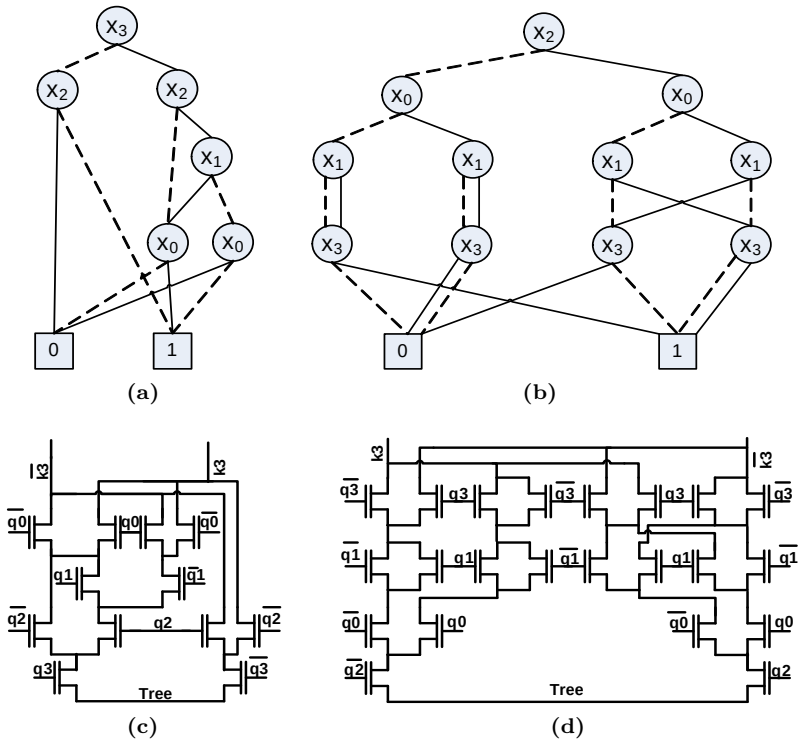


Fig. 2.14.: BDD of the function k_3^{-1} in the inversion $GF(2)^4$ block with the compact structure (a) and with the balanced structure (b). DPDN of the function k_3^{-1} in the inversion $GF(2)^4$ block with the compact structure (c) and with the balanced structure (d).

Evidently, not all balanced DPDNs of the remaining 5 functions feature 0 variance of the capacitance over the different input sequences. For example, the function K_1^{-1} in the inversion $GF(2)^4$ and both functions of the multiplication $GF(2)^2$ suffer a maximum capacitance variance of 0.89. Although the capacitance variance reduction brought by the balanced structures is evident, the majority of these structures are still inevitably unbalanced to a certain extent. As a result, an interesting research for the future would be to investigate other balancing techniques in order to reduce the capacitance variance to 0.

Nevertheless, to show the importance of having completely balanced structures we plot the current traces of the two version of the DDSLL S-box; namely, the one with the compact DPDNs and the one with the "balanced" DPDNs. By current traces, we mean the supply current resulting from the different operations of the S-box. In the SPICE simulations, we use n different input sequences where the inputs to the S-box under test transitions from the state of 0 to an arbitrary input state ranging from 0 to 255. The supply current is then extracted

for the evaluation period and divided into n current traces corresponding to the n different input sequences that took place during n clock cycles. Then, these n current traces are overlapped in order to visually inspect the differences resulting from applying the n input sequences. Finally, the standard deviation of the current traces over the different input sequences is plotted for all the time samples in order to numerically identify the sample that is most affected by the different transitions and the maximum value of this deviation. Details and further explanations of the choices we made in order to obtain the current traces are presented in Chapter 4. Of course, visually inspecting the current traces and calculating the standard deviations are not sufficient to claim a security enhancement of one implementation over another. In Chapter 4, we discuss the tools and metrics that we use to analyze the security of the DDSLL S-box implementation. Nevertheless, computing the standard deviations over the input sequences is considered an informal step in the security assessment and it also proved to be related to security metrics [44, 45].

In Fig. 2.15, we plot the current traces and the standard deviation over the input sequences for both DDSLL S-boxes. Simulations are done at typical conditions, $25°C$ temperature, 1.2 V supply voltage and without adding the extracted routing parasitics (only gate and junction capacitances are accounted for). The standard deviations (σ) are computed based on normal (Gaussian) and lognormal distributions. Normally, a normal assumption is sufficient, but due to the irregular shape at the end of the current traces (different termination times of the evaluation period), we plotted σ for the lognormal distribution as well. It appears that the sample that maximizes the σ for both S-boxes is almost the same for both distributions, despite the different values. Evidently, the use of balanced DPDNs renders the DDSLL S-box with slightly lower σ over most of the time samples compared to the compact version. However, the maximum lognormal σ dropped from 95 μA (compact DPDNs) to 30 μA (balanced DPDNs) which results in similar lognormal and normal distributions in the case of the balanced DDSLL S-box for all time samples.

These observations lead to an important conclusion; the use of balanced DPDNs does not enormously reduce the standard deviation (normal) with respect to the compact version of the DDSLL S-box. We also conclude that considering a lognormal distribution does not contribute to the final decision of whether the security enhancement of an implementation is substantial or not, compared to another, rendering it rather irrelevant. That is, we cannot claim that balancing the DPDNs lead to a significant reduction of standard deviation using the lognormal distribution while having only a slight improvement if we consider the normal assumption. This is simply because the standard deviation reduction brought by the use of a lognormal distribution is not coupled by the same reduction using a normal distribution. As a result, analyzing the current traces based on a normal assumption is enough. Also, it is important to state that the time sample that provides the maximum standard deviation is the critical time sample that the adversary will probably exploit (knowing that the standard deviation is related to security metrics [44, 45]). As a result, the lower standard

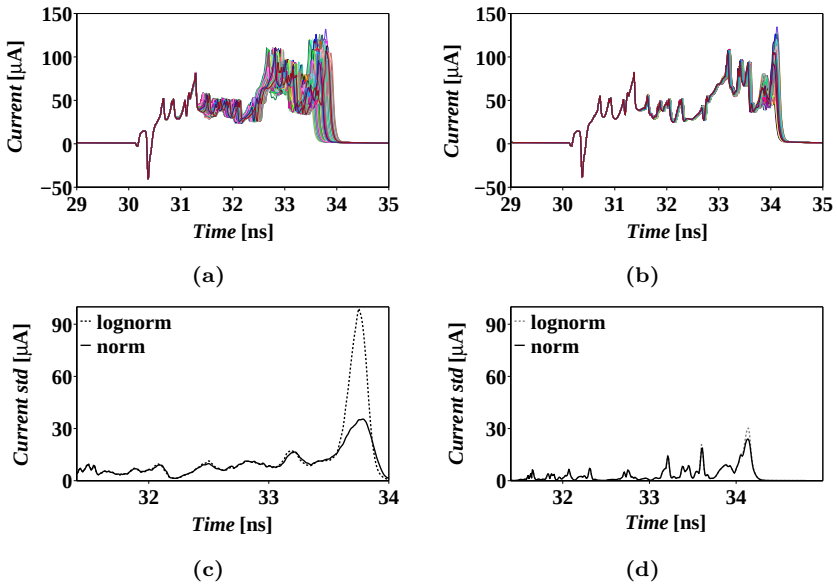


Fig. 2.15.: Simulated current traces and standard deviations of DDSLL compact (a, c) and balanced structures (b, d) for 256 different input transitions.

deviation over the inputs for most of the time samples of the balanced DDSLL S-box compared to the compact version is not important as long as there exists at least one time sample that features a similarly high standard deviation.

Furthermore, the balanced version of the DDSLL S-box caused a 7.15 % increase in the power consumption and a 12.6 % increase in its delay compared to the compact one. Therefore, these conclusions reinforce our believe that balancing the DPDNs is not a trivial task and that an interesting research topic is to investigate other balancing techniques in order to reduce the current standard deviations over the input sequences to a minimum level without increasing the performance overheads.

The use of balanced DPDNs does not enormously reduce the maximum standard deviation (normal) with respect to the compact version of the DDSLL S-box. Therefore, the security of the balanced DPDN is not expected to be improved. As a result, current balancing techniques are not sufficient and new methods need to be investigated. In the following discussions, we only use the compact form of the DDSLL S-box.

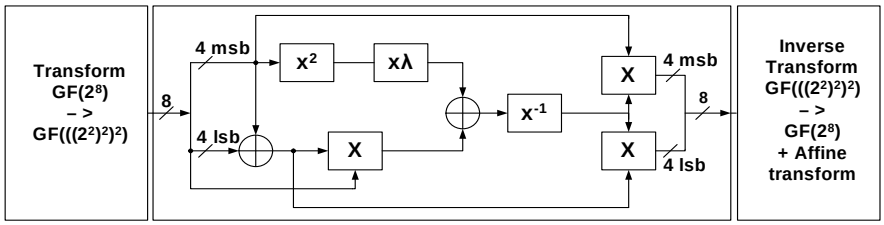


Fig. 2.16.: AES S-box block diagram [47].

Table 2.3.: Characteristics of the S-boxes.

Circuit	Logic gates	Logic depth	Transistors
2-input static CMOS	138	22	1530
4-input static CMOS	90	13	1099
SABL	90	13	1672
DyCML	90	13	1241
DDSLL	90	13	1275

2.7 CASE STUDY SUMMARY

We employ the AES S-box [20] as implemented in [22] and based on the proposed architecture of [46, 47]. The block diagram is shown in Fig. A.1. Details of the AES S-box implementation is described in Appendix A. We designed the S-box using two versions of static CMOS; namely the 2-input static CMOS that limits the gates to simple 2-input XOR/XNOR, AND/NAND gates and the 4-input static CMOS that uses 4-input functions as needed besides the before mentioned 2-input gates. Also, SABL, DyCML and DDSLL S-boxes are implemented and the characteristics of all S-boxes are summarized in Table 2.3. The circuit implementation of all the DDL styles is quite simple, each function can be realized using a single DPDN and all functions that operate at the same time can be grouped to share the common blocks as discussed in Section 2.5.4. Pre-layout simulation results are available for both versions of static CMOS and all DDL styles, whereas post-layout simulation results are available for the 2-input static CMOS S-box and the DDSLL styles only. It is worth mentioning that the 2-input static CMOS S-box was initially designed with the intention to work at sub-threshold voltages.

2.8 SIMULATION AND MEASUREMENT RESULTS

2.8.1 Test chip

A test chip was fabricated in order to investigate the power consumption, delay performances and the security of the DDSLL S-box and compare it to a reference static CMOS S-box [21]. In this chapter we study the power consumption and

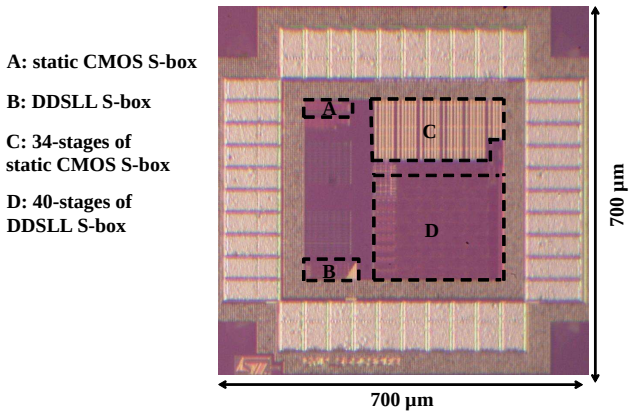


Fig. 2.17.: Test chip

delay performances as means only. The deviations of the power consumption and the delay of both S-boxes are detailed in Chapter 3. Whereas, the security assessment of the S-boxes is addressed in Chapter 4. Here, the static CMOS implementation uses SVT devices only with minimum gate length and 0.12/0.24 μm wide NMOS/PMOS transistors, while the DDSLL S-box is designed as explained in Section 2.5.2 with a target of minimizing both the area and the power consumption. As a result, the area of the DDSLL S-box is 1125 μm^2 which is only 12 % larger than its counterpart in static CMOS.

Fig. 2.17 shows the microphotograph of the test chip which has been fabricated using a 65 nm LP CMOS technology. The test chip comprises two blocks consisting of stand-alone S-boxes for power consumption measurements in static CMOS (block A) and DDSLL (block B) logics. Also to measure the delay performance of both logic styles, two blocks consisting of n-stage chained S-boxes in static CMOS (block C, n=34) and DDSLL (block D, n=40) styles are realized.

2.8.2 Power consumption test bench and measurement setup

To measure the power consumption, a single DDSLL S-box or static CMOS S-box is sufficient as the current sensitivity of the measuring equipment is very high which allows for measuring currents in the range of pA. The used test bench simply consists of input and output buffers that have an independent supply source, whereas the S-box under test has a separate supply source that also acts as a meter to measure the current of the S-box under test.

Simulations are done at typical conditions (V_{DD} = 1.2 V, temperature = 25°C, typical NMOS and PMOS transistors) using ELDO SPICE simulator. It is worth reminding that both static CMOS and SABL are full-swing logics while DyCML and DDSLL are low-swing logics. In order to have a fair comparison between DDSLL and DyCML styles, the transistors of the latter are sized so that the output voltage swing is the same as that of the DDSLL. Of course, having

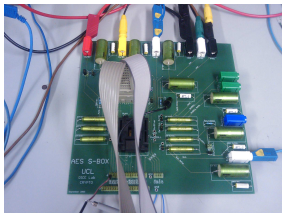
a less output swing for the DyCML is feasible, however in practice, we should maintain a sufficient swing to mitigate the impact of WID variability. As a result, we chose to keep the voltage swing of DyCML the same as for DDSLL ($0.6V$). The power consumption is extracted by averaging over 10 input transitions.

As for the measurement setup, it consists of a $100MHz$ digital waveform generator/analyzer (NI 6552) to provide the 8-bit input pattern to the S-boxes and the clock signal to the DDSLL S-box. It also consists of three power supplies for ESD, I/O buffers and for sourcing the voltage to the power supply of the S-box under test and measuring the current drawn from it. A Keithley 236 SMU is used here as its current sensitivity is $10fA$. Fig. 2.18 shows all the equipment used in the test bench to measure the power consumption of the S-boxes along with the PCB.

(a) National instruments $100MHz$ digital waveform generator/analyzer (NI 6552)

(b) Keithley K2400 source meter used for I/O buffers

(c) Keithley K236 source meter used to measure the current of the S-box under test



(d) The PCB under test

Fig. 2.18.: The measurement equipment used to measure the power consumption of the S-boxes.

In order to measure the power consumption, we use a 10 input transition pattern that is continuously repeated. In the case of 2-input static CMOS S-box, the leakage power consumption P_{leak} is measured by introducing constant inputs, then averaging over the corresponding leakage currents and multiplying by the supply voltage. The dynamic power is the result of subtracting the leakage power from the total power. In the case of DDSLL S-box, only the total power consumption can be measured. This is mainly due to the dynamic nature of the

DDSLL style where it has precharge and evaluation periods and also because of its floating nodes during evaluation. As a result, reproducing the same voltage levels as in normal operation during static mode (setting the clock and input signals to either GND or V_{DD}) is not possible. For both S-boxes, the measured power consumption results are averaged over 20 dies.

2.8.3 Power consumption performance results

First, we show the impact of sharing the common blocks on DDSLL in comparison to static CMOS logic using SPICE simulations. Fig. 2.19 plots the total power consumption of several blocks implemented using 2-input static CMOS and DDSLL with or without sharing *versus* the number of gates used by the 2-input static CMOS blocks. Simulations are done at typical conditions (V_{DD} = 1.2 V, temperature = 25°C, typical NMOS and PMOS transistors) without adding the extracted routing parasitics (only gate and junction capacitances are accounted for) at 100 kHz. It can be seen that the total power consumption of a single DDSLL XOR gate is almost 3× higher than that of a 2-input static CMOS XOR gate. Therefore, if DDSLL gates are implemented without sharing the common blocks, the power consumption is kept higher than that of the corresponding 2-input static CMOS blocks as seen for example, in the case of the S-box (1.4×). However, if the principle of sharing is applied to the DDSLL gates, then the power consumption tends to be lower than that of the 2-input static CMOS as the number of gates increases. As a result, the DDSLL S-box that uses the sharing principle consumes 1.9× less power than the 2-input static CMOS S-box and 2.7× less power than that without sharing. Knowing the fact that a 4-input static CMOS complex gate would have reduced dynamic and leakage power consumptions with respect to its 2-input static CMOS implementation, simulation results are conducted on the 4-input static CMOS S-box and its total power consumption is 24 % less than the 2-input static CMOS implementation. If the power of the DDSLL S-box with sharing is compared to that of the 4-input static CMOS S-box, the reduction will be a factor of 1.5× (instead of 1.9× if compared to the 2-input static CMOS S-box). However, the power reduction brought by the principle of sharing in the DDSLL style is evident regardless of the static CMOS implementation.

As the simulation of the DDSLL S-box demonstrates a significant reduction of power thanks to sharing the common blocks, here we apply the sharing principle to the SABL, DyCML S-boxes as well and compare their power consumption to that of the 2-input and 4-input static CMOS S-boxes. It is worth reminding that in this chapter we address the mean power consumption only and the analysis of the power consumption deviations is separately discussed in the following chapter.

Fig. 2.20 shows that the power consumption of SABL S-box is comparable to that of 2-input static CMOS at 100 kHz and starts to show a larger difference of about 30 % above 1 MHz. However when compared to 4-input static CMOS S-box, the SABL S-box consumes even more power, 25 % and 64 % at 100 kHz and

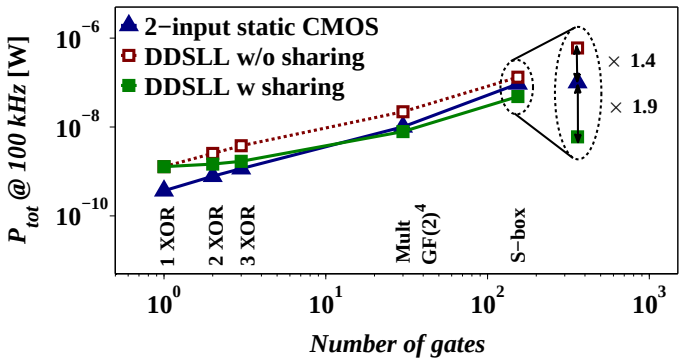


Fig. 2.19.: Evolution of total power consumption *versus* the number of gates due to the sharing principle.

above 1 MHz, respectively. Fig. 2.20 also shows that both the DyCML and the DDSLL S-boxes consume similar power which is almost 2× and 1.45× less than that of 2-input static CMOS S-box at 100 kHz and above 1 MHz, respectively. Nevertheless, when DyCML and DDSLL S-boxes are compared to 4-input static CMOS, the power reduction is limited to 1.5× and 11 % at 100 kHz and above 1 MHz, respectively. However, the power advantage of DDSLL and DyCML styles over SABL is clear; a reduction by a minimum factor of 1.85. Another interesting point is that above 1 MHz, the leakage power starts to be negligible and the P_{tot} represented as $\mu W/MHz$ is almost constant. However at 100 kHz, where the leakage power has a significant contribution, the P_{tot} represented as $\mu W/MHz$ is greater than that at higher frequencies. This is true for all logic styles, however for the DDL styles, the contribution of leakage power at 100 kHz seems to be less than that of the two versions of static CMOS. Using rough calculations, the leakage power is 40 nW, 28 nW, 16 nW, 11 nW and 9 nW for 2-input static CMOS, 4-input static CMOS, SABL, DyCML and DDSLL S-boxes, respectively.

The benefit of using a DDL style over static CMOS appears first in the reduction of the leakage power, more pronounced in DyCML and DDSLL due to the dynamic operation and in the use of a larger stack of transistors. However, not all DDL styles feature a reduced dynamic power. Although they all use the sharing principle discussed in Section 2.5.4 and they implement complex functions in single DPDNs as explained in Section 2.5.3, SABL S-box consumes 64 % more dynamic power than 4-input static CMOS S-box while DyCML and DDSLL S-boxes consume 12 % less at 100 kHz. This is because DyCML and DDSLL are low-swing logics while SABL is a full-swing logic. It is noteworthy that so far, simulations are done without adding the extracted routing parasitics (only gate and junction capacitances are accounted for).

As a further step to more realistic simulation results, extracted parasitic routing capacitances are added to the 2-input static CMOS and DDSLL S-boxes only. Table 2.4 demonstrates the effect of routing capacitance on the power con-

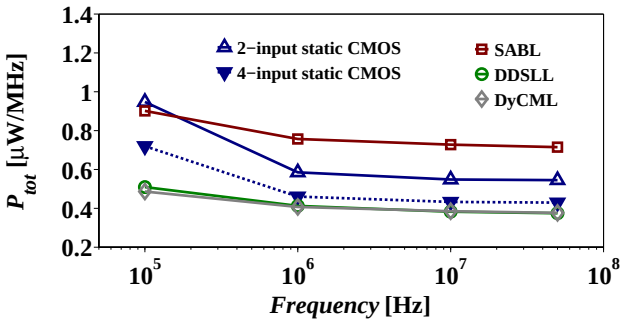


Fig. 2.20.: SPICE simulations of total power consumption (without adding the extracted routing parasitics) *versus* frequency of 2-input and 4-input static CMOS, SABL, DyCML and DDSLL S-boxes (average over 10 input transitions).

Table 2.4.: Power consumption simulation (pre- and post-layout) and measurement results of both 2-input static CMOS and DDSLL S-boxes.

	static CMOS			DDSLL		
	$No\ C_{rout}$	$Typical\ C_{rout}$	$Meas.$	$No\ C_{rout}$	$Typical\ C_{rout}$	$Meas.$
P_{stat} [nW]	46.6		20.3	-	-	-
P_{dyn}@100 kHz [nW]	54.2	107.9	106.4	-	-	-
P_{tot}@100 kHz [nW]	99.4	154.5	126.8	48.8	96.1	82.2
P_{dyn}@13.56 MHz [nW]	7,365	14,830	14,620	-	-	-
P_{tot}@13.56 MHz [nW]	7,412	14,877	14,640	5,246	10,856	9,042

sumption of 2-input static CMOS and DDSLL S-boxes. Results are shown at 100 kHz (the minimum clock frequency for throughput/latency constraint in passive RFID tags [6]) and 13.56 MHz (the clock frequency of contactless smart cards - ISO/IEC 14443). It shows that the dynamic power consumption of the 2-input static CMOS S-box increases almost by a factor 2× due to routing. As a result, the total power consumption of the static CMOS S-box increases by 1.55× due to the notable contribution of leakage power (46 % in case no routing parasitic capacitances are added and 30 % in case they are added) and 2× at a frequency of 100 kHz and 13.56 MHz, respectively. As for the DDSLL S-box, since it is a dynamic logic it is more practical to compute the total power which also increases by 2×.

Next, the power consumption of both 2-input static CMOS and DDSLL S-boxes are measured using the 10 input transition pattern and under the conditions described in Section 2.8.2. The comparison between measured and simulated power consumption of both S-boxes is also given in Table 2.4. Looking at the leakage power consumption of the 2-input static CMOS S-box, one can

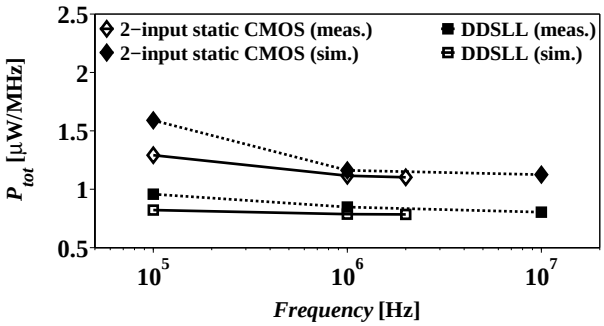


Fig. 2.21.: Measurement and simulation results (with extracted parasitic routing capacitances) of total power consumption *versus* frequency of 2-input static CMOS and DDSLL S-boxes (average of 20 dies, average over 10 input transitions).

fairly assume that the measured dies are between an SS corner (slow NMOS, slow PMOS) and the TT conditions (typical NMOS, typical PMOS). This is because at TT conditions the simulated leakage power is 46.6 nW, while at SS corner it is 9 nW. On the other hand, the measured dynamic power of the 2-input static CMOS S-box is accurately predicted by simulations as both TT conditions and SS corner render the same dynamic power. Accordingly, the measured total power of the 2-input static CMOS S-box is 18 % less than simulation at TT conditions. It is worth mentioning that according to the measurement results of the 2-input static CMOS S-box, the leakage power is 16 % of the total power at 100 kHz. As for the DDSLL S-box, the measured total power is 14.4 % less than the total power predicted by simulation at TT conditions.

Additionally, Fig. 2.21 shows the same trend as simulations for the measured total power consumption *versus* frequency for both S-boxes. As explained before, the leakage power can almost be neglected above 1 MHz, whereas at 100 kHz it has a substantial share for both 2-input static CMOS and DDSLL S-boxes. However for DDSLL, the contribution of leakage power at 100 kHz seems to be less than that of 2-input static CMOS, confirming the results obtained from simulations. Using the same calculations as for simulation, the leakage power is 19.5 nW and 3.9 nW for 2-input static CMOS and DDSLL S-boxes, respectively, which for the case of 2-input static CMOS matches the measured leakage power calculated for constant inputs in Table 2.4. To conclude, the measured power reduction brought by the DDSLL S-box is found to be 35 % which is fairly predicted by post-layout simulation (37.8 %), compared to 2-input static CMOS S-box at 100 kHz.

Moreover, the 2-input static CMOS S-box could be operated at a near-threshold supply voltage which would dramatically reduce its power consumption compared to DDSLL. However, we use the nominal 1.2 V supply for both S-boxes. Indeed, for cost concern, RFID tags and smart cards applications still fa-

vor operating at a nominal voltage to avoid multiple power domains on the chip, given that other circuitries such as the SRAM can hardly be operated at lower supply voltages. Another important aspect is security which will be discussed separately in Chapter 4. It is shown in [48] that operating at a near-threshold supply voltage greatly affects the security of the static CMOS S-box due to the rising impact of variability. As a result, the security evaluation of an implementation greatly depends on the choice of the supply voltage. In order to have a practical comparison from the application stand-point and a fair comparison of security, we thus maintain this choice of 1.2 V supply.

Finally, we conclude that reducing the power consumption of DDL styles that use the DPDN concept in general can be achieved by applying three main steps; share the common blocks, implement the functions as complex DPDNs when necessary and reduce the voltage swing. Indeed, all three steps are necessary, yet reducing the voltage swing renders the DDSLL and DyCML styles the most power efficient DDLs. Here, we compare DDSLL and DyCML to SABL and we conclude that the former styles reduce the power consumption by a minimum factor of 1.85. Also, DDL styles in the literature are known to consume a minimum of 2× more power compared to static CMOS; for example, [14, 16, 25]. In this section, we show that at frequencies higher than 1 MHz, DDSLL and DyCML consumes 12 % less power than static CMOS implemented with 4-input gates (similar to DDSLL). However at 100 kHz, the reduction factor is 1.5. As a result, low-swing DDLs (DDSLL and DyCML) are considered a potential solution for power constrained secure applications if they prove to enhance the security as well. Anyway, the security assessment of the DDSLL style is discussed in Chapter 4.

Reduction of the power consumption of DDL styles that use the DPDN concept is achieved by:

1. **Sharing the common blocks.**
2. **Using complex DPDNs to implement complex functions.**
3. **Reducing the voltage swing.**

2.8.4 Delay test bench and measurement setup

At the simulation level, the delay of either the static CMOS or DDSLL S-boxes is easily extracted from a single S-box using a test bench similar to the one described in Section 2.8.2. For the static CMOS S-box, the delay is calculated as the worst-case delay between the input and output signals of the S-box. However for the DDSLL S-box, the delay is calculated as the worst-case delay between the input and output clock signals of the S-box. Simulations are done using the same conditions described in Section 2.8.2.

As for the actual measurements, the delay of both S-boxes is measured over a chain of n-stages between two flipflops as this relaxes the requirements of the test equipment. The input/output transition that provides the worst-case delay in simulations is considered for both S-boxes. Actually, the input pin Din is the one that features the transition, whereas as the remained input pins of both S-boxes are hard-wired to either V_{DD} or GND. Fig. 2.23 shows the block diagram and the waveforms of the test bench used to measure the delay of either S-box. Here, two flipflops are clocked by the same Clk signal. The one at the beginning of the chain is used to toggle a single transition in Din signal through the n-stages of the S-box chain and the other one at the end of the chain is used to capture the transition of the output of the last S-box in the chain $Dout$. By reducing the clock period to the same range of the chain delay, the output flipflop will not be able to sample the transition of the output of the last S-box in the chain due to setup time violation which indicates that this clock period represents the delay of the chain. The Din and Clk signals are generated as data signals in order to avoid synchronization problems with the actual clock of the pattern generator. Of course, there could still be some skew between the two data signals, but it is expected to be much less than between a clock and a data signal. Therefore, the frequency of the Clk in the case of static CMOS is actually half the entered frequency to the pattern generator, while that of the DDSLL is 1/5 the entered frequency to the pattern generator since the precharge phase is considered 20 % of the clock period. To relax the requirements of the test equipment, the static CMOS chain is constructed of 34 stages, whereas the DDSLL S-box chain consists of 40 stages. Therefore, the delay of a single gate of DDSLL S-box is:

$$t_d|_{DDSLL} = (\frac{5}{f_{gen}})/40 \tag{2.8}$$

while the delay of a single gate of static CMOS S-box is:

$$t_d|_{static\ CMOS} = (\frac{2}{f_{gen}})/34 \tag{2.9}$$

where $t_d|_{DDSLL}$ and $t_d|_{static\ CMOS}$ are the delays of a single S-box in DDSLL and static CMOS styles, respectively and f_{gen} is the frequency of the NI 6552 pattern generator.

The measurement setup consists of a $100MHz$ digital waveform generator/analyzer (NI 6552) to provide the 1-bit input pattern that acts as the single transition of Din and the clock signal that feeds both input and output flipflops associated with the S-boxes. It also consists of three power supplies for ESD, I/O buffers and for the chain of S-boxes. To monitor the waveforms and hence calculate the delays of the S-boxes, MSO8104A 1 GHz real-time oscilloscope is used (see Fig. 2.22).

Fig. 2.22.: MSO8104 1 GHz real-time oscilloscope from Agilent.

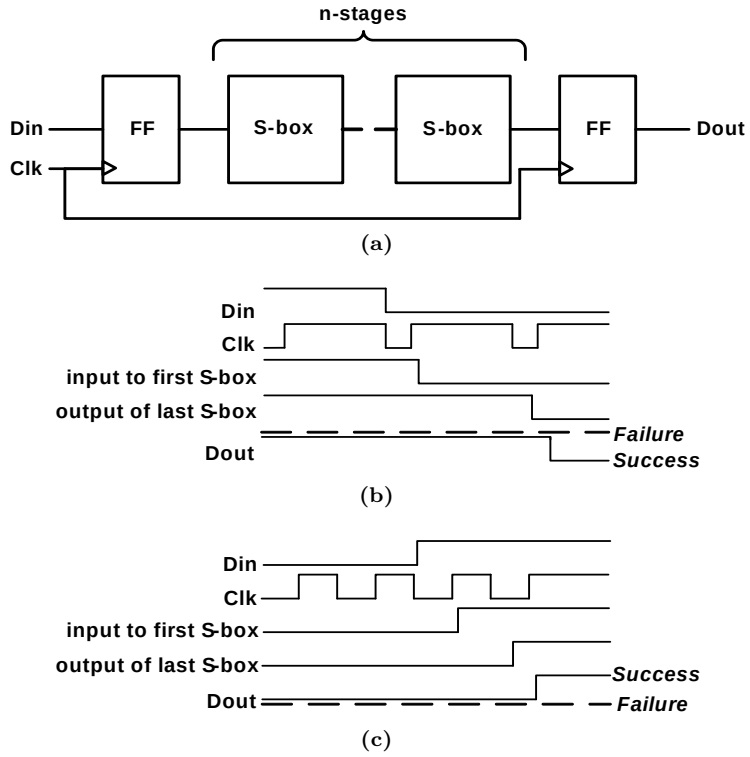


Fig. 2.23.: (a) Delay measurement test bench block diagram and related waveforms in case of success/failure for (b) DDSLL S-box and (c) static CMOS S-box.

2.8.5 Delay performance results

First, we compare the delay of the 2-input and 4-input static CMOS, SABL, DyCML and DDSLL S-boxes at the simulation level without adding the extracted routing parasitics (only gate and junction capacitances are accounted for). Fig. 2.24 plots the power consumption of these S-boxes at 100 kHz *versus* the delay and Table 2.5 details the numerical values. It shows that the DDSLL is the most suitable logic style because it has a 2× and 1.5× reduction in power consumption compared to 2-input and 4-input static CMOS, respectively and only a 2.5× increase in delay. DyCML also shows the same power consumption as DDSLL, but it has a greater delay penalty which is 6.8× higher than that of the two static CMOS versions. On the other hand, the SABL S-box consumes almost the same power as 2-input static CMOS with a 70 % increase in delay. These results further confirm the conclusion of [15] which clearly states that both DDSLL and DyCML consume comparable power while having a delay ratio varying from 2.2× to 2.5×, depending on the output swing of the DyCML.

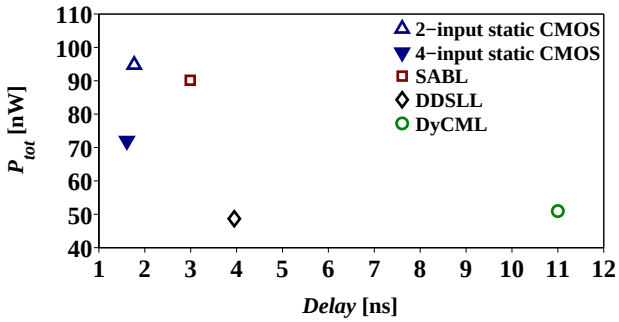


Fig. 2.24.: SPICE simulations (without adding the extracted routing parasitics) of total power consumption *versus* delay of 2-input and 4-input static CMOS, SABL, DyCML and DDSLL S-boxes (P_{tot} is at 100 kHz).

Table 2.5.: Simulation results of power consumption at 100 kHz and delay performances of 2-input and 4-input static CMOS versions, SABL, DDSLL and DyCML S-boxes at typical conditions (without adding the extracted routing parasitics).

S-box	P_{tot} @ 100 kHz [nW]	Delay [ns]
2-input static CMOS	94	1.7
4-input static CMOS	72	1.6
SABL	90	2.9
DDSLL	49	3.9
DyCML	51	11

As for the simulation results with parasitic routing capacitances and the measurement results, they are shown in Table 2.6 for the 2-input static CMOS and DDSLL S-boxes. The parasitic routing capacitances cause the delay of the 2-input static CMOS and DDSLL S-boxes to increase by a factor of $\sim$ 1.9 and 2, respectively. These results correspond well to the dynamic power consumption increase related to the parasitic routing capacitances, which is a factor of 2 for both S-boxes. Also, It is clear that the measured delays of both S-boxes are higher than that predicted by simulation with parasitic routing capacitances at TT condition which further proves the assumption made in Section 2.8.3. That is, the measured dies are closer to an SS corner than to a typical one. Numerically, the increase in the measured delay of the static CMOS S-box compared to simulation is 19 % while that of the DDSLL S-box is 6.8 % only.

Table 2.6.: The effect of parasitic routing capacitance on the delay of both static CMOS and DDSLL S-boxes.

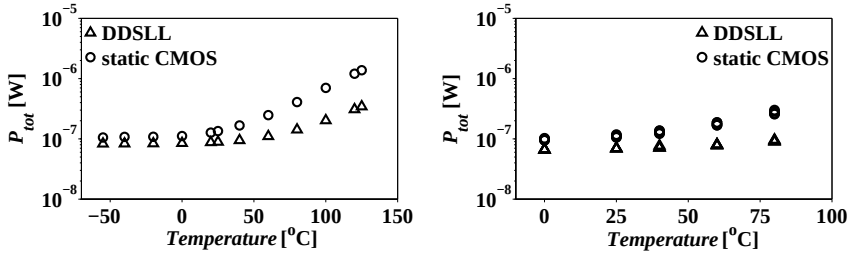
	static CMOS			DDSLL		
	$No\ C_{rout}$	$Typical\ C_{rout}$	$Meas.$	$No\ C_{rout}$	$Typical\ C_{rout}$	$Meas.$
delay [ns]	1.4	2.6	3.1	3.6	7.3	7.8

2.8.6 Effect of temperature on power and delay

In this section we demonstrate the impact of temperature on the power consumption and delay of the static CMOS and the DDSLL S-boxes. Simulation as well as measurement results in Fig. 2.25 illustrate the exponential increase in the total power consumption of both S-boxes at high temperatures. This is mainly attributed to the escalating contribution of the leakage power which is exponentially dependent on the temperature. Moreover, one can see that the gap between the power consumption of DDSLL and static CMOS S-boxes widens as the temperature increases (the ratio increases from 1.5× to 4× in favor of the DDSLL S-box as the temperature rises from $25°C$ to $125°C$). This reduction of power consumption trend brought by DDSLL with temperature is due to its reduced leakage power thanks to its large stack of transistors, its less number of stacked branches, its dynamic current source that cuts off the current when the gates evaluate their functions and the low-swing operation. Till now, we have considered only the 2-input version of the static CMOS S-box. However, if we compare the DDSLL to the 4-input static CMOS, these ratios are expected to be somehow less because it features large stack of transistors similar to DDSLL which decreases the leakage power by 40 %. Nevertheless, judging from the leakage power of both versions of the static CMOS and DDSLL S-boxes, it is still expected that the latter experiences less power consumption at high temperatures. It is worth mentioning that we were not able to measure the power consumption below $0°C$ due to problems with the measurement setup.

As for the delay, simulation results show that it is less sensitive to temperature variations for both static CMOS and DDSLL S-boxes. The static CMOS S-box features an increase of 18 % across the whole temperature range ($25°C$ to $125°C$) and the DDSLL S-box shows a similar trend (14 %). This is mainly because both logic styles operate at the super-threshold region where the $I_{on} \propto \mu \,.\, (V_{GS} - V_t)^n$. Now, $\mu \propto (T/T_0)^{-1.5}$, where T_0 is the room temperature and $V_t \propto T^{-1}$. As a result, the impact of temperature on both μ and V_t cancel out the effect of temperature on I_{on}.

An important point that may be seen as a disadvantage of DDSLL is the sensitivity of the floating nodes to temperature. Since the MOSFET subthreshold current increases exponentially with temperature, the floating nodes are more susceptible to being charged back to or close to V_{DD}. Therefore, this upsets



(a) Simulations **(b)** Measurements

Fig. 2.25.: Total power consumption *versus* temperature of static CMOS and DDSLL S-boxes at (a) the simulation level and (b) the measurement level at 1.2 V supply.

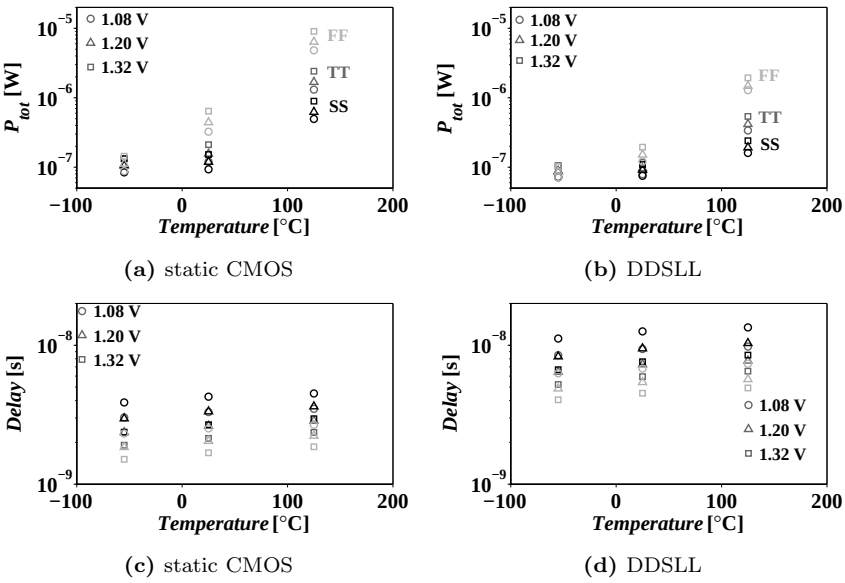
the equilibrium of the charges and may cause the DDSLL gate to fail and not function correctly. That is why for the latch and the clocked transistors; M_{10} and M_{11} of the precharge circuit, we use HVT devices because of their reduced subthreshold current compared to SVT devices.

2.8.7 PVT analysis

We further investigate the sensitivity of the DDSLL and static CMOS S-boxes to PVT corners. For the process corners, we consider the FF (fast NMOS and fast PMOS), TT (typical NMOS and typical PMOS) and SS (slow NMOS and slow PMOS). For the voltage, we use the nominal supply voltage and ± 10 %, i.e., 1.08 V, 1.2 V and 1.32 V. As for the temperature, we simulate at -55°C, 25°C and 125°C. SPICE simulations are performed using extracted parasitic routing capacitances.

First, we consider the effect of PVT on the power consumption as illustrated in Fig. 2.26 (a) and Fig. 2.26 (b). At -55°C, the power consumption of both S-boxes is insensitive to process corners. This is because leakage power that is greatly affected by process corners is severely reduced at such low temperatures and dynamic power becomes dominating.

Now, as the temperature increases, the effect of process on the power consumption of both S-boxes increases. Considering the 1.2 V supply, we report that at 25°C, the power consumption increases 206 % and 62 % for the static CMOS and DDSLL S-boxes, respectively which demonstrates the higher tolerance of DDSLL to process corners as the temperature increases. The main cause behind this tolerance is the reduced leakage power of DDSLL (∼ 18 % at TT) compared to static CMOS (∼ 30 % at TT). Therefore, process corners cause the gap of the power consumption between DDSLL and static CMOS to widen further with temperature as long as the leakage power difference is substantial. However, we can see that at 125°C, process corners affect both S-boxes in a similar way such that the power consumption increases 341 % and 307 % for the



(a) static CMOS **(b)** DDSLL

(c) static CMOS **(d)** DDSLL

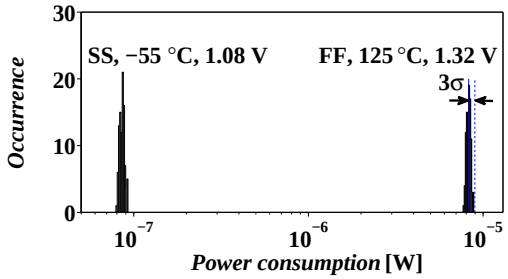
Fig. 2.26.: Impact of PVT on the power consumption of the S-boxes (a), (b) and on their delay (c), (d).

static CMOS and DDSLL S-boxes, respectively. We attribute this leap of power consumption fluctuations of the DDSLL S-box due to process corners (from 62 % at 25°C to 307 % at 125°C) to the considerable increase of leakage power at 125°C. Nevertheless, we see that for all process corners, the DDSLL S-box features a reduced power consumption compared to static CMOS (for example at TT, 1.2 V, 25°C, the power consumption ratio is 1.6× and at TT, 1.2 V, 125°C, the power consumption ratio is 4×).

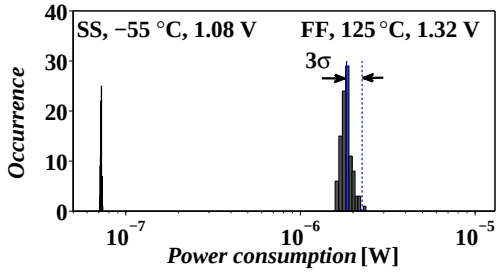
In addition, the impact of supply voltage variations is classic since it quadratically affects the dynamic power consumption of static CMOS (which is dominating in the case of the AES S-box: 84 % at 100 kHz - measurement results). Also, due to the low-swing operation of DDSLL, the supply voltage relation to the dynamic power consumption is linear. We can see the effect of supply voltage variations is moderate (compared to the impact of process corners) in the case of both S-boxes and it increases slightly with temperature.

As for the delay, Fig. 2.26 (c) and Fig. 2.26 (d) show that process corners, supply voltage and temperature have almost the same effect on both static CMOS and DDSLL S-boxes.

Finally, it is interesting to demonstrate the two extreme corners for the power consumption of both S-boxes; namely, the SS, -55°C, 1.08 V and the FF, 125°C, 1.32 V. Fig. 2.27 plots the power consumption of the static CMOS and DDSLL S-boxes at these extreme corners taking into account WID process variations as



(a) static CMOS



(b) DDSLL

Fig. 2.27.: Impact of the two extreme corners on the power consumption of both S-boxes while enabling WID variations.

well. For power closure, it is important to examine the worst-case corner while considering 3σ variations for high yield (assuming Gaussian distributions). We can clearly see the advantage of DDSLL over static CMOS as the $P_{tot}+3\sigma$ is $4\times$ less (2.25μW, 8.97μW for DDSLL and static CMOS S-boxes, respectively), although the span of the power consumption of DDSLL is larger than that of static CMOS. A complete study on the impact of process variations on DDSLL and static CMOS is discussed in Chapter 3.

2.9 SUMMARY

Although static CMOS is the traditional logic style used in building all kind of digital circuits because of its attractive features (e.g. robustness to voltage scaling, scalability with technology, automation in CAD tools, low power and small size compared to DDL styles), it fails to meet the one requirement of secure applications; the elimination of the data dependency of the power consumption. This requirement arose from the existence of recent types of attacks on cryptographic systems that are called power analysis attacks [11] which take advantage of this major drawback of static CMOS. Consequently, for the last decade researches have set their eyes on developing new logic styles with special attention to DDLs

[11]. However, designing logic styles for low-power secure applications is a challenging task. Indeed, currently existing solutions aim to enhance the security of the DDL styles at the expense of high performance costs. To the author's knowledge, the minimum power consumption of DDL styles is 2× higher than static CMOS coupled with the same penalty in die size.

Therefore, in this chapter we explore *the major design guidelines that lead to a potentially secure low-power DDL style*. We focus on DDSLL which was proposed in [15] to implement an AES S-box. The main challenge that we encountered was the design of the DPDNs of complex functions which is not trivial. Since we decided to implement complex functions with 4-input gates, we had to take care of balancing the DPDNs of these gates while trying to optimize the gate size and power consumption. Balancing simply means that the differential output capacitances and the DPDN internal node capacitances should be symmetric and independent on the data being processed. Of course, this is the ideal case. Realistically, partial balancing of the differential nodes is achievable using certain techniques that we detail in Section 2.5.3. Consequently we designed two versions of the DDSLL S-box, one with compact DPDNs (uses only some of the proposed balancing techniques) and another one with balanced DPDNs (uses most of the proposed balancing techniques). However, *the results of the balanced DDSLL S-box is not very encouraging*. By visually inspecting the current traces and calculating the standard deviation over the inputs, we see that the balanced DDSLL S-box brings only a slight reduction in the power consumption sensitivity to the input pattern. Therefore, an important open question is raised: ***"Is it possible to develop other balancing techniques that minimize the data dependency of the power consumption without adding additional performance overheads?"***

In the context of this study, we demonstrate three techniques that when combined, lead to a significant reduction of the power consumption relative to other DDL styles. *More specifically, sharing the common blocks, implementing complex functions with n-input DPDNs and reducing the voltage swing are the main contributors to the low power performance of DDSLL*. The sharing principle leads to a 2.7× reduction in power consumption of the DDSLL S-box with respect to a DDSLL S-box that does not employ sharing. On the other hand, the low voltage swing enabled DDSLL (as well as DyCML) to have less power consumption than SABL, which is a full-swing logic, by a factor of 1.9. Thanks to these techniques, the DDSLL S-box shows a 35 % reduction in power consumption at 100 kHz when compared to 2-input static CMOS S-box (measurement results) and almost the same power consumption (12 % - simulation results with only gate and junction capacitances) compared to 4-input static CMOS at frequencies higher than 1 MHz (the percentage is even higher if we consider 100 kHz). *The main drawback of DDSLL is its high delay which is 2.5× that of static CMOS*. However, DyCML shows even a greater delay penalty; 6.7× compared to static CMOS, whereas SABL features a modest delay increase of 70 %. Anyway, the delay overhead of DDSLL is considered moderate and still respects the loose timing constraints of the ultra-low-power applications. Also when operated at high tem-

peratures, both DDSLL and static CMOS S-boxes show an exponential increase in power consumption due to the escalating contribution of the leakage power (exponentially dependent on the temperature). *However, the power consumption reduction brought by DDSLL over static CMOS increases even more with temperature thanks to its lower leakage power.* In addition, the power consumption of DDSLL features less sensitivity to process corners as the temperature is raised compared to static CMOS. *As for the area, due to the compact implementation of the DDSLL S-box (using the sharing principle and implementing complex functions with 4-input DPDNs), its area is 1125 μm^2 which is only 12 % larger than its counterpart in static CMOS. However, the fact that DDSLL is designed in a full-custom fashion is considered another disadvantage.* ***Therefore, it would be an interesting topic for future research to implement the three techniques that DDSLL uses in a semi-custom flow.*** *Nevertheless, DDSLL represents a potential candidate for secure low-power applications if it proves to bring significant security enhancements with respect to static CMOS.*

Bibliography

[1] M. Roberti. (2006, May) A 5-cent breakthrough. [Online]. Available: http://www.rfidjournal.com

[2] T. Umeda, H. Yoshida, S. Sekine, Y. Fujita, T. Suzuki, and S. Otaka, "A 950-MHz rectifier circuit for sensor network tags with 10-m distance," *IEEE Journal of Solid-State Circuits, JSSC*, vol. 41, no. 1, pp. 35 – 41, Jan. 2006.

[3] Y. Hong, C. F. Chan, J. Guo, Y. S. Ng, W. Shi, L. K. Leung, K. N. Leung, C. S. Choy, and K. P. Pun, "Design of passive UHF RFID tag in 130nm CMOS technology," in *Proceedings of IEEE Asia Pacific Conference on Circuits and Systems, APCCAS*, Dec. 2008, pp. 1371 –1374.

[4] Sony. (2010) Near field communication (NFC) dynamic tag: RC S801. [Online]. Available: http://www.sony.net/Products/felica/business/tech-support/data/fp_rcs801_1.02.pdf

[5] Infineon. [Online]. Available: http://www.infineon.com/dgdl/SPI_SLE6636_1008.pdf?folderId=db3a304328c6bd5c0128d5df208b014f&fileId=db3a304328c6bd5c0128d5e58c240150

[6] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, "Strong authentication for RFID systems using the AES algorithm," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES*, 2004, pp. 357–370.

[7] Infineon. [Online]. Available: http://www.infineon.com/dgdl/my-d+vicinity_ProductBrief_2007-04.pdf?folderId=db3a30432662379201269d68cac1740f&fileId=db3a304328c6bd5c0128d63aa22a402e

[8] ISO. [Online]. Available: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=50942

[9] A. Ricci, M. Grisanti, I. De Munari, and P. Ciampolini, "Design of a 2μW RFID baseband processor featuring an AES cryptography primitive," in *Proceedings of the 15th IEEE International Conference on Electronics, Circuits and Systems, ICECS*, Sept. 2008, pp. 376 –379.

[10] M. Feldhofer and J. Wolkerstorfer, "Strong crypto for RFID tags - a comparison of low-power hardware implementations," in *Proceedings of IEEE International symposium of circuits and systems, ISCAS*, 2007, pp. 1839–1842.

[11] S. Mangard, E. Oswald, and T. Popp, *Power analysis attacks - revealing the secrets of smart cards*. Springer, 2007.

[12] S. Dziembowski and K. Pietrzak, "Leakage-resilient cryptography," in *FOCS*, 2008, pp. 293–302.

[13] L. Goubin and J. Patarin, "DES and differential power analysis (the "duplication" method)," in *Proceedings of the First International Workshop on Cryptographic Hardware and Embedded Systems, CHES*. London, UK: Springer-Verlag, 1999, pp. 158–172. [Online]. Available: http://portal.acm.org/citation.cfm?id=648252.752372

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

[15] I. Hassoune, F. Macé, D. Flandre, and J.-D. Legat, "Dynamic differential self-timed logic families for robust and low-power security ICs," *Integration*, vol. 40, no. 3, pp. 355–364, 2007.

[16] K. Tiri and I. Verbauwhede, "A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation," in *Proceedings of Design, Automation and Test in Europe, DATE*, 2004, pp. 246–251.

[17] ——, "Charge recycling sense amplifier based logic: securing low power security ICs against DPA [differential power analysis]," in *Proceeding of the 30th European Solid-State Circuits Conference, ESSCIRC*, Sept. 2004, pp. 179 – 182.

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

[19] P. Barreto and V. Rijmen. (2001) The khazad legacy-level block cipher, NESSIE project home page. [Online]. Available: https://www.cosic.esat.kuleuven.be/nessie

[20] (2001, Nov.) Federal information processing standards publication FIPS PUB 197 advanced encryption standard (AES). U.S. Department Of Commerce/National Institute of Standards and Technology (NIST). [Online]. Available: csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[21] D. Kamel, M. Renauld, D. Bol, F.-X. Standaert, and D. Flandre, "Analysis of dynamic differential swing limited logic for low-power secure applications," *Journal of Low Power Electronics and Applications, JLPEA*, vol. 1, no. 2, pp. 98–126, 2012. [Online]. Available: http://www.mdpi.com/2079-9268/2/1/98/

[22] D. Kamel, F.-X. Standaert, and D. Flandre, "Scaling trends of the AES S-box low power consumption in 130 and 65 nm CMOS technology nodes," in *Proceedings of IEEE International Symposium on Circuits and Systems, ISCAS*, May 2009, pp. 1385 –1388.

[23] S. Akers, "Binary decision diagrams," *IEEE Transactions on Computers*, vol. C-27, no. 6, pp. 509 –516, Jun. 1978.

[24] J. M. Rabaey, *Digital Integrated Circuits: A Design Perspective*. Prentice Hall, 1996. [Online]. Available: http://www.amazon.co.uk/Digital-Integrated-Circuits-Design-Perspective/dp/8120322576

[25] L. Giancane, P. Marietti, M. Olivieri, G. Scotti, and A. Trifiletti, "A new dynamic differential logic style as a countermeasure to power analysis attacks," in *Proceedings of the 15th IEEE International Conference on Electronics, Circuits and Systems, ICECS*, Sept. 2008, pp. 364 –367.

[26] M. Nassar, S. Bhasin, J.-L. Danger, G. Duc, and S. Guilley, "BCDL: A high speed balanced DPL for FPGA with global precharge and no early evaluation," in *Design, Automation Test in Europe Conference Exhibition, DATE*, Mar. 2010, pp. 849 –854.

[27] R. Soares, N. Calazans, V. Lomné, P. Maurine, L. Torres, and M. Robert, "Evaluating the robustness of secure triple track logic through prototyping," in *Proceedings of the 21st annual symposium on Integrated circuits and system design*, ser. SBCCI. New York, NY, USA: ACM, 2008, pp. 193–198. [Online]. Available: http://doi.acm.org/10.1145/1404371.1404425

[28] Z. Chen and Y. Zhou, "Dual-rail random switching logic: A countermeasure to reduce side channel leakage," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES*, 2006, pp. 242–254.

[29] T. Popp and S. Mangard, "Masked dual-rail pre-charge logic: DPA-resistance without routing constraints," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES*, 2005, pp. 172–186.

[30] Kasumi. (2002) Kasumi S-box function specifications. 3gpp 35.202 technical specification version 3.1.1. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/35202.htm

[31] M. Yamashina and H. Yamada, "An MOS current mode logic (MCML) circuit for low-power sub-GHz processors," in *IEICE*, 1992, pp. 1181–1187.

[32] F. Mace, F.-X. Standaert, I. Hassoune, J.-J. Quisquater, and J.-D. Legat, "A dynamic current mode logic to counteract power analysis attacks," in *Proceedings of the 19th Conference on Design of Circuits and Integrated Systems, DCIS*, Nov. 2004, pp. 186–191.

[33] F. Mace, F.-X. Standaert, J.-J. Quisquater, and J.-D. Legat, "A design methodology for secured ICs using dynamic current mode logic," in *Proceedings of Power and Timing Modeling, Optimization and Simulation, PATMOS*, ser. Lecture Notes in Computer Science, V. Paliouras, Ed. Springer, Sept. 2005, pp. 550–560.

[34] *Roadmap ITRS 2005*, ITRS, 2005.

[35] I. Hassoune, F. Mace, D. Flandre, and J.-D. Legat, "Low-swing current mode logic (LSCML): a new logic style for secure and robust smart cards against power analysis attacks," *Microelectronics Journal*, vol. 37, pp. 997–1006, May 2006.

[36] J. Cortadella, "Mapping BDDs into DCVSL gates," UPC/DAC Technical Report No. RR 95/04, Tech. Rep., Feb. 1995. [Online]. Available: ftp://gaudi.ac.upc.es/pub/reports/DAC/1995/UPC-DAC-95-04.ps.Z

[37] R. Bryant, "Graph-based algorithms for boolean function manipulation," *IEEE Transactions on Computers*, vol. C-35, no. 8, pp. 677 –691, Aug. 1986.

[38] J. Cong, Z. Pan, L. He, C.-K. Koh, and K.-Y. Khoo, "Interconnect design for deep submicron ICs," in *Proceedings of the 1997 IEEE/ACM international conference on Computer-aided design*, ser. ICCAD. Washington, DC, USA: IEEE Computer Society, 1997, pp. 478–485. [Online]. Available: http://dl.acm.org/citation.cfm?id=266388.266534

[39] K. Tiri and I. Verbauwhede, "Place and route for secure standard cell design," in *Proceedings of Smart Card Research and Advanced Application Conference, CARDIS*, 2004, pp. 143–158.

[40] K. Tiri, D. Hwang, A. Hodjat, B.-C. Lai, S. Yang, P. Schaumont, and I. Verbauwhede, "A side-channel leakage free coprocessor IC in 0.18/spl μm CMOS for embedded AES-based cryptographic and biometric processing," in *Proceedings of Design Automation Conference, DAC*, 2005, pp. 222–227.

[41] F. Cannillo and C. Toumazou, "Nano-power subthreshold current-mode logic in sub-100 nm technologies," *Electronics Letters*, vol. 41, no. 23, pp. 1268 – 1269, Nov. 2005.

[42] J. M. Musicer and J. M. Rabaey, "MOS current mode logic for low power, low noise CORDIC computation in mixed-signal environments," in *Proceedings of the International Symposium on Low Power Electronics and Design, ISLPED*, 2000, pp. 102–107.

[43] A. Tajalli, E. Brauer, Y. Leblebici, and E. Vittoz, "Subthreshold source-coupled logic circuits for ultra-low-power applications," *IEEE Journal of Solid-State Circuits, JSSC*, vol. 43, no. 7, pp. 1699 –1710, Jul. 2008.

[44] E. Brier, C. Clavier, and F. Olivier, "Optimal statistical power analysis," *IACR Cryptology ePrint Archive*, vol. 2003, p. 152, 2003.

[45] ——, "Correlation power analysis with a leakage model." in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES*, 2004, pp. 16–29.

[46] N. Mentens, L. Batina, B. Preneel, and I. Verbauwhede, "A systematic evaluation of compact hardware implementations for the Rijndael S-Box," in *CT-RSA*, 2005, pp. 323–333.

[47] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A compact Rijndael hardware architecture with S-Box optimization," in *ASIACRYPT*, 2001, pp. 239–254.

[48] M. Renauld, F.-X. Standaert, N. Veyrat-Charvillon, D. Kamel, and D. Flandre, "A formal study of power variability issues and side-channel attacks for nanoscale devices," in *EUROCRYPT*, 2011, pp. 109–128.

CHAPTER 3

IMPACT OF PROCESS VARIATIONS ON THE POWER, DELAY AND FUNCTIONALITY OF DDSLL AND STATIC CMOS STYLES

3.1 INTRODUCTION

The high demand for low-power secure and portable electronic applications motivates the use of advanced power-management techniques aiming at meeting tight power constraints and minimizing energy per operation. Amongst them, voltage scaling is the most straightforward. It leads to a quadratic reduction in the energy (power) consumed to switch internal capacitances at the expense of delay penalty [1]. Voltage scaling can either be static with a fixed V_{DD} reduction at design time or dynamic with on-demand V_{DD} lowering in low-power modes [2]. When speed performances are not critical, V_{DD} can ultimately be set at a value below the threshold voltage V_t, leading to the so-called subthreshold logic [3, 4].

On top of this, CMOS technology scaling brings increased speed performances and functionalities with reduced energy per operation. However, when reaching nanometer-scale geometries for MOSFET devices, variability becomes a serious concern [5, 6]. Nowadays, circuit designers face a challenging task to meet the market's constrained specifications in presence of variability. Indeed, the delay and the leakage power consumption of nano-CMOS circuits are inevitably influenced to a great extent. Consequently, it is important that the designers take these effects into account during early design stages. As a result, the variability impact on circuit performances should thus be properly characterized and modeled for allowing its prediction and minimization at design time. This motivated

CAD-tool providers to integrate statistical simulation tools that incorporate variability [7, 8]. These statistical simulation tools either target the timing issues [9] or the leakage power [10, 11].

On the first order, MOSFET variability sources can be classified into two main categories:

- spatially-correlated variations that equally affect all transistors from the same type on die-to-die (D2D), wafer-to-wafer (W2W) or lot-to-lot (L2L) basis,
- uncorrelated variations that affect each transistor independently on a within-die (WID) basis.

Let us precise here that we mean spatial randomness between manufactured circuits. Indeed, noise-induced temporal randomness does not come from MOSFET variability and is thus beyond the scope of this paper.

At circuit level, these sources induce delay variations. Cycle time margins are thus required to accommodate worst-case delay. This has traditionally been dealt with at design time by carrying out global process corner simulations. However, the recent increase of uncorrelated variability sources in nanometer CMOS technologies such as random dopant fluctuations (RDF) and line edge roughness (LER) has motivated the development of statistical static timing analysis tools [9]. They help to avoid overestimating margins, by taking averaging effect and parametric yield into consideration. Deep voltage scaling worsens the picture by magnifying the sensitivity against MOSFET variations [12]. Noticeably, V_t contribution to delay variations increases as V_{DD} is scaled down. This comes from I_{on} dependence on V_t, which increases with the reduction of the gate overdrive voltage ($V_{DD} - V_t$) and which ultimately becomes exponential in subthreshold regime. Also, WID delay variations might lead to hold time violations due to high timing uncertainties [13]. These effects get worst when entering the nanometer era [14].

Delay variations also impact the dynamic energy (power) in an indirect way. Till recently, dynamic energy / power has been considered as weakly sensitive to variability [5, 10], and thus it is not usually statistically modeled in circuit simulations [11, 12, 15] nor explicitly measured in circuit characterizations [16–18]. In the first part of this chapter, we thoroughly study the dynamic energy (power) variations in voltage-scaled logic circuits; namely ring oscillators and static CMOS S-box which represent two different types of logic circuits. We demonstrate that uncorrelated WID delay variability significantly magnifies the dynamic energy (power) variations at low voltages via the activity factor (α_F) because of the generated random glitches [19]. Meanwhile, we confirm that dynamic energy (power) variations remain low at 1 V. From this study, we show that dynamic energy variability cannot be neglected in combinatorial circuits at low voltages and that statistical simulations are required for properly capturing worst-case energy consumption.

In the second part of this chapter we aim at assessing the influence of process variations on the DDSLL style, with respect to static CMOS, as a potential

candidate for low-power secure applications, since it demonstrated an interesting power reduction compared to other dynamic differential logic (DDL) styles as explained in Chapter 2. First, we introduce the impact of process variations on the power consumption of both logic styles as an extension to the work that has been done in [20]. Indeed, variability sources strongly affect leakage power of logic circuits through variations of subthreshold leakage. Its exponential dependence on V_t and L_g through short-channel and DIBL effects makes it also highly sensitive to both D2D and WID variability sources [5]; thus promoting their incorporation in statistical simulation / prediction tools [10, 11]. However in this chapter, we do not address the leakage power because of the following reasons. First, DDSLL is a dynamic logic; thus, it consumes negligible leakage power (roughly 4.7 % according to measurement results Chapter 2). Second, even at 100 kHz, which is the lower bound on the clock frequency for throughput / latency constraint in passive RFID tags [21] (an example of power-constrained applications), the measured power consumption of the static CMOS S-box at 1.2 V in a 65 nm low-power CMOS process is dominated by 84 % dynamic power as shown in Chapter 2. Therefore, we decided to only focus on the dynamic power variability. Thanks to the deterministic α_F of the DDSLL style, its WID dynamic power variability averages out with the logic depth and/or circuit size and its D2D dynamic power variability is constant. On the other hand, static CMOS suffers from an accelerating WID and D2D dynamic power variabilities for increasing logic depth because of the glitch-induced α_F variations. Consequently, in case of tight power budgets, DDSLL is expected to meet power closure with minimum power deviations, whereas static CMOS can potentially jeopardize power closure due to its high dynamic power variability, not to mention the leakage power variability as well.

We also study the impact of process variations on the delay of both logic styles. DDSLL features high tolerance to WID process variations as manifested by the case of the 13 stage DDSLL S-box; whereas, the delay of the static CMOS S-box suffers from V_t variability. As a result, the DDSLL style reduces the probability of hold time violations by limiting the timing uncertainties. However, the delay of the DDSLL style suffers from high D2D process variations, which is still independent on the logic depth. This result is considered a drawback in the DDSLL style which causes parametric yield loss.

Finally, we address the functionality issues of both S-boxes at low supply voltages in light of WID process variations. Indeed, process variations lead to vanishing noise margins in SRAMs and register circuits [22]. Static CMOS combinatorial circuits (logic) are known to function correctly at low voltages despite the associated large process variations [23]. However, the functionality of the DDSLL style proves to be sensitive to WID process variations at low supply voltage in compliance with the claim in [23] that dynamic circuits are vulnerable to variability.

3.2 SOURCES AND TYPES OF VARIABILITY

Variability is generally classified according to its nature which can either be temporal or spatial [5]. Temporal variability can range from nanoseconds to years according to its source. Examples of these sources are: SOI history effect, self heating, aging induced deviations, Negative-bias temperature instability (NBTI), hot electron effects and electromigration (EM). However, temporal variability is out of the scope of this study and therefore we focus only on the second type of variability which is spatial in nature. Spatial variations exists between devices; within-die (WID), between dies; die-to-die (D2D), across wafers; wafer-to-wafer (W2W) and across lots; lot-to-lot (L2L) [5, 18]. Furthermore, spatial variations can be sub-categorized as random (due to random dopant fluctuation (RDF) [24, 25], line edge roughness (LER) [26, 27] and oxide thickness variations [28, 29]), systematic (e.g., well proximity effects [30] and wire thickness variation [31]) and environmental (such as, temperature and supply voltage variations [32]) [33].

Random variations are device specific because they are present at the atomic level. They mainly impact the threshold voltage, the gate length and the oxide thickness (t_{ox}) of the device. The primary contributor to the random variations is the atomistic nature of the doping profile in terms of the random number of dopants in the channel and the random positioning of the atoms that are present [6], collectively denoted by random dopant fluctuations (RDF). Consequently, V_t of the device is significantly affected by RDF. It was shown in [6] that the standard deviation of the device threshold voltage (σ_{V_t}) induced by RDF is inversely proportional to the square root of the channel area. Numerically, the $3\sigma_{V_t}$ edges up to 58 mV for minimum sized MOSFETs at the 65 nm technology node (SOI) and it is expected to worsen even more in future technologies due to further scaling [34].

The second influential contributor to random variations is the line-edge roughness which is caused by the statistical variations of lithographic-related causes [35]. Since LER is in the order of several nanometers, it was orders of magnitude lower than the critical dimension (CD) of MOS devices in old technology nodes. However as the technology aggressively scales below 100 nm, LER which does not scale accordingly, presents a substantial faction of the gate length [26]. Similar to RDF [24, 25], LER generates variations in the device threshold voltage that is inversely proportional to the square root of the channel length [25]. Moreover, σ_{V_t} due to LER increases as the device size is reduced, and it may even overweight the impact of RDF on V_t fluctuations for channel lengths below 30 nm [25, 36].

Another contributor to random variations is the atomic-scale oxide thickness variations (OTV). In sub-100 nm technologies, the physical gate oxide thickness is scaled down to 1 nm, giving rise to more than 50 % t_{ox} variations [6, 37]. Similar to RDF and LER, t_{ox} variations cause threshold voltage fluctuations [29]. They also induce variations in the surface roughness limited mobility and gate tunneling current [38]. The impact of t_{ox} variations on V_t is compared to that

of RDF using a 30× 30 nm MOSFET [6] where simulation results showed that V_t fluctuations due to RDF ranges between 2× and 5× those of t_{ox} variations. However, since both sources of variations are statistically independent, therefore the total standard deviation is given by:

$$\sigma_{V_t}|_{total} = \sqrt{(\sigma_{V_t}|_{OTV})^2 + (\sigma_{V_t}|_{RDF})^2} \tag{3.1}$$

As a result, σ_{V_t} increases by only 10 % and is still dominated by RDF [6, 33].

So far, we addressed spatial random variations while focusing on the effect of the three main sources; namely, RDF, LER and OTV on the V_t fluctuations. Indeed, these random variations are at the atomistic-level of the device; thus, they are associated with the WID spatial variations. Now, the remaining spatial variation classes are rather correlated, and they are subdivided into systematic and environmental variations as previously discussed. Nevertheless, the WID class also incorporates systematic and environmental deviations in the case of a large chip area. Systematic variations have different sources within the manufacturing process depending on the spatial class [33]. For example, as discussed in [33], D2D variability can be caused by variations in the rapid thermal annealing, in the photoresist development and in the etching [39], whereas W2W variability can be caused by focus and exposure variations where the mask position can shift from one wafer to another [40]. There exists also layout dependent variations that exhibit a spatial correlation, causing a significant systematic shift in the device parameters. For example, lithographic variations in poly pitch and orientation cause critical dimension (CD) variations [17]. Another well known example of layout dependent variations is the well-proximity effect that cause V_t variations [30]. Also [31, 41] discuss the interconnect variability that rise from the chemical mechanical polishing (CMP) process as another example of layout dependent systematic variations, causing fluctuations in the interconnect's resistance, capacitance and inductance. All these different types of correlated systematic variations correspond to changes in electrical parameters of devices and interconnects; thus they need to be integrated in the models for technologies below 100 nm.

Environmental variations are also spatial in nature and are mainly caused by deviations in the temperature and the supply voltage [32]. Indeed, power supply variations in advanced high-speed power-hungry ICs edge up to 10 % leading to a delay variability of also 10 % [33]. Also temperature variations causes the mobility and threshold voltage of transistors as well as the wire resistivity to fluctuate [33].

In the end, all these spatial variability sources ranging from WID random variations to correlated systematic and environmental variations impact the electrical parameters of both devices and interconnects, which directly affects circuit performance. Consequently, for the last few years, modeling these effects has been an important research field [10, 11, 42, 43] and the target of CAD-tool providers to integrate statistical simulation tools that incorporate them [7, 8].

Throughout the rest of this chapter we refer to spatial random variations as WID variability and correlated systematic variations as D2D variability although it also incorporates WID, W2W and L2L variabilities. The reason behind such naming convention is that the measurements are carried out using 20 dies from the same wafer, therefore, we do not encounter W2W or L2L variations. As a result, correlated systematic variations are actually D2D which inherently includes systematic WID variability if it exists. As for attributing the random variations to WID variability, this is because in our case the chip area is rather small (1 mm^2), therefore WID variability adequately presents the randomness between devices. It is important to mention that for the 65 nm CMOS technology we use, the interconnect fluctuations are not modeled in the design kit; therefore, they are not considered in our simulations, but included in measurements with only a limited effect. Also, environmental variations are not taken into account, only random WID and correlated systematic D2D variabilities are studied.

3.3 TEST CHIP AND MEASUREMENT SETUP

For monitoring dynamic power / energy variations of digital circuits, the same test chip described in Chapter 2 is used with extra structures implemented on the same chip as summarized here. In brief, it is fabricated in a 65nm low-power (LP) CMOS technology. It features logic circuits with different topologies and logic styles to highlight various variability effects. First, a fan-out 1 (FO1) inverter ring oscillator (RO) enables the characterization of simple effects. It comes in two versions with 53 (RO53) and 251 (RO251) stages to investigate the impact of averaging over the number of stages on variability. Second, the AES S-box with the architecture from [44] as implemented in [45] using static CMOS logic style as detailed in Appendix A. The static CMOS S-box represents a non-regular combinatorial circuit used to investigate complex effects. Third, the same AES S-box implemented using the DDSLL style as described in Chapter 2 emphasizes the advantage of such logic families in limiting the growing effects of variability on the dynamic power / energy of nano-CMOS digital circuits. The characteristics of all these logic circuits are described in Table 3.1.

In order to enable measurements for a wide range of supply voltages, the test chip contains buffers and level shifters. Table 3.1 summarizes the characteristics of the ring oscillators, the static CMOS and DDSLL S-boxes implemented on the test chip. Two copies (A and B) of RO251 test circuit are placed on each die to monitor within-die variability. A single copy of RO53, static CMOS S-box and DDSLL S-box test circuits is on each die. Measurement results are based on data from 20 dies. All circuits on the 20 measured dies are operational with scaled V_{DD} down to 0.2V except for the DDSLL S-box as will be explained later in Section 3.5.3. However, we limit our analysis to 0.4V, as the huge delay penalty below 0.4V strongly limits the possible applications.

Table 3.1.: Characteristics of the test chip components

Circuit	RO53	RO251	static CMOS S-box	DDSLL S-box
Architecture	-	-	8-bit input / 8-bit output	
Logic gates	52×FO1 INVs + 1 NOR2	250×FO1 INVs + 1 NOR2	138 XOR / XNOR, AND / NAND, INVs	90 XOR / XNOR, complex 4-in functions
Logic depth	-	-	22	13
Transistors	106	504	1530	1275
Device type	SVT	SVT	SVT	[a]
W_p/W_n	200 nm / 200 nm	200 nm / 200 nm	240 nm / 120 nm	[a]
L_g	60 nm	60 nm	60 nm	60 nm

[a]The device type and transistor sizing of the DDSLL S-box are detailed in Chapter 2

3.4 IMPACT OF VARIABILITY ON DYNAMIC POWER / ENERGY IN VOLTAGE-SCALED NANO-CMOS CIRCUITS

Till recently, dynamic energy / power has been considered as weakly sensitive to variability [5, 10] and thus it is not usually statistically modeled in circuit simulations [11, 12, 15] nor explicitly measured in circuit characterizations [16–18]. In [19], we thoroughly studied the dynamic energy variations in voltage-scaled logic circuits; namely ring oscillators and static CMOS S-box which represent two different types of logic circuits. In this study, we demonstrated for the first time that uncorrelated delay variability significantly magnifies the dynamic energy variations at low voltages because of the generated random glitches.

The dynamic energy is composed of two parts, the switching component (E_{SW}) due to charging of capacitive loads and the short circuit component (E_{SC}) due to the direct current path from the supply voltage to the ground. Short-circuit energy is usually 10 % of the total E_{dyn} at nominal V_{DD} and is even neglected at subthreshold operation [1]. We thus simplify the nominal E_{dyn} expression of an n-gate circuit to the summation of switching energies E_{SW} of all gates:

$$\begin{aligned} E_{dyn} &= \sum_{gate} E_{SW} \\ &= V_{DD}^2/2 \times \sum_{j=1}^{n} (\alpha_{F,j}.C_{L,j}) \end{aligned} \tag{3.2}$$

where $\alpha_{F,j}$ represents the activity factor and $C_{L,j}$ is the load capacitance of the j^{th} gate.

When it comes to variability, the distribution of E_{dyn} is modeled as a normal distribution which is verified by plotting the histograms of E_{dyn} as will be seen in the following sections. As a consequence, E_{dyn} can be considered as a summation of normally distributed random variables ($\alpha_F.C_L$), since V_{DD} is considered as fixed. Similar reasoning is applicable for the dynamic power at a fixed clock frequency. Let us precise here that we mean spatial randomness between manufactured circuits. Indeed, noise-induced temporal randomness does not come from MOSFET variability and is thus beyond the scope of this study.

In this context, we address the effect of variability on the dynamic energy / power as follows:

1. Characterize the simple effects of variability on the load capacitance through studying ring oscillators.
2. Investigate the impact of variability on the activity factor of complex combinatorial static CMOS circuits, taking the AES S-box as an example.

3.4.1 Load capacitance variability

In this section we aim at studying the effect of variability on the load capacitance component of Equation 3.2 through the characterization of ring oscillators. This

choice of circuitry is mainly motivated by the fact that in a ring oscillator α_F is always constant and it is equal to 1. Therefore, α_F is not affected by variability, allowing to study the impact of process variations on C_L only.

The two ring oscillators; RO53 and RO251 that are implemented on the test chip are used in this study. Here, the dynamic energy (E_{dyn}) is extracted from total power (P_{tot}), leakage power (P_{leak}) and frequency (f) data following:

$$E_{dyn} = \frac{P_{tot} - P_{leak}}{f} \tag{3.3}$$

The P_{tot} of the ring oscillators is measured at their natural oscillating frequency, while P_{leak} is measured when gating the oscillation through a NOR gate. First, we start by analyzing the histogram of E_{dyn} for copy A of RO251 test chip at 1 V as shown in Fig. 3.1 (a). For the sake of generality, the E_{dyn} distribution is modeled as Gaussian with a mean (μ) = 422 fJ and a standard deviation (σ) = 3.89 fJ. In this experiment, the total measured E_{dyn} variations come from three contributing variables: measurement noise, within-die (WID) and die-to-die (D2D) variability. We consider these variables as independent so that the measured total E_{dyn} standard deviation σ_{total} can be expressed as:

$$\sigma_{total} = \sqrt{\sigma^2_{noise} + \sigma^2_{WID} + \sigma^2_{D2D}} \tag{3.4}$$

In order to evaluate the importance of each contribution, we first perform noise measurement on a single die by simply repeating the measurement step 20 times. The resulting σ_{noise} is shown in Fig. 3.1 (b). It is an order of magnitude below σ_{total}. Second, we perform differential measurement of A and B copies of RO251 circuit on each die. For estimating the importance of within-die contribution, the standard deviation of the E_{dyn} difference between A and B copies are computed, denoted as $\sigma(\Delta\,|_{AB})$. Computed values are shown in Fig. 3.1 (b). From those measurements, we can isolate noise, WID and D2D contributions to total E_{dyn} variations with Equation 3.4. The relative standard deviation normalized to mean E_{dyn} for copy A of RO251 over the 20 dies are given in Table 3.2. The noise contribution is small, whereas the D2D variability is dominating. Similar contributions are observed at 0.4V.

It is important to note that in order to accurately characterize the within-die contribution to the variability of the RO251, several versions of the RO251 should have been implemented on the same die. However, for the sake of reducing the die area and hence the fabrication cost, we adopted the method of [46] to evaluate the WID variability of the RO. Only 2 copies, A and B are placed on each die as described in Section 3.3. After computing the difference between the standard deviations of the E_{dyn} of the two copies, it is then divided by $\sqrt{2}$ in order to reflect the actual value of the WID standard deviation because $\sigma(\Delta\,|_{AB})$ is $\sqrt{2}\times$ larger than the actual value.

Now, the results obtained from Table 3.2 show that the WID contribution is rather modest compared to the share of D2D variability. We try to explain this observation through the following analysis. Knowing that C_L is the only source

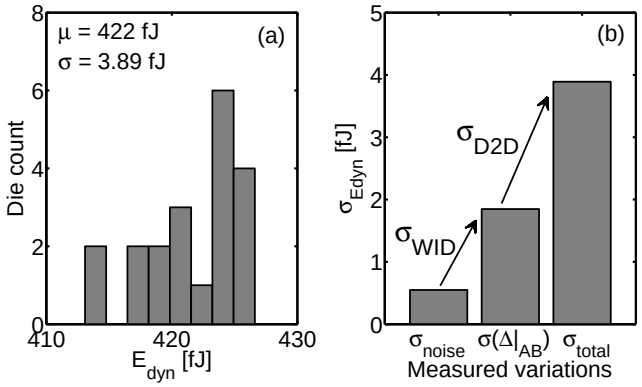


Fig. 3.1.: (a) Measured E_{dyn} of a 251-stage FO1 ring oscillator (RO251) for 20 dies at 1V and (b) standard deviations of the performed measurements.

Table 3.2.: Relative contributions to total E_{dyn} variations of RO251.

σ/μ [%]	noise	WID	D2D
@1V	0.13	0.42	0.80
@0.4V	0.23	0.53	0.76

of variability in the case of RO, we first identify the C_L components to be the gate output capacitance C_o, the input capacitance of the subsequent gate C_i and the routing capacitance C_{rout}. The first two components can be collectively considered as device capacitance. It was shown in [47] that WID device C_L variability is quite small and mainly due to random dopant fluctuations. For a device with 30nm channel length and 30nm width, the normalized standard deviation (σ/μ) is below 1% at 1V [47]. Also, the authors in [48] mention that the routing capacitance variation for a 90 nm CMOS technology is the dominant component, and it suffers from small variations, about 3 %. In this study, only the device capacitance fluctuations are studied, because the routing capacitance variations are not modeled in the design kit; therefore, they are not considered in our simulations, but included in measurements with only a limited effect (in the case of RO, there are only short interconnects that result in low C_{rout}).

Considering only the device capacitance fluctuations, Fig. 3.2 shows the instantaneous power and energy in three extreme Monte-Carlo runs of a single minimum-sized inverter under high-to-low input transition, when enabling WID variations at 0.4 V. It is clear that even though the instantaneous power strongly varies, E_{dyn} variations remain within 5% while including all variability sources; i.e., capacitances and currents. The variations of E_{dyn} in the case of the single inverter functioning at 0.4 V is somehow higher than the reported value in [47] of a single device operating at 1 V (< 1 %). The main reason of such a discrepancy is because the variability sources in the case of the inverter includes both

capacitance and short-circuit current fluctuations, where the current variations increase exponentially as the inverter enters the sub-threshold region [12, 14], whereas in [47] only the device capacitance is considered at 1 V. Moreover, as WID variations are uncorrelated, they are averaged out over the number of stages n in the ring oscillator [12] as:

$$\sigma_{WID}\,|_{RO} \quad = \quad \sqrt{n} \times \sigma_{WID}\,|_{inverter} \tag{3.5}$$

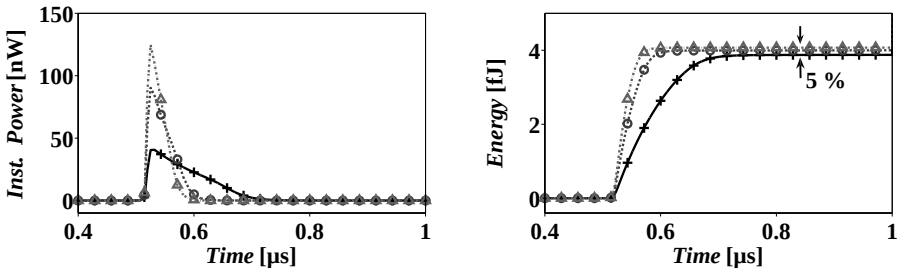
This observation is confirmed in Fig. 3.3 from Monte-Carlo SPICE simulations of an inverter chain with a varying number of stages. While the average E_{dyn} denoted as μ_{Edyn} increases linearly with the increase of gate count, the variations of E_{dyn} (which follow a Gaussian distribution) decreases as:

$$\mu\,|_{RO} \quad = \quad n \times \mu\,|_{inverter} \tag{3.6}$$

$$(\sigma/\mu)_{WID}\,|_{RO} \quad = \quad \frac{1}{\sqrt{n}} \times (\sigma/\mu)_{WID}\,|_{inverter} \tag{3.7}$$

In these simulations, only WID contribution is considered, which clearly demonstrates the averaging effect on E_{dyn} variations. Also, measurement results of the two versions of the ring oscillators, namely RO53 and RO251 are plotted. The μ_{Edyn} of the measured ROs are lower than the simulated curve, while the $(\sigma/\mu)_{Edyn}$ is rather constant (where E_{dyn} distributions of the measured ROs is assumed to be Gaussian). The measured $(\sigma/\mu)_{Edyn}$ of both ROs inherently includes noise, WID and D2D variations. This observation further manifests the dominating effect of the D2D variations, since the WID variations are averaged out and the noise contribution is relatively small as shown in Table 3.2. It also shows that the D2D variations are correlated in the sense that they do not average out with the increasing number of stages in the ring oscillator, they remain constant. To confirm the D2D contribution to the total variability, we perform corner simulations using a single device which show a global variability of C_L about 3.5 % at 1V. This is between FF (Fast NMOS, Fast PMOS) and SS (Slow NMOS, Slow PMOS) corners that implies W2W or even L2L variability. In the case of D2D variability, 3.5 % is a bit over-estimated. This validates the low E_{dyn} variability for ring oscillator structures (1 %) dominated by correlated D2D capacitance fluctuations.

Furthermore, in order to show how the variability of the ring oscillators is affected by the reduction of the supply voltage, Fig. 3.4 shows that relative E_{dyn} variations of both RO53 and RO251 circuits are small and roughly constant over the whole V_{DD} range. Again, this observation proves the dominancy of the D2D variability in the ring oscillators since the WID variations are averaged out as explained earlier, and this is valid over the whole V_{DD} range.



(a) (b)

Fig. 3.2.: SPICE simulations of 3 particular Monte Carlo runs of (a) dynamic power and (b) dynamic energy of a single inverter due to short circuit current at 0.4 V.

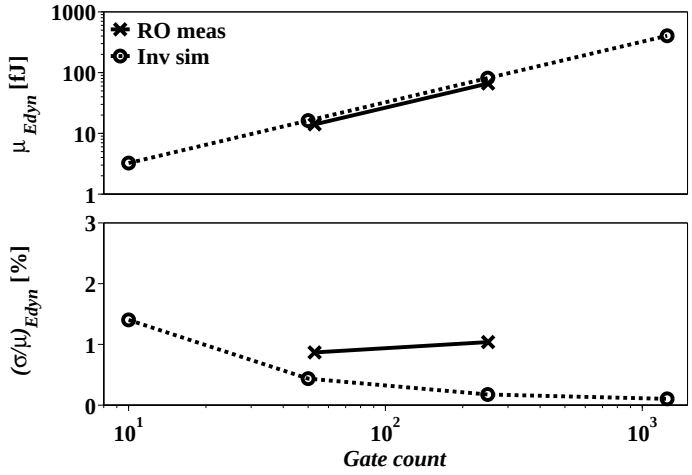


Fig. 3.3.: Measured and simulated average dynamic energy and variability of ring oscillators as a function of gate count at 1V (variability of measured ROs includes noise, WID and D2D variations).

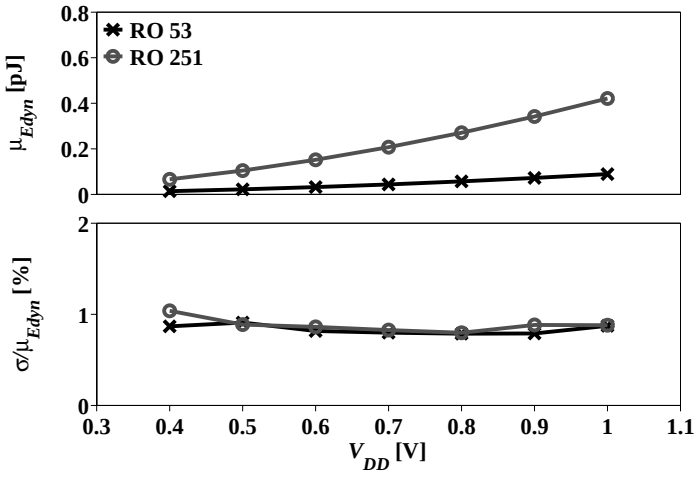


Fig. 3.4.: Measured mean dynamic energy and variability of ring oscillators with 53 and 251 stages.

The observations made in this section lead to the following conclusions:

1. **The ring oscillators represent a perfect test case to study the effect of variability on the C_L.**
2. **The WID variations of C_L are uncorrelated and hence are averaged out over the number of stages of the ring oscillator.**
3. **The correlated D2D variability of C_L is constant, independent on the number of stages of the ring oscillator and independent on the supply voltage.**

3.4.2 Glitch-induced activity factor variability

In the previous section we analyzed the impact of variability on ring oscillators in order to study the simple effects of process variations on the load capacitance that represents one of the two components of the dynamic energy affected by variability as expressed in Equation 3.2. In this section, we target the second component of the dynamic energy that is also proved to be dramatically affected by the process variations at low voltages [19]. For this purpose, we analyze the dynamic energy of a static CMOS S-box that represents a non-regular combinatorial circuit used to investigate the impact of variability on the activity factor α_F.

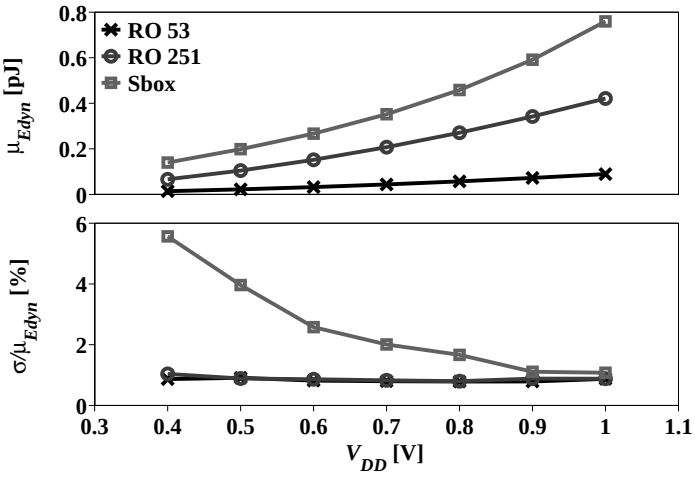


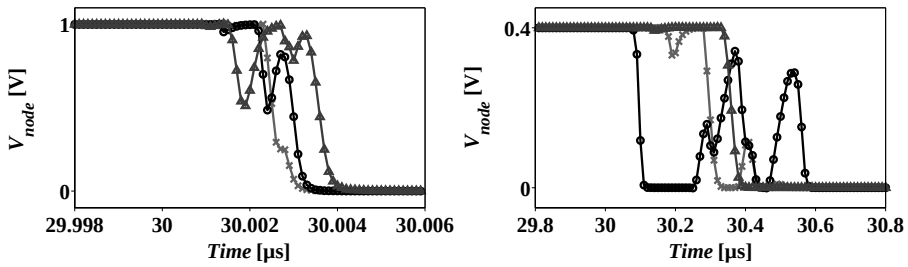
Fig. 3.5.: Measured mean dynamic energy and variability of ring oscillators with 53 and 251 stages and static CMOS S-box.

The P_{tot} of the static CMOS S-box is measured with a pseudo-random 2560-transition input pattern, at 100 kHz to avoid timing problems at low voltage. P_{leak} is measured as the average consumption over the 2^8 input vectors, under static conditions. Fig. 3.5 shows the mean measured dynamic energy (μ_{Edyn}) of the 20 dies for the static CMOS S-box and RO53, RO251 as references, and their normalized standard deviations in function of the supply voltage. It is seen that μ of the ring oscillators increases by the same ratio as the number of stages which was observed before from Fig. 3.3 in the previous section. Also, it is shown that the μ_{Edyn} of all the three circuits decreases quadratically with V_{DD}. On the other hand, σ/μ of both RO53 and RO251 are similar ($\sim 1\%$ and constant along the whole V_{DD} range). In contrast, σ/μ of the static CMOS S-box is similar to that of RO53 and RO251 at 1V, but it increases up to 5.6% as V_{DD} drops to 0.4V (where the E_{dyn} distribution of the measured S-box is Gaussian). As the dynamic energy variations depend on the C_L and α_F fluctuations and since C_L variations are merely dominated by correlated D2D variability, which is rather constant and independent on V_{DD}, then, only α_F is responsible for this behavior.

It is important at this stage to determine the causes behind the α_F variations. As explained in [1], α_F is a function of topology, signal statistics and spurious transitions or glitches associated to delay skew and logic depth. Notably, the glitches are rather random in nature and at the same time dependent on the input transition. That is, finite propagation delays of input signals traveling through different logic depths in a circuit and arriving at the inputs of an arbitrary gate, cause the output node of this gate to suffer multiple transitions in a single clock cycle before achieving the correct state. Indeed, these spurious transitions are well known to cause extra power dissipation over that required by a function to

perform the computation. For example in [1], it is stated that an 8-bit ripple-carry adder is expected to consume an extra 30 % of energy due to these spurious transitions. However until recently, α_F has been regarded as deterministic for a given circuit topology and input transition, disregarding the randomness of the glitches. However, measurement results of the static CMOS S-box tend to show that α_F becomes randomly distributed between manufactured circuits at low voltage [19] due to the glitch-induced random variations.

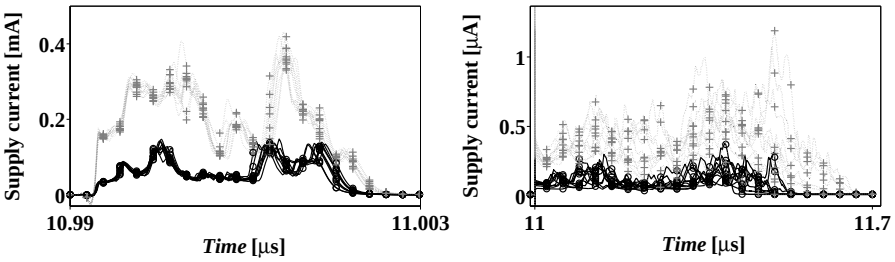
As a demonstration, we perform Monte-Carlo simulations of an internal node voltage of the static CMOS S-box, when considering WID variations. Fig. 3.6 shows the simulated waveforms of an arbitrary internal node for three distinct Monte-Carlo runs at 1 V and 0.4V, with a single input transition. Although the topology, signal statistics, input transition and logic depth are the same, the node activity considerably varies and this variation increases at 0.4 V. This comes from WID delay variability that is magnified at low voltage. More pronounced at 0.4 V, large delay skews between the inputs of a gate are present, which cause the WID variability to be more distinguished, generating random glitches; thus α_F variations. These random glitches affect the dynamic power consumption of the static CMOS S-box as seen from the supply current traces in Fig. 3.7, where 10 distinct Monte-Carlo runs are performed for two random input vectors. It illustrates the WID variability effect on α_F which is clearly magnified at 0.4 V. The glitch-induced WID α_F variations cause the current traces to behave differently with the Monte-Carlo runs and also with the input transition. This further manifests that some input transitions may suffer more from the α_F variations due to the different delay skews associated to these input transitions. Accordingly, the increase in the E_{dyn} variations of the static CMOS S-box at low voltages in Fig. 3.5 is a direct result of this phenomenon.



(a) V_{DD}= 1.0 V. **(b)** V_{DD}= 0.4 V.

Fig. 3.6.: SPICE simulations of 3 particular Monte Carlo runs of an internal node voltage of the static CMOS S-box at (a) V_{DD} = 1V and (b) V_{DD} = 0.4V.

Furthermore, the associated impact of WID variability on dynamic power ($E_{dyn} \times f$) variations becomes comparable to the measured D2D P_{leak} variations as shown in Table 3.3. At 1 V, σ/μ of dynamic power is about 1 % which is small



(a) V_{DD}= 1.2 V. **(b)** V_{DD}= 0.4 V.

Fig. 3.7.: Static CMOS S-box supply current traces of 10 Monte Carlo runs for two random input vectors at (a) $V_{DD} = 1.2$V and (b) $V_{DD} = 0.4$V.

Table 3.3.: Comparison of normalized standard deviation of P_{dyn} and P_{leak} of static CMOS S-box.

σ/μ [%]	P_{dyn}	P_{leak}
@ 1V	1.07	8.23
@ 0.4V	5.56	8.9

compared to the P_{leak} variations (8.23 %), whereas at 0.4 V, σ/μ of dynamic power rises to 5.5 % due to the glitch-induced WID α_F variability and becomes comparable to that of P_{leak} (8.9 %) which is dominated by D2D variations.

It is worth mentioning that E_{dyn} variations reported so far for the static CMOS S-box result from averaging the measured power consumption over a 2560-transition input pattern. However, worst-case input transitions also exist that feature higher E_{dyn} variations. For example, we measured relative E_{dyn} standard deviation σ/μ up to 8% for input patterns with 256 transitions at 0.4 V. As a result, averaging over a large number of input transitions can lead to a false sense of variability tolerance.

So far we have demonstrated that not only C_L variations contribute to the E_{dyn} or P_{dyn} variabilities, but also glitch-induced WID α_F variations have a substantial influence, which has not been taken into consideration until recently [19]. This new phenomenon is further magnified at low voltages. Due to the uncorrelated nature of the glitch-induced WID α_F variations, they are averaged out over the input transitions. As a result, it is important that the circuit designer takes the worst-case condition into consideration.

The observations made in this section lead to the following conclusions:

1. **E_{dyn} (P_{dyn}) variabilities of complex combinatorial circuits suffer from not only C_L variations, but also glitch-induced α_F variations.**
2. **Glitch-induced WID α_F variability is the dominant contributor to the E_{dyn} (P_{dyn}) variabilities since the WID variations of C_L are averaged out.**
3. **P_{dyn} variability at low voltages approaches the rising P_{leak} variability which is dominated by D2D variations.**

3.5 COMPARISON BETWEEN DDSLL AND STATIC CMOS STYLES

In this section we investigate the advantages and disadvantages brought by the use of the DDSLL style compared to static CMOS under the effect of variability. The reason behind our interest in how the DDSLL style is affected by variability is its inherently fixed α_F. It has been reported in the previous section how the glitch-induced WID α_F variability affects the dynamic power of combinatorial static CMOS circuits, specially at low voltages. Indeed, finite propagation delays of input signals of an arbitrary gate, traveling through different logic depths in a circuit, induce glitches at the output node before settling. This also leads to uncorrelated delay variability. As a result, combinatorial static CMOS circuits suffer glitch-induced delay variability as well as α_F variability. Consequently, DDSLL is expected to overcome the glitch-induced delay and dynamic power variabilities as a result of its glitch-free nature thanks to the self-timing scheme it utilizes.

Also, variability is well known to greatly impact leakage power [5]. However, we do not address it in our variability study because of two reasons. First, DDSLL is a dynamic logic; thus, it consumes negligible leakage power (roughly 4.7 % according to measurement results Chapter 2). Second, even at 100 kHz, which is the minimum clock frequency for throughput / latency constraint in passive RFID tags [21]), the measured power consumption of the static CMOS S-box in 65 nm LP CMOS process at 1.2 V is dominated by dynamic power (84 %). Therefore, we will focus only focus on the dynamic power variability.

Another important aspect is the effect of variability on the functionality, specially as the supply voltage decreases. Certainly, the functionality of combinatorial static CMOS circuits is known to withstand magnified variability at low supply voltages, whereas circuits such as SRAMs fail to function [23]. Also dynamic circuits are claimed to be vulnerable to process variations [23]. However, the static CMOS style suffers from possible parametric failure when operated in the sub-threshold region. For example, random WID variations might lead to

hold time violations due to high timing uncertainties [13]. In this analysis we aim at studying the impact of process variations under low supply voltages on the functionality of the DDSLL S-box compared to the static CMOS S-box.14

3.5.1 Dynamic power

In Section 3.4, the impact of process variations on the dynamic energy of ring oscillators and a static CMOS S-box, was studied. It was shown that the ring oscillator is a suitable test case to study the impact of variability on the load capacitance. On the other hand, the static CMOS S-box provided a good insight on the role of the glitch-induced α_F variations in the overall variability of its dynamic energy, specially in low-voltages. As expected, the C_L variations are dominated by D2D variability which is correlated and constant; thus, independent on the logic depth and the supply voltage. However, α_F is not deterministic as previously believed [5, 10]. It was proved in Section 3.4.2 that α_F is random in nature due to the presence of glitch-induced WID variations, associated to delay skew and logic depth.

The importance of the impact of WID variability on α_F is not limited to low voltage operation only, further advanced technology nodes are concerned as well. In this study, the investigated circuits are implemented on 65 nm LP CMOS technology. Indeed, variability cannot be neglected anymore in sub-100 nm technology nodes [49, 50], yet these studies focus on low-voltage and even ultra-low-voltage operations. Additionally, they are mainly concerned with the impact of variability on delay and minimum energy. However, E_{dyn} is expected to gain more sensitivity to the glitch-induced α_F variability in sub-100 nm technology even while the operation is at nominal supply voltage.

Consequently, in this section we investigate the advantage brought by the use of the DDSLL style in eliminating the glitch-induced α_F WID variability of complex combinatorial circuits as an extension to the work done in [20]. First, we define P_{dyn} of DDSLL as follows:

$$P_{dyn} = V_{DD}/2 \times \alpha_F \times f \times \sum_{j=1}^{n} (\Delta V_j . C_{L,j}) \tag{3.8}$$

where α_F is fixed thanks to the self-timing operation of DDSLL and ΔV_j is the voltage swing of the j^{th} gate. Certainly, the main objective of DDSLL as a DDL style is to maintain a constant ΔV that does not change with either the input sequence or process variations. This constraint is normal because if ΔV varies with the data being processed, then the power consumption will be data dependent to some extent (this issue will be further discussed in Chapter 4). Also, if ΔV changes with process variations, then it is expected to affect the data dependency of the power consumption as well. Consequently, the ideal case is to have a fixed ΔV. However, since DDSLL is based upon the principle of charging and discharging floating nodes, its ΔV is considered a random variable.

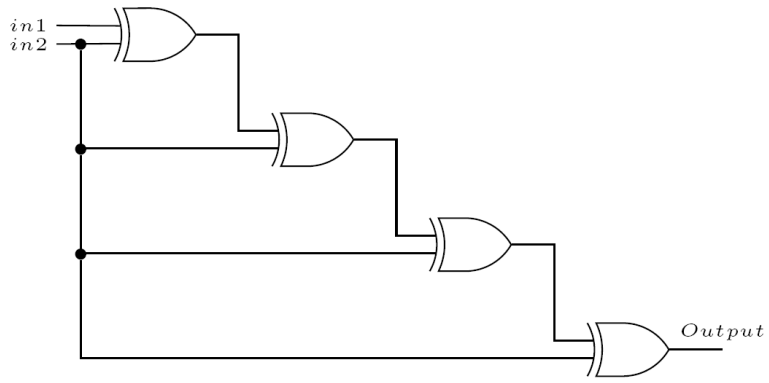


Fig. 3.8.: Block diagram of the XOR chain implemented using either static CMOS or DDSLL styles.

It is mainly affected by V_t variations that directly impact the *ON* current and resistance of the transistors.

This study is performed at the nominal supply voltage, knowing that the minimum V_{DD} of DDSLL is limited by functionality to only 1 V as will be explained later in Section 3.5.3. We start by analyzing the effect of WID variability on the dynamic power of both DDSLL and static CMOS styles followed by a similar analysis of D2D variability. The dynamic power of a static CMOS gate is calculated by subtracting the P_{leak} from the P_{tot}, where P_{leak} is extracted under static conditions. On the other hand, DDSLL being a DDL style, its dynamic power is not easily determined in the same way. As a result, in simulations, P_{dyn} is calculated by averaging the instantaneous power over the transition periods of precharge and evaluation. Although this method is not very accurate, it is used in order to be consistent while comparing DDSLL to static CMOS. Besides, the static power of DDSLL calculated using the same method is negligible compared to the total power (4.7 % according to measurement results) as previously explained in Chapter 2. Therefore, P_{tot} of DDSLL is dominated by its dynamic power which justifies the use of such method.

With-in-die (WID): In order to demonstrate the impact of WID variability on the dynamic power of both DDSLL and static CMOS gates, we use the classical chain of inverters as a reference. Fig. 3.8 shows the test-bench used in simulating either the static CMOS or the DDSLL XOR gate chain. To manifest the impact of increasing delay skew between the inputs of an arbitrary gate, the two inputs of any XOR gate in the chain are driven by the output of the previous gate and the same input as the first XOR gate in the chain. This way, the delay skew increases with the logic depth of the XOR chain; thus, giving rise to magnified glitches in the case of static CMOS.

In the case of the inverter chain, σ/μ of the dynamic power decreases with the increasing $\sqrt{n}$, where n is the logic depth as seen in Fig. 3.9 and previously

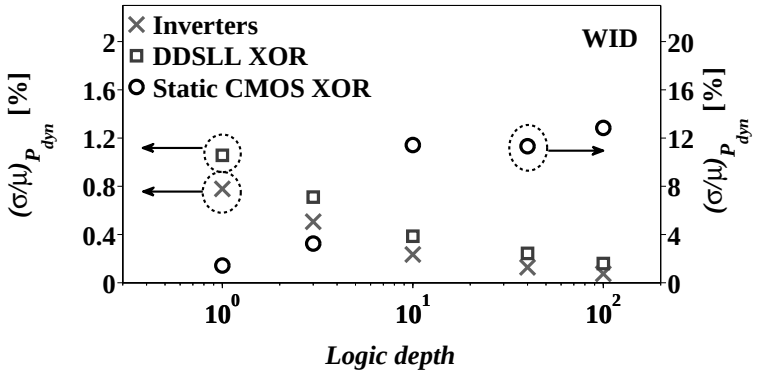


Fig. 3.9.: SPICE simulations of dynamic power WID variability of inverter chain, static CMOS XOR chain and DDSLL XOR chain with different logic depths 25°C, 1.2 V supply (100 Monte-Carlo runs with local process variations).

expressed in Equation 3.7. This relation was explained in Section 3.4.1 by the fact that the only source of variability in the case of the inverter chain is the WID C_L since α_F is fixed. Indeed, the WID C_L variations are uncorrelated; thus, they are averaged out over the logic depth. However, the chain of static CMOS XOR gates manifests the claim of dominating glitch-induced WID α_F variability. Fig. 3.9 shows that P_{dyn} variations of the static CMOS XOR chain (the P_{dyn} distribution is considered Gaussian) increases with the logic depth. That is, the WID variability of the $(\alpha_F.C_L)$ term in Equation 3.2 is no longer decreasing with the increase of logic depth, but it actually increases. Since, WID C_L variations average out, therefore, α_F is responsible for the increasing WID variations of P_{dyn} in static CMOS due to the magnified glitches resulting from the increasing delay skew with the logic depth. Contrarily, the P_{dyn} WID variability of the DDSLL XOR chain (the P_{dyn} distribution is considered Gaussian) decreases as the logic depth increases in accordance with the case of the inverter chain, i.e, with the increasing $\sqrt{n}$. The averaging out of P_{dyn} WID variability in the case of the DDSLL is a direct result of its fixed α_F, which indicates that $(\Delta V\ .\ C_L)$ is the only factor that count. This also implies that WID ΔV variations are uncorrelated, thus they are averaged out as well. In fact, ΔV variations are directly related to V_t deviations that are uncorrelated in the case of WID process variations. Certainly, being a DDL style, DDSLL is designed to avoid glitches, which are a major source of information leakage from the security perspective [51, 52]. Thanks to its self-timing operation, DDSLL is able to achieve this goal, and hence, it has a deterministic α_F.

Die-to-die (D2D): Although the D2D (global) variability is spatially-correlated, i.e, it affects all transistors on one die in the same way, it features dissimilar impact on the three test cases. Fig. 3.10 shows the effect of D2D variability on the dynamic power of the inverter chain, static CMOS and DDSLL

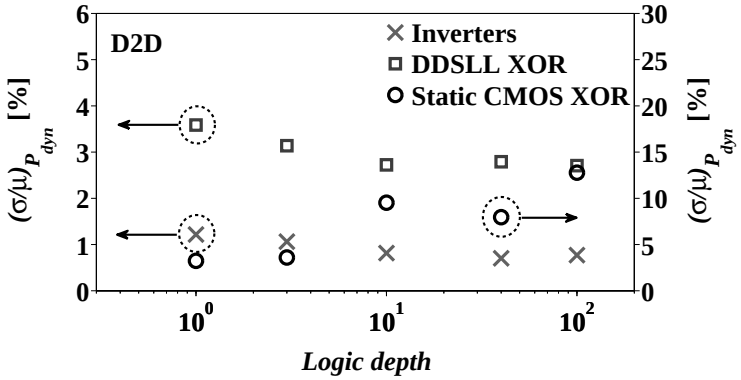


Fig. 3.10.: SPICE simulations of dynamic power D2D variability of inverter chain, static CMOS XOR chain and DDSLL XOR chain with different logic depths at 25°C, 1.2 V supply (100 Monte-Carlo runs with global variations).

XOR chains (the P_{dyn} distributions in all cases are considered Gaussian). The first conclusion is that in the cases where α_F is deterministic, the D2D C_L or D2D (ΔV . C_L) variability is independent on the logic depth as seen by the P_{dyn} variations of the inverter and the DDSLL XOR chains, respectively. Indeed, D2D C_L variations are correlated; hence σ/μ of P_{dyn} is fixed for all logic depths in the case of the inverter chain. As for the DDSLL XOR chain, D2D ΔV variations contribute as well. Since fluctuations in ΔV is a direct result of V_t deviations, then they are also correlated while considering D2D process variations. Consequently, σ/μ of P_{dyn} of the DDSLL XOR chain is fixed for all logic depths. For the inverter chain, σ/μ of dynamic power D2D variability is about 1 %, whereas it is around 3 % in the case of the DDSLL XOR chain. The reason behind the higher P_{dyn} D2D variability of the DDSLL XOR chain is the presence of ΔV D2D variations that add to the C_L variations since they are statistically independent. As for the static CMOS XOR chain, again σ/μ of the dynamic power is dependent on the D2D variability of α_F such that it increases with the logic depth as shown in Fig. 3.10. Therefore, the second conclusion is that glitch-induced α_F variations due to D2D variability is responsible for the increase in σ/μ of the dynamic power of the static CMOS XOR chain with logic depth.

It is worth reminding that in Monte-Carlo SPICE simulations, the D2D variability actually refers to global variations that includes W2W and L2L variations as well. Consequently, the numerical values demonstrated in Fig. 3.10 reflect a wide spatial spread of variations, whereas the measurement results, as demonstrated in Section 3.4, feature only D2D variability.

Thanks to the deterministic nature of α_F in the DDSLL style:

- **WID P_{dyn} variations are averaged out with the increase of the logic depth and circuit size, unlike these of the static CMOS that actually increase with logic depth.**
- **D2D P_{dyn} variations are independent on the logic depths and circuit size, also unlike static CMOS whose D2D P_{dyn} variations increase with logic depth.**

Effect of variability on the power consumption of the AES S-box: First, we focus on the impact of WID variability on the dynamic power consumption of the DDSLL and static CMOS S-boxes. It is important to analyze how the glitch-induced WID α_F variations affect the dynamic power of both S-boxes. Fig. 3.11 compares the normalized standard deviation of P_{dyn} of both static CMOS and DDSLL S-boxes, where P_{dyn} is averaged over different numbers of input transitions at 100 kHz. Clearly in the case of the static CMOS S-box, (σ/μ) of P_{dyn} varies between 1.6 % and 25 % for a 2-transition input pattern as a direct result of variations in α_F. However, the increase in the number of transitions in the input pattern leads to a reduced WID variability around 1.3 % which further manifests the effect of averaging over the number of input transitions. On the other hand, the DDSLL S-box shows almost a constant (σ/μ) of P_{dyn} around 1 % thanks to its deterministic α_F.

As a result, for strongly-power constrained applications such as RFID and contactless smart cards, the impact of variability on the dynamic power plays

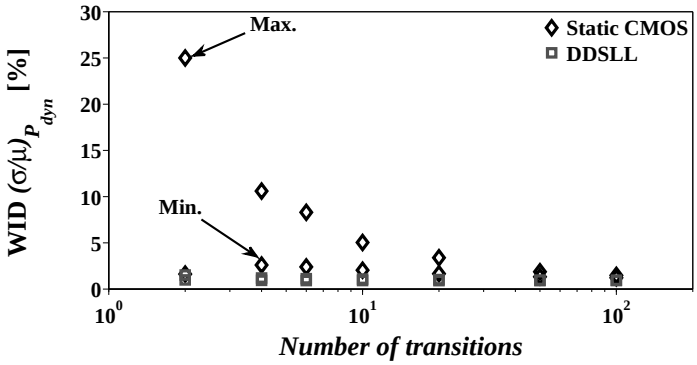


Fig. 3.11.: SPICE simulations of dynamic power WID variability of static CMOS and DDSLL S-boxes versus the number of transitions, showing the minimum and maximum P_{dyn} variability of static CMOS (for averaged 100 Monte-Carlo runs with local process variations).

an important role. In the case of high yield, the typical $P_{dyn} + 3\sigma$ should be less than the allocated power budget (dynamic). For a worst case scenario, a 2-transition input pattern is considered. For static CMOS, the typical P_{dyn} at 100 kHz is 108 nW and the worst case σ is 27 nW (derived from Fig. 3.11). Therefore, the typical $P_{dyn} + 3\sigma$ is evaluated as 189 nW. Now for DDSLL, the typical P_{dyn} at 100 kHz is assumed to be 96 nW (P_{tot}= 96 nW, however for simplicity and since the dynamic power dominates, we approximate P_{dyn} to be equal to P_{tot}) and the worst case σ is 0.96 nW (also acquired from Fig. 3.11). Therefore, the typical $P_{dyn} + 3\sigma$ would be 98.9 nW. Clearly the power advantage of DDSLL is magnified when taking the WID process variability into account (i.e. as its worst-case power consumption remains within the allocated budget). For static CMOS, process variations can cause the typical $P_{dyn} + 3\sigma$ to be almost 1.75× the typical P_{dyn}, whereas for DDSLL, the 3σ metric leads to a 3 % increase of P_{dyn} only.

It is worth reminding that the impact of the glitch-induced WID α_F variability on E_{dyn} of the static CMOS S-box is demonstrated at low voltages for a pseudo-random 2560-transition input pattern in Section 3.4.2. If a worst-case input pattern is considered, the typical $P_{dyn} + 3\sigma$ is expected to be further magnified, jeopardizing the power closure.

- **P_{dyn} variations of static CMOS combinatorial circuits decrease as the number of averaged input transitions increase. Therefore, worst-case input patterns should be considered (2-input transitions).**
- **For power closure, WID process variations cause the typical $P_{dyn} + 3\sigma$ of the static CMOS S-box to be 1.75× the typical P_{dyn}, whereas for DDSLL, the 3σ metric leads to a 3 % increase only.**

As for the measured results of both S-boxes, measurements are carried out on 20 dies at ambient temperature with 1.2 V supply for 10 input transitions in order to assess the variability of their power consumption. Fig. 3.12 (a) shows the histogram of the total power consumption of both S-boxes at 100 kHz. The total variability comprising both WID and D2D (σ/μ) of DDSLL S-box is 1.04 % and that of static CMOS is 2.78 % (the P_{dyn} distributions can be fairly assumed Gaussian for both S-boxes). It is worth mentioning that the D2D variability is far less important in this case as measurements are done on the same wafer. Therefore, these results complies with simulations as the DDSLL S-box shows dynamic power WID variability of about 1 % for all input patterns, while the static CMOS S-box shows dynamic power WID variability ranging from 2 % to 5 % for a set of 10 input transitions.

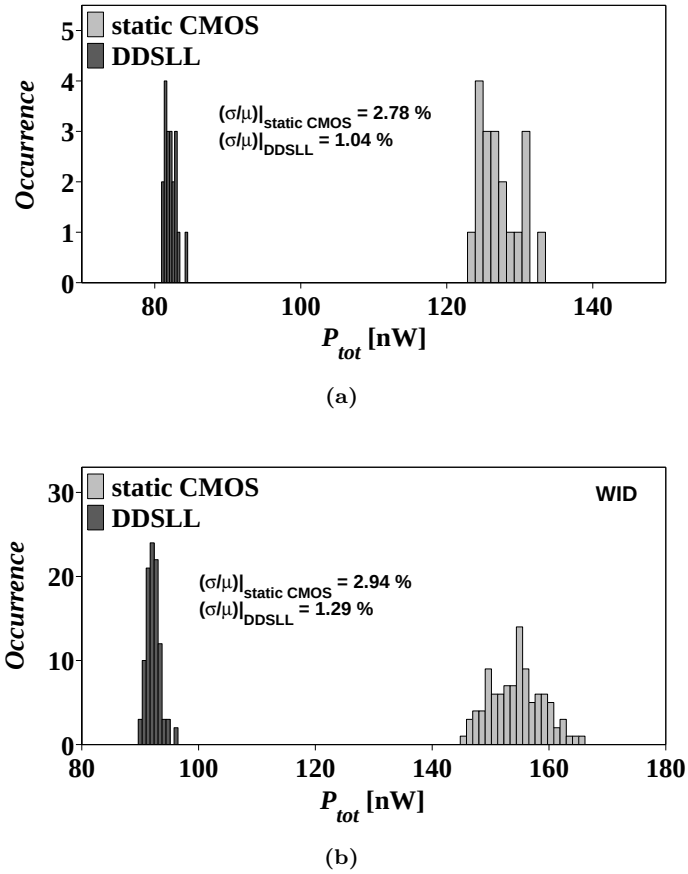


Fig. 3.12.: Histograms of the (a) measured and (b) simulated total power consumption of both static CMOS and DDSLL S-boxes at 100 kHz at 25°C and 1.2 V supply (average over 10 input transitions). Measurements are carried out on 20 dies, while SPICE simulations are performed while enabling local process variations (100 Monte-Carlo runs).

As a result, the worst case maximum power, considering 3σ for high yield, is 147.6 nW for static CMOS and 87.3 nW for DDSLL resulting in a worst case power ratio of 1.69× instead of the 1.54× ratio in the case of typical power consumption. The difference between the worst case and typical ratios is not very high because in the measurements we averaged over 10 input transitions, whereas if 2-transition inputs were considered as a worst-case scenario, the difference between the worst case and typical ratios would have been more significant as previously demonstrated with Monte-Carlo simulations.

Fig. 3.12 (b) reinforces the dominancy of WID variability with respect to D2D. The histograms of the total power consumption of both S-boxes simulated with the same conditions of the measurements while enabling local process variations, are plotted, where the distributions are again considered Gaussian. The normalized standard deviation of the DDSLL S-box is 1.29 % and that of static CMOS is 2.94 %, which is almost the same as the obtained measurement results.

3.5.2 Delay

Delay variability is becoming a major concern for timing-constrained applications, especially in sub-100 nm technologies. Therefore, it is important to investigate the impact of variability on the delay of the DDSLL style and compare it to that of static CMOS. How the process variations (local and global) affect the delay of static CMOS circuits and mitigation techniques have been an open area of research for decades [9, 13, 53]. Indeed, in sub-100 nm technologies, uncorrelated WID (local) variations are known to cause hold time violations in digital static CMOS circuits due to high timing uncertainties which are further magnified in the sub-threshold operation [13]. Also, process variations may cause the failure of the design to meet the target frequency and consequently, the minimum energy [54], leading to parametric yield loss [23, 54].

The delay in a logic path is generally expressed for static CMOS as in [55]:

$$\begin{aligned} t_d &\sim \sum_{j=1}^{n} t_{d,j} \\ &\sim \frac{1}{2} \times V_{DD} \sum_{j=1}^{n} \frac{C_{eff\ L,j}}{I_{on,j}} \end{aligned} \tag{3.9}$$

and for DDSLL as follows:

$$t_d \sim \frac{1}{2} \times \sum_{j=1}^{n} \frac{C_{L,j}.\Delta V_j}{I_{on,j}} \tag{3.10}$$

where $C_{eff\ L,j}$ represents the effective load capacitance, $I_{on,j}$ is the on current and ΔV (in DDSLL only) denotes the voltage swing of the j^{th} gate of the path. The effective load capacitance comprises the activity factor α_F as well as the physical load capacitance C_L. In this section we are mainly concerned with the effect of process variations on the delay of static CMOS and DDSLL S-boxes. First, we analyze the WID process variations.

With-in-die (WID): Fig. 3.13 shows the histograms of the delay of both S-boxes operating at 25°C and 1.2 V supply while enabling the WID variability during SPICE simulations (the WID delay distributions of both S-boxes are roughly assumed to be Gaussian). The WID delay normalized standard deviation of the static CMOS S-box is 6.2 %, whereas that of the DDSLL S-box is 2.05 %. Table 3.4 details the μ, σ and σ/μ of the delay for both S-boxes. It can be seen

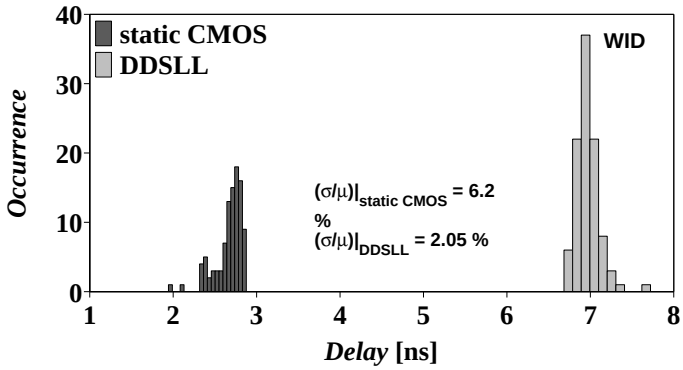


Fig. 3.13.: Histograms of the delay of static CMOS and DDSLL S-boxes at 25°C and 1.2 V supply using SPICE simulations (100 Monte-Carlo runs with local process variations).

that the delay standard deviations of both S-boxes are comparable, although the mean delay of the DDSLL S-box is 2.62x higher than static CMOS. This is an interesting result that favors the DDSLL style. In [23], the authors explained that one of the pitfalls of static CMOS is that when operating at low V_{DD}, the σ of the overall path delay distribution increases as well as the mean delay. As a result, more paths are affected by these variations leading to a degradation of the timing yield at low V_{DD}. Now, DDSLL already has a higher delay penalty; thus limiting the maximum speed, but it features a small delay WID standard deviation which allows for meeting the desired speed specification with high process variations tolerance. In addition, we expect the delay of the DDSLL style to maintain this high tolerance to process variations if it is capable of operating at lower supply voltages.

The reason behind the low impact of WID process variations on the delay of DDSLL is twofold. The first is related to the common averaging of uncorrelated WID C_L, I_{on} [48] and ΔV variabilities. For this reason, we consider the total number of transistors present in the critical path of both S-boxes which consists of n-stages and m-stack per stage. For the static CMOS S-box, the critical path

Table 3.4.: WID mean and standard deviations of the delay of static CMOS and DDSLL S-boxes at 25°C and 1.2 V supply (100 Monte-Carlo runs with local process variations).

Parameter	static CMOS	DDSLL
μ [ns]	2.67	6.96
σ [ns]	0.17	0.14
σ/μ [%]	6.20	2.05

contains 22 stages and 2 stacked NMOS transistors per stage, leading to a total of 44 NMOS transistors. As for the DDSLL S-box, its critical path consists of 13 stages and each stage is formed of 6 stacked NMOS transistors (forming the DPDN and the dynamic current source) and 1 NMOS transistor in the self-timing buffer, resulting in a total of 102 NMOS transistors. Therefore, the reduction of the WID delay variability of the DDSLL S-box with respect to static CMOS is approximately related to the increased number of transistors in the critical path. However, a second factor plays an important role in this relation; namely, α_F. Since DDSLL has a deterministic α_F, therefore it gives rise to the dominancy of the WID C_L, I_{on} and ΔV variabilities that average out with the logic depth. On the other hand, the delay of the static CMOS S-box suffers from glitch-induced WID α_F variations that causes the WID variability of $C_{eff\ L}$ to increase with the logic depth as in the case of E_{dyn} fluctuations previously explained in Section 3.4.2. Certainly, there are significant amounts of delay skews between the input signals of a static CMOS gate, such that WID process variations affect these delay skews in a random manner causing the output signal of the gate to fluctuate before settling on the correct state. Therefore, the delay of static CMOS gates is affected by the glitch-induced WID α_F variations which contribute to the increase of the WID delay variability of the static CMOS S-box compared to DDSLL. Of course, the impact will be further magnified for static CMOS gates at low voltages.

This is further illustrated in Table 3.5 taking the XOR chain shown in Fig. 3.8 as an example. The WID delay variability of DDSLL clearly averages out as the number of XORs in the chain increases from 1 to 10. However, the WID σ/μ of the delay of static CMOS increases from 7 % to 43 % moving from 1-XOR to 9-XORs instead of the common belief, which expects the reduction of the WID delay variability with the logic depth due to the averaging of uncorrelated WID C_L and I_{on} variabilities. Therefore, only α_F can be responsible for this phenomenon. Also, using a chain of 20 static CMOS XORs, we see a reduction in the WID delay variability compared to the 9-XOR static CMOS chain. This can be explained by the fact that WID α_F variations are still random in nature and according to the input sequence and the logic depth they can still feature some averaging. However, the WID delay variability of the 20 static CMOS XOR chain is still higher than that of the single XOR.

Table 3.5.: Mean and standard deviations of the delay of 1 and 10 series XOR DDSLL and static CMOS gates due to WID process variations at 25°C and 1.2 V supply (100 Monte-Carlo runs with local process variations).

		DDSLL		static CMOS		
	Parameter	1-XOR	10-XORs	1-XOR	9-XORs	20-XORs
WID	μ [ns]	0.102	1.011	0.057	0.274	0.826
	σ [ps]	3.960	12.28	0.004	0.119	0.195
	σ/μ [%]	3.870	1.210	7.160	43.60	23.68

- **WID delay variability of the DDSLL style decreases with the increase of the logic depth thanks to its deterministic α_F.**
- **WID delay variability of the static CMOS style depends and increases with the logic depth due to the glitch-induced WID α_F variations that impact the $C_{eff\,L}$ term in Equation 3.9.**

Die-to-die (D2D): Now, we analyze the impact of D2D process variations on the delay of both S-boxes. The same simulation conditions are used to produce the histograms of the delay of the S-boxes, while enabling global process variations as shown in Fig. 3.14 (the D2D delay distributions of both S-boxes are considered Gaussian). Unfortunately, the D2D process variations cause the delay of the DDSLL S-box to spread more than static CMOS such that the normalized standard deviations of both S-boxes are comparable (since the mean delay of the DDSLL S-box is already higher than static CMOS). Table 3.6 presents the extracted D2D μ, σ and σ/μ for both S-boxes. It can be seen that the value of σ for the delay of DDSLL S-box increases with the mean delay when compared to the static CMOS, thereby keeping σ/μ nearly the same.

Table 3.6.: D2D mean and standard deviations of the delay of static CMOS and DDSLL S-boxes at 25°C and 1.2 V supply (100 Monte-Carlo runs with global process variations).

Parameter	static CMOS	DDSLL
μ [ns]	2.67	6.96
σ [ns]	0.31	0.94
σ/μ [%]	11.6	13.5

The larger spread of the delay variability of DDSLL caused by D2D process variations is quite expected. The only contributors to the D2D delay variability

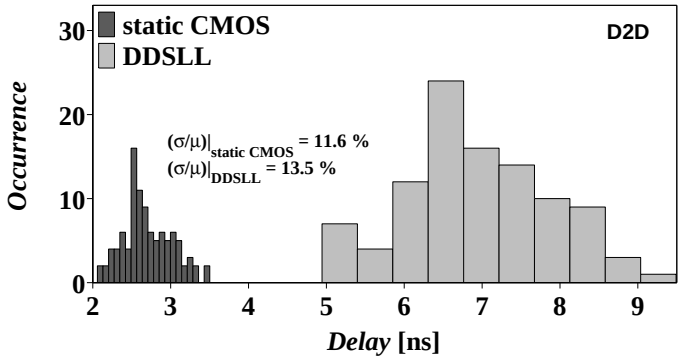


Fig. 3.14.: Histograms of the delay of static CMOS and DDSLL S-boxes at 25°C and 1.2 V supply using SPICE simulations (100 Monte-Carlo runs with global process variations).

are D2D C_L, I_{on} and ΔV variations because α_F of the DDSLL style is deterministic. It was explained in [48] that the fluctuations of the delay for tapered buffers due to D2D variability is independent on the number of stages. That is, the increase in the mean delay with the number of stages has to be matched by an increase in the standard deviation to maintain a constant delay variation. The case of the tapered buffers is analogous to the DDSLL style because both feature a deterministic α_F. Therefore, D2D C_L, I_{on} and ΔV variations are correlated and cause the delay variability to be independent on the logic depth.

This is further explained by using the same XOR chains in Table 3.5. Table 3.7 demonstrates how, due to the impact of D2D process variations, σ increases by the same factor as μ (x10) moving from a single DDSLL XOR gate to a chain of 10-XORs, so that σ/μ is kept constant. Therefore, the D2D delay variability of the DDSLL S-box is actually the same as that of a DDSLL XOR gate and that of a chain of 10 series DDSLL XOR gates. Now, the reason behind the higher D2D delay variability of DDSLL gates in general is the presence of D2D ΔV variations that add to the normal C_L and I_{on} variations since they are statistically independent.

Table 3.7 also shows the D2D delay variability of static CMOS XOR chains. Moving from a single XOR gate to a chain of 9 static CMOS XOR gates causes the D2D delay variability of the chain to rise from 14 % to 24 % as a direct result to the D2D α_F variability that is not correlated, but rather random. The 20 static CMOS XOR chain features a reduced D2D delay variability compared to its 9-XOR chain counterpart. Again, this is explained by the random nature of D2D α_F variability that can either increase or experience some averaging with logic depth.

Table 3.7.: Mean and standard deviations of the delay of 1 and 10 series XOR DDSLL and static CMOS gates due to D2D process variations at 25°C and 1.2 V supply (100 Monte-Carlo runs with global process variations).

		DDSLL		static CMOS		
	Parameter	1-XOR	10-XORs	1-XOR	9-XORs	20-XORs
D2D	μ [ns]	0.104	1.030	0.055	0.288	0.871
	σ [ns]	0.014	0.135	0.007	0.069	0.142
	σ/μ [%]	13.10	13.01	13.94	23.89	16.36

- **D2D delay variability of the DDSLL style is independent on the logic depth thanks to its deterministic α_F. However, it is relatively high (13.5 %).**
- **D2D delay variability of the static CMOS style depends and increases with the logic depth due to the glitch-induced α_F variations that impact the $C_{eff\ L}$ term in Equation 3.9.**

To conclude, the delay of the DDSLL style features high tolerance to WID process variations as manifested by the case of the 13 stage DDSLL S-box; whereas, the delay of the static CMOS suffers from glitch-induced WID α_F variations. As a result, the DDSLL style reduces the probability of hold time violations by limiting the timing uncertainties. However, the delay of the DDSLL style suffers from high D2D process variations. Parametric yield loss, such as, the failure to meet the target frequency and the energy increase above the target minimum are among the implications of the DDSLL sensitivity to D2D delay variations. Nevertheless, several design techniques are proposed to mitigate such sensitivity. For example, adaptive body biasing (ABB) [16, 56] compensation technique that adjusts the transistor threshold voltage in order to reduce process-related timing failures. This technique has been heavily investigated in the past decade for static CMOS circuits, such as microprocessors [16, 54, 56–58]. However to the author's knowledge, the ABB technique has not been applied to DDL styles before. This could be an interesting topic for future research.

3.5.3 Functionality

In the previous sections, the effect of process variations on the delay and the power consumption of DDSLL and static CMOS styles has been addressed, mainly at nominal supply voltage. In this section, we are interested in the impact of WID variability on the functionality of both logic styles at lower V_{DD}. Static CMOS combinatorial circuits (logic) are known to function correctly at

low voltages despite the associated large process variations [23]. However, it was also explained in [23] that dynamic circuits that operate on the principle of precharge and evaluation are susceptible to functional failure because of WID variability. Accordingly, we performed Monte-Carlo SPICE simulations, while enabling local process variation, on the static CMOS and DDSLL S-boxes as demonstrated in Fig. 3.15. Simulations are done at 25°C across a supply voltage that ranges from 0.3 V to 1.2 V. To check the functionality, the final outputs of the S-boxes are tested for each V_{DD} and compared with the expected outputs. The number of Monte-Carlo runs that fail the functionality test are counted and normalized to the total number of runs. Clearly, the static CMOS S-box is fully functional across the whole V_{DD} range. However, the DDSLL S-box has a 25 % failure rate at 1 V and it further increases to 100 % starting 0.8 V.

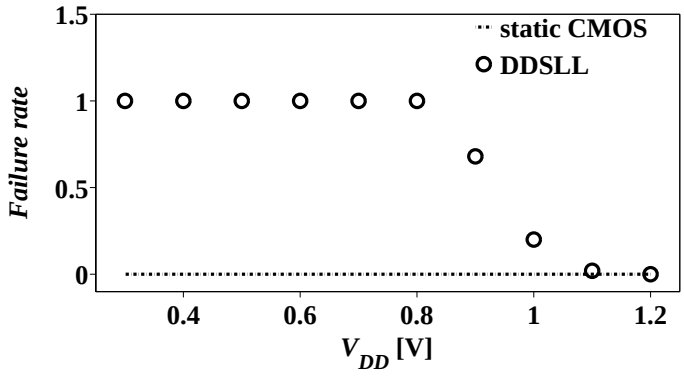
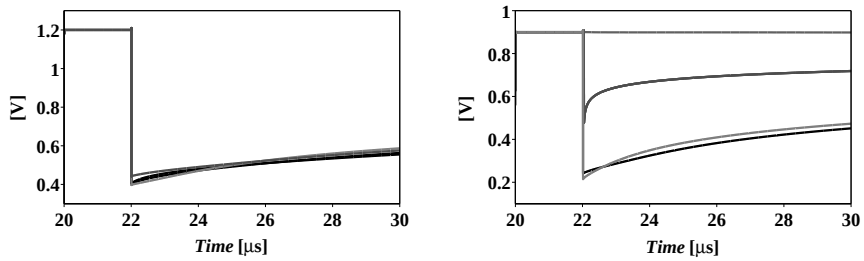


Fig. 3.15.: Normalized failure rates of static CMOS and DDSLL S-boxes with supply voltage at 25°C using SPICE simulations (50 Monte-Carlo runs with local process variations).

To understand why the DDSLL S-box fails to function at voltages below 1 V, we first need to observe one of the output signals that fails at 0.9 V as an example. Fig. 3.16 shows the waveforms of the DDSLL S-box's output signal k_1 at 1.2 V with nominal conditions and at 0.9 V with nominal and 1 particular Monte-Carlo run that fails. At 1.2 V, the average voltage swing is 650 mV and the variation caused by WID process variability is about 70 mV. However at 0.9 V, the voltage swing of the nominal run is about 480 mV and due to the WID process variations, this voltage swing may diminish such that the DDSLL S-box becomes nonfunctional. That is, due to the lower voltage swing at the inputs of a DDSLL gate, the DPDN of this gate becomes more sensitive to WID variability. Consequently, if V_t of an OFF transistor in the wrong path is lowered and that of another ON transistor in the correct path is raised due to WID variability, then a small voltage swing at the inputs of the gate can cause it to evaluate a wrong output. Another important reason that contributes to the failure of DDSLL gates to operate below 1 V is the fact that the DPDN



(a) @ 1.2 V **(b)** @ 0.9 V

Fig. 3.16.: SPICE simulations of the DDSLL S-box's output signal k_1 with nominal and 3 Monte-Carlo runs (a) at 1.2 V and (b) at 0.9 V.

switches the current between its branches based upon fully ON and partially ON transistors. At 0.9 V, all transistors are partially ON to a certain extent. This further reduces the tolerance of the DDSLL style to WID process variations.

The DDSLL S-box does not function correctly below 1 V due to the magnified impact of WID process variations at the lower operating voltage swings.

To conclude, the functionality of the DDSLL style proved to be sensitive to WID process variations at low supply voltage in compliance with the claim in [23] that dynamic circuits are vulnerable to variability because the information is saved as charge on the output node capacitor. In the case of DDSLL, the functionality is limited to near nominal supply voltages also due to low-swing operation. In order to overcome this deficiency, the voltage swing has to be maintained at lower V_{DD}, where ΔV_{max} is limited to V_{DD}, but then the low-swing advantage is lost. In addition, this technique may still fail below a certain V_{DD} due to the rising sensitivity of transistors to WID variability. Another approach is to increase the tolerance of the DPDN by increasing the sizes of the transistors and tweaking their V_t by applying the ABB technique [16, 56]. A different approach that could extend the operation of dynamic logic styles to near or even sub threshold is to maximize the I_{on}/I_{off} ratio of the transistors. That is, instead of having the NMOS transistors of the DPDN operate in the region where the slope of the $I_D - V_G$ curve is small $((V_{DD}) - (V_{DD} - \Delta V)$, thus low I_{on}/I_{off} ratio), we can shift the operation to the $((0) - (\Delta V))$ region which features a high I_{on}/I_{off} ratio. This can be achieved by having the DDL style discharge to GND during the precharge and charge to a predetermined voltage level during the evaluation phase. By maximizing the I_{on}/I_{off} ratio, the DDL style can withstand higher levels of WID process variations. However, the main draw back of this technique is its slower operation and its feasibility is still not

guaranteed. Anyway, all the above mentioned mitigation techniques represent an interesting research topic for the future.

Proposed solution to extend the functionality of the DDSLL style to lower supply voltages:
- **Maintain ΔV with V_{DD}.**
- **Increase transistor sizes and apply local ABB to the transistors of the DPDN.**
- **Increase I_{on}/I_{off} ratio.**

3.6 SUMMARY

As technology scales down, the device parameters are not deterministic anymore [6]. Indeed, with the technology entering the sub-100 nm, fluctuations in device parameters can no longer be neglected as they degrade the parametric yield by deeply jeopardizing the timing and power constraints of ICs [23]. Also, they lead to functionality failures in circuits such as SRAMs and registers [22, 23]. As a result, process variations, their impact on different design aspects and incorporating them in the simulation environment through statistical analysis tools have been a wide area of research for the past decades. Designers are mainly concerned with the effect of variability on the delay, the leakage power of CMOS circuits and on the functionality of sensitive circuits, such as, SRAMs and registers [9, 13, 22, 23, 53]. The first two design aspects are associated with timing uncertainties and total power fluctuations that lead to parametric yield loss, such as, failure to meet the target frequency, hold time violations, failure to achieve power closure and minimum energy. On the contrary, dynamic energy (power) has been considered till recently, as weakly sensitive to variability [5, 10].

In this chapter, we first address the impact of process variations on the dynamic energy of different types of voltage-scaled 65 nm CMOS circuits. We show that *the dynamic energy of α_F-sensitive circuits is deeply affected by WID process variations at low V_{DD}*. However, the dynamic energy of simple circuits such as ring oscillators are dominated by D2D C_L variability because WID C_L variations tend to average out with the increase of circuit size. Nevertheless for complex circuits, large delay skews between the inputs of a static CMOS gate are present, specially at low voltage, which cause the WID variability to be more distinguished, generating random glitches; thus α_F variations. *These random glitches cause the dynamic power of the static CMOS S-box (representing an example of complex combinatorial circuits) to have σ/μ about 5.6 % at 0.4 V, which then becomes comparable to that of P_{leak}, 8.9 % (dominated by D2D variations). This value is expected to get worse if a worst-case input test pattern is used, since the glitch-induced WID α_F variations tend to average out over the input transitions, and of course as the technology scales further.* As a result, E_{dyn} (P_{dyn}) variability of static CMOS circuits can no longer be neglected

and should strongly be considered for statistical circuit simulations, as it might change design paradigms at next technology nodes.

Also in this chapter, we provide a comparison between the DDSLL style and static CMOS with the goal of highlighting the advantages and disadvantages of the DDSLL style in light of the process variations. We start by addressing how variability influences the dynamic power of the DDSLL style. *Thanks to its self-timing operation, DDSLL is able to maintain a deterministic* α_F. Therefore, *WID* P_{dyn} *variability of DDSLL circuits is averaged out with the increase of logic depth*, since $\Delta V \ . \ C_L$ is the only factor that count and its WID fluctuations are random in nature. In addition, the dynamic power of the DDSLL S-box, under the effect of WID variability, proved to be insensitive to the number of averaged input transitions. However, P_{dyn} *of the static CMOS S-box suffers from a large dependency on the input pattern.* Averaging over a small number of input transitions leads to a wide range of σ/μ for its dynamic power. This is because WID input-dependent α_F variations of the static CMOS S-box affects P_{dyn} in a random manner. As a result, *considering a large number of input transitions (> 10) reduces the WID* P_{dyn} *variability of the static CMOS S-box and can lead to a false sense of variability tolerance.*

The importance of the dynamic power variability is the impact it has on power closure. The comparison between the DDSLL and static CMOS S-boxes is made at 1.2 V, where the WID variability is not as highly pronounced as for lower voltages. Yet, P_{dyn} *variability causes the* 3σ *yield metric in the case of static CMOS S-box to raise the worst-case dynamic power to 1.75× compared to the typical power, whereas the factor is only 1.03 for the DDSLL S-box considering a 10-input transition pattern.* Again, *the factor for the static CMOS S-box is expected to be higher if a worst-case 2-input pattern is used, and of course as the technology scales down.*

With respect to D2D variations, static CMOS demonstrated a significant sensitivity due to its glitch-induced α_F variations. A chain of static CMOS XOR gates feature a rising D2D P_{dyn} σ/μ with the logic depth. On the contrary, a chain of DDSLL XOR gates maintain a constant D2D P_{dyn} σ/μ with the logic depth thanks to its fixed α_F. This further impacts the power closure in the sense that *the D2D* P_{dyn} *variability of static CMOS circuits is uncontrollable and can reach high levels between 10 % and 15 %, depending on the logic depth. However for the DDSLL style, the D2D* P_{dyn} *variability is independent on the logic depth and it is limited to* $\sim$ *3 %.* As a result, *the usage of DDSLL style helps maintain the power consumption within the allocated budget; whereas, static CMOS requires a substantial margin to accommodate worst-case power consumption.*

Secondly, we consider the impact of the process variations on the delay of both S-boxes. *The delay of the DDSLL style features high tolerance to WID process variations as manifested by the case of the 13 stage DDSLL S-box (2.05 %), whereas the delay of the static CMOS suffers from glitch-induced WID* α_F *variations (6.2 %).* As a result, *the DDSLL style reduces the probability of hold time violations by limiting the timing uncertainties.* However, *the delay of the DDSLL style suffers from high D2D process variations*, although it is independent on the

logic depth due its deterministic α_F. The D2D delay variability of the DDSLL S-box is actually the same as that of a single DDSLL XOR gate. Consequently, *parametric yield loss, such as, the failure to meet the target frequency and the energy increase above the target minimum are among the implications of the DDSLL sensitivity to D2D delay variations.* Nevertheless, ***adopting the ABB compensation technique helps reduce process-related timing failures [16, 56].*** This technique has been heavily investigated in the past decade for static CMOS circuits, such as microprocessors [16, 54, 56–58], but has not been applied to DDL styles, to the author's knowledge. This can be an interesting topic for future research.

Finally, the functionality of both S-boxes at low V_{DD} under the impact of WID process variations is addressed. The static CMOS S-box has a 0 % failure rate along the whole V_{DD} range, as expected. However, *the DDSLL S-box starts to fail with 25 % failure rate at 1 V.* The DDSLL S-box completely fails to function below 0.9 V. This is attributed to the inherent dynamic nature of DDSLL (data is stored as a charge on the capacitance of the output nodes) along with the diminishing voltage swing at low supply voltages, besides the DDSLL principle of operation based on low I_{on}/I_{off} ratio of NMOS transistors. Several techniques are proposed to reduce the failure rate and extend the functionality of the DDSLL style to lower supply voltages. The first approach is to ***maintain a constant voltage swing to avoid the increasing sensitivity of the DPDN transistors at low supply voltages.*** However, this method will keep the power consumption at high levels, which contradicts the reason why we scale V_{DD} in the first place. Another approach is to ***increase the tolerance of the DPDN by increasing the sizes of the transistors (which results in an increased C_L, thus higher power consumption) and tweaking their V_t by applying the ABB technique [16, 56].*** A different approach that could extend the operation of dynamic logic styles to near or even sub threshold is to ***increase the I_{on}/I_{off} ratios of the NMOS transistors of the DPDN.*** Anyway, all the above mentioned mitigation techniques represent an interesting research topic for the future.

Bibliography

[1] A. Chandrakasan, S. Sheng, and R. Brodersen, "Low-power CMOS digital design," *IEEE Journal of Solid-State Circuits, JSSC*, vol. 27, no. 4, pp. 473 –484, Apr. 1992.

[2] T. Burd, T. Pering, A. Stratakos, and R. Brodersen, "A dynamic voltage scaled microprocessor system," *IEEE Journal of Solid-State Circuits, JSSC*, vol. 35, no. 11, pp. 1571 –1580, Nov. 2000.

[3] H. Soeleman and K. Roy, "Ultra-low power digital subthreshold logic circuits," in *Proceedings of the International Symposium on Low Power Electronics and Design, ISLPED*, 1999, pp. 94 – 96.

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

[5] D. Blaauw, K. Chopra, A. Srivastava, and L. Scheffer, "Statistical timing analysis: From basic principles to state of the art," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, TCAD*, vol. 27, no. 4, pp. 589 –607, Apr. 2008.

[6] A. Asenov, A. Brown, J. Davies, S. Kaya, and G. Slavcheva, "Simulation of intrinsic parameter fluctuations in decananometer and nanometer-scale MOSFETs," *IEEE Transactions on Electron Devices, TED*, vol. 50, no. 9, pp. 1837 – 1852, Sept. 2003.

[7] L. B. Marshall and E. A. Foreman, "Timing closure in 45-nanometer ASICs using statistical static timing analysis design methodology," in *ACM/IEEE Design Automation Conference, DAC*, 2010.

[8] "PrimeTime: Golden Timing Signoff Solution and Environment data sheet," Synopsys.

[9] S. Nassif, K. Bernstein, D. Frank, A. Gattiker, W. Haensch, B. Ji, E. Nowak, D. Pearson, and N. Rohrer, "High performance CMOS variability in the 65nm regime and beyond," in *IEEE International Electron Devices Meeting, IEDM*, Dec. 2007, pp. 569 –571.

[10] A. Srivastava, D. Sylvester, and D. Blaauw, *Statistical analysis and optimization for VLSI: timing and power*, ser. Series on integrated circuits and systems. Springer, 2005. [Online]. Available: http://books.google.be/books?id=WqsQTyOu5jwC

[11] M. Mani, A. Devgan, and M. Orshansky, "An efficient algorithm for statistical minimization of total power under timing yield constraints," in *Proceedings of the 42nd Design Automation Conference, DAC*, Jun. 2005, pp. 309 – 314.

[12] B. Zhai, S. Hanson, D. Blaauw, and D. Sylvester, "Analysis and mitigation of variability in subthreshold design," in *Proceedings of International Symposium on Low Power Electronics and Design, ISLPED*, Aug. 2005, pp. 20 – 25.

[13] N. Verma, J. Kwong, and A. Chandrakasan, "Nanometer MOSFET variation in minimum energy subthreshold circuits," *IEEE Transactions on Electron Devices, TED*, vol. 55, no. 1, pp. 163 –174, Jan. 2008.

[14] D. Bol, R. Ambroise, D. Flandre, and J.-D. Legat, "Interests and limitations of technology scaling for subthreshold logic," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, TVLSI*, vol. 17, no. 10, pp. 1508 –1519, Oct. 2009.

[15] ——, "Analysis and minimization of practical energy in 45nm subthreshold logic circuits," in *Proceedings of IEEE International Conference on Computer Design, ICCD*, Oct. 2008, pp. 294 –300.

[16] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter variations and impact on circuits and microarchitecture," in *Proceedings of Design Automation Conference, DAC*, Jun. 2003, pp. 338 – 342.

[17] K. Agarwal and S. Nassif, "Characterizing process variation in nanometer CMOS," in 44^{th} *ACM/IEEE Design Automation Conference, DAC*, Jun. 2007, pp. 396 –399.

[18] L.-T. Pang and B. Nikolic, "Measurements and analysis of process variability in 90 nm CMOS," *IEEE Journal of Solid-State Circuits, JSSC*, vol. 44, no. 5, pp. 1655 –1663, May 2009.

[19] D. Kamel, C. Hocquet, F.-X. Standaert, D. Flandre, and D. Bol, "Glitch-induced within-die variations of dynamic energy in voltage-scaled nano-CMOS circuits," in *Proceedings of European Solid-State Circuits Conference, ESSCIRC*, Sept. 2010, pp. 518 –521.

[20] D. Kamel, M. Renauld, D. Bol, F.-X. Standaert, and D. Flandre, "Analysis of dynamic differential swing limited logic for low-power secure applications," *Journal of Low Power Electronics and Applications*, vol. 1, no. 2, pp. 98–126, 2012. [Online]. Available: http://www.mdpi.com/2079-9268/2/1/98/

[21] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, "Strong authentication for RFID systems using the AES algorithm," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES*, 2004, pp. 357–370.

[22] J. Kwong and A. Chandrakasan, "Variation-driven device sizing for minimum energy sub-threshold circuits," in *Proceedings of International Symposium on Low Power Electronics and Design, ISLPED*, Oct. 2006, pp. 8 –13.

[23] S. Ghosh and K. Roy, "Parameter variation tolerance and error resiliency: New design paradigm for the nanoscale era," *Proceedings of the IEEE*, vol. 98, no. 10, pp. 1718 –1751, Oct. 2010.

[24] H.-S. Wong and Y. Taur, "Three-dimensional atomistic simulation of discrete random dopant distribution effects in sub-0.1 μm MOSFET's," in *International Electron Devices Meeting, IEDM, Technical Digest*, Dec. 1993, pp. 705 –708.

[25] A. Asenov, "Random dopant induced threshold voltage lowering and fluctuations in sub-0.1 μm MOSFET's: A 3-D atomistic simulation study," *IEEE Transactions on Electron Devices, TED*, vol. 45, no. 12, pp. 2505 –2513, Dec. 1998.

[26] P. Oldiges, Q. Lin, K. Petrillo, M. Sanchez, M. Ieong, and M. Hargrove, "Modeling line edge roughness effects in sub 100 nanometer gate length devices," in *International Conference on Simulation of Semiconductor Processes and Devices, SISPAD*, 2000, pp. 131 –134.

[27] C. Diaz, H.-J. Tao, Y.-C. Ku, A. Yen, and K. Young, "An experimentally validated analytical model for gate line-edge roughness (LER) effects on technology scaling," *IEEE Electron Device Letters*, vol. 22, no. 6, pp. 287 –289, Jun. 2001.

[28] M. Niwa, T. Kouzaki, K. Okada, M. Udagawa, and R. Sinclair, "Atomic-order planarization of ultrathin $\mathbf{SiO_2}$/**Si**(**001**) interfaces," *Japanese Journal of Applied Physics*, vol. 33, no. Part 1, No. 1B, pp. 388–394, 1994. [Online]. Available: http://jjap.jsap.jp/link?JJAP/33/388/

[29] M. Koh, W. Mizubayashi, K. Iwamoto, H. Murakami, T. Ono, M. Tsuno, T. Mihara, K. Shibahara, S. Miyazaki, and M. Hirose, "Limit of gate oxide thickness scaling in MOSFETs due to apparent threshold voltage fluctuation induced by tunnel leakage current," *IEEE Transactions on Electron Devices, TED*, vol. 48, no. 2, pp. 259 –264, Feb. 2001.

[30] J. Watts, K.-W. Su, and M. Basel, "Netlisting and modeling well-proximity effects," *IEEE Transactions on Electron Devices, TED*, vol. 53, no. 9, pp. 2179 –2186, Sept. 2006.

[31] B. Stine, D. Ouma, R. Divecha, D. Boning, J. Chung, D. Hetherington, C. Harwoo, O. Nakagawa, and S.-Y. Oh, "Rapid characterization and modeling of pattern-dependent variation in chemical-mechanical polishing," *IEEE Transactions on Semiconductor Manufacturing*, vol. 11, no. 1, pp. 129 –140, Feb. 1998.

[32] H. Chen and D. Ling, "Power supply noise analysis methodology for deep-submicron VLSI chip design," in *Proceedings of the 34th Design Automation Conference, DAC*, Jun. 1997, pp. 638 –643.

[33] S. Nassif, K. Bernstein, D. Frank, A. Gattiker, W. Haensch, B. Ji, E. Nowak, D. Pearson, and N. Rohrer, "High performance CMOS variability in the 65nm regime and beyond," in *IEEE International Electron Devices Meeting, IEDM*, Dec. 2007, pp. 569 –571.

[34] S. Hanson, B. Zhai, K. Bernstein, D. Blaauw, A. Bryant, L. Chang, K. K. Das, W. Haensch, E. J. Nowak, and D. M. Sylvester, "Ultra-low-voltage, minimum-energy CMOS," *IBM Journal of Research and Development*, vol. 50, no. 4.5, pp. 469 –490, Jul. 2006.

[35] T. A. Brunner, "Why optical lithography will live forever," *Journal of Vacuum Science Technology B: Microelectronics and Nanometer Structures*, vol. 21, no. 6, pp. 2632 –2637, Nov. 2003.

[36] A. Asenov, S. Kaya, and A. Brown, "Intrinsic parameter fluctuations in decananometer MOSFETs introduced by gate line edge roughness," *IEEE Transactions on Electron Devices, TED*, vol. 50, no. 5, pp. 1254 – 1260, May 2003.

[37] H. Sasaki, M. Ono, T. Yoshitomi, T. Ohguro, S. Nakamura, M. Saito, and H. Iwai, "1.5 nm direct-tunneling gate oxide Si MOSFET's," *IEEE Transactions on Electron Devices, TED*, vol. 43, no. 8, pp. 1233 –1242, Aug. 1996.

[38] E. Cassan, P. Dollfus, S. Galdin, and P. Hesto, "Calculation of direct tunneling gate current through ultra-thin oxide and oxide/nitride stacks in MOSFETs and H-MOSFETs," *Microelectronics Reliability*, vol. 40, pp. 585 – 588, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0026271499002656

[39] C. Hedlund, H. a. Blom, and S. Berg, "Microloading effect in reactive ion etching," *Journal of Vacuum Science Technology A: Vacuum, Surfaces, and Films*, vol. 12, no. 4, pp. 1962 –1965, Jul. 1994.

[40] Y. A. Borodovsky, "Impact of local partial coherence variations on exposure tool performance," in *International Electron Devices Meeting IEDM, Technical Digest*, vol. 2440, Feb. 1995, pp. 750 –770.

[41] X. Qi, A. Gyure, Y. Luo, S. C. Lo, M. Shahram, and K. Singhal, "Measurement and characterization of pattern dependent process variations of interconnect resistance, capacitance and inductance in nanometer technologies," in *Proceedings of the 16th ACM Great Lakes symposium on VLSI*, ser. GLSVLSI. New York, NY, USA: ACM, 2006, pp. 14–18. [Online]. Available: http://doi.acm.org/10.1145/1127908.1127914

[42] M. Abu-Rahma and M. Anis, "A statistical design-oriented delay variation model accounting for within-die variations," *IEEE Transactions on*

Computer-Aided Design of Integrated Circuits and Systems, TCAD, vol. 27, no. 11, pp. 1983 –1995, Nov. 2008.

[43] K. Okada, K. Yamaoka, and H. Onodera, "Statistical modeling of gate-delay variation with consideration of intra-gate variability," in *Proceedings of International Symposium on Circuits and Systems, ISCAS*, vol. 5, May 2003, pp. V–513 – V–516 vol.5.

[44] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A compact Rijndael hardware architecture with S-Box optimization," in *ASIACRYPT*, 2001, pp. 239–254.

[45] D. Kamel, F.-X. Standaert, and D. Flandre, "Scaling trends of the AES S-box low power consumption in 130 and 65 nm CMOS technology nodes," in *IEEE International Symposium on Circuits and Systems, ISCAS*, May 2009, pp. 1385 –1388.

[46] X. Yuan, T. Shimizu, U. Mahalingam, J. Brown, K. Habib, D. Tekleab, T.-C. Su, S. Satadru, C. Olsen, H. Lee, L.-H. Pan, T. Hook, J.-P. Han, J.-E. Park, M.-H. Na, and K. Rim, "Transistor mismatch properties in deep-submicrometer CMOS technologies," *IEEE Transactions on Electron Devices, TED*, vol. 58, no. 2, pp. 335 –342, Feb. 2011.

[47] A. Brown and A. Asenov, "Capacitance fluctuations in bulk MOSFETs due to random discrete dopants," *Journal of Computational Electronics*, vol. 7, pp. 115–118, 2008, 10.1007/s10825-008-0181-y. [Online]. Available: http://dx.doi.org/10.1007/s10825-008-0181-y

[48] M. Alioto, G. Palumbo, and M. Pennisi, "Understanding the effect of process variations on the delay of static and domino logic," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, TVLSI*, vol. 18, no. 5, pp. 697 –710, May 2010.

[49] M. Seok, D. Sylvester, and D. Blaauw, "Optimal technology selection for minimizing energy and variability in low voltage applications," in *ACM/IEEE International Symposium on Low Power Electronics and Design, ISLPED*, Aug. 2008, pp. 9 –14.

[50] D. Bol, D. Kamel, D. Flandre, and J.-D. Legat, "Nanometer MOSFET effects on the minimum-energy point of 45nm subthreshold logic," in *Proceedings of the International Symposium on Low Power Electronics and Design, ISLPED*, 2009, pp. 3–8.

[51] S. Mangard, T. Popp, and B. M. Gammel, "Side-channel leakage of masked CMOS gates," in *CT-RSA*, 2005, pp. 351–365.

[52] D. Suzuki, M. Saeki, and T. Ichikawa, "Random switching logic: A countermeasure against DPA based on transition probability," IACR ePrint on Transition Probability, Tech. Rep., 2004.

[53] D. Bol, "Robust and energy-efficient ultra-low-voltage circuit design under timing constraints in 65/45 nm CMOS," *Journal of Low Power Electronics and Applications*, vol. 1, no. 1, pp. 1–19, 2011. [Online]. Available: http://www.mdpi.com/2079-9268/1/1/1/

[54] S. Hanson, B. Zhai, M. Seok, B. Cline, K. Zhou, M. Singhal, M. Minuth, J. Olson, L. Nazhandali, T. Austin, D. Sylvester, and D. Blaauw, "Exploring variability and performance in a sub-200-mV processor," *IEEE Journal of Solid-State Circuits, JSSC*, vol. 43, no. 4, pp. 881 –891, Apr. 2008.

[55] J. M. Rabaey, A. P. Chandrakasan, and B. Nikolic, *Digital integrated circuits : A design perspective*, 2nd ed., ser. Prentice Hall electronics and VLSI series. Pearson Education, Jan. 2003. [Online]. Available: http://www.worldcat.org/isbn/0130909963

[56] J. Tschanz, J. Kao, S. Narendra, R. Nair, D. Antoniadis, A. Chandrakasan, and V. De, "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," *IEEE Journal of Solid-State Circuits, JSSC*, vol. 37, no. 11, pp. 1396 – 1402, Nov. 2002.

[57] M. Miyazaki, G. Ono, T. Hattori, K. Shiozawa, K. Uchiyama, and K. Ishibashi, "A 1000-MIPS/W microprocessor using speed adaptive threshold-voltage CMOS with forward bias," in *IEEE International Solid-State Circuits Conference, ISSCC, Digest of Technical Papers*, 2000, pp. 420 –421, 475.

[58] M. Miyazaki, G. Ono, and K. Ishibashi, "A 1.2-GIPS/W microprocessor using speed-adaptive threshold-voltage CMOS with forward bias," *IEEE Journal of Solid-State Circuits, JSSC*, vol. 37, no. 2, pp. 210 –217, Feb. 2002.

CHAPTER 4

ASSESSMENT OF SIDE-CHANNEL SECURITY OF DDSLL AND STATIC CMOS S-BOXES

4.1 INTRODUCTION

Since the apparition of side-channel attacks in the late 1990's, there has been an escalating attention in both the industry and the research fields to develop solutions that ideally prevent them. The strength behind these types of attacks is the fact that they are relatively cheap and easy to conduct compared to classical cryptanalysis based on a black box approach that explores the weaknesses of the algorithm (mathematical model). Independent on the type of algorithm, whether it is symmetric or asymmetric cryptographic algorithms, the main goal is to protect the secret information at all levels. Previously, the algorithm was considered to be "computationally" secure against classical cryptanalysis if breaking it requires impractical computing power [1]. However, side-channel attacks opened a whole new field of vulnerabilities that cause these computationally secure algorithms to leak information at the physical-level. Power analysis attacks, especially the differential power analysis (DPA) attacks [2], are particularly appealing in this field because they are easily mounted against the cryptographic device (basically they need an oscilloscope and a PC to monitor, capture and process the power information) and because they are considered very powerful.

The main cause of information leakage through the power consumption of a cryptographic device is related to the inherent use of the static CMOS logic to build its gates. Historically, static CMOS logic has been widely used in digital circuit design for low-power (LP) applications due to its intrinsic low power,

small size (compared to dynamic differential logic styles (DDL)), robustness to voltage scaling and scalability with technology. It has been also highly automated in CAD tools which renders it an ideal choice for most applications. However for secure cryptographic devices, static CMOS logic is not suitable, since its dynamic power consumption is highly correlated to the processed data. Such a deficiency leads to leaking secure information through power analysis attacks.

Meanwhile, countermeasures against power analysis attacks extend through various abstraction levels (protocol-level, e.g. [3], algorithm-level, e.g. [4], hardware-level, e.g. [5]). Hiding techniques, such as noise generation and the use of dynamic differential logic (DDL) styles (e.g. [5]) and masking techniques (e.g. [6]) are potential solutions [1]. However, these countermeasures cause significant performance degradations, such as large area, high power consumption overheads and low throughput. An important research goal is to determine the best tradeoff between security and performance specially for highly constrained low-power secure applications, such as RFIDs and smart cards.

In this study we are concerned with hardware-level countermeasures that are intuitively attractive as they tackle the problem directly where it lies. We are particularly interested in evaluating the security of the dynamic differential swing limited logic (DDSLL) style [7]. In Chapter 2, we discussed the different design aspects of the DDSLL style while concentrating on reducing the large area and power consumption overheads generally attributed to the use of DDL styles. For this purpose, we use the AES S-box designed using 65 nm CMOS technology as a test case. The first part of this study tackles an important question: What is the numerical amount of security gain brought by the use of DDL styles in general and more specifically, DDSLL? This question is brought up in the context of comparing different types of well known countermeasures. More specifically, masking is known to increase the security gain exponentially with the order of the mask (if the amount of noise present is large enough) [8]. On the other hand, noise addition shows a linear increase of security gain. Meanwhile, DDL styles increases the security gain by adding a constant compared to an unprotected device.

As a result, we try to find out the significance of this constant brought by the use of DDSLL [9] relative to the high security gain of masking and noise addition, specially that DDSLL reduces the power / area overhead, whereas masking for example is known to increase these overheads. We apply the unified security framework proposed in [10] to the DDSLL style. We start by performing an information theoretic analysis to assess the security of DDSLL in a worst-case scenario against the most powerful adversary who uses profiled template attacks [11]. The second step is to apply the security analysis in order to numerically evaluate the security enhancement (the value of the constant) achieved by DDSLL, which is reported to be one order of magnitude compared to static CMOS. This figure is rather modest compared to masking for example. As a result, an important open question is raised to determine whether we can do better. That is, either develop a logic style that sufficiently increases the security gain constant over static CMOS, or combine several countermeasures, e.g. DDL style

that features masking as an additional countermeasure. The second solution is particularly appealing, especially with the use of DDSLL because of its power / area overhead reduction that potentially gives room for implementing a masking scheme as well.

Another interesting issue that we address in this part of the study is the non-linearity of the leakage function of the DDSLL style (the leakage function is considered linear[1] if it can be estimated by well known and linear models, e.g., Hamming weight or by linear functions as in the case of the stochastic approach [12]). Indeed, current non-profiled distinguishers rely on the fact that the leakage functions of the attacked devices are linear [13]. Therefore, having a non-linear leakage function (where the leakage traces cannot be described by amplitude differences between states only) would make the success of these attacks more difficult to achieve. A good choice to reveal the linearity nature of the leakages is to use the stochastic approach [12] or its non-profiled version, "on-the-fly" [14]. This choice is motivated by the fact that the stochastic attack is a linear, regression-based approach that approximates the device's leakages with a linear function [12, 15]. Therefore, it evaluates the linearity of a leakage function by comparing the obtained estimated information leakage to the one acquired from the profiled template attack [11]. In the case of the DDSLL S-box, these leakages appear to be non-linear at the simulation level. Contrarily, actual (measured) leakages are rather linear except for a few samples that feature some non-linearity. This is attributed to the measurement artifacts (such as the different capacitances and inductances of the chip's package, the wire bonds, the ZIF socket, the PCB traces and the cables used) that are not taken into consideration during simulations. As a result, practical attacks are feasible against DDSLL even if only a few linear leakage samples are available. This is also true for static CMOS; the simulated traces reveal a non-linear leakage function, whereas the actual measured traces prove the leakage function to be linear for most of the time samples. Therefore, we raise an open question whether it is possible that designers consider non-linearity as a design guideline while developing secure logic styles.

The second part of this study targets the impact of increasing variability in advanced technologies on the security of the static CMOS logic [16]. Indeed, variability is a growing concern for circuit designers as explained in the previous chapter. Also from a cryptanalysis perspective, process variations present more challenges to actual adversaries. Here, we also study the AES S-box designed using 65 nm CMOS technology as a test case. In this part of the study, we first employ the information theoretic analysis using profiled templates in order to investigate two possible scenarios. The first is a worst-case scenario, where the adversary profiles and attacks the same die. The second and more realistic

[1]The processed data (either input, output or even internal signals) whose values are either 0 or 1 are used by the model / function to estimate the physical leakage of an implementation. Therefore, the resulting leakage model best describes amplitude differences between states. An example of current traces that feature a linear leakage function is given in Appendix E.

scenario is that the adversary profiles n dies and attacks the $n+1^{th}$ die. In this case, extending template attacks to infer process variations causes a significant loss of information as a result of profiling more than one die, so the template is not accurate for any die. Subsequently, actual adversaries using profiled attacks face a challenging task trying to build accurate templates, due to technology scaling. Nevertheless, the use of profiled template attacks is still important in the worst-case scenario because they provide the lower limit which the designer should consider while evaluating an implementation.

As a result, it is interesting to evaluate the effectiveness of the second category of distinguishers, namely the non-profiled attacks under the impact of process variations using the security analysis proposed in [10]. We employ the security analysis to determine the accuracy of the models used by these non-profiled attacks for the 65 nm CMOS technology. The correlation power analysis (CPA) that use Pearson's correlation coefficient [17] is a classical DPA tool that we use in our evaluation. Generally, CPA attacks use either the Hamming weight or the Hamming distance to model the leakage function of the cryptographic device. Another type of non-profiled attacks that we employ is the "on-the-fly" stochastic approach with a linear basis. We see that variability causes the CPA attacks using the Hamming weight model to fail due to the deficiency of the model to accurately estimate the leakage function of the static CMOS S-box at 65 nm CMOS technology and clearly beyond. Contrarily, the "on-the-fly" stochastic attack proves to be a particularly interesting approach in this context, since the leakage function of each die can be separately estimated by a linear function. This clearly indicates that the leakage functions of the dies implemented on a 65 nm CMOS technology in presence of process variations, are still linear. It is important to note that although the Hamming weight model used by the CPA attacks rely on the fact that the leakage functions are linear, it fails to estimate the correct leakage functions of the measured dies on 65 nm CMOS technology under the impact of process variations due to the fixed nature of the model.

4.2 PRELIMINARIES AND DEFINITIONS

In this section we list the necessary preliminary information needed by the circuit designer to understand the results obtained in this chapter. We start by defining the key elements in an attack environment where the described terminology will be used in the remainder of the chapter. A simple explanation of side-channel analysis is given to be aware of the various steps needed to mount a side-channel attack. The basic leakage models used in the case of static CMOS devices are introduced. Finally, the adversary's choice between two main categories of distinguishers is discussed along with an insight of the different types of attacks used in this chapter.

4.2.1 Attack environment: Definitions and terminology

Extracting secret information from a secure application can be done at different abstraction levels; either at the algorithmic level (mathematically), or at the hardware level (physically). However during the last decade, targeting actual hardware has proved to leak secret information from algorithm-secure cryptographic devices. Generally, physical attacks are categorized [1]:

- Invasive *vs.* non-invasive: The former is considered a tamper attack that damages the device it tries to extract the secret information from. While the latter exploits externally available information without physically modifying the device.
- Active *vs.* passive: Active attacks manipulates the functionality of the device, for example fault attacks [18] that try to induce errors in the computation. On the other hand, passive attacks just observe the inherent physical leakage of the device which can be the power consumption of the device [2], electromagnetic radiation [19, 20] or timing information [21].

In this study, we focus on the context of side-channel attacks which are considered non-invasive and passive in nature. First we define the attack environment as described in [10]. It basically consists of:

- The *device*: To define the device, we have to first introduce the term *primitive* which corresponds to the cryptographic algorithm. For example, the AES Rijndael is a cryptographic primitive. Now, the *device* denotes the physical realization of a cryptographic primitive using a specific implementation. It can be a smart card, an FPGA or even a μprocessor that features a cryptographic primitive. The *implementation* of the device corresponds to the way the device is designed, for example, the type of adopted logic style, the desired countermeasure technique, ... etc.
- The *side-channel*: It is the communication channel that the cryptographic device transfers its information through. However, this channel leaks some information about the processed data denoted by the *physical observable*. The power consumption and the electromagnetic radiation of a target device are examples of side-channel physical observables.
- The *measurement equipment*: It usually comprises an oscilloscope to monitor the physical observable and a PC to carry out communications between the device and the oscilloscope.
- The *leakage trace*: It refers to a set of power consumptions or electromagnetic radiations (as examples of the physical observable) consecutively captured across the operation of the device.
- The *leakage function*: It describes how the device actually leaks information through the side-channel. It also includes all the specificities of the measurement setup.

– The side-channel *adversary*: He is a malicious entity (attacker) that aims at analyzing the physical observable in order to obtain the leakage function. He is assumed to have full access to the physical observables.

– The *leakage model*: It is how the adversary interprets the physical observable. For example, in the case of the standard CPA attack [17], the adversary uses a Hamming weight leakage model in order to approximate the leakage function.

4.2.2 Side-channel analysis

As previously explained, the output of the side-channel can be the power consumption of the target device, its electromagnetic radiation or timing information. In this study we are mainly concerned with the information leaked by the power consumption and hence we briefly describe the types of power analysis.

First, we introduce how the power consumption of a device leaks information about the processed data. The power consumption consists of three components; the dynamic power, the short-circuit power and the leakage power. The first one is due to the charging and discharging of the load capacitance (C_L). The second one results from the short-circuit current that flows through simultaneously ON NMOS and PMOS transistors during an input transition. Finally, the leakage power is caused by the leakage currents of transistors and pn-junctions. The contributions of these components to the total power consumption depends on the technology and the operating supply voltage. Since the emergence of side-channel analysis and particularly the power analysis, the dynamic power consumption has been considered the main source of information leakage because of its direct relation to the activity of the device. Equation 4.1 expresses this relationship as follows:

$$P_{dyn} = \frac{1}{2} \times C_L \times V_{DD}^2 \times f \times \alpha_F \tag{4.1}$$

where V_{DD} is the supply voltage, f is the frequency of operation and α_F is the activity factor of the device. In static CMOS devices, α_F reveals which transition actually occurred. It can be concluded that when the input transitions from high to low, the measured power consumption will exhibit a high peak due to the charging of the load capacitance through the PMOS transistor. Whereas, when the input goes from low to high, thus turning on the NMOS transistor, C_L is discharged and the power consumption features a small peak due to the short-circuit currents. On the other hand, if the input remains unchanged (either low or high), there will be no changes in the power consumption. As a result, the switching information in the power trace of a static CMOS device is very useful from a side-channel view because the adversary can detect which input transition took place.

Now, the power analysis pioneered by Kocher in [2] is divided into simple power analysis (SPA) and differential power analysis (DPA). Simple power analysis depends on capturing a single or few power traces and analyzing the patterns

within one trace. It involves correctly assigning intermediate results occurring during the operation of the device to parts of the power trace caused by processing these results; thus interpreting the secret information. On the other hand, differential power analysis is more sophisticated because it is based on statistical analysis rather than visual inspection as in SPA. The main advantage of DPA compared to SPA is that there is no need to have a detailed knowledge about the cryptographic device, it is enough to just know the type of algorithm used. Further details of the DPA and how it is used in an attack are discussed in Section 4.2.5.2.

Anyway, using these types of power analyses to mount attacks against cryptographic devices has been an extensive field of research for the last decade. The steps that an adversary takes to perform a side-channel attack in general are introduced in [10] and tailored for our case study as follows. The adversary first selects the target algorithm (AES), the leakage source (dynamic power consumption) and the measurement setup (described in Section 4.5). The adversary then identifies the target signal. Since side-channel attacks are generally based on a divide-and-conquer strategy, the adversary chooses small parts of the key as the target of his attack. After that, he either selects or monitors the inputs to the device (plaintexts) depending on his accessibility. Then, the adversary builds matrices of predicted key-dependent internal values. In order to reduce the computational complexity, the key is subdivided into small sub-keys as dictated by the divide-and-conquer strategy. These prediction matrices are used later during the execution phase. Following, the adversary chooses and applies a leakage model (is described in Section 4.2.3). Hamming weight and Hamming distance models are widely used in the context of power analysis to predict the leakage of the target device. Now, the adversary is ready to perform the actual measurements of the leakage source, which is the power consumption in this study. He then builds leakage matrices that contain leakage traces. After selecting the relevant leakage samples, the adversary compares the predicted leakages to the measured ones based on the type of distinguisher (as discussed in Section 4.2.5) he adopts.

4.2.3 Models

The most well known leakage models that are derived for and best suit devices designed using the static CMOS style are the Hamming weight and the Hamming distance. The first model simply counts the number of non-zero bits present in the analyzed value. It assumes that the side-channel leakage (the power consumption in this study) is correlated to the Hamming weight of the value computed in a device. On the other hand, the Hamming distance model counts the number of different bits present in two values of equal length. It assumes that when a value in a static CMOS device switches, the side-channel leakage is correlated with the Hamming distance of these values.

These models in the above mentioned form are considered raw because the Hamming weight model assumes that all bits have equal contributions to the

overall power consumption and the Hamming distance model considers the transition from high to low and vice versa to have the same impact to the overall power consumption as well. Therefore, to improve the efficiency of the side-channel attack, these leakage models are improved. For example, in [22] different leakages for the high to low transition and vice vera are proposed. Also in [23], the authors proposed the use of the weighting sum technique for the predictions of the different leaking signals in a device based upon the fact that these signals leak differently depending on their load capacitances.

Although the modified Hamming weight or Hamming distance models can accurately predict the leakage of a device, they are only suitable for static CMOS devices. Devices that use other logic styles; for example, DDL styles, have different power consumption behavior that cannot be simply modeled using Hamming weights or Hamming distances as shown latter in the chapter.

4.2.4 Terminology

The following are the general guidelines for the notation undertaken in the remaining sections of this chapter.

- Capital letters are generally assigned to random variables, while lower cases refer to specific samples of the random variable.
- In symmetric cryptography where the key K_s is kept secret, the plaintext is denoted by P while the ciphertext is referred to as C_c.
- In the simulation / measurement environment, L represents the model of the leakage function whose output l is the power trace.
- Generally, the leakage function has three input arguments: the discrete random variable X which denotes the value of the processed data under investigation, the discrete random variable C which refers to the index of the die under investigation and the continuous random variable N represents the noise in the measurements.
- The power trace is composed of t time samples.

4.2.5 Distinguishers: profiled versus non-profiled

Generally, the term *distinguisher* refers to the type of attack that an adversary mounts against the target device. Distinguishers can be classified as profiled and non-profiled attacks. Each class is equally important, however, depending on the context, one class might be favored over the other.

Profiled attacks are simply divided into two phases; a characterization (offline) phase and an exploitation (online) phase. In the characterization phase, the adversary studies the target device with the goal of obtaining a precise knowledge of the leakage functions. This step is done offline where the adversary is expected

to have access to all the necessary resources enabling him to fully profile the leakage distributions of the target device. The second phase involves exploiting the knowledge obtained from the characterization phase to actually mount the attack. This step is done online with the goal of recovering the secret key. Despite the fact that profiling the leakages prior to mounting the actual attack increases the efficiency of the adversary, it requires a lot of computational resources as well as time and storage capabilities. This makes the class of profiled attacks a worst-case scenario where the adversary who employs them is considered an optimal adversary because the tedious profiling phase and the huge amount of allocated resources are traded for the higher achieved efficiency. As a result of their high efficiency, profiled attacks are considered good candidates for evaluating the security of a target implementation.

On the other hand, non-profiled attacks do not feature an offline characterization phase since actual adversaries do not have the required capabilities to fully profile the leakages of the target device. In this context, the actual adversary is considered suboptimal. This class of distinguishers is further divided into two subclasses; non-profiled attacks that involve the use of predefined models such as the Hamming weight and Hamming distance or predefined selection functions such as the single-bit DPA attack [2], and the non-profiled attacks that feature limited profiling of the leakages which in this case is done on-the-fly. The second subclass of distinguishers have gained particular attention since the use of typical models has proved to be not completely adequate in certain contexts, such as the presence of process variations [16, 24] and the use of DDL styles [13]. Another important point in favor of the "on-the-fly" non-profiled distinguishers is that they allow performing optimized attacks that lead to a higher efficiency in obtaining the secret key [10].

In general, these two categories of distinguishers may lead to different conclusions. However, profiled and non-profiled attacks are considered complementary in the sense that they shed different lights on the security of devices.

4.2.5.1 Profiled attacks

Template attacks [11] are considered a good example of profiled attacks because they are useful for estimating the worst-case scenario with the most powerful adversary. The adversary is assumed to have the prior knowledge of the target implementation that allows him to profile the leakage function (offline - during a preparation phase) and capture all the information available [25, 26]. This is essentially important for evaluating the security of a certain implementation, since the designer should base his assessment on a worst-case scenario. However, this requires a lot of computational efforts in order to first profile and characterize the leakages which is also time and memory consuming. Nevertheless, while this drawback is considered a limitation for an adversary, it is not a problem for a designer, since the designer is assumed to have all the computational, time and memory capabilities to perform the profiling phase of the template attack.

The basic principle as described in [11], comes from the signal processing domain, where the measured signal is divided into two parts; an intrinsic signal and

noise. The intrinsic signal is considered deterministic, whereas the noise is actually random. Now, the template consists of the mean signal and noise probability distributions. During the characterization phase of the template attack, the adversary builds 2^n templates corresponding to the 2^n values of the unknown n-bit key. Each template is a Gaussian distribution $\mathcal{N}(l|\hat{\mu}_{x,N}, \hat{\sigma}^2_{x,N})$, where x refers to the processed data and N denotes the noise random variable. In the exploitation phase, the adversary uses the maximum likelihood approach [27] where the candidate key maximizes the probability of the noise in the measured signal as in:

$$\tilde{x} = \underset{x^*}{\operatorname{argmax}}\, \hat{Pr}_{model}[x^*|l] \tag{4.2}$$

Building templates is done in an iterative manner because at first the choice for the key bits and the processed data is limited to a small set. Then, it is repeated with an increasingly longer portions of the data and the corresponding templates to limit the space of possible key guesses. Thus, the limiting strategy is key to the success of the template attacks because it depends on how effectively the adversary can limit the combinatorial explosion during the iterative process.

As for the stochastic attack [12], it is basically introduced as a type of profiled distinguishers that depend on linear models (that best describe amplitude differences between states) during the characterization phase. It works as follows: First, during the characterization phase, the adversary chooses the base vectors which in our case are the S-box output bits (i.e., a 9-element basis, with 8 output bits and a constant term). Then he tries to approximate the leakages with a linear function $\hat{L}_t = \Sigma_i\, \beta_{i,t} \,.\, g_i(x)$, where the coefficients $\beta_{i,t}$ are determined by regression and $g_i(x)$ represent the basis. Intuitively, stochastic attacks are less precise than template attacks because of their dependency on linear models with a small set of base vectors as in our case. In this context, the stochastic attack can be mounted rapidly and thus provides a "simpler to estimate" approximation of the leakage function [25, 26]. Of course, if the degree of the stochastic model increases to 8 with 256 coefficients, then it would be equivalent to a template. Also, stochastic attacks are very convenient in evaluating the non-linearity of the leakage function.

4.2.5.2 Non-profiled attacks

We start by describing Kocher et al.'s single-bit DPA attack [2]. The basic idea is to magnify the effect of a well chosen single bit within an n-bit word in the target device. The first step in the single-bit DPA involves the measurement of m encryption operations and capturing power traces $l_{1..m}[1..t]$ containing t samples each. Also, the adversary is assumed to have access to the ciphertexts $C_{c\ 1..m}$. Next, the adversary chooses a key-dependent selection function $D(C_c, b, K_s)$ which is defined by computing the value of bit b (1..n of an n-bit word) of an intermediate signal in the device for a given ciphertext C_c, where K_s represents the secret key associated to bit b. In other words, the output of $D(C_c, b, K_s)$ is the value of the selection bit b corresponding to fixed C_c and K_s.

The adversary then computes the differential trace $\Delta_D[1..t]$ which is defined as the difference between the average of the traces for which the selection bit is 1 and the average of the traces for which the section bit is 0. The resulting curve will feature a peak corresponding to the correct key guess and a noise floor approaching 0 as the number of measurements m increases. As a result, at the peak of the differential trace, the selection function is correlated to the value of the processed bit b. Thus, the adversary is capable of differentiating between the correct key guess and the rest of the key guesses.

The second type of non-profiled attacks is the correlation power analysis (CPA) attack [17, 28]. Similar to the single-bit DPA attack, the CPA attack depends on statistical analysis. However, they feature different decision criteria. CPA attacks differentiate the correct key by analyzing the correlation coefficient of the processed data and their measured power consumption. It depends on the fact that the power consumption associated to the processed data is proportional to the Hamming weight of this data. In [17], the authors suggest the use of the Hamming distance in order to obtain a more accurate power model. To mount the attack, the adversary gathers l power traces, such that L represents the random variable of the measured power traces. Using either model, the adversary builds the leakage model, where H refers to the random variable of the Hamming weight / distance of the processed data X. Then the correlation coefficient $\rho_{L,H}$ between L and H is expressed as:

$$\begin{aligned} \rho_{L,H} &= \frac{Cov(L,H)}{\sigma_L \sigma_H} \\ &= \frac{E((L-\mu_L)(H-\mu_H))}{\sqrt{V}ar(L)\sqrt{V}ar(H)} \end{aligned} \tag{4.3}$$

where E is the expected value function, $Cov(L,H)$ is the covariance between L and H and Var is the variance of either L or H. The maximum value of $\rho_{L,H}$ indicates a correct key guess since the measured power consumption is correlated to the hypothetical power model.

Also, the stochastic approach can be implemented as a non-profile distinguisher by applying a variant version described in [14]. In this case, the adversary builds the leakage model "on-the-fly" using a linear basis. The main idea of this variant is that it assumes that the most accurate model should correspond to the correct key hypothesis under the condition that the base vectors used to build this model reasonably match the actual leakages.

4.3 COUNTERMEASURES

Since side-channel attacks rely on the fact that the side-channel is correlated to the processed data, therefore, the goal of countermeasures is to break this correlation or at least reduce it. Countermeasures exist at different abstraction levels (e.g. protocol, algorithm, hardware). Generally, they are grouped into two main categories; namely, masking and hiding as detailed in [1].

Masking aims at decorrelating the power consumption of a cryptographic device and its intermediate values by randomizing these intermediate values. As a potential candidate for countermeasures against side-channel attacks, it has been a popular research area for the last decade. Masking can be implemented at the algorithm level, for example in [6, 29–31], without changing the power consumption characteristics of the cryptographic device. It is also implemented at the hardware-level which further contains architecture- and gate-level solutions [32–34].

On the other hand, the goal of hiding is to make the power consumption of the cryptographic device independent on the processed data. Hiding can be achieved either by randomizing the power consumption of the device in each clock cycle, or by having the device consume exactly the same power in each clock cycle. The first method, which is the randomization of the power consumption, is addressed at the algorithmic level via time randomization, in which case the operations of the cryptographic device are performed in an arbitrary order [35] or by the random insertion of dummy operations. A second approach is to randomize the power consumption of the cryptographic device at the hardware architecture level by performing random switching activities in parallel to the actual operation with the goal of increasing the switching noise variance ($Var(P_{sw.\,noise})$), also known as algorithmic noise, in order to reduce the signal-to-noise ratio (SNR) given by:

$$SNR = \frac{Var(P_{exp})}{Var(P_{sw.\,noise} + P_{el.\,noise})} \tag{4.4}$$

where $Var(P_{exp})$ refers to the exploitable power consumption variance and $Var(P_{el.\,noise})$ denotes the electronic noise variance. Also, increasing $Var(P_{el.\,noise})$ through the use of noise engines to achieve higher background noise has been proposed in [1]. Another approach to reduce the SNR is to reduce the signal; i.e, reduce $Var(P_{exp})$ which falls under the category of having the cryptographic device consume exactly the same power during each clock cycle. This can be achieved at the hardware gate-level by employing dynamic differential logic (DDL) which is also referred to as dual-rail precharge (DRP) logic.

Despite all the efforts to avoid the dependency of the power consumption on the processed data through various types of countermeasures, this goal is still considered ideal. The above mentioned solutions only reduce the dependency; hence, increase the difficulty of performing the side-channel attacks, but do not fundamentally prevent them.

4.3.1 Masking

As mentioned earlier, masking represents a potential candidate as a countermeasure against side-channel attacks, thus it has been a wide research area for the last decade. The basic idea is to use a random value m that is called *mask* to conceal the intermediate value v as follows:

$$v_m = v * m \tag{4.5}$$

The mask is generated within the cryptographic device, hence it is not known to the adversary. The operation $*$ is most often the Boolean exclusive-or function $\oplus$, the modular addition $+$, or the modular multiplication $\times$, depending on the type of masking, whether it is Boolean or arithmetic (further details of the different types of masking and their fields of application is beyond the scope of this study). The security proof of masking is based on the assumption that if the masked value v_m is independent on v and m, then the power consumption of v_m is likewise independent on v [36]. However, this assumption is not always valid since the power consumption normally depends on several values of v that, in the case of a single mask, share the same mask m. Hence, the power consumption can be correlated to the unmasked intermediate values [1]. As a result, increasing the order of masking d (where $d+1$ is the number of shares), i.e., using more than one mask is an appealing solution to enhance the security of this countermeasure and makes it less prone to standard DPA attacks. Now, higher-order DPA attacks are necessary to retrieve the secret information from higher-order masking schemes [6]. However, extending the order of the masking comes at the cost of an inevitable significant performance overhead as well as a large increase of area because masking requires the computation of the different shares individually. In Appendix F, we illustrate some examples of masking schemes implemented at both the algorithm and gate (logic style) levels.

Using dedicated masked logic styles is an interesting solution because it enables the optimization of the masking scheme which can reduce the performance overhead. Such a solution still needs certain precautions to be effective, for example the problem of glitches jeopardizes the security of masked logic styles [37, 38] and it needs to be addressed. Also, the early propagation effect weakens the security of masked logic styles [39–41] and it has to be avoided. Another important issue is that a higher-order DPA attack (e.g. [6, 42]) can be mounted against cryptographic devices that feature masked logic styles. In this case the adversary processes the leakages such that he combines the ones corresponding to the shares that use the same mask in order to obtain key-dependent information. Thus, combining the mask scheme with a hiding solution, such as noise addition or the use of dynamic differential logic further enhances the security of the cryptographic device [8, 13].

Consequently, it is important to quantify the security gain that a masking scheme provides in function of the order of the scheme (d). Since, the effectiveness of the masking scheme is dependent on the amount of noise [8] proposed to plot the evaluating metrics with its respect. It was shown in [8] that the data complexity of a side-channel attack increases exponentially with the number of shares in the masking scheme if the amount of noise present is large enough. In Section 4.3.4, the impact of masking in function of the order of the scheme along with other types of countermeasures is expressed in the preliminary plot of data complexity versus the noise variance.

4.3.2 Noise addition

Among the methods to implement hiding is the noise addition with the goal of reducing the SNR as mentioned earlier. Performing random switching activities in parallel to the actual operation in the cryptographic device increases $Var(P_{sw.\,noise})$, whereas the use of noise engines help increase $Var(P_{el.\,noise})$ [1]. Noise engines fundamentally rely on random number generators that are connected to a network of large capacitors to be able to have a big impact on the power consumption. The charging / discharging of these capacitors is done in a random fashion which leads to the increase of $Var(P_{el.\,noise})$.

Independent on the technique used to increase the noise variance, the impact of noise addition is manifested as a linear increase in the security against side-channel attacks. Since it is considered as a generic countermeasure, [8] suggested to plot the evaluating metrics with its respect. In Section 4.3.4, the impact of noise addition along with other types of countermeasures is expressed in the preliminary plot of data complexity versus the noise variance. Also, the authors in [8, 43] showed that using a stand-alone countermeasure, such as masking or time randomization is not effective unless there exists a sufficient amount of noise. As a result, for small cryptographic devices, where there is a big chance its noise level is low, additional noise generation should be considered to further enhance the security gain [43]. The impact of noise addition as a complementary countermeasure to masking of even high orders is shown to significantly improve the resistance of the cryptographic device against DPA attacks by an order of magnitude [31] depending on the amount of generated noise.

4.3.3 DDL styles

The use of DDL styles as a countermeasure against side-channel attacks has been a popular research area for the last decade [5, 7, 44–46]. They are considered a type of hiding technique that aims at reducing the SNR at the hardware gate level, where the instantaneous power consumption of a cryptographic device is ideally the same in each clock cycle. As detailed in Chapter 2 and mentioned here as a reminder, equalizing the effect of different events on the power consumption is done as follows:

- The differential operation eliminates the difference in the power consumption between a 1 - 0 and a 0 - 1 event as well as between a 0 - 0 and a 1 - 1 event. However, 1 - 0 / 0 - 1 events consume power, whereas 0 - 0 / 1 - 1 do not.

- The dynamic operation (precharge) eliminates the difference between a 0 - 0 and a 1 - 0 event and between a 1 - 1 and a 0 - 1 event. Nevertheless, the 1 - 1 / 0 - 1 are the only events that consume power due to the precharging.

As a result, combining the differential and the dynamic operations into a single logic style is essential to have the same instantaneous power consumption every

clock cycle. However, it is not sufficient as other factors play an important role causing variations in the power consumption that are dependent on the processed data. Key factors, such as the presence of glitches [38], early propagation effects [39, 47] and capacitance imbalances in the differential nodes cause the DDL style to leak information as its power consumption is still correlated to the processed data.

Since glitches are a main source of information leakage in side-channel attacks, they are usually avoided in the design of DDL styles. On the other hand, early propagation effects that appear in some DDL styles, such as wave dynamic differential logic (WDDL) [39, 44], masked dual-rail with precharge logic (MDPL) [33, 40] and dual-rail random switching logic (DRSL) [34] should be avoided by simple self-timing techniques. Sense amplifier based logic (SABL) [5], dynamic current mode logic (DyCML) [45] and dynamic differential self-timed logic families developed by Hassoune et al. [7] are good examples of DDL styles that are designed to avoid both glitches and early propagation effects.

However, capacitance imbalances of the differential nodes are considered a serious threat to the security of the DDL styles. The capacitances of the differential outputs consist of three components; namely, the output capacitance of the DDL gate C_o, the sum of the input capacitances of the subsequent gates C_i and the routing capacitance to the subsequent gates C_{rout}. As the technology scales down below 100 nm, the contribution of C_{rout} becomes dominating [48, 49]. Therefore, balancing the differential wire capacitance is the most essential task according to [1]. A differential routing methodology is proposed in [50] and its details are previously discussed in Chapter 2.

In order to understand to which extent these DDL styles add to the security of the cryptographic device, some sort of security assessment is needed. Unfortunately, no unified method is used to quantify the security gain of these DDL styles, each one or group are characterized individually. This makes it extremely difficult to point out a single DDL style that is superior to all others. A well known method uses the normalized energy deviation and the normalized standard deviation metrics proposed in [5], which are given by:

$$NED = \frac{max(E) - min(E)}{max(E)} \tag{4.6}$$

$$NSD = \frac{\sigma(E)}{\mu(E)} \tag{4.7}$$

where E is a random variable representing the energy per cycle of an operation. The NED is claimed to be useful, since the smaller the NED value (the variations of the energy), the more measurements will be necessary for an attack to succeed [5]. Moreover, NSD is used in order to take into account the normal distribution of the energy due to the different input patterns. However, these two metrics tend to overestimate the security gain achieved by logic styles generating a high power consumption due to the normalization by either $max(E)$ or $\mu(E)$. Consequently, NED and NSD cannot be considered reliable metrics [1, 51]. Also, it was recalled

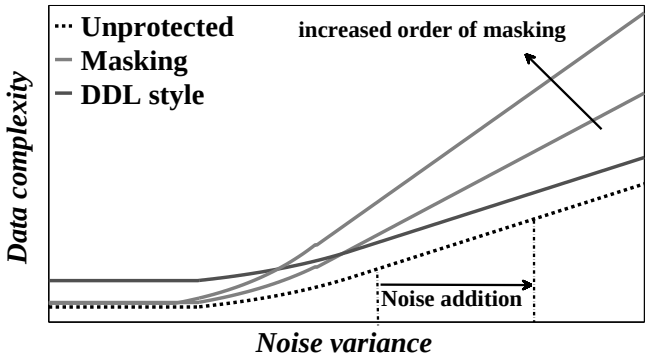


Fig. 4.1.: Data complexity in function of the noise for different types of countermeasures, namely masking with different orders, noise addition and using DDL styles.

in [51] that the standard deviation is related to the result of a CPA attack in an optimal statistical analysis [17, 52] which further justifies the importance of the standard deviations and not the normalized deviations.

Another popular approach to assess the efficiency of a DDL style is to mount a real attack at the measurement level (for example, [53]) or at the simulation level (for example, [33]). However, this approach is attack dependent, whereas the metric should depend on the implementation only. Also, using the SNR as a metric sounds appealing, specially for DDL styles, because it shows how efficiently the DDL style reduces the variance of the signal with respect to the variance of the noise. Nevertheless, SNR does not quantify the security strength of the implementation because it deals with the signal and noise variances only.

4.3.4 The big picture: Comparison between the different types of countermeasures

In order to evaluate the security gain of any countermeasure and compare it to previously introduced ones, it requires the unification of the assessment framework. This includes the benchmark circuits that are suggested to be standard encryption algorithms, such as data encryption standard (DES), AES and elliptic curve cryptography (ECC) [53]. The authors of [53] also propose mounting actual attacks either at the simulation level, using adequate power simulation models and correct assumptions on the attack or at the measurement level. They also suggest the use of the number of measurements to disclosure (MTD) as an evaluation metric. This guarantees a trustworthy comparison between countermeasures not only from a security point of view, but also from the performance perspective (power consumption, execution time, ... *etc.*).

In this section we show the general trend of the three aforementioned categories of countermeasures against side-channel attacks and specifically power

analysis attacks. In Fig. 4.1 we plot the data complexity expressed in MTD resulting from an arbitrary side-channel attack mounted against the different countermeasures as well as an unprotected device versus the noise variance. As explained earlier, this choice of plot is justified by the need to study the security gain of a countermeasure with respect to the noise because generally these countermeasures are known to be ineffective if the noise level is too low. For the masking scheme, Fig. 4.1 illustrates the exponential increase in data complexity with the order of masking only if the level of noise is high enough, otherwise the data complexity is sufficiently low for all orders of masking to mount a successful attack. As for the noise addition countermeasure, Fig. 4.1 manifests its impact as a linear increase in the data complexity. Finally, the DDL styles feature an increase in the data complexity by the addition of a constant compared to the unprotected device. It remains an open question to determine the value of this constant and whether it is significant or not in order to endure the high cost of implementing the DDL style.

We try to answer this question by studying the security of DDSLL as a representative of DDL styles in Section 4.6. DDSLL is particularly interesting because of its low performance overhead as detailed in Chapter 2. Nevertheless, it is still a full-custom design that requires more design efforts and time-to-market than semi-custom DDL styles.

With respect to an unprotected implementation:

1. **Masking increases the security exponentially (based on the order of the mask).**
2. **Noise addition increases the security linearly.**
3. **DDL styles add a constant.**

It is important to quantify the value of this constant to evaluate the efficiency of the DDSLL style.

4.4 SECURITY TOOLS AND METRICS

In this section we explain the tools and metrics we used in the security analysis of the DDSLL and static CMOS S-boxes. These tools and metrics can be used by both the circuit designer and the adversary to assess the quality of the countermeasure he adopts and to mount a successful attack against the cryptographic device, respectively. The following subsections represent the framework of our study where we start by a visual inspection of the leakage traces and a basic standard deviation analysis. Then, we follow the framework of [10] which is divided into two categories, a (worst-case) information theoretic analysis and a security analysis.

4.4.1 Notations

Using the terminology described in Section 4.2.4, the model of the leakage function denoted by $L(\ ,\ ,\ )$ contains either random variable arguments or fixed arguments. Since the target of our study is the assessment of AES S-boxes under different conditions, therefore, the random variable X which refers to the processed data, will represent the input value of the S-box under investigation. As an example, $L(x, c, N)$ is a random variable representing the noisy traces corresponding to input x and die c. We also denote the t^{th} time sample in a leakage trace as $L_t(\ ,\ ,\ )$.

In the measurement environment, it is useful to consider noise-free mean traces:

$$\overline{L_t^{meas}}(X, C) = \underset{n}{\mathbf{E}}\, L(X, C, n) \tag{4.8}$$

where $\mathbf{E}$ denotes the mean operator. Now in the simulation environment, the provided traces are noise-free by default. In this case, and in order to analyze the impact of noise on the security of the S-boxes, we added a Gaussian noise (the Gaussian assumption is considered a good choice from a side-channel point of view because it renders almost the same results when it is used with the simulated traces, as the actual measurements) in the evaluating tools. As a result, the simulated leakage function model is given by:

$$L_t^1(X, C, N) = L_t^{sim}(X, C) + N \tag{4.9}$$

By contrast, when the actual power traces are considered, the noise is directly present in the obtained measurements from the oscilloscope and the corresponding leakage function model is given by:

$$L_t^2(X, C, N) = L_t^{meas}(X, C, N) \tag{4.10}$$

However, we also need to assess the security of the measured S-boxes under the impact of noise. Therefore, it is useful to use the noise-free mean traces in Equation 4.8 to achieve a hybrid leakage function model, where the noise is represented by the additive Gaussian noise:

$$L_t^3(X, C, N) = \overline{L_t^{meas}}(X, C) + N \tag{4.11}$$

4.4.2 Leakage traces and standard deviation analysis

The first step a designer (adversary) takes in order to assess the security of (perform an attack on) an implementation is to produce the power traces corresponding to all possible input transitions. Therefore, it makes sense to observe and analyze these traces with hope of gaining some insight about the qualitative information regarding the security of an implementation. Of course to be able to benefit from such a preliminary analysis, the target implementation has to be compared to a reference one. For assessing the security of the DDSLL S-box, we

analyze its traces and compare it to those of the static CMOS S-box operating in the same conditions. For evaluating the impact of variability on the static CMOS S-box, we capture power traces at 0.5 V supply and compare it to those operating under a 1.2 V supply. In any case, the measured power traces are practically misaligned, unlike in the simulation environment where the traces are always aligned. The main cause of misalignment is directly related with the way the actual power traces are captured. In measurements, the power traces are recorded by an oscilloscope via a triggering signal that initiates the capture of traces every clock cycle. In the case of the static CMOS S-box, there is no clock signal, therefore one of the input signals of the S-box is used as a trigger signal. As for the DDSLL S-box, its clock signal is directly used as a trigger. We used the NI 6552 signal generator to produce the input signals and the clock of the S-boxes. In case the clock signal is used for triggering the oscilloscope, there is always a certain amount of jitter that is present and it causes the misalignment of the captured traces. In case an input signal is used for triggering the oscilloscope, it is not always synchronized to the clock signal even if the clock had minimum jitter. This is also explained in [1]. Now, producing power traces for all possible input transitions ($2^8 \times 2^8$) is very time and memory consuming. As a result and in order to have a sufficient number of traces at the same time, we consider 256 transitions between 0 and an arbitrary S-box input for both static CMOS and DDSLL styles.

4.4.3 Information theoretic analysis

The need for an information theoretic analysis in the evaluation and comparison of different implementations was explained in [10]. It is mainly motivated by the lack of concrete unified evaluation techniques in the literature. As described in [10], the present techniques used for comparing the security against side-channel attacks of an implementation to another unprotected one, are typically adversary (attack) dependent. This means they evaluate the security of the implementations against a certain type of adversary, for example the CPA attack [17]. Such techniques may result in the failure of an attack due to either the quality of an implementation or the inefficiency of an adversary, which can lead to a false sense of security. Therefore, claiming a security gain of an implementation should be based only upon the strength of an implementation in protecting the information and not on the type of adversary as well. Consequently, the information theoretic analysis in [10] aims at theoretically evaluating the security of an implementation using a worst-case scenario of an attack, namely the template attacks [11], summarized in Section 4.2.5.1, which represent the most powerful adversary. Hence, the information theoretic analysis is considered independent on the type of adversary.

Another important benefit of using the information theoretic analysis is that performing the actual attacks is very resource consuming (time and storage) for reaching a high security level [8]. This implies the need for such an analysis at the design phase of a countermeasure in order to evaluate the quality of an

implementation against a worst-case adversary before needing to prototype and mount tedious, time and memory consuming attacks.

It is worth noting that the information theoretic metric (mutual information [10]) was found to be related to the correlation coefficient of the CPA attack [54] which allows to link currently used metrics for the security analysis to evaluate standard DPA attacks (for example, the measurements to disclosure (MTD) metric as will be explained in Section 4.4.4), to the information theoretic metric. Also in [8], it was shown that the information theoretic metric is correlated with the MTD metric required to achieve a certain success rate. As a result, these meaningful relations confirm the relevance of using the information theoretic analysis as an important step while evaluating the security of an implementation during the design phase.

The information theoretic metric we use to evaluate the leakage of an implementation is the perceived information (PI) introduced in [16]:

$$PI(X;L) = H[X] - \sum_{x \in X} Pr[x] \sum_{l \in L} \hat{Pr}_{impl}[l|x] \; . \; log_2(\hat{Pr}_{model}[x|l]) \qquad (4.12)$$

This metric mainly captures the accuracy of the adversary's leakage model estimate (given by $\hat{Pr}_{model}[x|l]$) at predicting the leakage distribution estimate of an implementation (denoted as $\hat{Pr}_{impl}[l|x]$). These leakage estimates are used because in practice, the actual leakage distributions are unknown. The perceived information introduced in [16] is in fact equal to the mutual information metric from [10] that actually captures the worst-case information leakage only if the adversary's model is perfect. In some contexts this is not the case, i.e the adversary's model is not perfect. For example, in [16] the effect of inter chip variability is studied and the main conclusion was that the adversary's model and the actual chip's leakage distribution can strongly differ. For this purpose, the perceived information metric is considered as a generalization of the mutual information that allows the quantification of an adversary's degraded leakage model. Indeed, the perceived information could be either lower or equal to the mutual information.

The remaining terms in Equation 4.12 are the $H[X]$ term which is defined as the entropy of the implementation's input random variable X before any side-channel attack has been performed. The $Pr[x]$ term denotes the probability of each input hypothesis before taking into account the side-channel information.

Indeed, the information theoretic analysis is an important step in the evaluation of an implementation's security. However, it has to be completed with a security analysis in order to quantify the security gain of using a certain implementation in terms of the security metric; the MTD as discussed in the following section.

In the context of this chapter, the information theoretic analysis is used to study the security gain of DDSLL that presents a type of DDL styles as discussed in Section 4.6.2. Also, it is used to investigate the impact of variability on the

security of cryptographic devices (static CMOS S-box) designed using 65 nm CMOS technology as explained in Section 4.7.2.

4.4.4 Security analysis

The security analysis is considered the second complementary step of the evaluation framework proposed in [10]. It evaluates the strength of a particular adversary to mount a successful attack using the MTD metric at a required success rate. Also, the efficiency of the attack can be assessed using the security analysis. Indeed, an attack is considered the most efficient one if it achieves a minimum MTD at a certain success rate compared to other mounted attacks. Furthermore, it allows mounting actual attacks on either simulated or measured traces. As a result, the security analysis represents a useful tool to quantify the security gain of an implementation. The attacks considered in this study vary between profiled template attacks [11] (see Section 4.2.5.1) and non-profiled attacks (see Section 4.2.5.2 that include standard DPA attack [2], CPA attack [17] and "on-the-fly" stochastic attacks [12, 14].

In this context, we try to quantify the security enhancement achieved by the use of the DDSLL style compared to static CMOS as detailed in Section 4.6.3. We also employ the security analysis to quantify the degradation of mounting successful attacks in the presence of variability (Section 4.7.3).

4.5 SIMULATION AND MEASUREMENT SETUPS

The analysis provided in this chapter is based on simulated and measured power traces of full-custom designed AES S-boxes using static CMOS and DDSLL styles implemented on a LP 65 nm CMOS technology as described in Chapter 2. Since only the S-boxes are implemented without being xored with the key, therefore an additional post-processing step is needed to mount the attacks. That is a key byte K_s is chosen and xored with a set of plaintext bytes P, where each plaintext byte corresponds to a power trace L. During an attack, L is chosen such that it corresponds to the output of the S-box $S(P\ xor\ K_s)$.

Measurements were performed on 20 prototype dies where the voltage drop on a resistor (1kΩ) introduced in the path of the power supply of the measured S-box is captured using a differential probe. We used a high sampling rate oscilloscope (1 Gsample/second), while running the chips at 2 MHz (motivated by interface constraints of our prototype board) to store the voltage drop samples. Further processing using Matlab was done in order to convert the voltage drop sample values into power and be able to prepare the measured power traces. On the other hand, simulated power traces are obtained using post-layout SPICE simulations. We directly probed the current flowing into the supply voltage without adding a resistor in its path. Also for simplicity, measurement specificities, such as the different capacitances and inductances of the chip's package, the wire bonds, the ZIF socket, the PCB traces and the cables used, are not included in the simulation environment. While assessing the impact of variability on static

CMOS, Monte-Carlo simulations are done using SPICE models for 100 runs, where only with-in die (WID) variability is considered. The reason behind not considering the die-to-die (D2D) process variations in simulations is that measured dies are implemented on the same wafer, therefore they do not experience wafer-to-wafer (W2W) and lot-to-lot (L2L) variations that are incorporated in the global process variations simulation environment and hence would magnify the impact of D2D process variations. The test-bench of the S-boxes features buffered inputs and outputs without registers and the DDSLL S-box has an additional buffered clock[2]. Each S-box is provided with its own V_{DD} source and the input / output buffers are fed from a different power supply. This test-bench is the same for the simulation and measurement environments. Both simulations and measurements are done at ambient temperature using a nominal supply voltage of 1.2 V and an additional V_{DD} of 0.5 V which is necessary in the case of static CMOS to study the effect of variability at low supply voltages.

4.6 ASSESSING THE SECURITY OF THE DDSLL STYLE

In this section, we analyze the security of the DDSLL style, as a potential DDL candidate because of its hardware efficiency in terms of power consumption and chip area compared to static CMOS that is considered the unprotected implementation in this case. We follow the framework discussed in Section 4.4 by first, visually inspecting the power traces, and then apply the information theoretic and security analyses [10].

4.6.1 Leakage traces and standard deviation analysis

In Section 4.4.2 we mention that in order to generate the power traces, we use 256 transitions between 0 and an arbitrary S-box input instead of all input combinations ($2^8 \times 2^8$) for both static CMOS and DDSLL S-boxes (to save time and memory). Although this choice of input transitions is theoretically not mandatory for DDSLL because it is a precharge logic, it is essential in order to have a fair comparison between both logic styles. In practice, the power trace of the DDSLL style is divided into a precharge phase and an evaluation phase, where the precharge phase of the trace is slightly dependent, in an indirect way, on the input transition. This slight dependency is the result of the DDSLL being a low-swing logic. That is, the evaluated discharged output node of a gate is kept floating at V_{DD}-ΔV, where ΔV is the voltage swing that varies from one input transition to the other because of the imbalances of the intrinsic capacitance of the DPDNs (assuming the differential routing capacitances are perfectly matched) and also because of the WID variability (it is not studied in this context). Appendix C provides a brief explanation about how the voltage

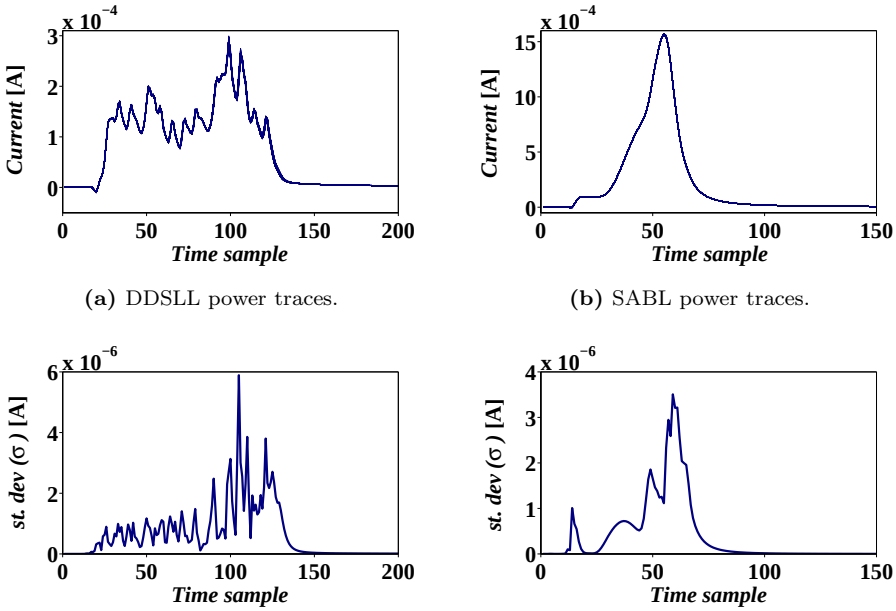
[2]For the sake of perfectly aligning the inputs of the static CMOS S-box, on-chip registers at the input of the S-box would avoid potential skew from the measurement setup, the PCB, the wire bond, ... etc.

swing depends on the input transition. Consequently, the precharge power is dependent on the value of ΔV of the previous evaluation phase, i.e on the amount of resident charge. This dependency is demonstrated in Fig. 4.2 (c) which shows the standard deviation over the inputs (σ_{in}) for the precharge phase of the DDSLL S-box. However, σ_{in} is not insignificant, its maximum value is about 6×10^{-6}.

An interesting comparison would be between the traces of the DDSLL S-box and that of SABL [5]. The SABL style was proposed to avoid having to balance the differential nodes of the gates as explained in Chapter 2. It uses an always on, key transistor that connects the output nodes of the DPDN of a gate. This transistor guarantees the discharge of all internal nodes during the evaluation phase such that when the next precharge phase arrives, all the discharged nodes will be precharged. Hence, SABL theoretically avoids having a varying resident charge during each precharge phase (the same reasoning applies for the evaluation phase), i.e. its power traces are presumably independent on the input transition. As a result, it is considered a good candidate from the security point of view, at least theoretically. It is shown in Fig. 4.2 (d) that the maximum standard deviation over the inputs during the precharge phase is about 3.5×10^{-6} which is similar to that of DDSLL. The main reason behind this similarity is that SABL aims at minimizing the ratio σ_{in}/μ, where μ is the mean over the inputs of the instantaneous power consumption which is $5\times$ that of DDSLL, whereas DDSLL targets an overall reduced instantaneous power consumption (both μ and σ_{in}). The main drawback of SABL is its relatively high total power consumption (about $2\times$ that of static CMOS) as shown in Fig. 2.24 in Chapter 2. Consequently, it is worth comparing both logic styles in terms of security as was done in Chapter 2 with respect to performance.

On the other hand, Fig. 4.3 shows the simulated power traces of the different input transitions (from 0 to an arbitrary state) and the standard deviation for the evaluation phase of the DDSLL S-box and those of the static CMOS S-box. It is clear from the standard deviations of the precharge and evaluation phases of the DDSLL S-box in Fig. 4.2 (c) and Fig. 4.3 (c), respectively that the information is mainly leaked during the evaluation phase since its maximum standard deviation over the inputs is $6\times$ that during the precharge phase. Therefore, only the evaluation part of the power traces will be considered throughout the rest of this chapter. It can also be seen from Fig. 4.3 that the maximum standard deviation for the static CMOS is about $4\times$ larger than that of DDSLL. This result gives an insight of the potential security gain achieved by using the DDSLL style.

To complete the comparison between DDSLL and SABL, we extend the analysis to the evaluation phase. The simulated traces and standard deviation over the inputs of the SABL S-box during evaluation are illustrated in Fig. 4.4. Comparing these figures to those of DDSLL in Fig. 4.3, it can be seen that both the DDSLL and the SABL S-boxes feature significant time shifts, specially near the end of the evaluation phase and minor amplitude shifts as well, although it is more amplified in the case of DDSLL. As for the standard deviation, it is almost



(a) DDSLL power traces.

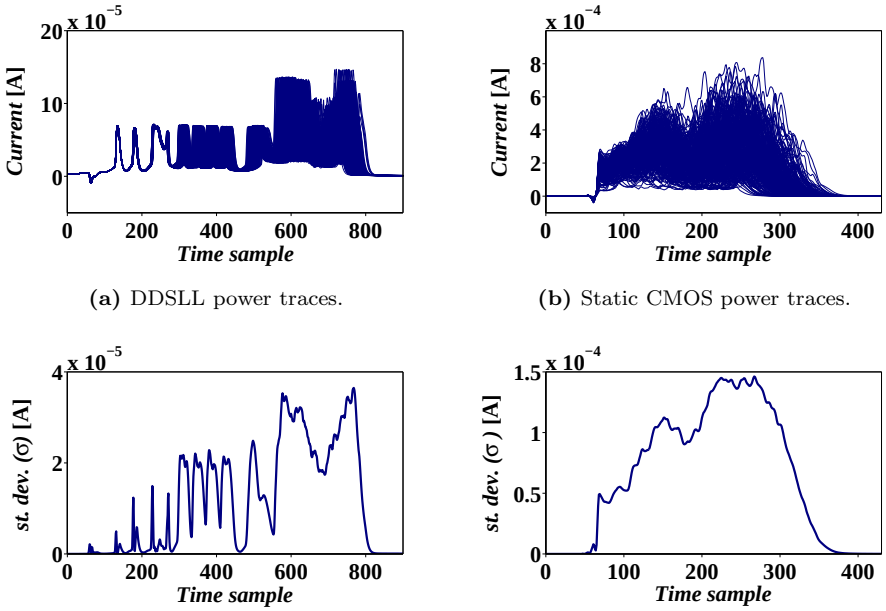
(b) SABL power traces.

(c) DDSLL standard deviation over the inputs.

(d) SABL standard deviation over the inputs.

Fig. 4.2.: Simulated power traces and their corresponding standard deviation over the inputs of the DDSLL and SABL S-boxes during the precharge phase (inputs transition from an arbitrary state to 0).

the same for both logic styles and mainly due to the existence of amplitude and time shifts in the traces. It is worth mentioning that in the case of DDSLL, the routing parasitic capacitances are taken into account, while for SABL, only gate capacitances are accounted for. Nevertheless, it is expected that if routing parasitic capacitances are added to the SABL S-box, then similar amplified time and amplitude shifts to that of DDSLL would be seen. These observations, lead to a rather important conclusion. That is, although the design of SABL theoretically guarantees the independency of the power traces on the the input transitions, practically, data dependent time shifts are clearly seen in the power traces of SABL that lead to a similar standard deviation to that of DDSLL. Although this is contradictory to the results obtained in [5], it can be attributed to the scaling of technology. In [5], the authors implemented SABL using a 0.18 μm CMOS technology, whereas we use the 65 nm CMOS technology to design the S-boxes. Indeed, technology scaling implies smaller capacitances and more sensitivity to small capacitance imbalance. Also, it was recalled in [51] that the standard deviation is related to the result of a CPA attack in an optimal statistical analysis [17, 52] which justifies the need to observe these standard deviations as a preliminary step. Nevertheless, the approach of SABL achieves a standard



(a) DDSLL power traces.

(b) Static CMOS power traces.

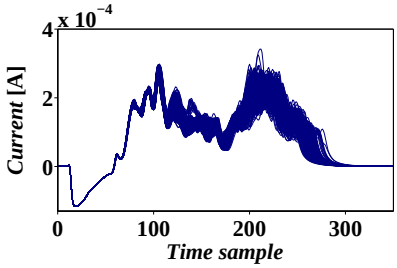
(c) DDSLL standard deviation over the inputs.

(d) Static CMOS standard deviation over the inputs.

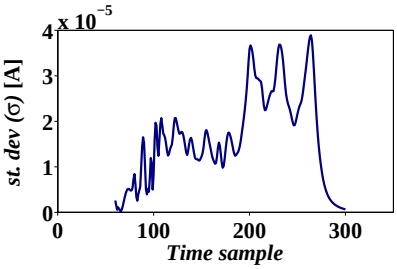
Fig. 4.3.: Simulated power traces and their corresponding standard deviation over the inputs of the DDSLL S-box during the evaluation phase and the static CMOS S-box (inputs transition from 0 to an arbitrary state).

deviation over the inputs that is almost 10× less than the mean of the instantaneous current consumption as can be observed from the different y-axes of the current and the standard deviation in Fig. 4.4. On the other hand, DDSLL features a standard deviation over the inputs $\sim 4\times$ less than the mean of the instantaneous current consumption (Fig. 4.3 (a) and Fig. 4.3 (c)). Clearly, the SABL approach is more effective in reducing the standard deviation over the inputs although the absolute value is comparable to that of DDSLL.

Now to reinforce the result obtained from the simulated traces, measured power traces and standard deviations over the input of both DDSLL and static CMOS S-boxes are plotted in Fig. 4.5. The same conclusion is drawn from the measurement traces, which is, the DDSLL S-box features a decrease in the standard deviation over the inputs by a factor of 4.5 compared to static CMOS. Nevertheless, one can observe that the shapes of the measured power traces are significantly different from the simulated ones, in particular for the DDSLL case. We notice that the time shifts present in the simulated power traces of the DDSLL nearly diminish in actual measurements and instead amplitude shifts are clearly seen. Also, there is a reduction in the amplitudes of the measured power traces and standard deviations about 7× for both static CMOS and DDSLL



(a) SABL power traces.



(b) SABL standard deviation over the inputs.

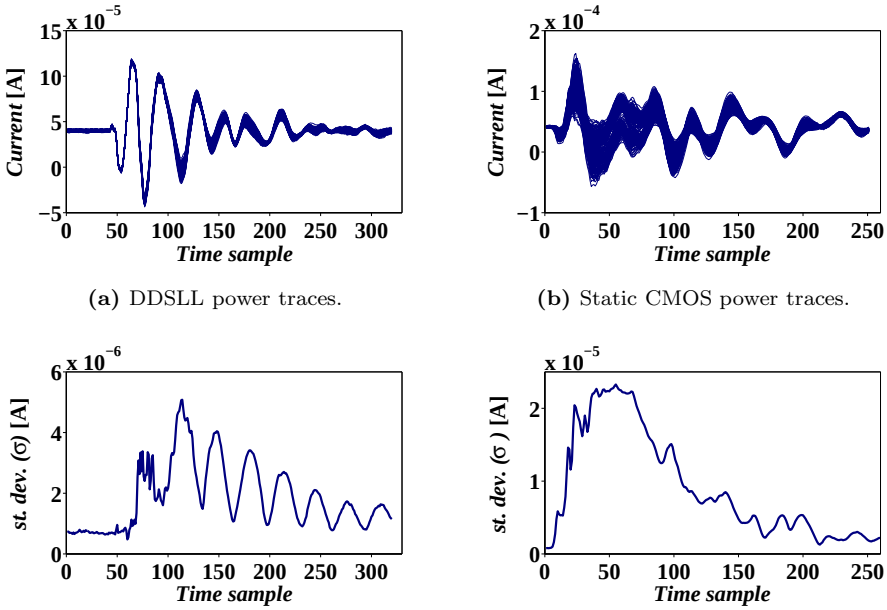
Fig. 4.4.: Simulated power traces and their corresponding standard deviation over the inputs of the SABL S-box during the evaluation phase (inputs transition from 0 to an arbitrary state).

S-boxes. One possible reason for such differences is the specificities of the measurement setup that are not included in the simulation environment.

However, observing these traces is an informal step that gives an intuition about, but does not confirm or quantify the security gain of using DDSLL over static CMOS. Accordingly, a thorough analysis is needed as explained in the following two sections.

4.6.2 Information theoretic analysis

In Section 4.3.4 an important question was raised; whether using a DDL style as a countermeasure against side-channel attack adds a significant constant to the data complexity of the unprotected implementation. In order to answer this question, we need to perform both an information theoretic analysis and a security analysis using template attacks for both the DDSLL and static CMOS S-boxes. Additionally, in the information theoretic analysis phase we also consider simulated leakages from a SABL and DyCML S-boxes as other DDL examples to show where the DDSLL style stands in terms of security. It is worth mentioning that in Chapter 2, DDSLL proved to be more power and area efficient compared to SABL and it achieves the same power with a much less delay penalty



(a) DDSLL power traces.

(b) Static CMOS power traces.

(c) DDSLL standard deviation over the inputs.

(d) Static CMOS standard deviation over the inputs.

Fig. 4.5.: Measured power traces and their corresponding standard deviation over the inputs of the DDSLL S-box during the evaluation phase and the static CMOS S-box (inputs transition from 0 to an arbitrary state).

with respect to DyCML. Therefore, it is interesting to investigate the security of DDSLL, compared to these alternative logic styles.

First, some hypothesis need to be defined [9]:

H1. The measurement noise is assumed to be normally distributed.

H2. Only 256 transitions (between 0 and an arbitrary S-box input) are considered for both S-boxes (among the $2^8 \times 2^8$ possible ones) as formerly indicated in Section 4.6.1.

H3. Only the leakages of the evaluation phase for the DDSLL S-box are considered, because they exhibited the largest standard deviations over the input as shown in Section 4.6.1.

H4. All our analyses are univariate, i.e. they consider leakage samples one by one. This choice was motivated by the goal to compare the amount and nature of the leakage function observed for the different samples.

In order to assess the available leakages, the information theoretic metric (perceived information) described in Equation 4.12 is used. The three probability distributions are computed as follows. First, $Pr[x]$ which denotes the probability

of each input hypothesis before taking into account the side-channel information, is considered uniform ($Pr[x] = \frac{1}{256}$). Next, $\hat{Pr}_{impl}[l|x]$ is the estimated conditional probability of observing a leakage l given an input x. We consider two cases: either real measurements with simulated noise in which case we have $\hat{Pr}_{impl}[l|x]$ computed from the leakage probability density functions, or real measurements with real noise, in which case this distribution is sampled from the actual chip, i.e. we use $\sum_{x \in X} Pr[x] \sum_{l \in L} \hat{Pr}_{impl}[l|x]=1\frac{1}{q}$, with q the number of traces measured [9]. Finally, $\hat{Pr}_{model}[x|l]$ is the conditional probability of the input given the leakages, derived using the adversary's model. This probability is computed from the template distributions estimated during the profiling. Both for the DDSLL and static CMOS S-boxes, our estimations are based on sets of 100 measurements for each of the input events x, both for the profiling and the attack phases.

We started the analysis by investigating the time sample that maximized the perceived information, against a worst-case template adversary, for the different implementations. Interestingly, the time sample corresponding to the maximum standard deviation between the different input transitions in the simulations, gives the maximum perceived information. Also, it was shown in [9] that the template-based perceived information calculated for all the time samples of the measured traces follows the same trend of the standard deviation curves illustrated in Fig. 4.5. Indeed, this trend was also confirmed by the analysis of standard DPA attacks in [54]. As a result, an informal yet interesting conclusion can be made; reducing the standard deviation over the inputs simply implies a reduction in the perceived information, against a worst-case template adversary. Consequently, the designer can set his primary goal to reduce the standard deviation over the inputs using advanced techniques, among which some are explained in Chapter 2.

After choosing the best time sample that maximizes the value of the perceived information metric, the next step is to study the different implementations using the perceived information curves for different noise levels. Fig. 4.6 (a) shows the perceived information computed with template attacks for the simulated traces of static CMOS, SABL, DDSLL and DyCML S-boxes along with those computed with stochastic attacks for static CMOS and DDSLL S-boxes only. First we address the curves obtained from the template attacks. These curves clearly illustrate the information leakage reduction obtained using the three different DDL styles over static CMOS against a worst-case template adversary. However, differences between the DDL styles can be observed. The DyCML S-box features the minimum perceived information, whereas implementing the S-box using SABL results in a maximum perceived information among the three DDL styles. Meanwhile, the DDSLL S-box presents the middle curve between SABL and DyCML S-boxes. Again, these results fit well with the standard deviation over the inputs (σ_{in}), where SABL leads to the largest value of σ_{in}, followed by DDSLL, whereas DyCML shows the lowest value of σ_{in}. This observation is simply explained by the relation between σ_{in} and the instantaneous power con-

sumption; the lower the power consumption, the lower the standard deviation. It is seen in Chapter 2 that the maximum instantaneous power among the three DDL styles is consumed by SABL, followed by DDSLL and finally DyCML. It is worth reminding that the DDSLL S-box addressed by these analyses is the one that uses the compact form of the DPDNs that represent the complex functions. Similar information theoretic analysis is performed for the balanced version of the DDSLL S-box and the PI along the whole range of σ_{noise} is found to be nearly identical to the one of the compact version. This result aligns well with the conclusion obtained in Chapter 2, which is: the maximum standard deviation over the inputs of the balanced DDSLL S-box is similar to that of the compact version; thus there is no security improvement. Again, these observations reinforce the significance of evaluating the standard deviation over the inputs as an informal step during the design phase to gain a qualitative intuition about the security strength of an implementation when compared to an unprotected one.

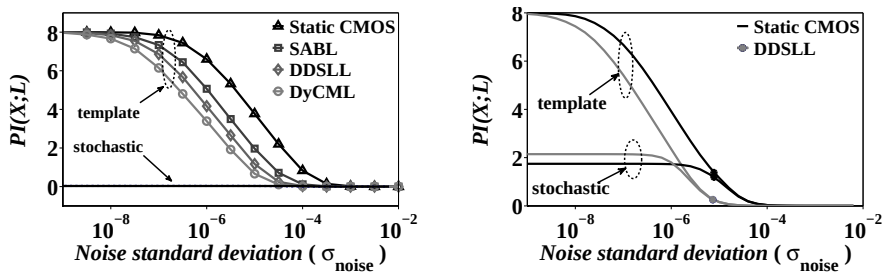
As for the measurement results, the perceived information computed with template and stochastic attacks for the traces of the static CMOS and DDSLL S-boxes are plotted against the simulated noise standard deviation in Fig. 4.6 (b). Also, the real measurement noise values are represented by the stars in the figure. Compared to the perceived information curves of the simulated traces calculated with the template attacks, we observe the following:

1. Both simulations and actual measurements of the DDSLL S-box show reduced information leakage compared to the static CMOS S-box.

2. The amount of information gained using measurements is reduced for both S-boxes compared to simulations. This difference is the result of the various specificities of the measurement setup that are not included in the simulation environment. This observation was already noted from the differences in the power traces between simulations and measurements in Section 4.6.1.

3. Quantitatively, the information leakage reduction brought by the use of DDSLL seen in simulations (Fig. 4.6 (a)), is decreased in actual measurements. Again the specificities of the measurement setup play an important role. Consequently, an interesting study for further research would be to determine the cause behind the impact of these measurement specificities on static CMOS and DDSLL S-boxes. Nevertheless, such an observation implies the importance of performing the analysis using the actual measurements and not to rely on the simulation results for accurate data.

From the above mentioned observations, we conclude that the experimental data still shows a security improvement by implementing the S-box using the DDSLL style compared to static CMOS even at a 65 nm CMOS technology.



(a) Simulation results. **(b)** Measurement results.

Fig. 4.6.: Perceived information using the template (WC) and stochastic attacks in function of the noise for (a) simulated traces of static CMOS, SABL, DDSLL and DyCML S-boxes (stochastic attacks are available for static CMOS and DDSLL only) and (b) measured traces of static CMOS and DDSLL S-boxes. In (b), the PI with real noise is marked by stars (*).

All three DDL styles under investigation feature a security improvement with respect to static CMOS as seen from the perceived information curves. The relative PI reduction follows similar trend to the σ_{in} reduction, confirming the relationship between the standard deviation over the inputs and the perceived information metric.

Until now we have only considered the optimal adversary that uses template attacks in a worst-case scenario to compare the security of DDSLL against static CMOS. However, to take into account suboptimal adversaries as well, we perform "on-the-fly" stochastic attacks [14] summarized in Section 4.2.5.2. This choice is particularly appealing in assessing the non-linearity of the DDSLL style because the stochastic attack is a linear, regression-based approach that approximates the device's leakages with a linear function [12, 15]. Therefore, it evaluates the linearity of a leakage function by comparing the obtained estimated information leakage to the one acquired from the profiled template attack [9]. The main motivation behind studying the linearity of the leakages is to shed a light upon the difficulties that could encounter a suboptimal adversary while trying to exploit (model) such leakages with non-profiled attacks [13]. It is worth mentioning that according to [13], none of the non-profiled distinguishers so far can perform successful attacks when the leakage function becomes highly non-linear[3]. In this context, the information theoretic analysis is an appropriate tool for investigating the linearity of the leakage functions and how it is accurately captured by the adversary's model.

[3]Non-profiled distinguishers rely on the assumption that the leakage function of an implementation is linear (the resulting leakage model best describes amplitude differences between states only). An example of current traces that feature a linear leakage function is given in Appendix E.

We analyze the perceived information curves calculated using the stochastic approach for the simulated and measured power traces of the static CMOS and DDSLL S-boxes illustrated in Fig. 4.6. We observe the following:

1. The leakage function of the simulated traces of the static CMOS S-box cannot be estimated by a linear stochastic model. Hence, the leakage function in this case is considered non-linear. This unexpected observation is explained by the nature of the power traces illustrated in Fig. 4.3 (b). There exists not only amplitude shifts, but also glitches and time shifts between the power traces of different input transitions. The amplitude shifts are easily explained by the charging activity of the nodes. In a static CMOS gate, when the input transitions from high to low or vice versa, the output node is charged or discharged, respectively, whereas if the input does not change states, the output remains unchanged. On the other hand, the glitches represent an inherent phenomenon in static CMOS associated to the delay skew between signals at a gate's input and to the logic depth [55]. Finally, the time shifts are mainly the result of signals traveling through different logic depths, which is directly related to the architecture of the S-box, such that the computational time for different input transitions vary accordingly. Now, the linearity of the leakage function is mainly captured in the amplitude of the traces for the different transitions (See Appendix E which shows am example of current traces that feature a linear leakage function). Therefore, having glitches and time shifts in the power traces changes the nature of the leakage function to a non-linear one. This important observation is also reported in [16], where stochastic models with low-degree bases could not capture the leakage function of a static CMOS S-box implemented using 65 nm CMOS technology. As a consequence, the former assumption that independent computations lead to independent leakage does not hold anymore for advanced technologies [16]. That is, linear leakage models that only depend on the input/output bits of an S-box are not accurate because of the presence of data dependent glitches.

2. For the DDSLL S-box, the linear stochastic model fails to estimate the leakage function of the simulated traces as discussed in [9, 13], similar to the case of simulated traces of static CMOS. Nevertheless, the simulated power traces of DDSLL shown in Fig. 4.3 (a), feature rather limited amplitude shifts and noticeable time shifts whereas there are no glitches present. The same reasoning performed for static CMOS is valid in the case of DDSLL, only here the time shifts are solely responsible for the non-linearity of the leakage function. Although glitches were responsible for the leakage dependencies on inter-computations within a static CMOS circuit [16], DDSLL does not suffer from glitches, yet it features a non-linear leakage function. We suggest that the existing time shifts in the traces are responsible, since as mentioned before, the linearity of the leakage function is mainly observed in the amplitude of the traces for the different transitions. Now, a reasonable question would be: why DDSLL exhibits these amplitude and

time shifts, despite being a DDL style? That is, although the charging / discharging of the nodes are theoretically independent of the input transitions, we still see these shifts. The answer is simply because in practice, all DDL styles suffer from some kind of imbalances between the differential nodes and in the case of DDSLL (and all DDLs that constitute of complex DPDNs), it features imbalances in the complex DPDNs as well. Ideally, if there were no amplitude or time shifts in the power traces of the DDSLL S-box, the leakage function would hypothetically still be non-linear.

3. Interestingly, actual measurements show that the leakage functions of both the static CMOS and DDSLL S-boxes are reasonably linear such that they can be estimated by a linear stochastic model. This observation contradicts the one obtained from the simulation results. Nevertheless, it can be related to the specificities of the measurement setup that are not included in the simulation environment. Namely, the resistor in the path of the supply voltage to monitor the traces, the different capacitances and inductances of the chip's package, the wire bonds, the ZIF socket, the PCB traces and the cables used. Hence, these measurement artifacts contribute to the low-pass filtering that leads to smoothing the shape of the traces by filtering out the high frequencies. This is true for both static CMOS and DDSLL S-boxes as can be observed by the different shapes of the power traces between simulations and measurements in Fig. 4.3 (a), Fig. 4.3 (b), Fig. 4.5 (a) and Fig. 4.5 (b). We try to obtain close results to the measurements by simulating the DDSLL S-box with an inserted resistor of 1kΩ in the V_{DD}'s path. The resulting perceived information curves calculated using template and stochastic attacks are shown in Fig. 4.7. Evaluating the PI curves calculated using the stochastic approach, we see that the insertion of the resistor got the PI curve of the simulated traces closer to the measurements. That is, the leakage function of the simulated DDSLL S-box is not non-linear anymore and it can be modeled by a linear stochastic approach, even if the maximum retrieved information is only 0.4 bit as illustrated in the figure.

4. Finally, using a linear stochastic model in the case of the actual measurements can lead to a false sense of security in the case of static CMOS with respect to DDSLL at low noise levels. This observation is explained by the inherent inaccuracy of the stochastic model in the case of using an incomplete linear basis [16]. By extending the basis (e.g. using quadratic, cubic, ... terms as well), the stochastic model approaches the profiled templates where the DDSLL clearly preserves its security improvement over static CMOS at low noise levels.

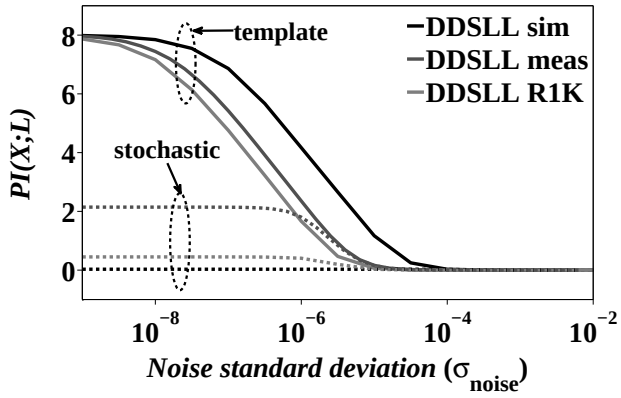


Fig. 4.7.: Perceived information using the template (WC) and stochastic attacks in function of the noise for simulated traces (sim), simulated traces with 1kΩ in the V_{DD} path (R1K) and measured traces (mess) of the DDSLL S-box.

Measurement artifacts contribute to the low-pass filtering of the traces of both static CMOS and DDSLL S-boxes. It is important to include models of the measurement specificities in the simulation environment to be able to draw concrete conclusions at the design phase.

These observations are rather disappointing in the sense that DDSLL was expected to have a non-linear leakage function even in actual measurements. Yet, it features a non-linear leakage function only in simulations as also shown in [9, 13], similar to static CMOS, due to the presence of time shifts in the traces that are smoothed and filtered out in actual measurements leading to completely linear leakage functions. The main cause of such a disappointment is that having amplitude shifts, glitches and time shifts is inherent in the case of static CMOS, whereas the main goal of DDL styles in general is to eliminate such sources of information leakage. However, DDL styles have failed in this context in which they only limited these deficits and have not completely eliminated them.

As a result, equalizing the differential routes and balancing the complex DPDNs for all input transitions of the DDSLL S-box are essential requirements that would lead to the ideal situation where no amplitude or time shifts are observed. Generally, DDL styles try to achieve these requirements in various ways. In the case of DDSLL, since the layout of all the blocks and routes are done in a full-custom manner, special care was taken to balance the differential routes and symmetrize the DPDNs. However, the DPDNs of the DDSLL S-box are not fully balanced as discussed in Chapter 2. Another example is SABL [5]. The authors introduced a new technique such that all nodes are discharged during the evaluation phase (charged during the precharge phase), i.e. equalizing the capacitance to be discharged (charged) independent on the input transition. However in prac-

tice, there are still differences between the capacitances depending on the input transition which is why time shifts also exist in the simulated traces of SABL as shown in Fig. 4.4. In an ideal case where there is neither amplitude shifts nor time shifts in the traces, i.e. the traces for all input transitions are identical, the leakage function would be hypothetically non-linear for both simulated and measured traces, if no other artifacts contribute to the linearity of the leakages in actual measurements.

Furthermore, different time samples are studied in the case of simulated traces of the DDSLL S-box. More specifically, at the beginning (no time shifts observed), at the middle and at the end of the traces (maximum time shifts observed and maximum standard deviation over the input transitions). All studied time samples rendered the same conclusion; the linear stochastic model is incapable of estimating the leakage function of the DDSLL S-box due to its high non-linearity nature. This is particularly important for the time sample at the beginning of the traces where there is no time shifts. Accordingly, the disappearance of both amplitude and time shifts lead to a non-linear leakage function confirming the hypothesis obtained in the ideal case.

Finally, these results raise an important open question whether it is possible to develop a logic style that is free of amplitude and time shifts. More generally, can designers consider non-linearity as a design guideline while developing the secure logic style?

One way to preserve the non-linearity of the leakage function of the DDSLL S-box is to suppress all time and amplitude shifts (as they are eventually smoothed and transformed to amplitude shifts at the measurement level). Therefore, it is important to include non-linearity as a design guideline.

4.6.3 Security analysis

The main goal of the security analysis is to quantitatively answer the question raised in Section 4.3.4; whether using a DDL style as a countermeasure against side-channel attack adds a significant constant to the data complexity of the unprotected implementation. The focus here is the DDSLL as a representative of DDL styles, since it represents the best power - delay tradeoff as explained in Chapter 2 and at the same time maintains a noticeable security improvement over static CMOS as discussed in the previous section.

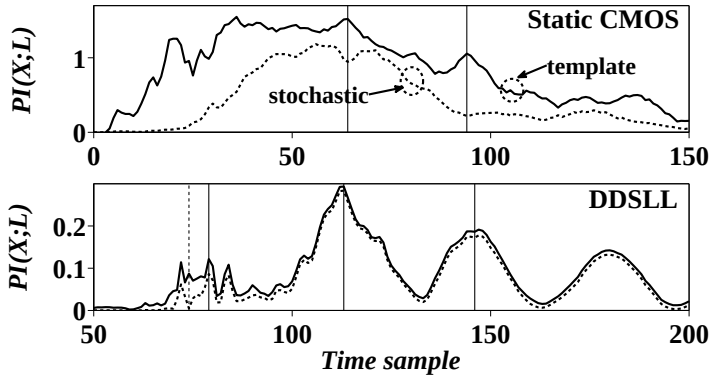
First, we start by choosing the appropriate time samples. Fig. 4.8 (a) shows the perceived information calculated using the template and "on-the-fly" stochastic attacks mounted against static CMOS and DDSLL S-boxes (measured power traces) for the different time samples. We can clearly see a reduction in the information extracted from the DDSLL S-box compared to the static CMOS one across the whole range of time samples for both templates and linear stochastic models. The template curves certainly provide the worst-case PI which clearly

illustrates the benefit of using profiled attacks. Indeed, the information extracted with templates is always higher than that of the "on-the-fly" stochastic model which uses a linear basis. However, some time samples feature accurately predicted information using "on-the-fly" linear stochastic models. We use the marked time samples (vertical lines) to mount template and "on-the-fly" stochastic attacks against the static CMOS and DDSLL S-boxes and plot the resulting success rates as a function of the data complexity measured in MTD in Fig. 4.8 (b). Clearly, template attacks provide the minimum MTD compared to the linear stochastic approach for both S-boxes. Also at the time sample that maximizes the PI of the template attacks for both S-boxes (for static CMOS: $t_s = 64$ and for DDSLL: $t_s = 113$), we can see one order of magnitude increase in the MTD of the DDSLL S-box compared to the static CMOS one computed from the template attacks. Since the MTD metric reveals how many measurements are needed to identify the correct key (with a certain confidence level), it is clear that an adversary who mounts template attacks against DDSLL will need about $10\times$ more measurements than static CMOS to recover the key[4]. This observation reinforces the correlation existing between the data complexity required to perform a successful template attack (security metric) and the perceived information computed with profiled templates (information theoretic metric).

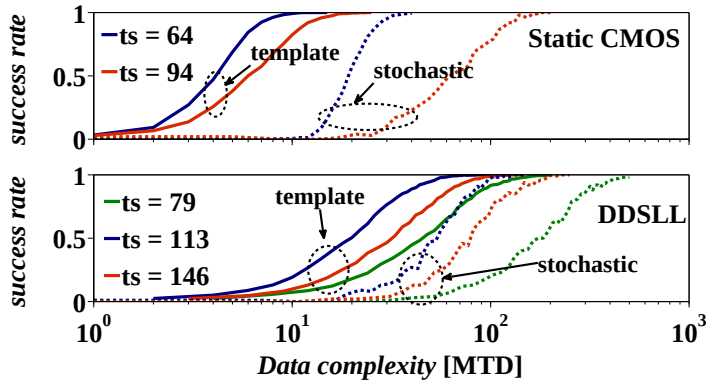
DDSLL features one order of magnitude of security improvement in terms of MTD with respect to static CMOS.

One can also conclude that for both S-boxes, there are samples that are sufficiently linear (i.e. strong linearly dependent on the S-box output bits) for being easily exploited with non-profiled attacks. To further manifest the significance of the linearity of the chosen sample in mounting a successful non-profiled attack, we plot in Fig. 4.9 the success rates of different non-profiled attacks mounted against both S-boxes using the most informative sample (highly linear) and the less informative one (less linear). We can see from Fig. 4.9 (a) and Fig. 4.9 (c) that for the highly linear sample taken from either S-box, non-profiled attacks (single-bit DPA and CPA with Hamming weight leakage model) can successfully retrieve the secret key with moderate data complexity similar to the "on-the-fly" stochastic approach. In the case of static CMOS, the single-bit DPA attack slightly outperforms the "on-the-fly" stochastic approach such that it requires less data complexity and it gets closer to the worst-case template curve. Whereas for DDSLL, the CPA attack with the Hamming weight model represents a good choice as it also features less MTD compared to the "on-the-fly" stochastic approach. As for the less linear sample of the static CMOS S-box, we can see from Fig. 4.9 (b) that the CPA attack with Hamming weight model fails to recover the key, while the single-bit DPA and "on-the-fly" stochastic attacks using a

[4]A security improvement of $10\times$ is not enough for a stand-alone countermeasure because static CMOS is usually really easy to attack as seen from Fig. 4.8 (b), where an adversary that mounts template attacks against static CMOS needs only 10 MTD to achieve 100 % success rate.



(a)



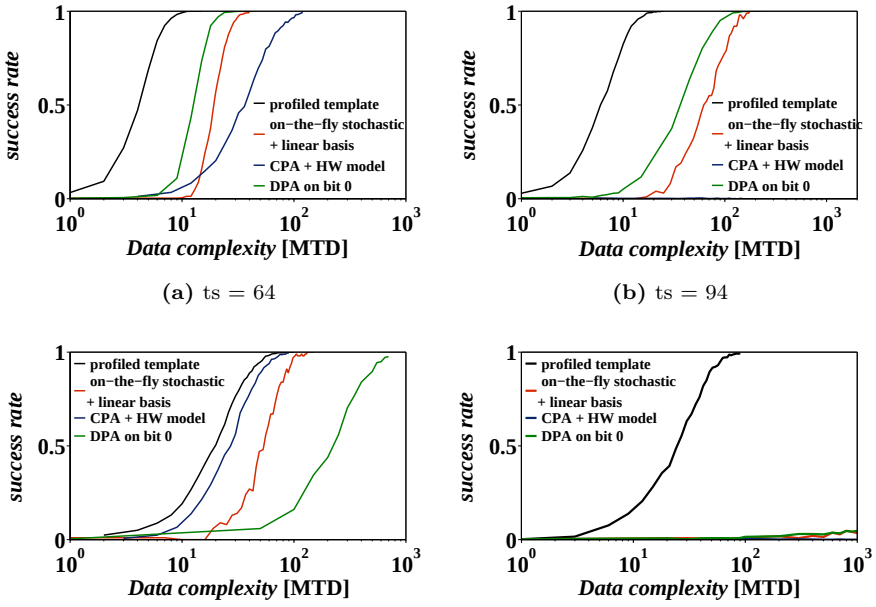
(b)

Fig. 4.8.: (a) perceived information computed using template and "on-the-fly" stochastic attacks against measured power traces of static CMOS and DDSLL S-boxes for different time samples, (b) success rates calculated for the marked time samples (vertical lines).

linear basis succeed. It is worth mentioning at this point that we could verify that the the S-box output bits having high weight in the stochastic models are the ones for which a single-bit DPA succeeds. This observation explains why the single-bit DPA attack succeeds, similar to the "on-the-fly" stochastic approach if a less linear sample is used, while the CPA attack does not. Anyway, we confirm the previous observation in [16] that the stochastic approach represents a good choice for dealing with leakages in advanced technology nodes that feature possible variability issues and less linearity. However, if the sample is completely non-linear as in the case of DDSLL in Fig. 4.9 (d), only profiled side-channel attacks succeed in retrieving the secret key.

These results confirm the conclusion drawn by the information theoretic analysis in Section 4.6.2 where the measured DDSLL S-box manifested linear leakages despite the fact that simulated power traces are highly non-linear. This was attributed to the measurement artifacts that were not taken into consideration during simulations. This important observation explains the discrepancy between simulated and measured leakages and reflects the importance of relying on measured leakages to obtain accurate results. Nevertheless, there are some samples that are highly non-linear in the measured traces of the DDSLL S-box. However, these non-linear leakage samples do not limit practical attacks since the existence of even a few linear leakage samples is sufficient to mount a successful non-profiled attack. Therefore, the same open question is raised; whether the designer can consider non-linearity as a design guideline while developing the secure logic style.

Now the answer to the question of whether using a DDL style (DDSLL) as a countermeasure against side-channel attack adds a significant constant to the data complexity of the unprotected implementation (static CMOS); is unfortunately no. The data complexity increase, brought by DDSLL, is limited and not



(a) ts = 64 **(b)** ts = 94

(c) ts = 113 **(d)** ts = 74

Fig. 4.9.: Template and non-profiled attacks against (a) the static CMOS S-box with a highly linear sample, (b) the DDSLL S-box with a highly linear sample,(c) the static CMOS S-box with a less linear sample, and (d) the DDSLL S-box with a highly non-linear sample.

sufficient (only one order of magnitude) to claim a large security gain compared to static CMOS. As a result, it is an interesting question to determine whether we can do better. That is, either develop a logic style that sufficiently increases the data complexity constant over static CMOS, or combine several countermeasures, e.g. DDL style that features masking as an additional countermeasure (See Appendix F for some examples of how masking can be implemented at the algorithm and gate levels). The first solution is still limited in the sense that it tries to maximize a constant gain in terms of data complexity. Anyway, developing more secure DDL styles can be achieved by either one or a combination of the following:

- Implementing new balancing techniques to overcome the current deficiency of the current techniques to fully balance the DPDNs.
- Reducing the standard deviation over the inputs by lowering the instantaneous power consumption.
- Applying the same method as SABL, that is; reduce the sensitivity of the DPDN to the imbalance of the differential nodes by charging and discharging the same total capacitances every clock cycle.

However, the second solution, which is combining the DDL style with a masking scheme, is particularly appealing, specially if DDSLL is the DDL candidate. Clearly, the power consumption reduction brought by the use of DDSLL compared to other DDL styles (whose power consumption is 2× that of static CMOS in the best case), as shown in Chapter 2, gives room for adding other countermeasures. This is of great importance to low-power secure applications that demand power-efficient, high-security solutions.

Additionally, using masking as an added countermeasure leads to an exponential increase in the data complexity as explained in Section 4.3.4. In [13], simulation results of a masked version of the DDSLL S-box implemented using 65 nm CMOS technology node features a data complexity improvement about two orders of magnitude with respect to the unmasked version of the same DDSLL S-box. However, no information is provided about the power performance of the masked version of DDSLL in [13]. Yet, it is a well known fact that masking is a power-hungry, low-performance solution [1], although it is considered as a good added countermeasure candidate to the DDSLL style from a security point of view. Therefore, combining countermeasures still poses a security - performance tradeoff.

Improving the security of an implementation can be achieved by:

- **Develop more secure DDL styles that are capable of increasing the MTD security constant.**
- **Combine DDSLL with a masking scheme which takes advantage of the power efficiency of DDSLL and the security efficiency of masking.**

4.7 ANALYZING THE EFFECT OF VARIABILITY ON THE STATIC CMOS LOGIC

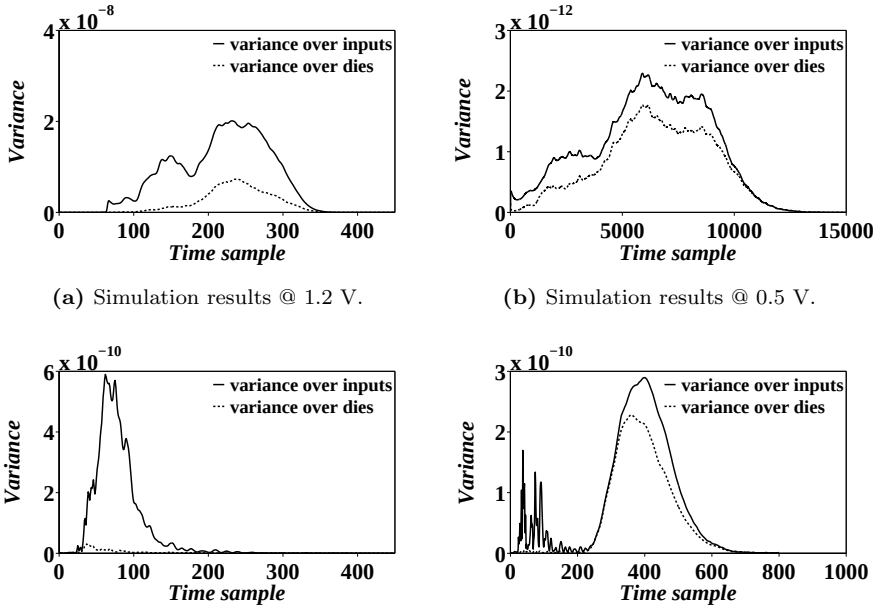
In the previous section, we analyzed the security of the DDSLL style as a potential countermeasure against side-channel attacks. We observed how the nonlinearity of the leakage function can be considered an obstacle for a suboptimal adversary. In this section, we tackle another obstacle which is the process variability. The effect of variability on power - delay performances and on functionality of digital circuits, namely, the S-box implemented using static CMOS and DDSLL styles, is studied in details in Chapter 3. However in this section, we focus on the effect of process variability on the security of digital circuits.

The scaling of the CMOS technology beyond 100 nm implies an increased variability concern [56–58] that can be spatially categorized as with-in-die (WID), die-to-die (D2D), wafer-to-wafer (W2W) and lot-to-lot (L2L). This implies that even circuits implemented on the same die (chip) can have different power performances and hence different leakage functions. Indeed, the dynamic power is usually the most studied power component in SCA because of its strong data dependency, although there has been efforts to exploit the input dependency of static power as well [59, 60], yet it is not as mature as the analysis using dynamic power. Meanwhile, process variations were reported to affect the dynamic energy in nanoscale devices due to data-dependent glitch-induced variations of the activity factor associated to delay skew and logic depth in complex combinatorial circuits [61]. This phenomenon is significantly magnified at low supply voltages, where large delay skews can be seen, leading to the generation of more pronounced random glitches. As a result, we mainly aim at studing how the process variability affects the security assessment of a static CMOS S-box implemented using 65 nm CMOS technology at nominal and low supply voltages.

It is worth reminding that the measurements of the static CMOS S-box are performed on 20 dies, thus experiencing both WID and D2D variabilities, whereas Monte-Carlo simulations are done using SPICE models for 100 runs, where only WID variability is considered. The reason why the D2D variability is not taken into account in the simulations is because the Monte-Carlo analysis spatially differentiates between random variations that affect each transistor independently (WID) and systematic variations that affects all transistors in the same way (D2D, W2W and L2L). Accordingly, Monte-Carlo simulations would have exaggerated the effect of D2D variability because it takes the W2W and L2L variabilities also into account. Both simulations and measurements are done using a nominal supply voltage of 1.2 V and an additional V_{DD} of 0.5 V to study the effect of variability at nominal and low supply voltages.

4.7.1 Leakage trace and standard deviation analysis

As a first informal step we observe the power traces and plot the variance over the different dies along with the variance over the different input transitions, in order to demonstrate the relevance of studying the process variability, at nominal



(a) Simulation results @ 1.2 V.

(b) Simulation results @ 0.5 V.

(c) Measurement results @ 1.2 V.

(d) Measurement results @ 0.5 V.

Fig. 4.10.: Variances of the power traces over the input transitions and dies. (a) simulation results at 1.2 V, (b) simulation results at 0.5 V, (c) measurement results at 1.2 V and (d) measurement results at 0.5 V.

V_{DD} and at 0.5 V as illustrated in Fig. 4.10. It can be seen that the variance over the dies is present at 1.2 V, at least in simulations. Moreover, it is magnified for the 0.5 V supply when process variability becomes more influential. However, the variance over the dies at 1.2 V is negligible for actual measurements first because at 1.2 V, the glitch-induced variations are less significant than at 0.5 V, second because measurements are done using 20 dies only, whereas in simulations we consider 100 Monte-Carlo runs. Consequently, one can speculate that having a variance over the dies close to that over the inputs makes the adversary's job more difficult whether he uses profiled or non-profiled attacks.

4.7.2 Information theoretic analysis

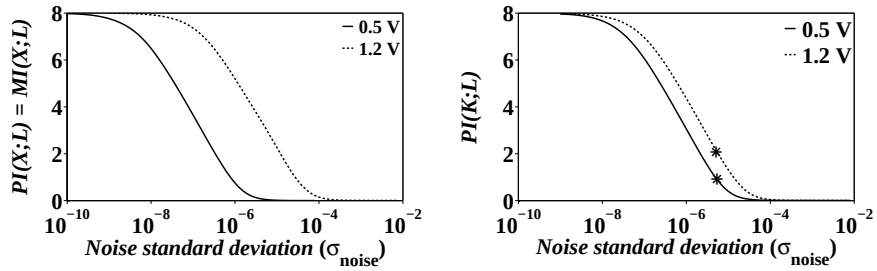
As seen in the previous section, process variability can pose a potential threat to side-channel attacks when it becomes in the same order of magnitude as the variance over the input transitions. Indeed, in a 65 nm CMOS technology and beyond, variability causes the same circuit implemented on different dies, or even with-in the same die, to have different leakage models. As a result, the goal of this section is to investigate how process variability affects the application of side-channel attacks from an information theoretic point of view. As detailed in

[10] and previously used in Section 4.6.2 to assess the security of the DDSLL, the information theoretic analysis allows to quantify the security of an implementation against an optimal adversary who can perfectly profile the leakage distribution. Due to the presence of process variations, the assumption of the optimal adversary may not hold anymore. Consequently, two different scenarios arise; the first is to profile and attack the same die, while the second and more realistic one is to profile n dies and attack a $n+1^{th}$ die. The first scenario represents the worst case in which the adversary is capable of attacking the same die he characterized during the offline phase. Therefore, this scenario does not consider the process variability and assumes that all dies features the same leakage model, which is valid for older technology nodes when process variations were not an issue. The second and more realistic scenario infers the effect of process variations in advanced technologies. In this case, the perceived information proposed in [16] is considered a good metric because it captures the degradation of the adversary's leakage estimate due to the process variations.

The results of the first scenario, which is profile and attack the same die, are shown in Fig. 4.11 for the static CMOS S-box operating at 1.2 V and 0.5 V. The information leakage of the simulated traces at 1.2 V fairly corresponds to the one of the measured traces, whereas at 0.5 V (variability is more pronounced), the simulated leaked information is quite less than that of the measurements. This can be explained by the deficiency of the simulation models that do not include the measurement specificities. Anyway, the results obtained from simulations are qualitatively sufficient to conclude that decreasing the supply voltage leads to a reduction in the information leakage (thanks to the reduction of the instantaneous power consumption and hence the instantaneous standard deviation over the inputs) at the expense of a larger delay penalty, but actual measurements are needed to quantify this reduction. Again, we can see that the average measurement results plus simulated noise (L_t^3) matches exactly the actual measurements with real noise (L_t^2). Therefore, we will only consider L_t^3 leakages as they provide similar behavior to the actual measurements while allowing the sweep of the noise levels.

The reduction of the supply voltage increases the security of the implementation against the template attacks due to the reduction of the instantaneous standard deviation over the inputs where speed is traded for better security (worst-case scenario - profiling and attacking the same die).

Now we move on to the more realistic scenario, where the adversary profiles and attacks different dies. Fig. 4.12 shows the information leakage for different situations analyzed using the template attacks mounted against the static CMOS S-box at 1.2 V supply. First, we plot as a reference the perceived information for the worst-case scenario, denoted by "1 vs 1" (same die), where the attack is mounted on the profiled die. In this case, the perceived information curve follows the same pattern as previous information theoretic analyzes; i.e, the perceived



(a) Simulation results. **(b)** Measurement results.

Fig. 4.11.: Perceived information (mutual information) using templates (WC) in function of the noise for the static CMOS S-box using (a) simulated traces ($L_t^1(X, N)$ averaged over 100 MC runs) and (b) measured traces ($L_t^3(X, N)$ averaged over 20 dies). In (b), the PI with real noise (corresponds to $L_t^2(X, N)$) is marked by stars (*).

information is positive for all noise levels. We also illustrate the actual attack scenario in which an adversary profiles and attacks different dies, defined as "1 vs 1" (different chips). For this situation, the adversary clearly looses information as the process variability implies the use of a degraded profile to attack a die. As a result, the perceived information has a negative value at low noise levels, meaning that the leakage is misinterpreted by the adversary's model. In order to improve the accuracy of the template attack, the adversary profiles n dies, where n varies from 5 to 15 dies, to attack a different die. It can be seen that characterizing a large set of dies allows the adversary to avoid having a false leakage model (where the perceived information is negative). However, the penalty of such an approach is a reduction in the information leakage as observed by the lower value of the perceived information for the "10 vs. 1" curve as an example, compared to the worst-case scenario. Consequently, this observation indicates that process variability causes the obtained profile, formed by characterizing a large set of dies, to be less accurate for any single die. Finally, there is a maximum number for the characterized dies, in our case it is 10 dies, after that the adversary does not gain more information. Similarly, using a 0.5 V supply instead of 1.2 V would render the same observations.

Extending the template attacks to infer the impact of process variations by profiling n dies to attack the $(n+1)^{th}$ die avoids the false leakage model at the expense of reduced accuracy for any single die.

Another important factor that has to be taken into account is the sample choice. We could see a huge difference in the contributions of the die variance and the input variance across the different time samples in Fig. 4.10 (d). We plot

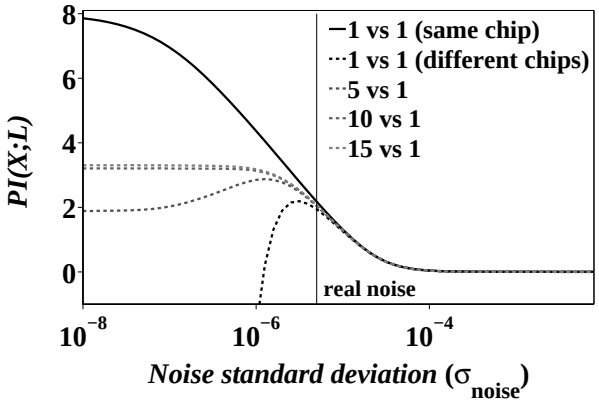


Fig. 4.12.: Perceived information using the template attacks, mounted on the static CMOS S-box, in function of the noise for various sets of learning dies at a supply voltage of 1.2 V.

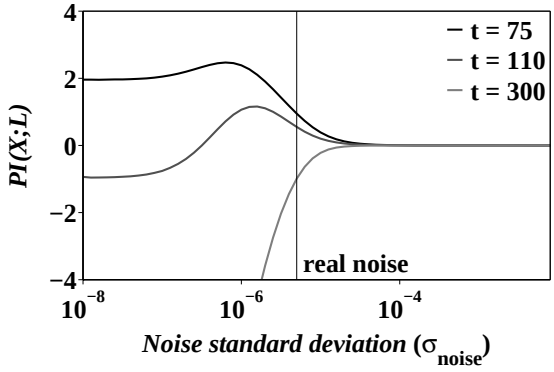


Fig. 4.13.: Perceived information using template attacks mounted on the static CMOS S-box in function of the noise for various time samples at a supply voltage of 0.5 V (5 vs 1).

the information leakage calculated using templates against the static CMOS S-box at 0.5 V in function of the noise for 3 distinct time samples in Fig. 4.13, where we consider the case of profiling 5 dies in order to attack 1 die denoted by "5 vs 1". It is clear that the amount of information is reduced as the input variance decreases (moving from time sample 75 to 110) and also as the die variance increases (moving from time sample 75 to 300).

As a result, modeling process variations with a straightforward extension of template attacks leads to significant information losses. This important conclusion stresses the need to develop new side-channel distinguishers, that can better cope with such situations. For example, the use of non-profiled distinguishers that

perform "on-the-fly" model estimations which is discussed in the next section. Nevertheless, the use of profiled attacks in such advanced technologies is not obsolete, it is actually equally important in the sense that it is used to determine the security limits of an implementation against the most powerful adversary (capable of profiling and attacking the same chip). The good news for the designer is that as the technology scales, this limit is stretched further away for actual adversaries.

4.7.3 Security analysis

In the previous section we discussed the effect of variability on the static CMOS S-box implemented on a 65 nm CMOS technology node using profiled attacks from the information theoretic perspective. In this section we tackle the effectiveness of the non-profiled adversary's model which usually depends on Hamming weights (or distances) for static CMOS in light of the increased process variations in advanced technologies. More specifically, we use the security analysis to determine the accuracy of such models for the 65 nm CMOS technology. For this purpose, we use the Pearson's correlation coefficient [17] since it is a classical DPA tool. We also perform "on-the-fly" stochastic attacks [14] in order to compare the "on-the-fly" model estimation to the fixed model (Hamming weight) used by the CPA attack.

However, we start first by employing the security analysis to quantify in terms of MTD at a 0.9 success rate, the different situations explored in Fig. 4.12 for template attacks mounted against the static CMOS S-box operating at 1.2 V in Fig. 4.14. The lower bound corresponds to the worst-case scenario where the adversary profiles and attacks the same die, whereas the upper bound reflects the actual situation where the adversary profiles and attacks different dies. In between the two boundaries, it is shown that the number of MTD is low once the adversary profiles a set of 4 dies to attack another die. Therefore, using a set of 10 dies renders similar results as can be noted from Fig. 4.12, considering noise levels above 10^{-6}, which is meaningful to take as a lower limit for actual noise levels. These results reinforce the conclusions deduced from the information theoretic analysis in the previous section. First, the worst-case scenario, which is profiling and attacking the same die, represents the lower limit which the designer should consider while evaluating an implementation. Second, extending template attacks to infer process variations, by profiling a large set of dies to attack another die, causes a significant reduction in data complexity.

Now, we address the non-profiled attacks. In Fig. 4.15 (a) we plot the success rates of the CPA attacks, using a Hamming weight model, mounted on each of the 20 measured dies. In this figure, only two dies were able to reach a 100 % success rate. Clearly, the Hamming weight leakage model does not lead to successful attacks on all dies due to the inadequacy of such a fixed model as previously demonstrated in [13]. This important observation indicates that CPA attacks with a Hamming weight model are no longer suitable for advanced technologies featuring high process variations such as the 65 nm CMOS technology [16].

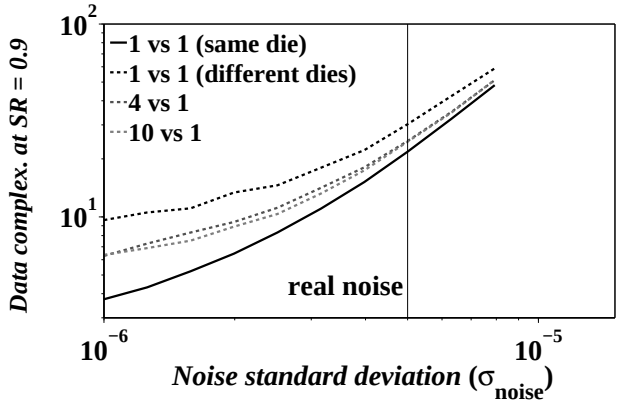


Fig. 4.14.: Data complexity to reach a 0.9 success rate using the template attacks, mounted on the static CMOS S-box, in function of the noise at a supply voltage of 1.2 V.

Furthermore, we study the second type of non-profiled attacks which is the "on-the-fly" stochastic approach with a linear basis. Fig. 4.15 (b) illustrates the average success rates (over the 20 dies) of both the stochastic and CPA attacks. The average success rate of the correlation attack is below 30 % which is a direct result of the failure of the Hamming weight model to estimate the leakage function of most dies. Notably, the inadequacy of the CPA attacks in presence of variability indicates the need to adopt new side-channel distinguishers [13]. However, the "on-the-fly" stochastic approach easily reaches the 100 % average success rate indicating the success of the attack mounted on the individual dies, since the leakage function of each die can be separately estimated by a linear function. In addition, the success of the stochastic approach clearly indicates that the leakage functions of the dies implemented on a 65 nm CMOS technology in presence of process variations, are still linear.

CPA attacks using the Hamming weight model fail under the impact of process variations due to the deficiency of the fixed model to correctly estimate the leakage function of the measured static CMOS S-box. On the other hand, "on-the-fly" stochastic attacks are 100 % successful, which indicates that the leakage functions are still linear at 65 nm CMOS technology in presence of process variations.

Meanwhile, it was illustrated in [24] that process variability affects the security of both static CMOS and SABL styles (as a representative of DDL styles) in the same way. However, the authors claim that the security of both logic styles deteriorate with process variations according to their evaluation metric. Anyway, DDSLL is expected to follow the same trend as static CMOS in light

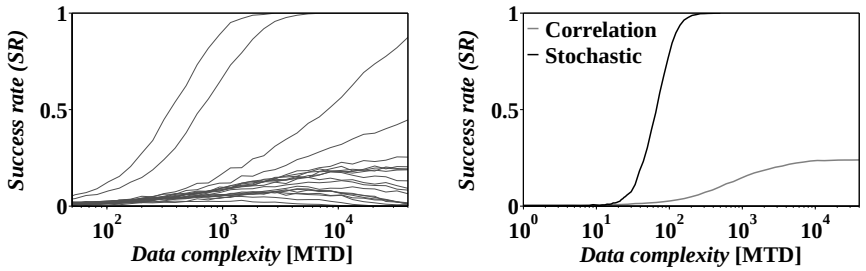


Fig. 4.15.: Success rates of (a) the CPA attacks for each of the 20 dies and (b) non profiled attacks: CPA and stochastic "on-the-fly" attacks, averaged over 20 dies, for different data complexities (1.2V supply).

of the increasing effect of variability in advanced technologies. Finally, an important open question is to determine how will these effects evolve in the future for both static CMOS and DDL styles. That is, investigate whether the security advantage of DDL styles over static CMOS diminishes in further advanced technologies. This question arises mainly from two facts. First, technology scaling renders static CMOS cryptographic devices harder to attack with side-channel analysis. Second, balancing the differential routes of DDL styles is expected to become more difficult due to the randomness of variability (we specifically mean the WID process variations). However, if the DDL style is well designed to properly balance the differential routes and overcome the effect of variability, then it could remain an interesting countermeasure that is capable of reducing the information leakage and increasing the non-linearity nature of the leakages.

In presence of process variations, it is important to study whether the security advantage of the DDL styles over static CMOS diminishes in further advanced technologies.

4.8 SUMMARY

In this chapter we address two important issues that face cryptographic hardware designers and physical cryptanalysis adversaries. First, we try to answer the question; *what is the amount of security gain brought by the use of the DDSLL style that is designed with low power and area overheads for highly power-constrained secure applications, such as RFIDs and contactless smart-cards?* Second, *we study the rising impact of variability in advanced technologies on the security of cryptographic devices implemented using static CMOS (AES S-box).*

With respect to the first issue, we demonstrated in Chapter 2 that DDSLL can be considered a good DDL candidate thanks to its low power and area overheads compared to other DDL styles. Since countermeasures generally cost a

lot in terms of power, area and throughput, it is interesting to assess the security of a potential countermeasure with low overheads in order to optimize the tradeoff between security and performance. Thus, we are motivated to investigate the security of DDSLL in order to numerically evaluate the enhancement it provides over the traditional use of static CMOS. For this purpose we follow the framework proposed in [10] which includes an information theoretic analysis complemented with a security analysis. In the worst-case scenario that uses template attacks, the information leakage reduction of the DDSLL S-box corresponds to *one order of magnitude in terms of MTD* with respect to the static CMOS S-box. *This reduction is rather insufficient and the reason is attributed to the high challenge of balancing the differential routes.* Thus, an important open question is raised ***whether we can design a DDL style that can maximize the information leakage reduction over static CMOS, or whether combining DDSLL with a masking scheme would render a modest tradeoff between security and performance.*** The latter option takes advantage of the power consumption and area reduction brought by the use of DDSLL compared to other DDL styles, which gives room for adding a masking scheme as an additional countermeasure.

We also investigate the linearity of the leakage function of the DDSLL S-box using the "on-the-fly" stochastic approach. This choice of non-profiled attacks is motivated by the method the stochastic attack uses to model the actual leakage function, which in this case is approximated by a linear function of the target bits in the attack. *The importance of the linearity study rises from the fact that current non-profiled distinguishers rely on the linear nature of the leakages (where the leakage model best describes amplitude differences between states). Thus, having a non-linear leakage function would jeopardize the success of these attacks.* Unfortunately, *the measured leakages of the DDSLL S-box turn out to be strongly linear against the expectations of simulations due to measurement specificities that are not accounted for in simulations.* As a result, an interesting question for further research is to ***determine the possibility of including non-linearity as a design guideline while developing secure logic styles.***

Finally, the increasing concern of process variability in sub-100 nm technologies and its impact on the security of cryptographic devices is analyzed. Measurements of 20 dies implementing static CMOS AES S-boxes on 65 nm CMOS technology show that *extending template attacks to infer process variations causes a significant loss of information. Thus, adversaries who use profiled templates face a challenging task trying to build accurate templates due to technology scaling.* These measurements also prove the inadequacy of the currently used models by the non-profiled DPA attacks, represented by the Hamming weight model used by the CPA attack, to evaluate the security of an implementation in light of high process variations. Clearly, *CPA attacks with Hamming weight model fail to retrieve the secret information from most dies.* On the other hand, *"on-the-fly" stochastic attacks are completely successful since in presence of process variations, the leakages of each die are still linear and can be separately estimated by a linear function.* Similarly, DDSLL is expected to follow the same trend as static

CMOS in light of the increasing effect of variability in advanced technologies. Consequently, an interesting study would be to ***investigate whether the security advantage of DDL styles in general over static CMOS further diminish with technology scaling.***

Bibliography

[1] S. Mangard, E. Oswald, and T. Popp, *Power analysis attacks - revealing the secrets of smart cards.* Springer, 2007.

[2] P. C. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *CRYPTO*, 1999, pp. 388–397.

[3] S. Dziembowski and K. Pietrzak, "Leakage-resilient cryptography," in *FOCS*, 2008, pp. 293–302.

[4] L. Goubin and J. Patarin, "DES and differential power analysis (the "duplication" method)," in *In Proceedings of the First International Workshop on Cryptographic Hardware and Embedded Systems, CHES*. London, UK: Springer-Verlag, 1999, pp. 158–172. [Online]. Available: http://portal.acm.org/citation.cfm?id=648252.752372

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

[6] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi, "Towards sound approaches to counteract power-analysis attacks," in *CRYPTO*, 1999, pp. 398–412.

[7] I. Hassoune, F. Macé, D. Flandre, and J.-D. Legat, "Dynamic differential self-timed logic families for robust and low-power security ICs," *Integration*, vol. 40, no. 3, pp. 355–364, 2007.

[8] F.-X. Standaert, N. Veyrat-Charvillon, E. Oswald, B. Gierlichs, M. Medwed, M. Kasper, and S. Mangard, "The world is not enough: Another look on second-order DPA," in *ASIACRYPT*, 2010, pp. 112–129.

[9] M. Renauld, D. Kamel, F.-X. Standaert, and D. Flandre, "Information theoretic and security analysis of a 65-nanometer DDSLL AES S-Box," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES*, 2011, pp. 223–239.

[10] F.-X. Standaert, T. G. Malkin, and M. Yung, "A unified framework for the analysis of side-channel key recovery attacks," in *Proceedings of the 28th Annual International Conference on Advances in Cryptology: the Theory and Applications of Cryptographic Techniques*, ser. EUROCRYPT. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 443–461. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-01001-9_26

[11] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES*, 2002, pp. 13–28.

[12] W. Schindler, K. Lemke, and C. Paar, "A stochastic model for differential side channel cryptanalysis," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES, Springer, LNCS 3659*. Springer, 2005, pp. 30–46.

[13] N. Veyrat-Charvillon and F.-X. Standaert, "Generic side-channel distinguishers: Improvements and limitations," in *CRYPTO*, 2011, pp. 354–372.

[14] K. Lemke-rust, "Models and algorithms for physical cryptanalysis," Ph.D. dissertation, University of Bochum, Bochum, 2007. [Online]. Available: http://www.emsec.rub.de/media/crypto/attachments/files/2010/04/thesis_lemke_rust.pdf

[15] J. Doget, E. Prouff, M. Rivain, and F.-X. Standaert, "Univariate side channel attacks and leakage modeling," *J. Cryptographic Engineering*, vol. 1, no. 2, pp. 123–144, 2011.

[16] M. Renauld, F.-X. Standaert, N. Veyrat-Charvillon, D. Kamel, and D. Flandre, "A formal study of power variability issues and side-channel attacks for nanoscale devices," in *EUROCRYPT*, 2011, pp. 109–128.

[17] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES*, 2004, pp. 16–29.

[18] D. Boneh, R. A. DeMillo, and R. J. Lipton, "On the importance of checking cryptographic protocols for faults (extended abstract)," in *EUROCRYPT*, 1997, pp. 37–51.

[19] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, "The EM side-channel(s)," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES*, 2002, pp. 29–45.

[20] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES*, no. Generators, 2001, pp. 251–261.

[21] P. C. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," in *CRYPTO*, 1996, pp. 104–113.

[22] E. Peeters, F.-X. Standaert, and J.-J. Quisquater, "Power and electromagnetic analysis: Improved model, consequences and comparisons," *Integr. VLSI J.*, vol. 40, no. 1, pp. 52–60, Jan. 2007. [Online]. Available: http://dx.doi.org/10.1016/j.vlsi.2005.12.013

[23] F.-X. Standaert, F. Macé, E. Peeters, and J.-J. Quisquater, "Updates on the security of FPGAs against power analysis attacks," in *ARC*, 2006, pp. 335–346.

[24] L. Lin and W. Burleson, "Analysis and mitigation of process variation impacts on power-attack tolerance," in *Proceedings of the 46th ACM/IEEE Design Automation Conference, DAC*, Jul. 2009, pp. 238 –243.

[25] B. Gierlichs, K. Lemke-Rust, and C. Paar, "Templates vs. stochastic methods," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES*, 2006, pp. 15–29.

[26] F.-X. Standaert, F. Koeune, and W. Schindler, "How to compare profiled side-channel attacks?" in *ACNS*, 2009, pp. 485–498.

[27] H. L. V. Trees, *Detection, Estimation, and Modulation Theory, Part I.* John Wiley Sons, 1968.

[28] J.-S. Coron, P. C. Kocher, and D. Naccache, "Statistics and secret leakage," in *Financial Cryptography*, 2000, pp. 157–173.

[29] L. Goubin, "A sound method for switching between Boolean and Arithmetic masking," in *Proceedings of the Third International Workshop on Cryptographic Hardware and Embedded Systems, CHES*. London, UK, UK: Springer-Verlag, 2001, pp. 3–15. [Online]. Available: http://dl.acm.org/citation.cfm?id=648254.752571

[30] J.-S. Coron and A. Tchulkine, "A new algorithm for switching from arithmetic to Boolean masking," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES*, 2003, pp. 89–97.

[31] K. Schramm and C. Paar, "Higher order masking of the AES," in *CT-RSA*, 2006, pp. 208–225.

[32] D. Suzuki, M. Saeki, and T. Ichikawa, "Random switching logic: A countermeasure against DPA based on transition probability," on Transition Probability, IACR ePrint, Tech. Rep., 2004.

[33] T. Popp and S. Mangard, "Masked dual-rail pre-charge logic: DPA-resistance without routing constraints," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES*, 2005, pp. 172–186.

[34] Z. Chen and Y. Zhou, "Dual-rail random switching logic: A countermeasure to reduce side channel leakage," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES*, 2006, pp. 242–254.

[35] D. May, H. L. Muller, and N. P. Smart, "Random register renaming to foil DPA," in *Proceedings of the Third International Workshop on Cryptographic Hardware and Embedded Systems, CHES*. London, UK, UK: Springer-Verlag, 2001, pp. 28–38. [Online]. Available: http://dl.acm.org/citation.cfm?id=648254.752574

[36] J. Blömer, J. Guajardo, and V. Krummel, "Provably secure masking of AES," in *Selected Areas in Cryptography*, 2004, pp. 69–83.

[37] E. Oswald, S. Mangard, and N. Pramstaller, "Secure and efficient masking of AES - a mission impossible?" *IACR Cryptology ePrint Archive*, vol. 2004, p. 134, 2004.

[38] M. Nassar, S. Bhasin, J.-L. Danger, G. Duc, and S. Guilley, "BCDL: A high speed balanced DPL for FPGA with global precharge and no early evaluation," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, Mar. 2010, pp. 849 –854.

[39] D. Suzuki and M. Saeki, "Security evaluation of DPA countermeasures using dual-rail pre-charge logic style," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES*, 2006, pp. 255–269.

[40] T. Popp, M. Kirschbaum, T. Zefferer, and S. Mangard, "Evaluation of the masked logic style MDPL on a prototype chip," in *Proceedings of the 9th international workshop on Cryptographic Hardware and Embedded Systems, CHES*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 81–94. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-74735-2_6

[41] E. D. M. and Benedikt Gierlichs and Bart Preneel and Ingrid Verbauwhede, "Practical DPA attacks on MDPL," Cryptology ePrint Archive, Report 2009/231, 2009, http://eprint.iacr.org/.

[42] M. Joye, P. Paillier, and B. Schoenmakers, "On second-order differential power analysis," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES*, 2005, pp. 293–308.

[43] F. Durvaux, M. Renauld, F.-X. Standaert, L. van Oldeneel tot Oldenzeel, and N. Veyrat-Charvillon, "Cryptanalysis of the CHES 2009/2010 random delay countermeasure," *IACR Cryptology ePrint Archive*, vol. 2012, p. 38, 2012.

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

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

[46] F. Macé, F.-X. Standaert, I. Hassoune, J.-J. Quisquater, and J.-D. Legat, "A Dynamic Current Mode Logic to Counteract Power Analysis Attacks," in *Proceedings of the 19th Conference on Design of Circuits and Integrated Systems, DCIS*, Nov. 2004, pp. 186–191.

[47] S. Guilley, L. Sauvage, J.-L. Danger, T. Graba, and Y. Mathieu, "Evaluation of power-constant dual-rail logic as a protection of cryptographic applications in FPGAs," in *Second International Conference on Secure System Integration and Reliability Improvement, SSIRI*, Jul. 2008, pp. 16 –23.

[48] J. Cong, Z. Pan, L. He, C.-K. Koh, and K.-Y. Khoo, "Interconnect design for deep submicron ICs," in *Proceedings of the IEEE/ACM international conference on Computer-aided design*, ser. ICCAD. Washington, DC, USA: IEEE Computer Society, 1997, pp. 478–485. [Online]. Available: http://dl.acm.org/citation.cfm?id=266388.266534

[49] *Roadmap ITRS 2007*, ITRS, 2007.

[50] K. Tiri and I. Verbauwhede, "Place and route for secure standard cell design," in *CARDIS*, 2004, pp. 143–158.

[51] F. Macé, "Physical design of cryptographic applications: constrained environments and power analysis resistance," Ph.D. dissertation, Université catholique de Louvain, Louvain-la-Neuve, 2008.

[52] E. Brier, C. Clavier, and F. Olivier, "Optimal statistical power analysis," *IACR Cryptology ePrint Archive*, vol. 2003, p. 152, 2003.

[53] K. Tiri, D. Hwang, A. Hodjat, B.-C. Lai, S. Yang, P. Schaumont, and I. Verbauwhede, "A side-channel leakage free coprocessor IC in 0.18 μm CMOS for embedded AES-based cryptographic and biometric processing," in *Proceedings of Design Automation Conference, DAC*, 2005, pp. 222–227.

[54] S. Mangard, E. Oswald, and F.-X. Standaert, "One for all and all for one: unifying standard differential power analysis attacks," *Information Security, IET*, vol. 5, no. 2, pp. 100 –110, Jun. 2011.

[55] A. Chandrakasan, S. Sheng, and R. Brodersen, "Low-power CMOS digital design," *IEEE Journal of Solid-State Circuits, JSSC*, vol. 27, no. 4, pp. 473 –484, Apr. 1992.

[56] D. Blaauw, K. Chopra, A. Srivastava, and L. Scheffer, "Statistical timing analysis: From basic principles to state of the art," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 27, no. 4, pp. 589 –607, 2008.

[57] S. Nassif, K. Bernstein, D. Frank, A. Gattiker, W. Haensch, B. Ji, E. Nowak, D. Pearson, and N. Rohrer, "High performance CMOS variability in the 65nm regime and beyond," in *IEEE International Electron Devices Meeting, IEDM*, Dec. 2007, pp. 569 –571.

[58] K. Bowman, S. Duvall, and J. Meindl, "Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration," *IEEE Journal of Solid-State Circuits, JSSC*, vol. 37, no. 2, pp. 183 –190, Feb. 2002.

[59] A. Zadeh and C. Gebotys, "Leakage power and side channel security of nanoscale Cryptosystem-on-Chip (CoC)," in *IEEE Computer Society Annual Symposium on VLSI, ISVLSI*, May 2009, pp. 31 –36.

[60] L. Lin and W. Burleson, "Leakage-based differential power analysis (LDPA) on sub-90nm CMOS cryptosystems," in *Proceedings of IEEE International symposium of circuits and systems, ISCAS*, 2008, pp. 252–255.

[61] D. Kamel, C. Hocquet, F.-X. Standaert, D. Flandre, and D. Bol, "Glitch-induced within-die variations of dynamic energy in voltage-scaled nano-CMOS circuits," in *Proceedings of European Solid-State Circuits Conference, ESSCIRC*, Sept. 2010, pp. 518 –521.

CHAPTER 5

CONCLUSIONS

Highly power-constrained secure applications present a great challenge for circuit designers who try to determine the best tradeoff between security and performance. They feature tight power consumption and area budgets while relaxing the timing constraint. Circuit designers are directly involved since "computationally" secure applications are not secure anymore due to the physical leakage of information that is exploited by side-channel attacks. Consequently, it is their responsibility to develop new techniques that maximize the security of the implementation with minimum cost in terms of power consumption and area in light of the growing concern of variability associated to nano-CMOS technologies. In this dissertation, we decided to focus on hiding techniques, more specifically the use of dynamic differential logic (DDL) styles because they tackle the problem directly where it lies and they potentially allow hardware optimizations to reduce the power consumption and the area. Certainly, as a hardware countermeasure, DDL styles address the source of information leakage, that is the power consumption dependency on processed data. Their goal, as a hiding technique, is to minimize the signal-to-noise (SNR) ratio given as:

$$SNR = \frac{Var(P_{exp})}{Var(P_{sw.\,noise} + P_{el.\,noise})} \tag{5.1}$$

where $Var(P_{exp})$ refers to the exploitable power consumption variance, $P_{sw.noise}$ is the switching noise variance and $Var(P_{el.noise})$ denotes the electronic noise variance. DDL styles target the reduction of $Var(P_{exp})$ by trying to consume exactly the same power during each clock cycle.

Generally, in order to evaluate the security of a countermeasure and compare it to other techniques, the assessment framework has to be unified. This includes the benchmark circuits (AES S-box in our case), the testing methods (using actual attacks in an information theoretic sense and in a security analysis as proposed in [1]) and the evaluation metrics (the perceived information in the case of the information theoretic analysis and the measurements to disclosure (MTD) in the case of the security analysis). In our assessment framework of DDSLL we chose the AES S-box as a case study in compliance with the suggestion in [2] that the benchmark circuits should be standard encryption algorithms. The testing methods and accordingly, the evaluation metrics are decided upon as previously mentioned because of their relevance. Indeed, until recently, designers of DDL styles favored the normalized energy and standard deviations metrics (NED and NSD). However, these two metrics tend to overestimate the security gain achieved by logic styles generating a high power consumption due to the normalization by either $max(E)$ or $\mu(E)$, which renders them as unreliable metrics [3, 4]. On the other hand, considering actual attack scenarios is a direct approach to assess a countermeasure. Also, it was proved in [5, 6] and also in this dissertation that the standard deviation over the inputs is related to the results of actual attacks in an optimal statistical analysis. This further justifies the importance of the standard deviations, and not the normalized deviations, as a preliminary step that the designer can use during the design phase to speculate the strength of his countermeasure, but not enough to draw concrete conclusions. The designer still has to employ the suggested assessment framework, where he finally computes the MTD in a worst-case scenario (for example, mount template attacks) and compare it to that of a reference implementation.

In this dissertation we investigated the different aspects of the dynamic differential swing-limited logic (DDSLL) style as a potential solution, starting from its design phase till evaluating its performance and security through measurements of a test chip implemented in a LP 65 nm CMOS technology. Table 5.1 summarizes the advantages and disadvantages of the DDSLL style. In the design methodology, we demonstrated three techniques that when combined lead to the reduction of the DDSLL power consumption with respect to static CMOS. The first is the use of n-input differential pull-down networks (DPDN)s to implement complex functions (n is limited to 4 in our case). The second is the sharing of the common blocks; the dynamic current source, the self-timing buffer and some transistors from the precharge circuitry. Finally, the third technique is the inherent low-swing operation of the DDSLL style. Combined, these techniques lead to a power consumption reduction factor of 1.9 compared to sense amplifier based logic (SABL) [7] along the studied frequency range and 11 % less power than static CMOS above 1 MHz (at nominal conditions).

The use of the 65 nm CMOS technology empowered us to experience the growing impact of process variations on the performance and the security aspects of the DDSLL style. We demonstrated how process variations threaten the power closure of static CMOS combinatorial circuits via their dynamic power sensitivity to glitch-induced activity factor (α_F) variations. On the other hand, we proved that DDSLL solves this problem thanks to its deterministic α_F. This manifests the importance of using self-timed gates in solving the problem of desynchronized input signals that eventually lead to glitches which increase with the logic depth of the circuit. In this context, we considered the 3σ metric for high yield to evaluate the performance of the DDSLL S-box in presence of process variations. As a result, DDSLL featured a 3 % increase with respect to the typical dynamic power consumption, whereas the static CMOS S-box experienced a 75 % increase in its dynamic power consumpiton. As for the security aspect, we showed that process variations actually harden the circuits against typical attacks (we used the static CMOS S-box as a case study). In the case of profiled distinguishers, we demonstrated that extending template attacks [8] to infer process variations (i.e. profile n dies to attack another die) causes a significant loss of information. Thus, adversaries who use profiled templates face a challenging task trying to build accurate templates due to technology scaling. In the case of non-profiled distinguishers, we showed that process variations render the models currently used by the non-profiled differential power analysis (DPA) attacks (more precisely, Hamming weight and Hamming distance) inadequate. On the other hand, "on-the-fly" stochastic attacks [9, 10] are completely successful since in presence of process variations, the leakages of each die are still linear and can be separately estimated by a linear function.

During the security assessment of the DDSLL S-box, we observed the non-linearity nature of the leakages at the simulation level. Despite the fact that actual leakages are linear (due to the presence of measurement specificities that led to smoothing the shape of the traces by filtering out the high frequencies as illustrated in Fig. 4.7 in Chapter 4), non-linearity of the leakages presents an appealing method to adopt in order to render the adversary's job more challenging. The importance of the non-linearity rises from the fact that current non-profiled distinguishers rely on the linear nature of the leakages. Thus, having a non-linear leakage function would jeopardize the success of these attacks.

Table 5.1.: Advantages and disadvantages of the DDSLL style.

Aspect	Advantages	Disadvantages
Design	• Compact gates thanks to the sharing principle and the use of complex DPDNs.	• Full-custom, long time-to-market, high design cost.
	• Small area.	• Difficult to fully balance the complex DPDNs (with the current techniques).
	• Easy to interface with static CMOS.	• Achieving the balance between the differential routes and the complex DPDNs is expected to become more difficult due to WID process variations.
Performance	• Low power consumption thanks to the sharing principle, the use of complex DPDNs and the low-swing operation (1.9× lower than SABL along the studied frequency range and 11 % less than static CMOS above 1 MHz).	• Large off currents at high temperatures jeopardize the functionality by charging back the low-voltage floating nodes to V_{DD} during evaluation.
	• Low leakage power (4.7 % compared to 16 % for static CMOS in the case of the S-box).	• High delay penalty (2.5× static CMOS).
	• Power consumption is less sensitive to temperature.	• D2D delay variability is high, despite being independent on the logic depth (13.5 %).
	• Dynamic power consumption is not susceptible to glitch-induced α_F variations thanks to its deterministic α_F - Its WID variations average out over logic depth and circuit size and its D2D variations are constant.	• It is not functional at low V_{DD} in presence of WID process variations (25 % failure rate at 1 V and 100 % at 0.8 V).
	• For high yield of the S-box, typical P_{dyn}+3σ is only 3 % higher than typical P_{dyn} (the factor is 75 % in the case of static CMOS).	
	• WID delay variability averages out with logic depth.	

Continued on next page

Table 5.1 – continued from previous page

Aspect	Advantages	Disadvantages
Security	• Its MTD is an order of magnitude higher than static CMOS in a worst-case scenario using profiled template attacks. • In presence of process variations, security assessment of cryptographic devices is harder to attack with side-channel analysis (True for static CMOS and it is expected that DDSLL follows the same trend).	• The security enhancement is not sufficient compared to other types of countermeasures (masking - exponential gain and noise addition - linear gain). • The measured leakage function is linear for almost all the time samples (DDL styles in general are expected to achieve non-linear leakage functions).

Prespectives

Basically, there are three ways to reduce the signal variance (exploitable power consumption variance, $Var(P_{exp})$) in order to eventually minimize the SNR ratio.

1. ***DDL styles try to balance the differential routes and the differential internal nodes of the complex DPDNs*** (in case the DDL style is composed of DPDNs). Of course, as we demonstrated throughout this dissertation, it is an extremely difficult task with the current available techniques. As a result, completely balancing the differential routes and the capacitances of the DPDN internal nodes requires the development of new balancing methods.

2. ***DDL styles try to achieve moderate balancing of the differential routes and the capacitances of the DPDN internal nodes (e.g. the compact DDSLL S-box) and then reduce the overall power consumption such that the standard deviation of the instantaneous power over the input sequences decreases accordingly (e.g. low-swing operation)***. This method is more realistic as it compromises the tradeoff between security and power consumption and it aims at further improving the security through the reduction of the power consumption. This is the method that low-swing DDL styles (DyCML [11] and DDSLL) utilize. Nevertheless, accepting a reduced standard deviation over the inputs instead of 0 is risky because current side-channel attacks are capable of revealing the secret information from implementations that feature very low standard deviations over the inputs. Therefore, designers of DDL styles have to be aware of the competence of the side-channel attacks and target a standard deviation over the inputs that is below their ability to detect it. In the case of the DDSLL style, it achieved one order of magnitude of

security gain over static CMOS in terms of MTD in a worst-case scenario using template attacks.

3. ***DDL styles use the technique proposed in [7]*** where SABL is designed to guarantee that in every clock cycle it charges and discharges a total capacitance with a constant value despite the fact that during each clock cycle different capacitances are switched. That is, despite the existence of unbalanced differential capacitances, SABL charges and discharges the sum of all capacitances to avoid having to fully balance the differential routes and the capacitances of the DPDN internal nodes. Although this method sounds promising, it does not guarantee a signal variance equal to 0 as we demonstrated in Chapter 4. One possible reason is the scaling of technology; the authors of [7] designed SABL circuits using a 0.18 μm CMOS technology and we used the 65 nm LP CMOS technology. Indeed, technology scaling implies smaller capacitances and more sensitivity to small capacitance imbalance. From its power traces we are able to see the huge reduction in the standard deviation over the input sequences with respect to the average instantaneous power consumption (10×, while DDSLL for example features a reduction of about 4×). However, the absolute value of the standard deviation over the inputs is similar to that of DDSLL (4×10^{-5}).

The first method, despite being the most straightforward, is theoretical and not feasible up to now, especially with the growing concern of process variations of nano-scale CMOS technologies. The second and third methods are more appealing because of their practicality, but at the same time, care has to be taken during the design phase in order to actually achieve a high security gain over static CMOS as explained above. To overcome the moderate security gain of DDSLL compared to static CMOS (10× in terms of MTD), a possible solution is to try to combine the second and third methods in one solution. That is, ***develop a new DDL style that achieves moderate balancing, low power consumption (consider low-swing operation) and charge (discharge) the sum of all capacitances during each clock cycle.*** This way, the designer has three levels of freedom to try to optimize the security - power consumption and area tradeoff. Another possible solution is to ***add another countermeasure, for example masking, to the DDSLL style as its low power consumption and area overheads give room for considering a combination of countermeasures.*** These solutions target the most powerful adversary in a worst-case scenario (template attacks).

Adversaries that are actually suboptimal and use non-profiled distinguishers, are also considered threatening and have to be dealt with adequately. In light of the importance of the non-linearity of the leakages that threatens the success of non-profiled attacks, we raise an interesting question for further research: ***Is it possible to include non-linearity as a design guideline while developing secure logic styles?***

Another important question is to determine ***how will these effects evolve in the future for both static CMOS and DDL styles.*** That is, investigate whether the security advantage of DDL styles over static CMOS diminishes in further advanced technologies. This question arises mainly from two facts. First, technology scaling renders static CMOS cryptographic devices harder to attack with side-channel analysis. Second, balancing the differential routes of DDL styles is expected to become more difficult due to the randomness of variability (we specifically mean the WID process variations). However, if the DDL style is well designed to overcome the power consumption dependency on the processed data, then it could remain an interesting countermeasure that is capable of reducing the information leakage and increasing the non-linearity nature of the leakages.

Finally, we conclude that the techniques that we proposed throughout this dissertation to optimize the security - performance tradeoff of the DDSLL style renders it a potential solution for highly power-constrained secure applications. Nevertheless, DDSLL has its list of disadvantages; most importantly its inability to break the power consumption dependence on the processed data such that the security gain over static CMOS is only one order of magnitude. Also, the sensitivity of its functionality towards WID process variations at supply voltages below 1 V makes it less robust, especially with technology scaling. Therefore, ***the techniques we proposed serve more as guidelines that pave the way to developing new, low-power, variability-tolerant, secure DDL styles.***

Bibliography

[1] F.-X. Standaert, T. G. Malkin, and M. Yung, "A unified framework for the analysis of side-channel key recovery attacks," in *Proceedings of the 28th Annual International Conference on Advances in Cryptology: the Theory and Applications of Cryptographic Techniques*, ser. EUROCRYPT. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 443–461. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-01001-9_26

[2] K. Tiri, D. Hwang, A. Hodjat, B.-C. Lai, S. Yang, P. Schaumont, and I. Verbauwhede, "A side-channel leakage free coprocessor IC in 0.18μm CMOS for embedded AES-based cryptographic and biometric processing," in *Proceedings of Design Automation Conference, DAC*, 2005, pp. 222–227.

[3] F. Macé, "Physical design of cryptographic applications: constrained environments and power analysis resistance," Ph.D. dissertation, Université catholique de Louvain, Louvain-la-Neuve, 2008.

[4] S. Mangard, E. Oswald, and T. Popp, *Power analysis attacks - revealing the secrets of smart cards.* Springer, 2007.

[5] E. Brier, C. Clavier, and F. Olivier, "Optimal statistical power analysis," *IACR Cryptology ePrint Archive*, vol. 2003, p. 152, 2003.

[6] ——, "Correlation power analysis with a leakage model," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES*, 2004, pp. 16–29.

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

[8] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES*, 2002, pp. 13–28.

[9] W. Schindler, K. Lemke, and C. Paar, "A stochastic model for differential side channel cryptanalysis," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES, Springer, LNCS 3659.* Springer, 2005, pp. 30–46.

[10] K. Lemke-rust, "Models and algorithms for physical cryptanalysis," Ph.D. dissertation, University of Bochum, Bochum, 2007. [Online]. Available: http://www.emsec.rub.de/media/crypto/attachments/files/2010/04/thesis_lemke_rust.pdf

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

APPENDIX A

AES S-BOX STATIC CMOS DESIGN

The AES S-box mainly consists of a multiplicative inverse in a Galois field $GF(2^8)$ and an affine transformation. Its gate complexity (and power consumption) is greatly reduced when composite field arithmetic is employed as proposed by [1]. However, this requires a transformation matrix to map the elements of the original field $GF(2^8)$ to the field $GF(((2^2)^2)^2)$ and an inverse transformation to move back to the original field. In the following, we use the optimized S-box description given in [2]. In Fig. A.1, the AES S-box architecture is shown and the details of each block is illustrated in the subsequent figures. Most of these block diagrams are reused from [3] which is a useful report that also helped deduce the remaining blocks.

$$X_3^{-1} = x_3 \oplus x_3x_2x_1 \oplus x_3x_0 \oplus x_2 \tag{A.1}$$

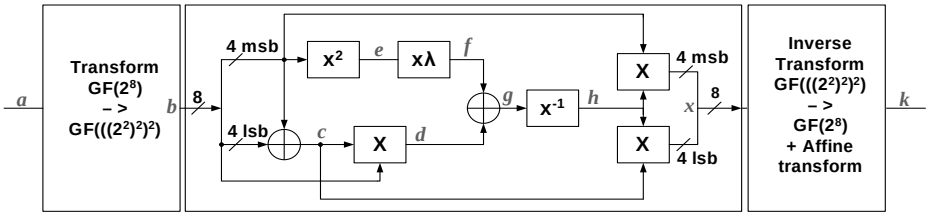
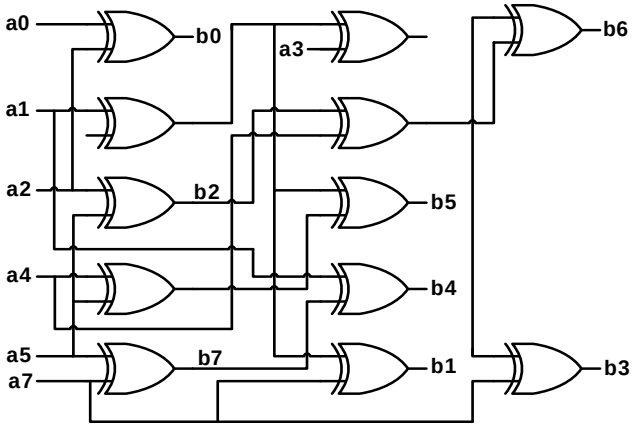


Fig. A.1.: AES S-box block diagram [1].

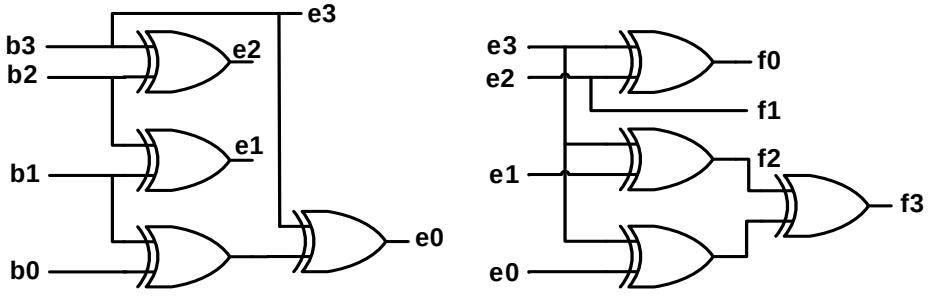
$$\begin{bmatrix} b0 \\ b1 \\ b2 \\ b3 \\ b4 \\ b5 \\ b6 \\ b7 \end{bmatrix} = \begin{bmatrix} 1\,0\,1\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0\,1\,1 \\ 0\,0\,1\,0\,0\,1\,0\,0 \\ 0\,1\,0\,1\,0\,0\,1\,1 \\ 0\,1\,0\,0\,0\,1\,0\,1 \\ 0\,1\,0\,0\,1\,1\,1\,0 \\ 0\,1\,1\,1\,1\,1\,1\,0 \\ 0\,0\,0\,0\,0\,1\,0\,1 \end{bmatrix} \begin{bmatrix} a0 \\ a1 \\ a2 \\ a3 \\ a4 \\ a5 \\ a6 \\ a7 \end{bmatrix}$$

(a)



(b)

Fig. A.2.: Matrix (a) and block diagram (b) of the transform from $GF(2)^8$ to $GF(((2^2)^2)^2)$ function.



(a) [3] (b)

Fig. A.3.: Block diagrams of the (a) Squaring function (X^2) in $GF(2^4)$ and the (b) Multiplication with λ function ($\times\lambda$) in $GF(2^4)$.

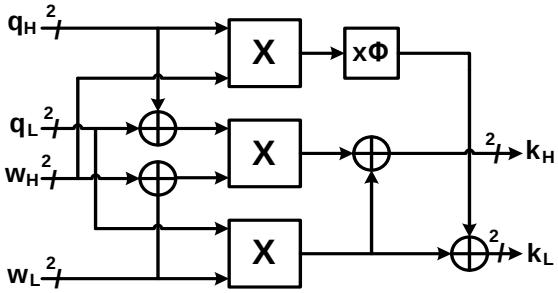


Fig. A.4.: Block diagram of the Multiplication function in $GF(2^4)$ [3].

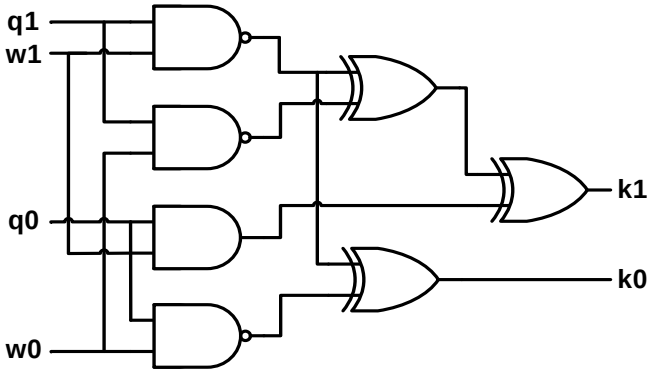


Fig. A.5.: Block diagram of the Multiplication function in $GF(2^2)$.

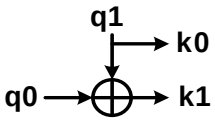


Fig. A.6.: Block diagram of the Multiplication by ϕ function ($\times\phi$) in $GF(2^2)$ [3].

$$X_2^{-1} = x_3x_2x_1 \oplus x_3x_2x_0 \oplus x_3x_0 \oplus x_2 \oplus x_2x_1 \tag{A.2}$$

$$X_1^{-1} = x_3 \oplus x_3x_2x_1 \oplus x_3x_1x_0 \oplus x_2 \oplus x_2x_0 \oplus x_1 \tag{A.3}$$

$$\begin{aligned} X_0^{-1} = {} & x_3x_2x_1 \oplus x_3x_2x_0 \oplus x_3x_1 \oplus x_3x_1x_0 \oplus x_3x_0 \oplus x_2 \\ & \oplus x_2x_1 \oplus x_2x_1x_0 \oplus x_1 \oplus x_0 \end{aligned} \tag{A.4}$$

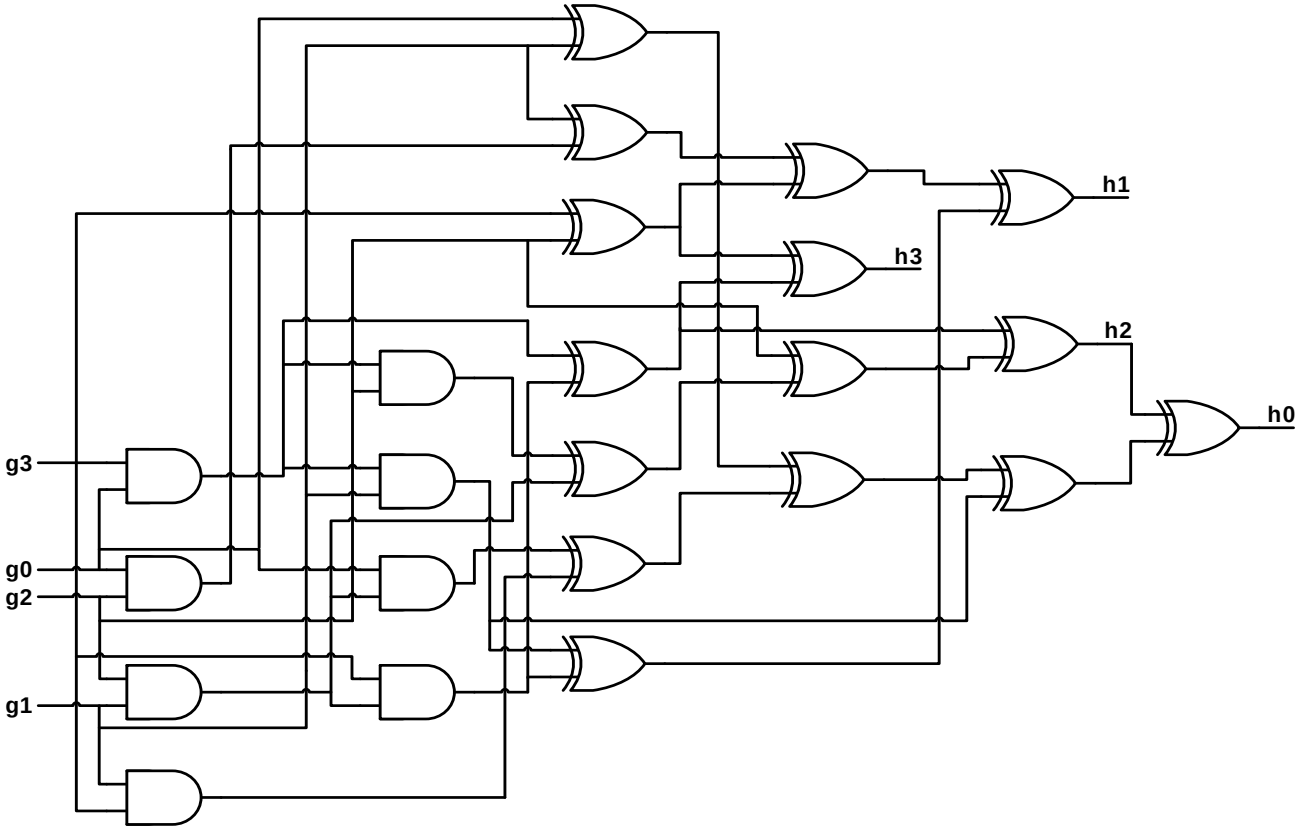


Fig. A.7.: Block diagram of the Inversion $GF(2^4)$ function as explained by the equations in [4].

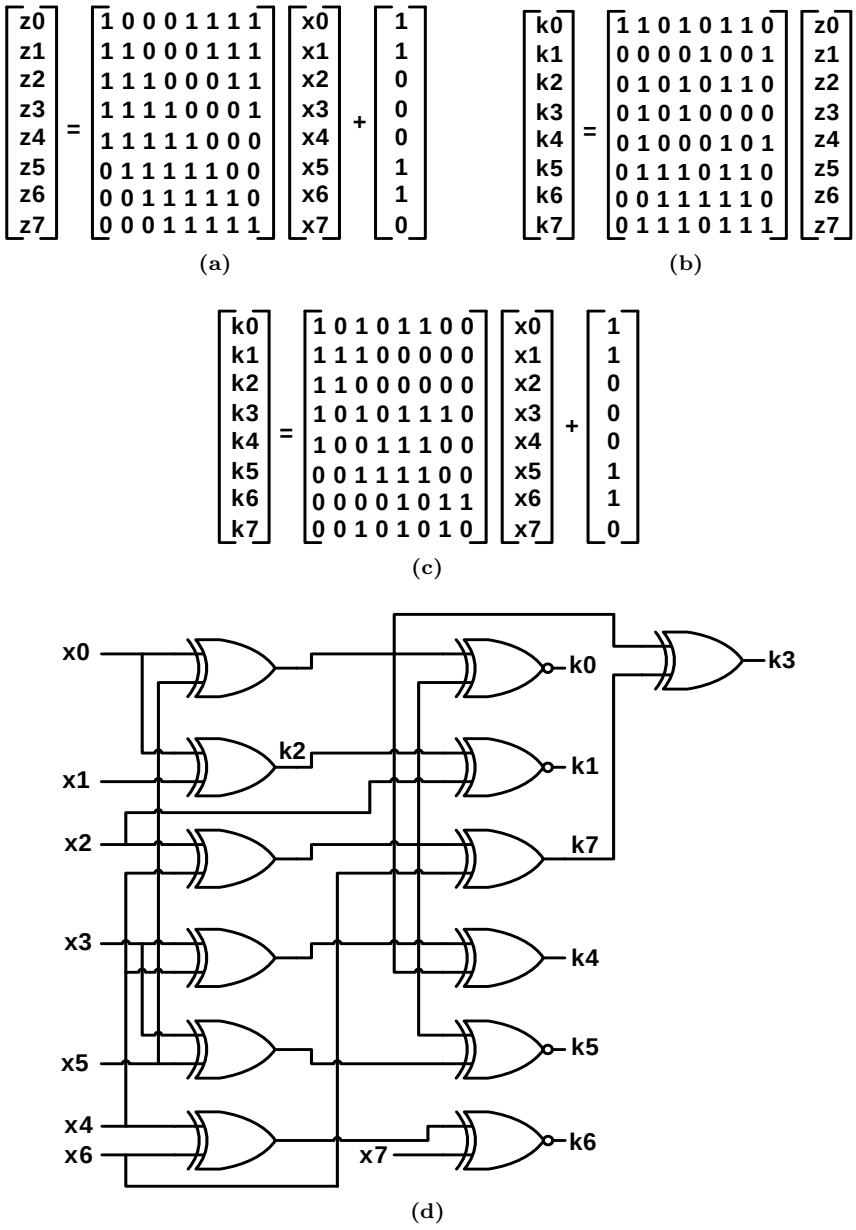


Fig. A.8.: Matrices of the (a) Affine transform function, (b) Inverse transform from $GF(((2^2)^2)^2)$ to $GF(2)^8$ function and (c) merged Affine transform and Inverse transform matrices. (d) Block diagram of the merged Affine transform and Inverse transform matrices.

Bibliography

[1] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A compact Rijndael hardware architecture with S-Box optimization," in *ASIACRYPT*, 2001, pp. 239–254.

[2] N. Mentens, L. Batina, B. Preneel, and I. Verbauwhede, "A systematic evaluation of compact hardware implementations for the Rijndael S-Box," in *CT-RSA*, 2005, pp. 323–333.

[3] E. N. C. Mui, "Practical implementation of Rijndael S-Box using combinational logic," http://www.xess.com/projects/Rijndael_SBox.pdf, XESS Corp., xess.com, Tech. Rep., 2007. [Online]. Available: http://www.xess.com/projects/Rijndael_SBox.pdf

[4] X. Zhang and K. Parhi, "High-speed VLSI architectures for the AES algorithm," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, TVLSI*, vol. 12, no. 9, pp. 957 –967, Sept. 2004.

APPENDIX B

DPDNS OF DDSLL S-BOX COMPLEX FUNCTIONS

In this appendix, we present the BDD and schematic diagrams of the complex functions in the AES S-box implemented using DDSLL styles, namely, the four functions of the Inversion block in $GF(2)^4$ and the two functions of the Multiplication block in $GF(2)^2$. As a reminder, the functions in the Inversion $GF(2)^4$ block are:

$$X_3^{-1} = \overline{x_3}x_2\overline{x_1x_0} + \overline{x_3}x_2\overline{x_1}x_0 + \overline{x_3}x_2x_1\overline{x_0} + \overline{x_3}x_2x_1x_0 + x_3\overline{x_2x_1x_0} + x_3\overline{x_2}x_1\overline{x_0} + x_3x_2\overline{x_1}x_0 + x_3x_2x_1\overline{x_0} \tag{B.1}$$

$$X_2^{-1} = x_3\overline{x_2x_1}x_0 + x_3\overline{x_2}x_1x_0 + x_3x_2\overline{x_1}x_0 + x_3x_2x_1\overline{x_0} + x_3x_2x_1x_0 + x_3x_2\overline{x_1x_0} + \overline{x_3}x_2\overline{x_1x_0} + \overline{x_3}x_2\overline{x_1}x_0 \tag{B.2}$$

$$X_1^{-1} = \overline{x_3x_2}x_1\overline{x_0} + \overline{x_3x_2}x_1x_0 + \overline{x_3}x_2\overline{x_1x_0} + \overline{x_3}x_2x_1x_0 + x_3\overline{x_2x_1x_0} + x_3\overline{x_2x_1}x_0 + x_3\overline{x_2}x_1x_0 + x_3x_2\overline{x_1}x_0 \tag{B.3}$$

$$X_0^{-1} = \overline{x_3x_2x_1}x_0 + \overline{x_3x_2}x_1\overline{x_0} + \overline{x_3}x_2\overline{x_1x_0} + \overline{x_3}x_2x_1\overline{x_0} + \overline{x_3}x_2x_1x_0 + x_3\overline{x_2}x_1x_0 + x_3x_2\overline{x_1x_0} + x_3x_2x_1\overline{x_0} \tag{B.4}$$

In the Multiplication $GF(2)^2$ block:

$$k_0 = \overline{q_1w_1}q_0w_0 + \overline{q_1}w_1q_0w_0 + q_1\overline{w_1}q_0w_0 + q_1w_1\overline{q_0w_0} + q_1w_1\overline{q_0}w_0 + q_1w_1q_0\overline{w_0} \tag{B.5}$$

$$k_1 = \overline{q_1}w_1q_0\overline{w_0} + \overline{q_1}w_1q_0w_0 + q_1\overline{w_1q_0}w_0 + q_1\overline{w_1}q_0w_0 + q_1w_1\overline{q_0w_0} + q_1w_1q_0w_0 \tag{B.6}$$

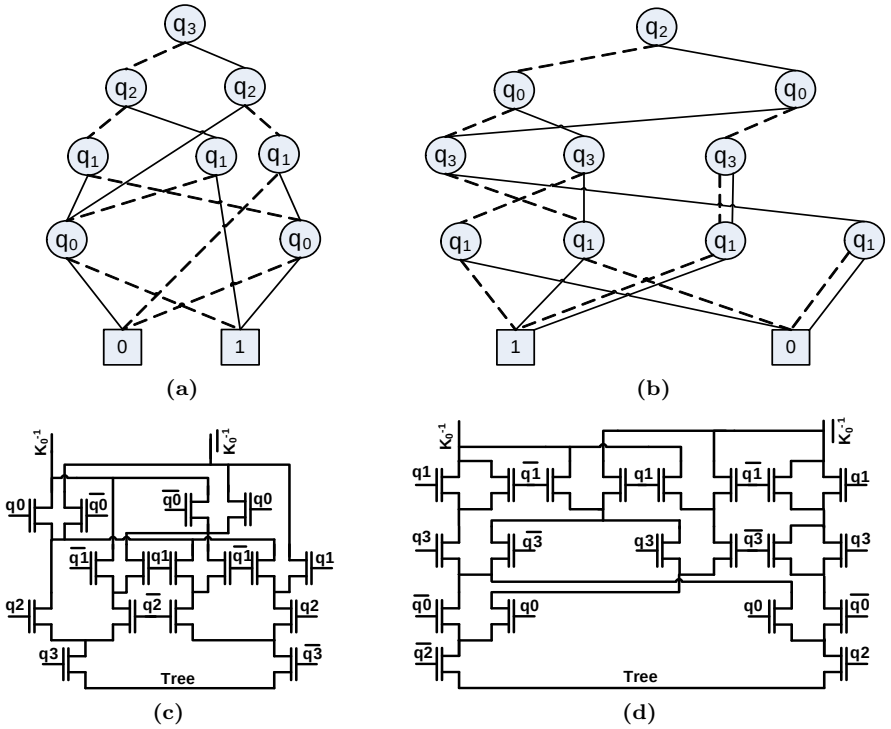


Fig. B.1.: BDD of the function k_0^{-1} in the Inversion $GF(2)^4$ block with the compact structure (a) and with the balanced structure (b). DPDN of the function k_0^{-1} in the Inversion $GF(2)^4$ block with the compact structure (c) and with the balanced structure (d).

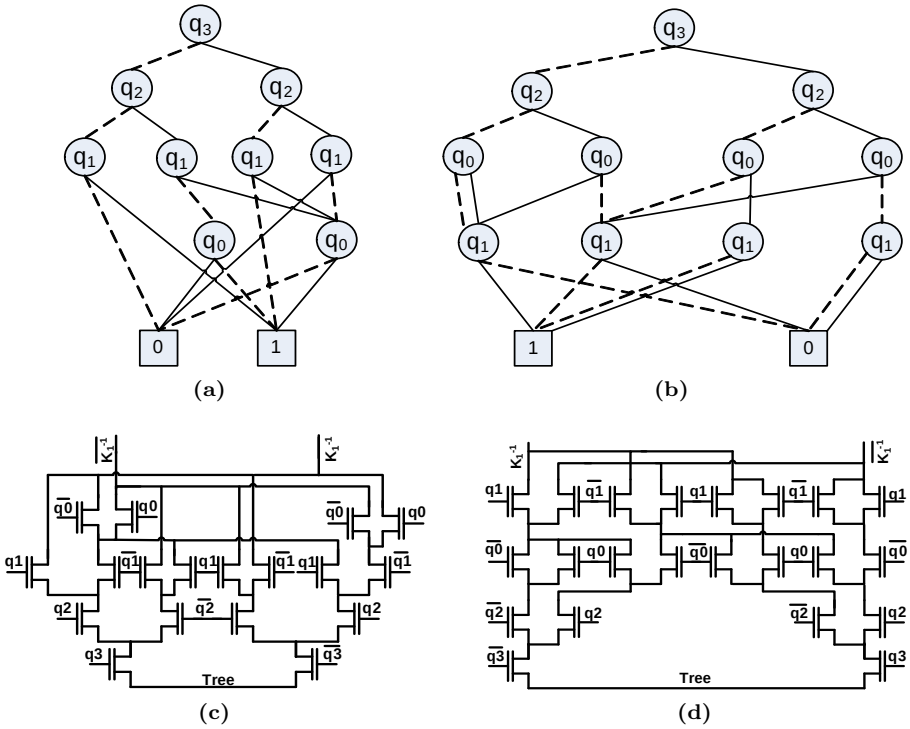


Fig. B.2.: BDD of the function k_1^{-1} in the Inversion $GF(2)^4$ block with the compact structure (a) and with the balanced structure (b). DPDN of the function k_1^{-1} in the Inversion $GF(2)^4$ block with the compact structure (c) and with the balanced structure (d).

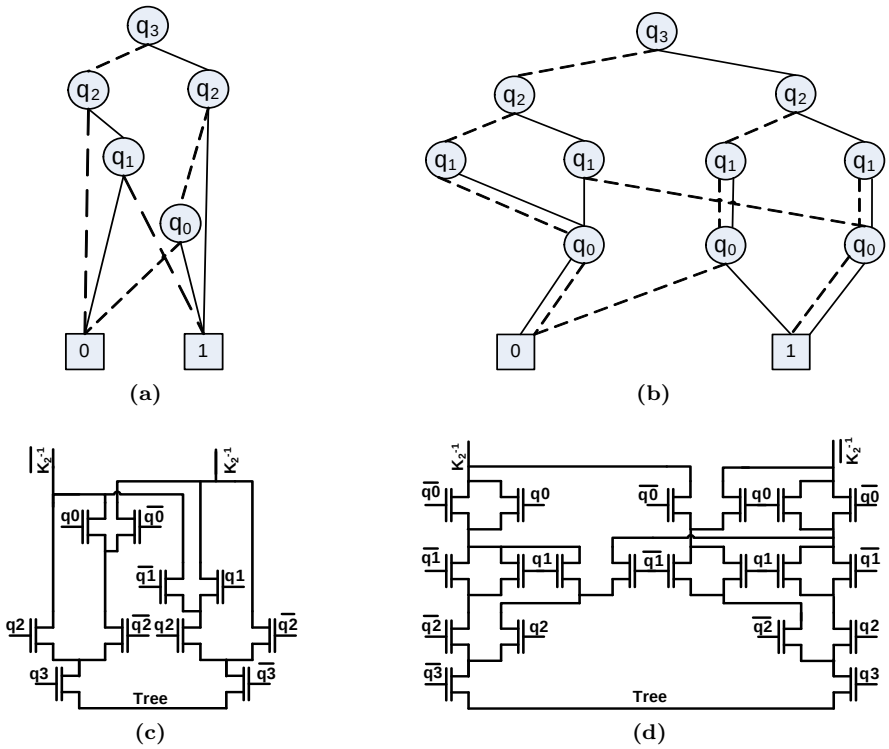


Fig. B.3.: BDD of the function k_2^{-1} in the Inversion $GF(2)^4$ block with the compact structure (a) and with the balanced structure (b). DPDN of the function k_2^{-1} in the Inversion $GF(2)^4$ block with the compact structure (c) and with the balanced structure (d).

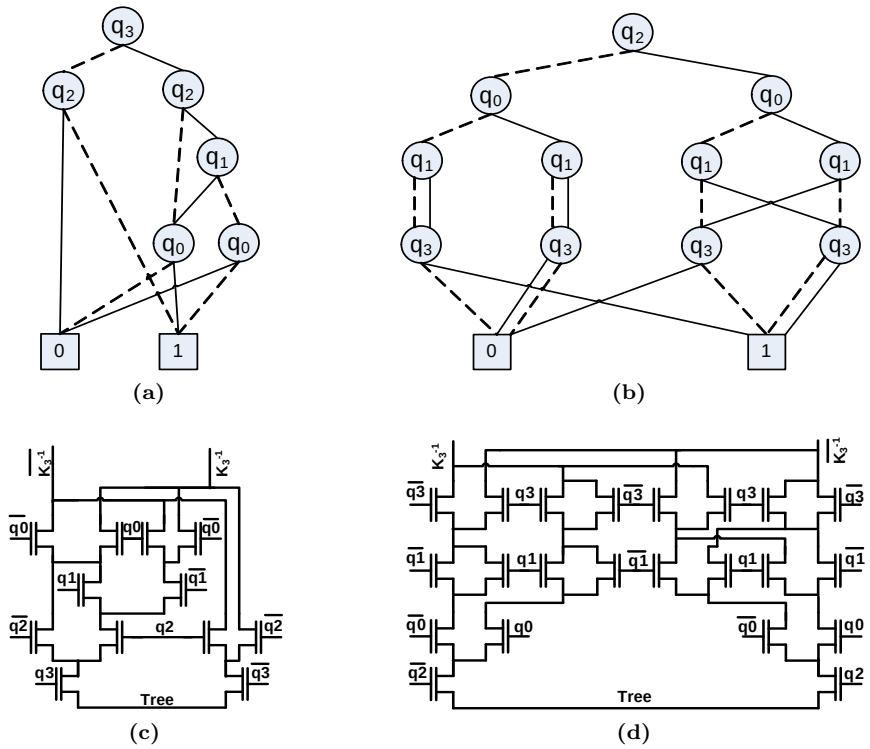


Fig. B.4.: BDD of the function k_3^{-1} in the Inversion $GF(2)^4$ block with the compact structure (a) and with the balanced structure (b). DPDN of the function k_3^{-1} in the Inversion $GF(2)^4$ block with the compact structure (c) and with the balanced structure (d).

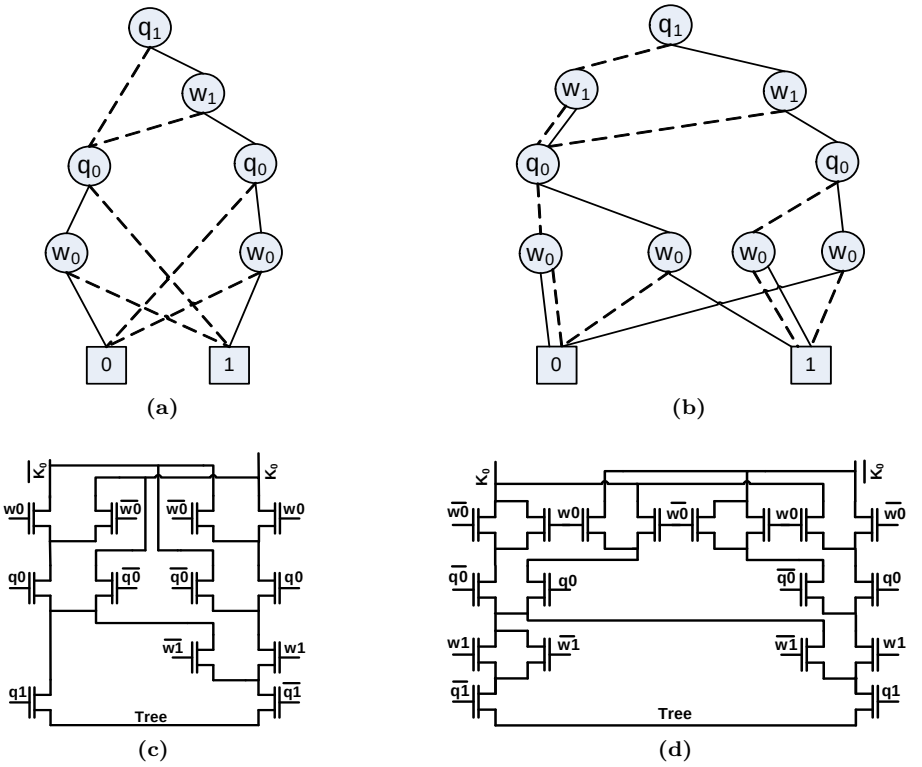


Fig. B.5.: BDD of the function k_0 in the Multiplication $GF(2)^2$ block with the compact structure (a) and with the balanced structure (b). DPDN of the function k_0 in the Multiplication $GF(2)^2$ block with the compact structure (c) and with the balanced structure (d).

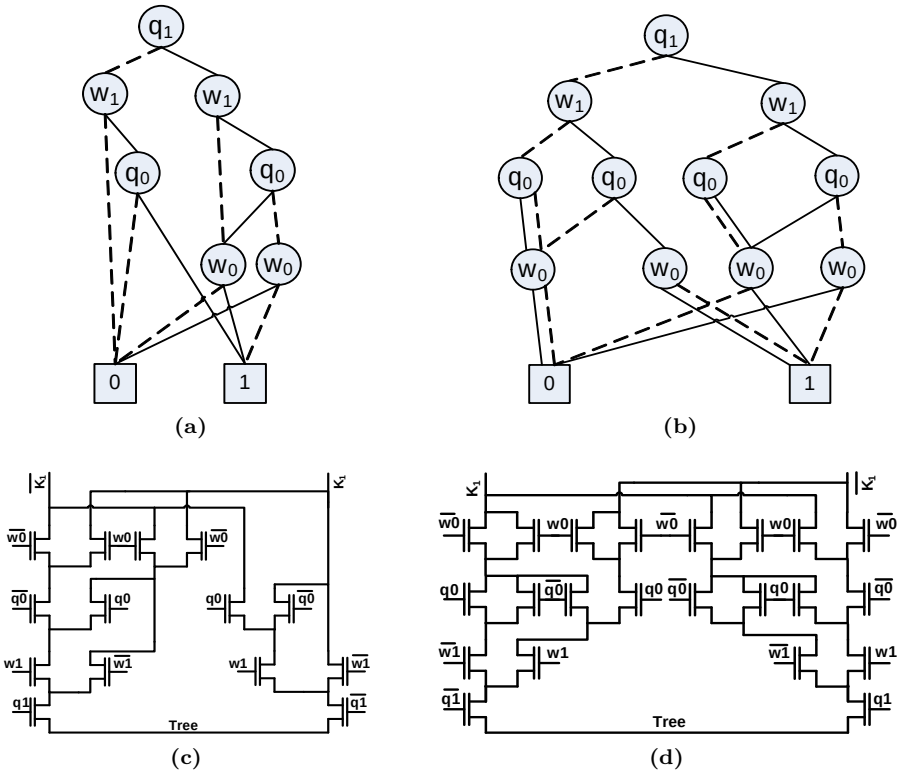


Fig. B.6.: BDD of the function k_1 in the Multiplication $GF(2)^2$ block with the compact structure (a) and with the balanced structure (b). DPDN of the function k_1 in the Multiplication $GF(2)^2$ block with the compact structure (c) and with the balanced structure (d).

APPENDIX C

DDSLL DESIGN DETAILS

The main goal of selecting the standard V_t (SVT) devices of the 65 nm LP CMOS technology for the majority of the DDSLL S-box transistors is derived from the study performed in [1] using a static CMOS S-box. Table C.1 gives the power consumption and delay simulation results (only gate and junction capacitances are accounted for) at 1.2 V for the static CMOS S-box implemented using the three different device types offered by the 65 nm LP CMOS technology, namely, low V_t (LVT), SVT and high V_t HVT devices. Using LVT devices causes the leakage power P_{leak} to dominate at 100 kHz, which is the minimum clock frequency for throughput/latency constraint in passive RFID tags [2] (79 %). Now, the use of SVT devices reduces the leakage power 10×, leading to a reduction of the total power (P_{tot}) by a factor of 3.6 at 100 kHz. As a result, SVT devices causes the dynamic power (P_{dyn}) to dominate at 100 kHz (27.8 %)[1]. Also, HVT devices minimize P_{leak} of the static CMOS S-box (P_{leak} is reduced 6.9× compared to the SVT version). However, P_{leak} of the SVT S-box is not dominating anymore, therefore, the total power consumption reduction at 100 kHz brought by the use of HVT devices is limited to 1.34×. As a result, the main benefit is obtained by using SVT devices which reduces the leakage power with a small delay penalty (29 % compared to the LVT S-box), whereas the HVT S-box only brings 1.34× P_{tot} reduction compared to the SVT version at the expense of 55 % increase in delay. Therefore, the use of SVT devices is considered a good choice that optimizes the power consumption - delay tradeoff. This is particularity important if the S-box were to operate at subthreshold since the delay penalty of the HVT S-box may violate the timing constraint of the target application. As a result of this study, we decided to employ the SVT devices in the design of the DDSLL S-box as well.

However, gate-level simulation results (only gate and junction capacitances are accounted for) at high temperature ($125°C$) show that the DDSLL S-box

[1] These results do not consider the parasitic routing capacitances that will definitely increase the contribution of P_{dyn}

Table C.1.: Power consumption and delay of a static CMOS S-box implemented using different types of transistors in 65 nm LP CMOS technology node at 1.2 V (only gate and junction capacitances are accounted for - extracted routing parasitics are not added).

Device type	P_{tot} at 10 MHz	P_{tot} at 1 MHz	P_{tot} at 100 kHz	P_{leak}	Delay
LVT	7.06	938 nW	326 nW	258 nW	1.82 ns
SVT	6.57	680 nW	90.6 nW	25.2 nW	2.35 ns
HVT	6.35	639 nW	67.2 nW	3.65 nW	3.65 ns

that uses only SVT devices, fail to function correctly. The main cause of this failure is linked to the basic mechanism of DDSLL which relies on floating output nodes during the evaluation phase. At high temperature, these output floating nodes are charged back to values close to V_{DD}, specially if the clock period is long enough (SPICE simulations are performed at 100 kHz and the evaluation period is 80 % (8 μs)). This leads to the failure of the output buffer to interpret the correct value of the output states. Consequently, we exchanged the SVT PMOS transistors of the latch (M_{12} and M_{13}) and the precharge transistors M_{10} and M_{11} with HVT devices as shown in Fig. C.1 to limit the charging of the floating output nodes.

The target of the design is to minimize the power consumption with tolerable delay penalty and good security improvement. Therefore, with respect to the transistor sizing, we chose the gate length (L_g) to be equal to the minimum feature size (0.06μm) in order to reduce the gate capacitance and hence the dynamic power consumption. The transistor widths of the DPDN and the dynamic current source transistors are $W = 0.3\mu m$ to obtain an average voltage swing of 0.6 V. The PMOS transistors of the inverters have $W = 0.24\mu m$ which is double that of the NMOS transistors of the inverter to maintain the duty cycle of the input clock (80 %). In all other transistors the width is chosen to be the minimum feature size which is $0.12\mu m$.

Another important challenge while designing the AES S-box with DDSLL is how to maintain a constant voltage swing independent on the input transition in order to satisfy the main security requirement, which is maintaining equal power traces each clock cycle. This is extremely difficult to achieve because of the imbalances present in the DPDNs. In order to explain further, first we need to revise how a DDSLL gate functions during evaluation. When the Clk_i signal is high, the dynamic current source is active and ready to discharge one of the output nodes (the inputs to the DPDN are already evaluated by the previous DDSLL gates). At the beginning, node $Tree$ is discharged and then one of the output nodes follows. When the output node discharges to a value below $V_{DD} - V_{tp}$, where V_{tp} is the threshold voltage of the PMOS transistors in the feedback circuit, the current is cut off from the dynamic current source keeping the discharged output and the $Tree$ nodes floating. Now, the voltage value of the node $Tree$ is determined by the resistance drop over the ON transistors of the

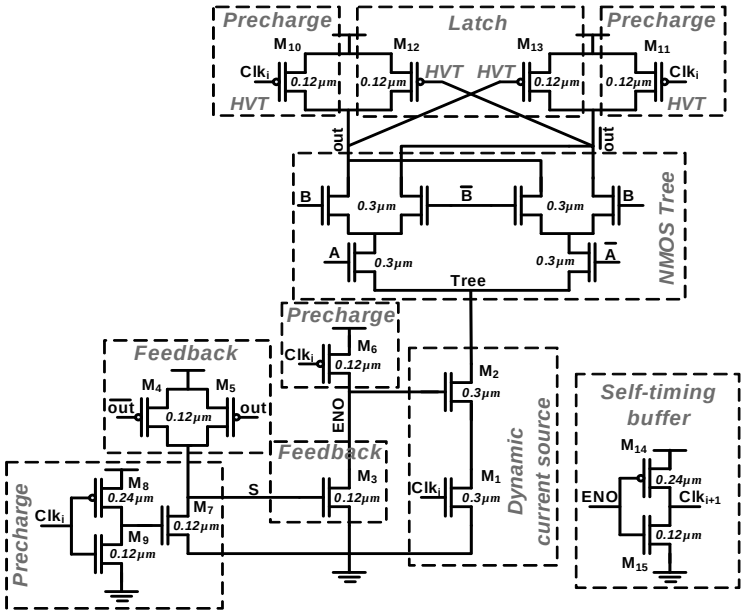


Fig. C.1.: DDSLL XOR schematic.

DPDN (*ON* path) which is different from one input state to the other. Therefore, unbalanced DPDNs lead to deviations in the voltage swing. This further demonstrates the difficulties encountered while designing DDSLL circuits.

Bibliography

[1] D. Kamel, F.-X. Standaert, and D. Flandre, "Scaling trends of the AES S-box low power consumption in 130 and 65 nm CMOS technology nodes," in *Proceedings of IEEE International Symposium on Circuits and Systems, ISCAS*, May 2009, pp. 1385 –1388.

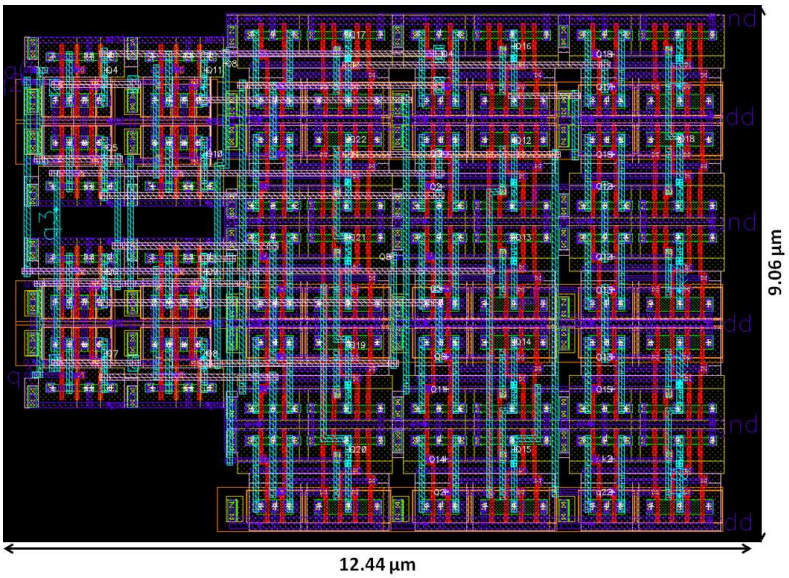
[2] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, "Strong authentication for RFID systems using the AES algorithm," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES*, 2004, pp. 357–370.

APPENDIX D

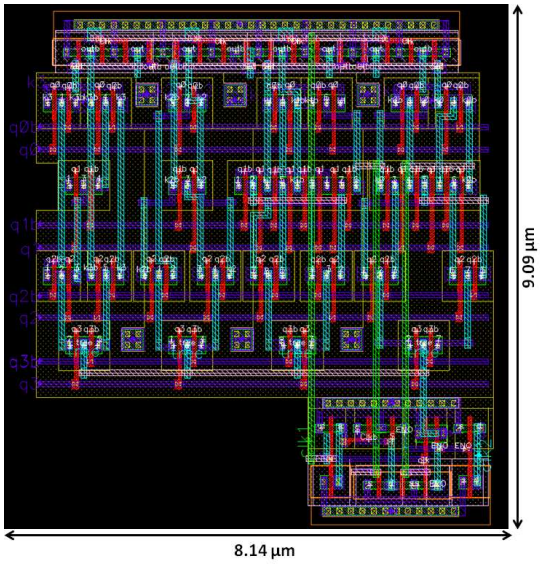
LAYOUT DESIGN

The placement and routing of all blocks of the static CMOS and DDSLL S-boxes are done in a full custom fashion. The layouts are implemented on a 65 nm LP CMOS technology using Cadence® Virtuoso®. All the layouts of the basic building blocks are designed in a full-custom manner. The layouts of the static CMOS S-box blocks are designed with the primary goal of optimizing the routing length such that the routing capacitances are minimized. On the other hand, the layouts of the DDSLL blocks are not that simple. Since DDSLL is a differential logic, special care has to be taken while routing the differential signals to maintain the symmetry between them. This is particularity important to avoid imbalance of the capacitances of the differential routes. Therefore, all differential wires are routed using the same metal layer, the same number of vias while connecting between metal layers and for similar distances. We also took care that the differential routes are at equal distances from other wires such that they experience similar crosstalk (although this is also a function of the signal activity carried by the neighboring wires). As for the clock signal, it is always routed using higher metal layers (unless otherwise needed) to minimize the crosstalk between the strong clock signal and the sensitive differential signals.

In Fig. D.1, we illustrate an example of the layout of a static CMOS and a DDSLL block which is the Inversion in $GF(2^4)$ function. Also, the layouts of the whole S-box implemented using static CMOS and DDSLL are shown in Fig. D.2.

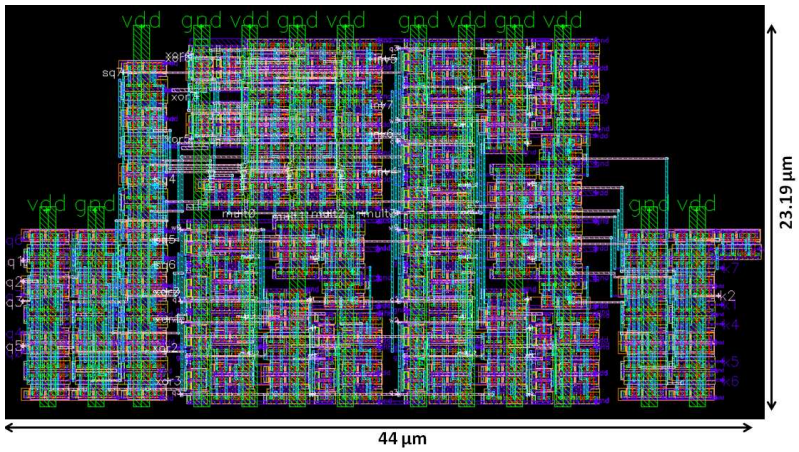


(a) Static CMOS.

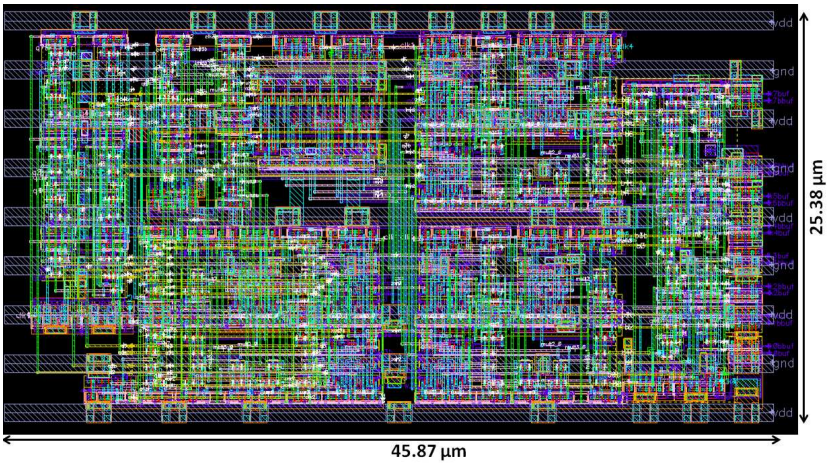


(b) DDSLL.

Fig. D.1.: Layouts of the Inversion $GF(2^4)$ block in (a) static CMOS and (b) DDSLL using 65nm LP CMOS technology.



(a) Static CMOS.



(b) DDSLL.

Fig. D.2.: Layouts of the S-box block in (a) static CMOS and (b) DDSLL using 65nm LP CMOS technology.

APPENDIX E

EXAMPLE OF A PHYSICAL LEAKAGE LINEAR FUNCTION

Usually the linearity of the leakage function is mainly observed in the amplitude of the traces for the different transitions. In Fig. E.1, the power traces of an arbitrary circuit are plotted for n different transitions. It shows that the power traces are distinguishably centered around 9 different groups which clearly indicates that the leakage function of this arbitrary implementation is linear.

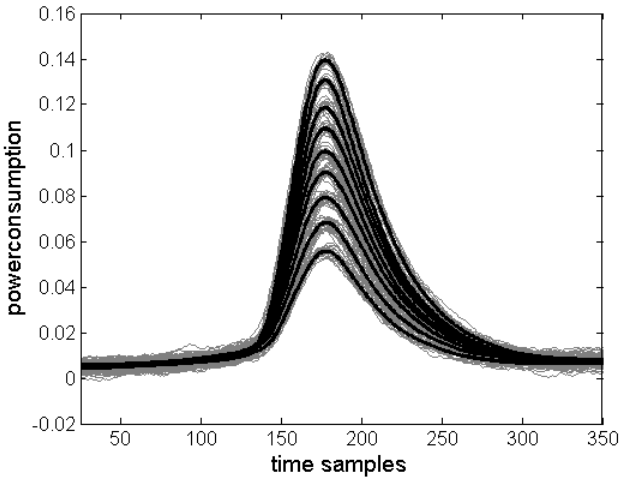


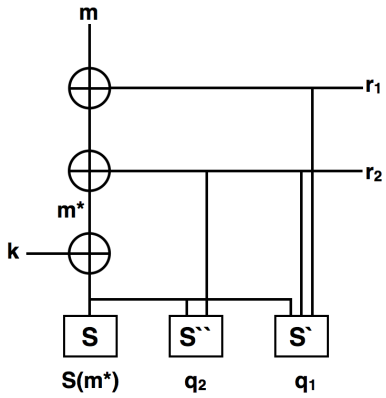
Fig. E.1.: Example of power traces of an arbitrary circuit showing n different transitions centered around 9 different groups to manifest the linearity of the physical leakage.

APPENDIX F

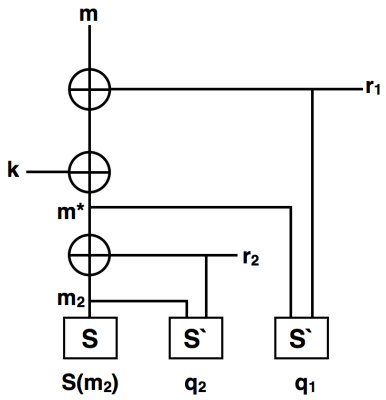
EXAMPLES OF MASKING SCHEMES

The main goal of masking is to randomize the bits of the masked implementation to increase its security against side-channel attacks. Generally, masking can be extended to d-order, i.e. d number of masking bits which leads to $d+1$ shares. According to [1, 2], a simplified leakage model of a higher-order masking scheme is proposed where a bit b is masked using a d random bits ($r_1,\ r_2,\ \ldots,\ r_d$) such that the masked bit is defined as: $q\ =\ b \oplus r_1 \oplus r_2 \ldots \oplus r_d$. The model is based on the assumption that the adversary is provided by $d+1$ leakage variables, since there are $d+1$ different shares. Using this leakage model, the number of samples required by the adversary to succeed in his attack (assuming all possible side-channel distinguisher) is lower bounded by an exponential function of the masking order d (only if sufficient noise exists). Masking can be done at different abstraction levels (e.g., the algorithm and gate levels). At the algorithm level, it can be implemented using software or hardware methods. Since in this study we focus only on hardware implementations, therefore we show some examples of hardware masking schemes at the algorithm level.

In Fig. F.1, we illustrate two examples of masking schemes at the algorithm level proposed in [3]. The first example (Fig. F.1 (a)) is a basic version where the cipher's S-box S is unchanged and mask updated functions (S' and S'' in the case of second-order masking scheme and of course it increases for higher orders of mask) are used to compute the output mask from the masked input to the original S-box and the input mask. The main drawback of this scheme is that a d-mask scheme requires the computation of functions with input space of size up to $(d+1).n$ bits (n is the number of input bits to the original S-box), which becomes infeasible as d grows. The second example in Fig. F.1 (b) demonstrates a more efficient masking scheme which limits the input space size to $2n$ bits instead of $(d+1).n$ bits, i.e., the mask updated functions are the same (S'). The main drawback of this scheme is that an attacker who is capable of observing the leakages relative to m^* and q_1 can retrieve the secret key.



(a) Basic version.



(b) More efficient version.

Fig. F.1.: Examples of higher order hardware masking schemes (order = 2) at the algorithmic level [3].

At the gate level, masking can be implemented by modifying in the logic style in order to incorporate the masking bits. For example, masked dual-rail with precharge logic (MDPL) [4] is a DDL logic that features masking at the gate level. The basic idea of MDPL is to use masked cells to randomize the logic signals in the circuit which is a different approach from trying to perfectly balance the differential signals. All signals in an MDPL circuit are masked with the same mask that is generated by a pseudo random-number generator (PRNG) and it changes every clock cycle. To implement a function, MDPL needs to duplicate the block in order to have both true and false outputs. Also, each 2-input function is transformed to a 3-input function to accommodate the mask bit. The design of MDPL gates merely depends on an MDPL AND gate shown in Fig. F.2 (a) that uses the majority function shown in Fig. F.2 (b).

However, practical SCAs against MDPL have been reported in literature [5, 6]. It is claimed in [7] that the MDPL logic suffers from the well known weakness of early propagation effect, which proves to be the cause of side-channel information leakage in [5, 6]. As a result, MDPL is not an optimum choice for a masking scheme at the gate level, but it serves as an example of how masking can be incorporated in a logic style.

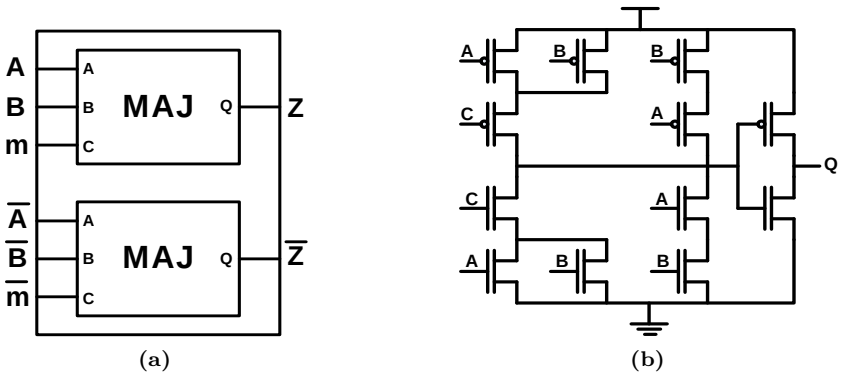


Fig. F.2.: The schematics of (a) a MDPL AND gate and (b) a static CMOS majority gate.

Bibliography

[1] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi, "Towards sound approaches to counteract power-analysis attacks," in *CRYPTO*, 1999, pp. 398–412.

[2] M. Rivain and E. Prouff, "Provably secure higher-order masking of AES," in *Proceedings of the 12th international conference on Cryptographic hardware and embedded systems, CHES*. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 413–427. [Online]. Available: http://dl.acm.org/citation.cfm?id=1881511.1881549

[3] G. Piret and F.-X. Standaert, "Security analysis of higher-order Boolean masking schemes for block ciphers (with conditions of perfect masking)," *Information Security, IET*, vol. 2, no. 1, pp. 1 –11, Mar. 2008.

[4] T. Popp and S. Mangard, "Masked dual-rail pre-charge logic: DPA-resistance without routing constraints," in *Proceedings of Cryptographic hardware and embedded systems, CHES*, 2005, pp. 172–186.

[5] T. Popp, M. Kirschbaum, T. Zefferer, and S. Mangard, "Evaluation of the masked logic style MDPL on a prototype chip," in *Proceedings of the 9th international workshop on Cryptographic Hardware and Embedded Systems, CHES*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 81–94. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-74735-2_6

[6] E. D. Mulder, B. Gierlichs, B. Preneel, and I. Verbauwhede, "Practical DPA attacks on MDPL," Cryptology ePrint Archive, Report 2009/231, 2009, http://eprint.iacr.org/.

[7] D. Suzuki and M. Saeki, "Security evaluation of DPA countermeasures using Dual-Rail Pre-charge Logic style," in *Proceedings of Cryptographic Hardware and Embedded Systems, CHES*, 2006, pp. 255–269.