



UNIVERSITÉ CATHOLIQUE DE LOUVAIN
LABORATOIRE DE MICROÉLECTRONIQUE
Louvain-la-Neuve

Design and optimization of digital circuits for low-power and security applications

Ilham Hassoune

*Thèse présentée en vue de l'obtention du grade de
docteur en Sciences Appliquées*

Jury

Prof. Jean-Didier Legat (UCL/DICE)-promoteur
Prof. Denis Flandre (UCL/DICE)-promoteur
Prof. Christian Pigué (EPFL et CSEM, Suisse)
Prof. Amara Amara (ISEP, France)
Dr. Amaury Neve (DG Recherche, CE)
Prof. Luc Vandendorpe (UCL/TELE, président)

JUNE 2006

Abstract

Since integration technology is approaching the nanoelectronics range, some practical limits are being reached. Leakage power is increasing more and more with the continuous scaling, and design of clock distribution systems needs to be reconsidered as it becomes difficult to deal with performance and power consumption specifications while keeping a correct synchronisation in modern multi-GHz systems. The ongoing technology trend will become difficult to maintain unless dedicated library cells, new logic styles and circuit methods are emerging to prevent the drawbacks of future nanoscale circuits.

In this thesis we investigate a new class of dynamic differential logic family that features a self-timed operation and low output logic swing. The latter contributes to reduce dynamic power, while the self-timing scheme alleviates the drawbacks of synchronous circuits and systems.

Furthermore, the dynamic and differential nature of LSCML class brings advantages in terms of reduction of the power consumption variation and thus gives LSCML an additional potential for implementation of secure encryption devices against attacks based on power analysis.

We investigate dynamic and leakage power reduction at the cell level through the application of low-power low-voltage techniques to a new hybrid full adder structure. The 8b RCA circuit based on the ULPFA (ultra low power full adder) version of this full adder, achieves a total power and a leakage power, which are both reduced by 50% compared to the 8b RCA implemented with conventional static CMOS full adder, while featuring better power delay product.

Acknowledgment The research on the application of the LSCML class in hardware implementation of secure devices was done in collaboration with the crypto-group at UCL, Belgium.

Acknowledgments

First of all, I would like to thank my promoters, Prof. Jean-Didier Legat and Prof. Denis Flandre, who offered me the opportunity to carry out this research and allowed me to spend 42 months at the microelectronics laboratory at UCL. This thesis would certainly not have been what it is without the numerous advices and comments they brought to me.

My gratitude to Prof. Christian Piguet and Prof. Amara Amara for the time they spent reading this thesis and for their fruitful comments on this research. It is a real pleasure to have them as members of my evaluation committee.

I also thank Dr. Amaury Neve for his advices on branch based design and for the time he spent reading my thesis. I would also like to thank Prof. Luc Vandendorpe for chairing the evaluation committee.

I sincerely thank Aimad Saib for helping me with L^AT_EX.

It is a real pleasure for me to thank members of DICE labs for making it an enjoyable place of work. I thank also my colleagues and all persons I worked with.

I would like to heartily thank Lucien Pierret for his support and encouragement.

Finally I would like to thank my family and friends for their encouragement, with a deep thought to my mother.

TABLE OF CONTENTS

Abstract	5
Scientific publications	1
1 Introduction	1
1.1 Motivation of the work: Low power challenges	1
1.2 High performance arithmetic circuits	3
1.3 Low power architectures: Clocking considerations	4
1.4 Security of encryption components	6
1.5 Thesis contributions	8
1.6 Outline	9
2 State of the Art	13
2.1 Introduction	13
2.2 Power-aware design criteria	13
2.2.1 Low leakage power circuit techniques	14
2.2.1.1 Leakage control by body biasing	14
2.2.1.2 Leakage control by MTCMOS technique	14
2.2.1.3 Leakage control by MVMOS technique	15
2.2.2 Dynamic power management: Clock gating	15
2.2.3 Power-aware logic styles	16
2.2.4 Power-aware architectures: asynchronous ver-	
sus synchronous	27
2.3 Arithmetic adders	31
2.3.1 Ripple carry adder	31
2.3.2 Conditional sum adder	32
2.3.3 Carry lookahead adder	32
2.3.4 Power aware full adders	34
2.3.4.1 Static CMOS and pass-transistor logic	
full adders	34
2.3.4.2 Hybrid full adders	38
2.4 Security ICs criteria	43

2.4.1	Self-timed design to prevent DPA	43
2.4.2	Dynamic differential logic to prevent DPA	44
2.5	Conclusion	48
3	Class of low Swing Current Mode Logic styles	53
3.1	Introduction	53
3.2	LSCML structure and operation	53
3.3	LSCML gate cascading	54
3.3.1	Clock buffering with simple inverters	54
3.3.2	Clock buffering with ST1 scheme	56
3.3.3	Clock buffering with ST2 scheme	56
3.4	Interfacing with full-swing logic styles	61
3.5	Functioning behaviour of LSCML circuits	61
3.6	Modified LSCML gate (IFLSCML)	64
3.7	Further optimizations of the LSCML	64
3.8	Stability of LSCML class operation	65
3.9	conclusion	70
4	Investigated applications of the LSCML style	73
4.1	Introduction	73
4.2	Power analysis attacks	73
4.3	Figures of merit	74
4.4	Khazad S-box	76
4.5	Logic styles selected for comparison	77
4.5.1	DDCVSL(ST) versus DDCVSL(CD)	80
4.5.2	DyCML versus SABL	82
4.6	LSCML based logic gates	83
4.7	8b RCA circuit	84
4.8	8b CLA circuit	86
4.9	Khazad S-box comparisons	89
4.9.1	Impact of the output swing variation	92
4.9.2	ST2 versus ST1	93
4.9.3	Impact of the fanout mismatch	95
4.9.4	Impact of the output swing value	95

4.10 Conclusion	97
5 ULPFA: a power aware hybrid full adder	101
5.1 Introduction	101
5.2 Branch-Based design	102
5.3 The BBL-PT hybrid full adder	102
5.4 The BBL-PT FA vs the static CMOS FA	107
5.5 MTCMOS and DT MOS circuit techniques	107
5.6 Application to the BBL-PT FA: MTCMOS technique . .	109
5.7 The BBL-PT full adder with DT MOS devices	110
5.8 The BBL-PT FA with DT MOS devices and high V_t . . .	114
5.9 Advances in the Hybrid full adder	115
5.9.1 Conventional level restorer	115
5.9.2 ULP diode based level restorer	116
5.9.3 Feasibility in $0.13\mu\text{m}$ PD SOI/CMOS	120
5.9.4 LP XOR/XNOR gates	122
5.9.5 The Ultra Low Power Full adder	123
5.9.6 Static power in the ULPFA	130
5.9.7 ULPFA based 8-bit RCA	132
5.10 Conclusion	136
6 Conclusions	139
A LSCML circuits evaluation: Transistor sizes	145
B Hybrid full-adder evaluation: Transistor sizes	149
C 4 bit branch based CLA Synthesis	155

Scientific Publications

Articles in periodicals

- [1] I. Hassoune, F. Mace, D. Flandre, and J.-D. Legat. Dynamic differential self-timed logic families for robust and low-power security ICs. *Accepted for publication in Integration, the VLSI Journal*, 2006.
- [2] 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. *Article in press in the Microelectronics Journal*, 2006.
- [3] I. Hassoune, A. Drummond, A. GAUDISSERT, D. Bol, D. Levacq, J.-D. Legat, and D. Flandre. A new multi-valued current-mode adder based on negative-differential resistance using ULP diodes. *Journal Of Solid-State Electronics*, Volume 49/7, pp.1185–1191, July 2005.

patent

I. Hassoune and J.-D. Legat. Low swing current mode logic family. *Patent number: WO2005112263, Priority number: US20040571383P 20040514*, November 2005.

Conference Proceedings

- [1] I. Hassoune, J.-D. Legat, and D. Flandre. Hybrid full-adder cell in $0.13\mu\text{m}$ PD SOI CMOS for low-voltage low-power applications. In *in the Workshop of EUROSIOI 2005*.
- [2] I. Hassoune, D. Levacq, A. Drummond, A. GAUDISSERT, J.-D. Legat, and D. Flandre. A hybrid full-adder cell for high-performance and low-power design. In *Proceedings of the 11th EDS 2004 conference*.

- [3] I. Hassoune, A. Neve, J.-D. Legat, and D. Flandre. Branch-based design of carry calculation cell for ultra-low power and high-temperature applications. In *Proceedings of HITEC 2004 conference*.
- [4] I. Hassoune, A. Neve, J. Legat, and D. Flandre. Investigation of low-power low-voltage circuit techniques for a hybrid full-adder cell. In *Proceedings of PATMOS 2004 conference*, pages 189–197.
- [5] F. Mace, F.-X. Standaert, I. Hassoune, J.-D. Legat, and J.-J. Quisquater. A dynamic current mode logic to counteract power analysis attacks. In *Proceedings of DCIS 2004 conference*, pages 186–191.

CHAPTER 1

INTRODUCTION

1.1 Motivation of the work: Low power challenges

Power-aware design is becoming the major trend for mobile and embedded applications nowadays. Moreover, since the technology is moving toward the $0.1\mu\text{m}$ generations and beyond, embedded VLSI systems are meeting some issues that affect power consumption. Continuous technology scaling is increasing more and more the on-chip integration density, hence raising the cost of packaging and cooling. Moreover, the scaling of V_{dd} requires a subsequent V_t reduction to maintain sufficient speed performance. As a consequence, the leakage current I_{off} is steadily increasing and static power consumption is expected to become as significant as dynamic power in technology generations ahead [1]. Besides the previously reported circuit techniques for leakage power management and/or minimization of dynamic power, dedicated library cells and logic styles should also be considered to face this issue.

In CMOS digital circuits, the total power dissipation includes two main components: the dynamic power and static power. Nose et al. [2] have investigated the power consumption trend in VLSI design by carrying out an estimation through closed-form formulas for V_{dd} and V_{th} using typical device parameters in VLSI design. This study sets the dynamic power consumption to more than 70% of total power and the static power to less than 30% of total power (for a channel length range of $0.07 - 0.05\mu\text{m}$). However, in CMOS digital circuits, these predicted amounts may vary according to the chosen logic style, architectural topology and the possible circuit techniques used to manage leakage power. Each component of total power in turn includes many components [3]. The dynamic power is due to three sources: the switching power, the short-circuit power and the glitching power.

The switching power is the power consumed by a logic gate to charge the

parasitic capacitance during power-consuming transitions. This component is expressed as follows [3]:

$$P_{dyn} = \alpha_{0 \rightarrow 1} \cdot C_L \cdot V_{dd} \cdot V_{swing} \cdot f \quad (1.1)$$

where $C_L = \Sigma C_G + \Sigma C_J + \Sigma C_{INT}$ (C_G , C_J and C_{INT} denote gate, junction and interconnection capacitances), V_{dd} is the supply voltage, V_{swing} the output logic swing ($V_{swing} = V_{dd}$ for the logic families with a full output swing), f the operating frequency and $\alpha_{0 \rightarrow 1}$ the activity factor. The short-circuit power results from the direct path short-circuit current which occurs when both the NMOS and PMOS transistors are turned ON during logic transitions thereby creating a direct path from the supply voltage V_{dd} to ground. The expression that shows the dependency of short-circuit power on some operation and device parameters in a CMOS inverter is given by [3]:

$$P_{sc} = \frac{\mu C_{ox}}{12} \frac{W}{L} \cdot (V_{dd} - 2V_{th})^3 \cdot \tau \cdot f \quad (1.2)$$

where μ is the carrier mobility, C_{ox} the oxide capacitance, τ is the rise/fall time of the input signal. This equation is valid under two assumptions: $V_{th_n} = V_{th_p} = V_{th}$ and $\beta_n = \beta_p = \mu C_{ox} \frac{W}{L}$.

The contribution of short-circuit currents to the overall power consumption is limited but not negligible ($\approx 8 - 10\%$), except for very low V_{dd} . This contribution is expected to decrease with the ongoing technology trend due to the decrease of $(V_{dd} - V_t)$.

The third component of the dynamic power is the glitching power. This is due to internal delay discrepancies from one logic block to the next. This problem can be solved by balancing the delay paths on logic block inputs, i.e. by inserting buffers on fast paths and by a careful layout of input signals lines [3].

The static power is the power consumed when no transition occurs. It is due to two sources: the leakage power and the DC power. Leakage power arises from the subthreshold current, tunnelling gate current and substrate injection. The subthreshold current is the dominant component

of the off-state leakage in deep-submicron devices. This current occurs for gate-to-source voltage (V_{gs}) values below V_{th} . The drain current in subthreshold region is expressed as follows [4]:

$$I_s = I_0 \exp\left(\frac{V_{gs} - V_{th}}{nV_T}\right) \left(1 - \exp\left(\frac{-V_{ds}}{V_T}\right)\right) \quad (1.3)$$

where V_{ds} is the drain to source voltage, $V_T = \frac{KT}{q}$ is the thermal voltage, $I_0 = \mu_0 C_{ox} \frac{W}{L} V_T^2 e^{1.8}$ and $n = 1 + \frac{C_d}{C_{ox}}$ where C_d is the depletion layer capacitance of the source/drain junction [4]. The DC power component is due to the existence of a constant direct path between V_{dd} and ground, which is only present in few logic styles like pseudo-NMOS logic or MOS current mode logic (MCML).

1.2 High performance arithmetic circuits

In many synchronous implementations of microprocessors, the adder lies in the critical path because it is a key element in a wide range of arithmetic units including ALU, multiplier and many other arithmetic operations. Because the critical path determines the overall synchronous system performance, dedicated adder or multiplier architectures have been investigated particularly for digital signal processors in embedded applications where a high-speed capability must cope with low-power management.

Better adder performance can be achieved by efficiently implementing the carry propagation chain. This can be addressed by either improving the structure of 1-bit full-adder which is one of the basic cells in some adders like carry select and carry skip adders and the building block of the ripple carry adder (RCA) since a n-bit RCA is formed by $n \times 1$ -bit full-adder, or by using improved fast adder architectures like carry lookahead (CLA) and conditional sum (CSA) adders.

Speed performance has often been addressed in some multiplier architectures by the use of specific fast adders such as those mentioned above in design of the final adder that is needed to sum the reduced partial prod-

ucts. Among fast architectures for multiplier, Booth and Wallace tree multipliers are well-known and often used in DSP applications. Booth multiplier is composed of multiplier array containing partial products generation and 1-bit (half and full) adders, the Booth encoder and the final stage adder. In Wallace tree multiplier, 1-bit full-adders form a tree of several layers to reduce the partial products before a final carry propagation in the final stage adder. In these multipliers, the 1-bit full-adder is a main building block and enhancing its structure and performance helps to meet the required capabilities at the architecture level.

1.3 Low power architectures: Clocking considerations

There are two classes of clocking strategies in digital VLSI design. Synchronous systems define a class of circuits that are controlled by a global distributed periodic signal, namely the clock signal. The operations of all individual modules are executed in a certain order when the clock-event occurs. This global clocking strategy introduces temporal constraints to each element to insure a correct operation. Conversely, in asynchronous systems, there is no global clock and then there is no temporal link between individual modules operation. Asynchronous systems operate according to synchronization of events occurrences whatever the time at which they occur.

In modern multi-Gigahertz technologies, in order to meet the power consumption constraints, the clocking has become one of the most important considerations when designing digital systems. Indeed, in large chips, the clock distribution is a major contributor in power dissipation [5] [6].

Synchronous digital VLSI design is still so popular because it has reached such maturity in terms of automation of design methodologies and synthesis tools [7]. As a consequence, industry is still reluctant to adopt a new class of digital VLSI design. Nevertheless, over the last twenty years, asynchronous architectures and systems have raised a great deal

1.3 LOW POWER ARCHITECTURES: CLOCKING CONSIDERATIONS

of interest in the research community. Due to the rapid growth of chip sizes and packing, it becomes difficult to synchronize perfectly all the part of SoCs (Systems-On-Chip). Moreover, design of clock distribution trees for synchronous systems will not be optimized anymore to be tailored to multi-GHz frequencies. Hence, asynchronous architectures are being emerging as a solution to synchronization failures.

A fundamental feature that characterizes asynchronous circuits is the communication protocol used to exchange information between individual computation blocks. This communication protocol allows a local synchronization between connected blocks independently of the other parts of the chip while each block operates at its own rate. This offers a larger flexibility to designers in comparison to synchronous design where it is difficult to deal with temporal constraints.

Unlike synchronous systems where the sequence of tasks of individual blocks is controlled by a global clock, asynchronous systems generally use two signals: “Request” and “Acknowledge” for local synchronization (Figure 1.1). An asynchronous module starts the data processing once it receives the “Request” signal and generates an “Acknowledge” signal to report the end of the evaluation and that output data are available. This requires both a computation block which starts the data processing once the “Request” signal is received and generates a completion signal when the computation is achieved, and an eventual additionnal circuitrie (interconnection blocks) that controls the synchronization between the computation blocks. The handshaking protocol is the most basic and common implementation of these interconnection blocks. Among the solutions used to generate the completion signal in the computation blocks in asynchronous modules, one of these called “delay scheme” consists in estimating the timing on the critical path of the computation block that achieves the data processing. This timing is afterwards used to size the delay (generally in form of two inverters in series) to be introduced between the request and the complete signals. Though it is a well known and often used technique in asynchronous modules, the delay scheme introduces timing constraints since the completion signal is

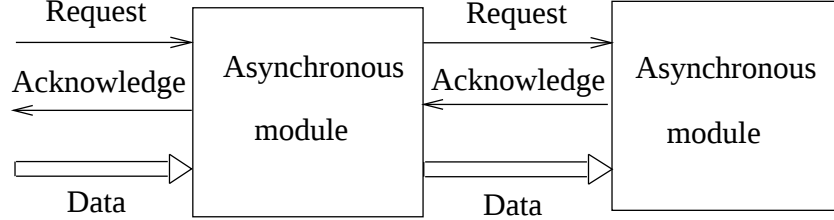


Figure 1.1: Communication between asynchronous modules in form of request/Acknowledge signaling.

generated at a constant time without any consideration to the possible variation of the evaluation time. This makes the delay scheme similar to the synchronous approach. The use of self-timed logic to implement the computation blocks is an efficient way to generate the completion signal because this signal is inherent to this class of logic. Unlike in the delay scheme, the timing delay of the completion signal in self-timed logic may vary according to the processed data and the path they take without affecting the functionality. This attributes to the self-timing approach more robustness to variation of some parameters like environmental parameters (supply voltage, temperature) or device parameters. It should be emphasized that both the delay scheme and self-timed logic can be used in either asynchronous or synchronous architectures.

1.4 Security of encryption components

Cryptography components such as smart cards have found their way in a broad range of applications, and the topic of their security at the hardware level has recently gained importance in the research community. Encryption circuits can indeed leak information, exploitable through attacks such as power consumption analysis, timing analysis and electromagnetic emission analysis. Kocher et al. [8] have indeed shown that variations in some electrical characteristics of the encryption device can

be correlated to the input data and hence to the secret key. These links can thus be used to mount particularly efficient attacks against physical implementations of cryptographic algorithms. These attacks are referenced to as side-channel attacks (SCAs). Among electrical characteristics that can leak information in cryptosystems, three of these are particularly well-known:

- The power consumption of the encryption circuit. It can be either an SPA (simple power analysis) or a DPA (differential power analysis) attack. SPA attack can reveal information about the sequence of instructions being executed. If the execution path is data dependent, this information can be used to break the security of the cryptosystem algorithm [8]. DPA attack uses the correlation of variation in power consumption to the processed data. Even small data dependent variation can be used to recover the secret key. This is achieved by using statistical functions that are tailored to the targeted cryptographic algorithm.
- The execution time of a cryptographic algorithm can leak information if it is data dependent [9].
- Electromagnetic emissions of the encryption circuit can be used as a side-channel information through an EMA (Electromagnetic analysis) attack which is an equivalent of the power analysis attack [10].

Power analysis attacks that rely on measurements of instantaneous power consumption and statistical analysis are cheap and easy to mount [8]. The threat of DPA which was reported in the literature as the most powerful among side-channel attacks [11] has generated a growing focus of cryptographers and hardware designers on the material security issue. Indeed, the few steps to achieve are the collection of many power consumption traces of the chip realizing the cryptographic operation and their correlation with prediction of this power consumption to reveal the secret key. There is no need to know details about the sequence of instructions being executed as it is the case in SPA attack. To counteract

this attack, several software countermeasures were proposed such as data masking using random boolean values, or the addition of random delays by using random processes, pseudo-instructions and random interrupts. These random delays introduce a shifting of operations in time, thereby making the statistical analysis more difficult. Other solutions like the adjunction of random noise by connecting noisy structures to the power supply, have been suggested to mask the power consumption signature. However, it was also reported in the literature that such “randomization” may be overcome by efficient attacks.

Other works have proposed to use specific circuit design methodologies [12] [11] or logic families [13] [14] to face efficiently this issue at the cell level.

1.5 Thesis contributions

The major contributions of the present thesis are summarized in the outline hereafter:

- We propose a new dynamic differential self-timed logic style. The low swing current mode logic (LSCML) family features a low swing operation that helps to reduce the dynamic power. This latter is meaningful in dynamic logic based circuits. Several progressive optimizations at the schematic level have been carried out from the first proposed version of the LSCML [15] in order to improve the power delay product. Simulations in $0.13\mu\text{m}$ PD SOI CMOS technology that we carried out on adders and Khazad S-Box, based on this logic family have shown a good behaviour with regard to power consumption, speed performance and the reduction of the differential power signature. We believe that the LSCML logic family can be interesting for implementation of secure and robust smart cards against differential power analysis (DPA) attacks.
- We propose a new design of full adder cell that combines branch-based logic (BBL) and pass transistor logic. The hybrid cell namely

the BBL-PT FA proposed in [16] has shown encouraging simulation results when compared with other state-of-the art full adder designs. Further advances in the hybrid cell have achieved more enhancements regarding to both power and speed.

1.6 Outline

This thesis is structured as follows. Chapter 2 presents a state-of-the-art related to the ongoing research on power management techniques. We discuss circuit techniques reported in the open literature, used to save power at the architecture and logic levels. More specifically, we discuss the constraints brought by synchronous design versus its asynchronous counterpart. At the logic level, a qualitative comparison of the prior art in logic styles and their capability regarding to the application targets. We also present a brief review of some popular adder architectures before we carry out a survey of previously reported binary full adder designs. Finally, we end this chapter with a brief overview of the most common countermeasures undertaken at the logic level in order to counteract power analysis attacks.

Chapter 3, 4 and 5 constitute our personal contribution. In chapter 3, we propose a new logic style called LSCML. We describe the progressive optimizations carried out on the LSCML logic family for further improvements to achieve trade-off between power dissipation and speed. In chapter 4 we evaluate it for both low power adders and security ICs applications.

Chapter 5 presents a new full adder design that we evaluate through qualitative and quantitative comparison versus other state-of-the-art full adders. Its evolution from the original proposed version up to an ultra-low-power cell is also described. We also investigate some common low power low voltage circuit techniques through their application at the cell level.

In chapter 6, we discuss some challenges brought by the continuous technology scaling and its impact on digital VLSI design before we conclude

CHAPTER 1. INTRODUCTION

this thesis by an overview of possible further prospects of our work.

References

- [1] N. S. Kim, T. Austin, D. Blaauw, T. Mudge, K. Flautner, J. S. Hu, M. J. Irwin, M. Kandemir, and V. Narayanan, "Leakage current: Moore's law meets static power," *IEEE J. Computer*, vol. 36, pp. 68–75, December 2003.
- [2] K. Nose and T. Sakurai, "Optimization of V_{dd} and V_{th} for low-power and high speed applications," in *Proceedings of the conference on Asia South Pacific design automation*, ACM Press, 2000, pp. 469 – 474.
- [3] M. Anis and M. Elmasry, *Multi-Threshold CMOS digital circuits, managing leakage power*, Kluwer Academic Publishers, 2003.
- [4] R. X. Gu and M.-I. Elmasry, "Power dissipation analysis and optimization of deep submicron CMOS digital circuits," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 5, pp. 707–713, May 1996.
- [5] J. Schutz, "A 3.3V 0.6 μ m BiCMOS superscalar microprocessor," in *Proceedings of ISSCC Dig. Tech. Papers*, February 1994, pp. 202–203.
- [6] Suresh Rajgopal, "Challenges in low-power microprocessor design," in *Proceedings of the 9th Inter. Conference on VLSI Design: VLSI in mobile communication*, January 1996, pp. 329–330.
- [7] M. Renaudin, B. El Hassan, and A. Guyot, "A new asynchronous pipeline scheme: Application to the design of a self-timed ring divider," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 7, pp. 1001–1012, July 1996.
- [8] P. C. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proceedings of CRYPTO conference, LNCS 1666*, Springer-Verlag, 1999, pp. 388–397.
- [9] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, RSA, DSS and other systems," in *Proceedings of CRYPTO conference, LNCS 1109*, Springer-Verlag, 1996, pp. 104–113.

- [10] J. J. Quisquater and D. Samyde, “Electromagnetic analysis (EMA): Measures and countermeasures for smart cards,” in *Proceedings of E-smart conference, LNCS 2140, Springer-Verlag 2001*, 2001, pp. 200–210.
- [11] Z. C. Yu, S. B. Furber, and L. A. Plana, “An investigation into the security of self-timed circuits,” in *Proceedings of Int’l Symp. Asynchronous Circuits and Systems (ASYNC)*, 2003, pp. 206–215.
- [12] D. Sokolov, J. Murphy, A. Bystrov, and A. Yakovlev, “Design and analysis of dual-rail circuits for security applications,” *IEEE Transactions on Computers*, vol. 54, no. 4, pp. 449–459, April 2005.
- [13] 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 ESSCIRC Conference*, 2002.
- [14] F. Mace, F-X. Standaert, I. Hassoune, J-D. Legat, and J-J. Quisquater, “A dynamic current mode logic to counteract power analysis attacks,” in *Proceedings of DCIS conference*, Bordeaux, France, 2004, pp. 186–191.
- [15] I. Hassoune and J-D. Legat, “Low swing current mode logic family,” *Patent number: WO2005112263, Priority number: US20040571383P 20040514*, November 2005.
- [16] I. Hassoune, A. Neve, J.D Legat, and D. Flandre, “Investigation of low-power low-voltage circuit techniques for a hybrid full-adder cell,” in *Proceedings of PATMOS conference, Springer-Verlag, LNCS 3254, Santorini, Greece, 15-17 September 2004*, pp. 189–197.

CHAPTER 2

STATE OF THE ART

2.1 *Introduction*

This chapter presents a state-of-the-art related to the ongoing research on power management techniques. We also survey some popular adder architectures and previously reported binary full adder designs. Regarding the security concept at the hardware level, we briefly overview some countermeasures undertaken at the logic level, reported to be efficient against power analysis attacks.

2.2 *Power-aware design criteria*

Since Silicon technology is moving toward scaled down CMOS devices, this results in reduced parasitic capacitances, reduced power consumption per gate and improved delay performance. This results also in an increased packing density. Consequently, power consumption is increasing with the increased number of transistors per chip, which in turn introduced several power management constraints. These constraints are not limited to mobile and embedded applications since cooling is becoming a serious issue. Many reviews have reported the extensive research that has been made at all the design levels to achieve power management. Moreover, due to the broad range of applications incorporating digital signal processing, high speed is also a concern in the research community. To speed up the microprocessors data-paths, dynamic logic styles are often used, thereby making the data-path circuits high power consumers. Many works have reported circuit techniques or logic styles that can be applied to implement data-paths in microprocessors and achieve trade-offs between speed and power dissipation. This section reviews some ongoing applied circuit techniques or dedicated logic styles for power saving at both logic and architecture levels.

2.2.1 Low leakage power circuit techniques

Although dynamic power is continuously being reduced with technology scaling, leakage power tends to increase and is expected to become a large component in total power in few technology generations ahead. Leakage power is becoming a real issue and needs a suitable power management particularly as systems spend most of time in the standby mode. A high leakage power can thus be critical for portable electronic devices. Over the last decade, numerous leakage power management techniques have been developed and reported in the open literature. In this section we describe the most common amongst these.

2.2.1.1 Leakage control by body biasing

The body biasing technique consists in increasing the V_t of NMOS and PMOS devices during the standby mode by varying the body bias voltage. Thereby reducing leakage. The body of the NMOS device is biased to a voltage lower than ground in order to obtain a V_{th} higher than V_{th0} (threshold voltage for a source-to-body voltage $V_{SB} = 0$) and similarly the body of the PMOS device is biased to a voltage higher than V_{dd} in order to increase V_{th} . In the active mode, the body of the NMOS transistor is biased to ground whereas the body of the PMOS transistor is biased to V_{dd} , in order to obtain the normal low V_{th} and achieve normal speed.

Dynamic threshold CMOS (DTMOS) is a variant of the leakage control by body biasing since the body is tied to the gate and a variable threshold voltage is achieved according to the device state.

2.2.1.2 Leakage control by MTCMOS technique

In multi-threshold CMOS (MTCMOS) technique, high V_t switch transistors are used to control leakage. These transistors are controlled by a signal “SLEEP” in order to turn ON or OFF the switch transistors according to the state of the logic block [1]. During the standby mode, the switch transistors are turned OFF and limits leakage current thanks

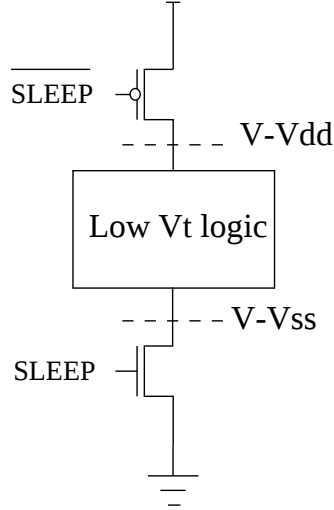


Figure 2.1: MVCMOS technique.

to their high V_t . In the active mode, the switch transistors are turned ON and act as virtual V_{dd} and ground.

2.2.1.3 Leakage control by MVCMOS technique

Multi-voltage CMOS (MVCMOS) [1] is depicted in Figure 2.1. Unlike in MTCMOS where the “SLEEP” transistors have high V_t , the MVCMOS technique uses “SLEEP” transistors with low V_t whose gates are controlled by a voltage which is larger than V_{dd} for the PMOS transistor and lower than ground for the NMOS transistor during the standby mode. This results in negative V_{gs} values for both NMOS and PMOS and subsequently in smaller leakage. Unlike in MTCMOS technique, the “SLEEP” transistors do not need to be oversized in order to avoid an excessive speed degradation since they have low V_t in the active mode.

2.2.2 Dynamic power management: Clock gating

Clock gating is a widely used low power technique. The idea behind this technique is to enable the clock signal in individual logic blocks

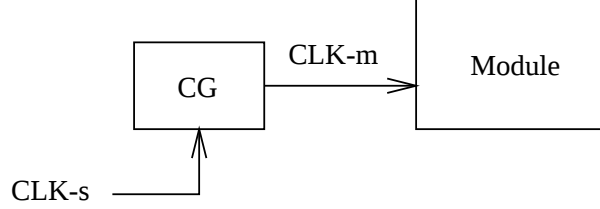


Figure 2.2: Gated clock in an individual module.

only when necessary and thus to save power thanks to switching activity reduction. From the system clock, other clocks (gated clocks) are derived. These gated clocks can be slowed down or disabled under certain conditions [2]. Figure 2.2 illustrates the basic principle of this technique, where the block “CG” depicts the clock-gating circuit, “CLK-s” the global system clock and “CLK-m” the gated clock of an individual module.

2.2.3 *Power-aware logic styles*

The choice of logic style to design basic gates and more complex circuits strongly influences the circuit performances. The delay time depends on the transistors size, the number of transistors per stack, the parasitic capacitance that includes intra-cells node capacitances and capacitances due to intra and extra-cells routing, and the logic depth (i.e. number of logic gates on the critical path). The power consumption depends on the switching activity, the number of transistors and on parasitic capacitances. The die area is influenced by the number of transistors, their sizes and the routing complexity. The choice of the macro-cells schematics is therefore the first important step to design low power circuits. This choice closely depends on the used design style.

Conventional static CMOS has long been popular in VLSI circuits for its high noise margin, low power consumption, compactness since it allows a quite regular layout topology and robustness to technology and

to voltage scaling. The drawback of the static CMOS is the low packing density that appears in multiple-inputs complex gates. Moreover, the driving capability decreases and the circuits operation can be slowed down when the number of series transistors in stacks increases [3]. Static CMOS logic is still nowadays so popular due to the wide availability of design methodology, synthesis tools and digital cells libraries supporting the static CMOS logic design. Generally, those CAD tools are not available for less conventional logic designs.

Logic CMOS families using the pass-transistor circuit technique to improve power consumption have long been proposed [4] [5] and have gained a great interest besides dynamic logic styles particularly for critical-path implementation in high performance microprocessors due to their lower power-delay product [6]. This logic style has the advantage to use only NMOS transistors, thus eliminating the large PMOS transistors used in conventional CMOS style. However, pass-transistor logic suffers from the degraded output logic level and a level restoration is needed to avoid static currents. Among pass-transistor logic families, complementary pass-transistor logic (CPL) proposed by Hitachi [7] is a well-known design style. The advantages of CPL design are the efficient implementation of XOR and multiplexer gates, its dual-rail structure which provides true and inverted signals, the good output driving capability thanks to the output inverters. Many studies in the literature have reported a substantial advantage in the power-delay product of CPL adders and multipliers in comparison to their counterparts in other logic families [7] [4]. Nonetheless, CPL gates show generally an overhead in wiring complexity which increases the layout sizes as it was demonstrated in [8]. CPL gates feature a high number of internal nodes (i.e. internal connections) and an overhead in transistor count since it needs two NMOS networks to implement the dual-rail. This increases the power consumption especially in complex gates. Particularly, it was shown in [9] that the conventional CMOS full adder outperforms the CPL one in terms of power consumption while a better delay is obtained in the CPL full adder.

Branch-based logic (BBL) proposed by CSEM [10] can be considered as a restricted version of static CMOS where a logic gate is only made of branches that contain a few transistors in series. The branches are connected in parallel between the power supply lines and the common output node. It was demonstrated in [11] that by using the branch-based concept, it is possible to minimize the number of internal connections and isolated diffusions, and thus the parasitic capacitances associated to the diffusions, interconnections and contacts. Moreover, it allows a quite regular layout topology. As it belongs to the family of static CMOS, it presents high noise margins and robustness to voltage scaling.

Dynamic logic features the use of few transistors (either NMOS or PMOS logic tree) to implement logic functions. This reduces the amount of the switched capacitance and benefits to the power-delay product. However, dynamic logic requires extra-circuitry to increase the noise margin that can be degraded by charge-sharing and avoid erroneous evaluations. The major drawbacks featured by dynamic logic in comparison to static logic lie in the high switching activity and the need of clock buffers to drive precharge and evaluation transistors. This make dynamic logic a high power-consumer in comparison to the static CMOS logic styles.

Domino logic style (Figure 2.3) owns its advantages in its high speed and its compactness since it contains a low number of transistors. When pipelining logic structures, the static inverter placed between adjacent logic blocks enhances its driving capability and avoids the clock race problem. The drawback of domino logic lies in its low noise margin because an input signal as low as V_t can turn on the pull-down NMOS transistor, thereby causing an erroneous state at the output.

Differential cascode voltage switch logic was proposed by IBM [12]. Over the past twenty years, many versions of differential CVS logic (DCVSL) have been proposed. Dynamic differential CVSL (DDCVSL) [13] (Figure 2.4) is one of the most popular versions of the CVS logic due to its high speed operation, compactness and differential structure.

Logic styles that feature a dual-rail structure (i.e. each input/output is represented by its true and inverted signal) has the advantage to allow

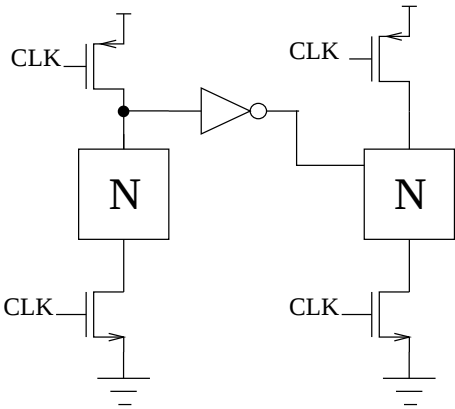


Figure 2.3: Domino circuit.

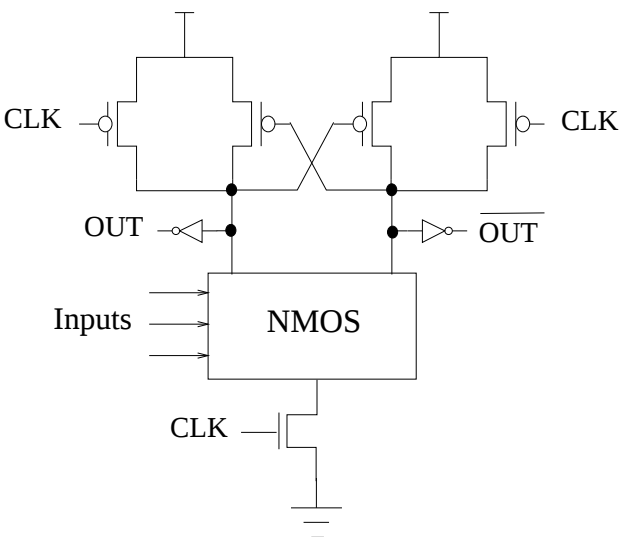


Figure 2.4: DDCVSL gate.

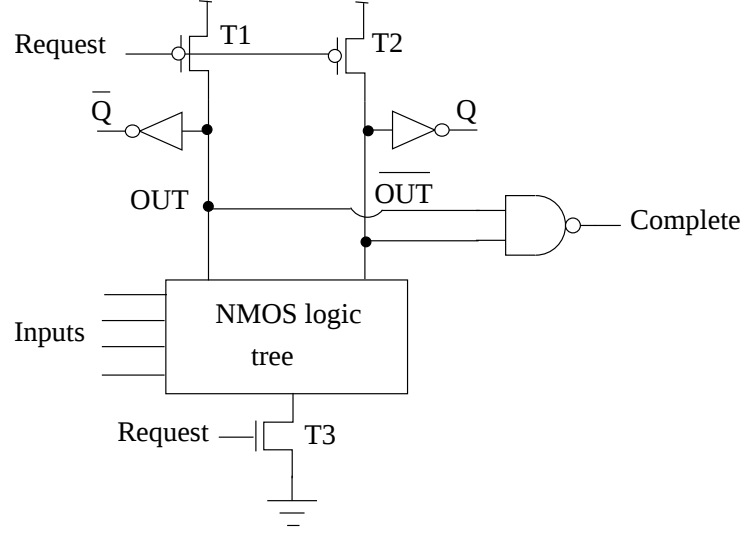


Figure 2.5: DDCVS logic for completion signal generation.

the completion signal generation. This potential is inherent to differential logic. DCVS logic has long been the most popular logic family used to implement the computation blocks in self-timed arithmetic circuits [14], [15], [16] thanks to its high speed performance and its low transistor number. Figure 2.5 shows the DDCVS logic with completion circuit. This latter is simply formed by a NAND gate.

When the signal Request is low, the two output nodes OUT and $\overline{OUT}$ are precharged to V_{dd} through the precharge circuit formed by the PMOS transistors (T1,T2) and the complete signal is pulled down to 0. When Request goes high (evaluation phase), the precharge circuit (T1,T2) turns OFF while a current path to ground is created through the NMOS transistor T3. According to input data values, one of the differential outputs will be discharged to 0 while the other is still precharged to high. The signal complete will then go to high. The signal complete is used as a Request signal for the next computation stage.

More emergent logic families use an aggressive reduction of the logic swing to achieve low dynamic power design according to equation 1.1.

2.2 POWER-AWARE DESIGN CRITERIA

MOS current mode logic (MCML) has been proposed by M. Yamashina in [17]. It consists in a differential pair operating with a constant current source (Figure 2.6). An MCML gate operates as follows. The transistor Q1 acts as a DC current source controlled by a constant voltage V_{ref} , R_1 and R_2 are pull-up resistors. The flow of current through the two differential branches is controlled by the values of input variables of the NMOS tree. Unlike in full swing logic circuits, transistors are either fully or partially ON in reduced voltage swing circuits. In current mode logic, the value of the output logic depends on the difference between currents flowing through the differential branches. According to the input values, the voltage at one output node (OUT or $\overline{\text{OUT}}$) will begin to drop until it reaches a steady state while the other output node is pulled up to V_{dd} through the pull-up resistor (R_1 or R_2). As a consequence, the voltage at one node is V_{dd} while $V_{dd} - R_i \cdot I$ is the voltage at the other one, where I is the current drained by the DC current source Q1 and R_i is the pull-up resistor. The output voltage swing is the voltage difference between OUT and $\overline{\text{OUT}}$ at steady state. Because delay time in MCML circuits is expressed by $\frac{C\Delta V}{I}$ [17] where ΔV is the output voltage swing at steady state, the delay can be reduced by reducing ΔV . Moreover, because the scaling of supply voltage does not increase the delay-time in MCML circuits, the power consumption can then be reduced by reducing the supply voltage without any speed penalties. MCML based circuits have shown a higher speed when compared to their CMOS counterparts [17]. The power dissipation in MCML circuits is constant and equals $V_{dd} \cdot I$. Hence, the power dissipation is still constant even when increasing frequency. Therefore, MCML logic should be interesting in very high-frequencies applications. However, it becomes power-hungry at low frequencies unless the supply voltage is aggressively scaled. Since $\Delta V = R_i \cdot I$, the output voltage swing ΔV may be adjusted by choosing the proper value of the pull-up resistor in MCML circuits.

DyCML current mode logic (DyCML) based circuits [18] (Figure 2.7) feature the same advantages than those of MCML circuits. Moreover, unlike MCML circuits, DyCML circuits use a dynamic current source

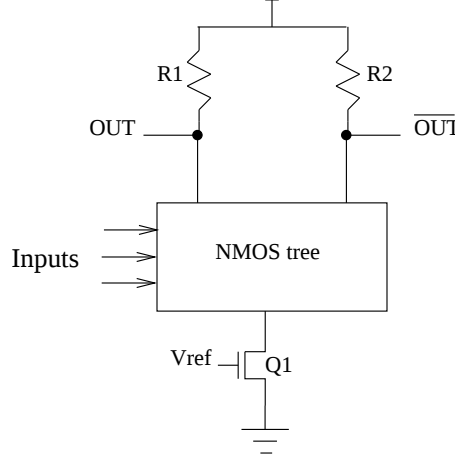


Figure 2.6: MCML gate.

instead of a DC current source. This avoids the DC power and makes DyCML circuits more prone than MCML to achieve low power consumption even at low frequencies. Moreover, the pull-up circuit in DyCML gate is formed by PMOS transistors instead of the pull-up resistors used in MCML gates, which leads to a gain in die area.

The structure of a DyCML gate is shown in Figure 2.7. It consists in a precharge circuit (Q2,Q3,Q6), a dynamic current source (Q1,C1) where the transistor C1 acts as a capacitor, a latch (Q4,Q5) to maintain logic output value after evaluation and a NMOS tree for logic function evaluation. When CLK is low (precharge phase), the precharge circuit (Q2,Q3,Q6) is turned ON. The capacitor C1 is discharged, thereby pulling down the node d to ground, while the output nodes OUT and $\overline{\text{OUT}}$ are precharged to V_{dd} . During this time, transistor Q1 is turned OFF. Hence, there is no DC path from V_{dd} to ground as it is the case in MCML logic. When CLK goes high (evaluation phase), the precharge circuit switches OFF and the transistor Q1 turns ON. Two current paths are then created through the two branches. These two paths have differ-

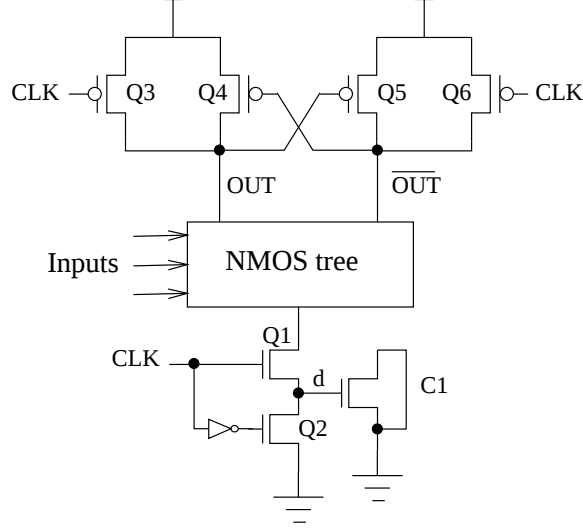


Figure 2.7: DyCML gate.

ent impedances that depend on the values of the input variables. Hence, one output node will discharge faster than the other one. In the latch circuit (Q4,Q5), the transistor whose gate is connected to the node which discharges faster will switch ON once the voltage at this node is less than $V_{dd} - |V_{tp}|$ (V_{tp} is the threshold voltage of the PMOS transistor), and will pull-up the other output node to V_{dd} . The current flowing through Q1 charges the capacitor C1. When the voltage at node d reaches a value such as $V_{ds}(Q1) \simeq 0$, the transistor Q1 switches OFF and limits the amount of charge transferred from the output node. The voltage at node d is then about $V_{dd} - \Delta V$. The size of transistor C1 is calculated using the following equations:

$$\Delta V \cdot C_L = W_{C1} \cdot L_{C1} \cdot C_{ox} \cdot (V_{dd} - \Delta V) \quad (2.1)$$

$$W_{C1} \cdot L_{C1} = \frac{\Delta V \cdot C_L}{C_{ox} \cdot (V_{dd} - \Delta V)} \quad (2.2)$$

where C_{ox} is the gate oxide capacitance per unit area and W_{C1} and L_{C1} are respectively the width and length of transistor C1 and C_L is the

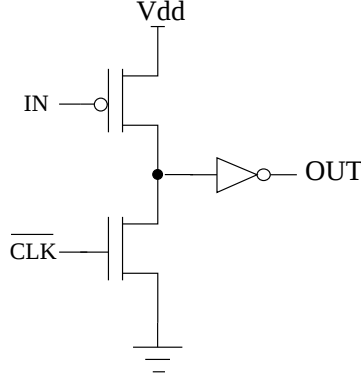


Figure 2.8: Buffering circuit for level conversion of the output signal.

total parasitic capacitance per output node. DyCML gates can be used in conjunction with full-swing logic. The circuit shown in Figure 2.8 is used to convert the reduced output voltage swing to a full-swing output signal. DyCML gates can be cascaded efficiently thanks to the signal generated at node d. This signal can be used as a completion signal. However, because this signal goes up $V_{dd} - \Delta V$ during the evaluation phase, a special level converter as shown in Figure 2.9 is used to generate a full-swing completion signal that will be used as a clock-signal for the next DyCML stage.

DyCML gates cascading is shown in Figure 2.10. The buffering of the completion signal works as follows. When CLK0 is low, the transistor Q1 (of the buffering circuit for completion signal, shown in Figure 2.9) switches ON. Meanwhile, the node d is discharged to 0. The node i is charged to V_{dd} through Q1. Hence, the transistor Q3 switches OFF while Q4 is turned ON, which discharges the node CLK1 to ground. When CLK0 is high, the transistor Q1 is switched OFF while the voltage at node d is charged up to $V_{dd} - \Delta V$. Q2 turns ON and discharges the node i to 0. Meanwhile Q4 is switched OFF. Q3 turns ON and charges CLK1 up to V_{dd} .

Short-circuit current logic (SC²L) [19] is a dynamic differential limited

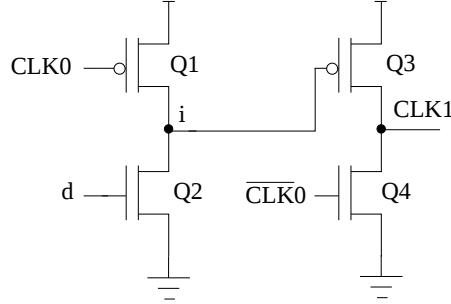


Figure 2.9: Buffering circuit for the completion signal in DyCML gates cascading.

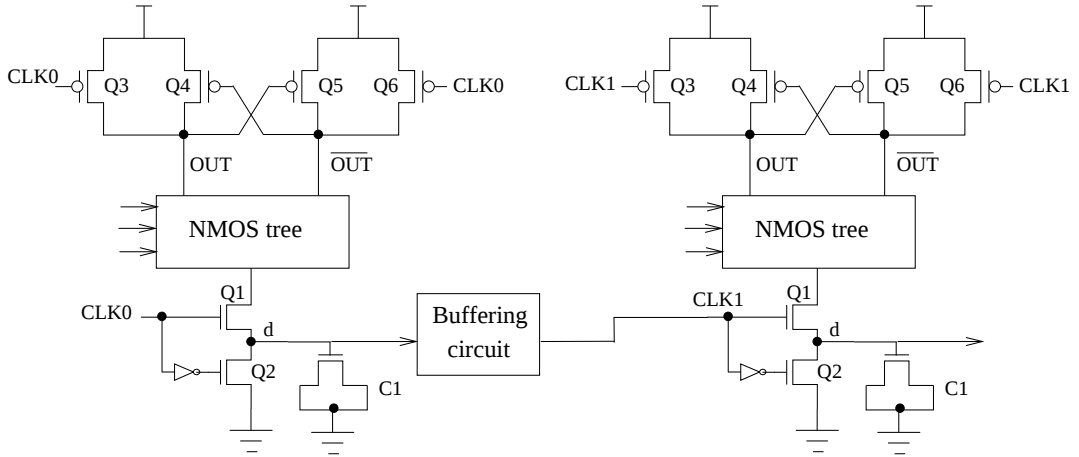


Figure 2.10: DyCML gates cascading. The “Buffering circuit” is detailed in Figure 2.9.

swing logic style (Figure 2.11). It features high performance in terms of energy-delay product [20] because of an aggressive reduction of output voltage swing. When CLK is high (precharge phase), the output (node q) of the inverter formed by (MN1,MP1) is discharged to 0. Transistors MP2 and MP3 turn ON and precharge the nodes F1 and F2 to V_{dd} . Meanwhile, transistors MP5 and MP4 that act as a latch transistors, are turned OFF. When CLK goes low (evaluation phase), during the transition $1 \rightarrow 0$ of the signal CLK, a short-circuit current will flow through (MP1,MN1) during a very short time. This short-circuit current will then discharge one node of the NMOS logic tree outputs according to the values of input variables. MP4 and MP5 turn ON and drive the evaluated signal to OUT and $\overline{\text{OUT}}$ respectively. The amount of voltage drop across the output depends on the short-circuit current, which in turn depends on clock signal slope and transistor sizing in the inverter gate. This amount depends also on charge sharing between parasitic capacitances at node q and F_i (F_1 or F_2). Because the maximum voltage at node q is $V_{dd} - V_{tp}$, body terminal in transistors MP2 and MP3 must be biased by a boosted supply voltage (V_{WELL}) to turn them completely OFF during the evaluation phase. Moreover, because the reduced swing voltage at node q is used as a completion signal for the next stage in pipelined SC^2L gates (Figure 2.12), transistors MP4 and MP5 must also have their body biased by a boosted supply voltage (V_{WELL}) to turn them completely OFF during the precharge phase and avoid erroneous evaluation in the next stage. The basic operation of this pipelining scheme is illustrated in Figure 2.13 through a timing diagram of three SC^2L based pipelined stages. Before we explain the basic operation of this pipelining scheme, let us explain the general principle of pipelining of arithmetic operations.

Pipelining is often used to speed up the execution of successive identical operations. Unlike in nonpipelined design that executes a single operation on one data set of operands at a time, pipelining design allows to partition a circuit into several subcircuits that operate independently on consecutive data sets. For that sake, storage elements (i.e. latches)

are added between adjacent stages in a such way that when a stage is processing one data set, the previous stage can process the next data set. Nonetheless, pipelining of arithmetic operations like addition is advantageous only if addition of several successive data sets is needed. In Figure 2.13, the signal $CLK1$ represents the inversion of the clock-signal $CLK0$ and $CLK0'$ is the inversion of the latter. SC^2L Pipelining alternates evaluation and precharge stages. When the stage0 finishes the evaluation of the input data $D1$, the output signal $Out0(D1)$ is valid and available at the input lines of stage1. The latter is in precharge phase, hence no processing is performed. On the other hand, since the latch transistors are switched OFF, no data is available at its output nodes and then at the input lines of stage2. When stage1 goes into the evaluation phase, the data $Out0(D1)$ is then processed and the evaluation result is made available at the input lines of stage2. When the latter goes into evaluation phase, the data $Out1(D1)$ is processed and data output $Out2(D1)$ is transferred to the next stage. Meanwhile, the stage0 starts the evaluation of the next data set ($D2$). In [19], an 8-bit asynchronous SC^2L based ripple carry adder (RCA) has been implemented using this pipelining scheme. However, due to the large load on node q , this slows down the propagated clock-signal and subsequently the circuit operation as well. In [21], a cascading approach has been proposed instead of a pipelining approach in order to speed up the generation of the completion signal for a SC^2L cascaded gates strategy.

As reported in [20], the reduced output voltage swing in SC^2L gates is very sensitive to the CMOS inverter sizing, the clock slope, process and supply voltage variation. This may cause a significant variation in delay and power consumption. Moreover, the SC^2L logic family does not scale well with V_{dd} .

2.2.4 Power-aware architectures: asynchronous versus synchronous

Among dedicated techniques to power saving at the architecture level, clock gating (see section 2.2.2) has usually been reported as one of the

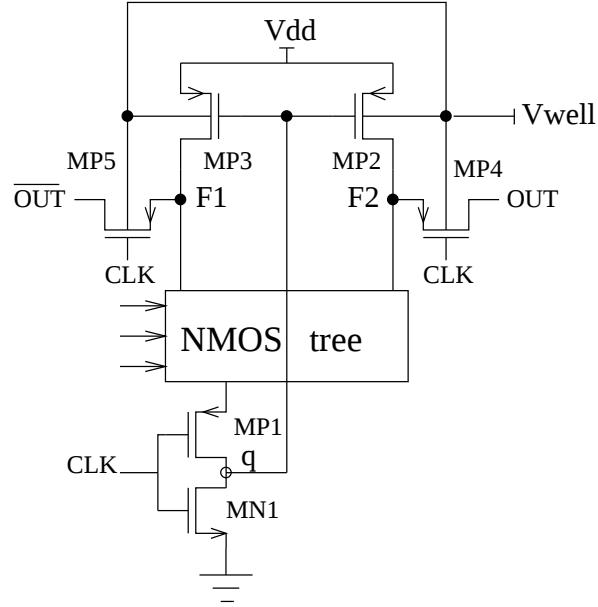


Figure 2.11: SC²L gate.

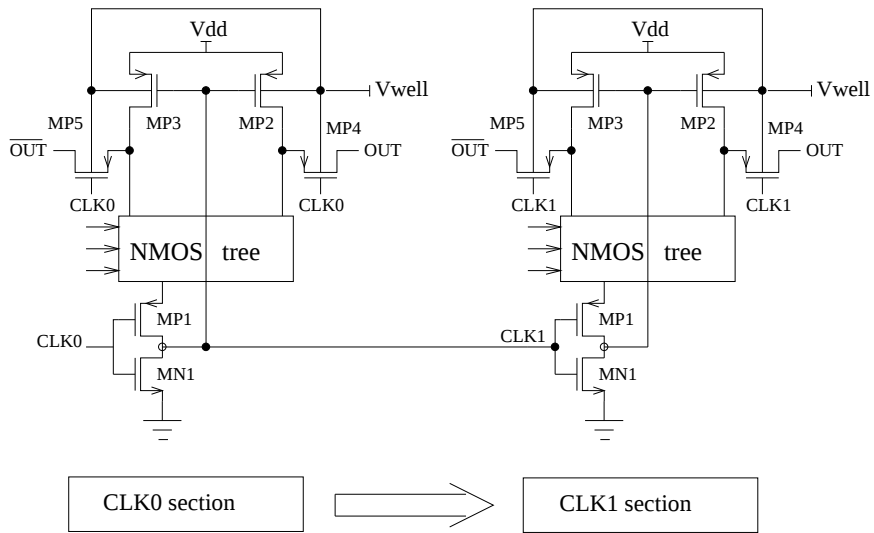


Figure 2.12: SC²L gates pipelining.

2.2 POWER-AWARE DESIGN CRITERIA

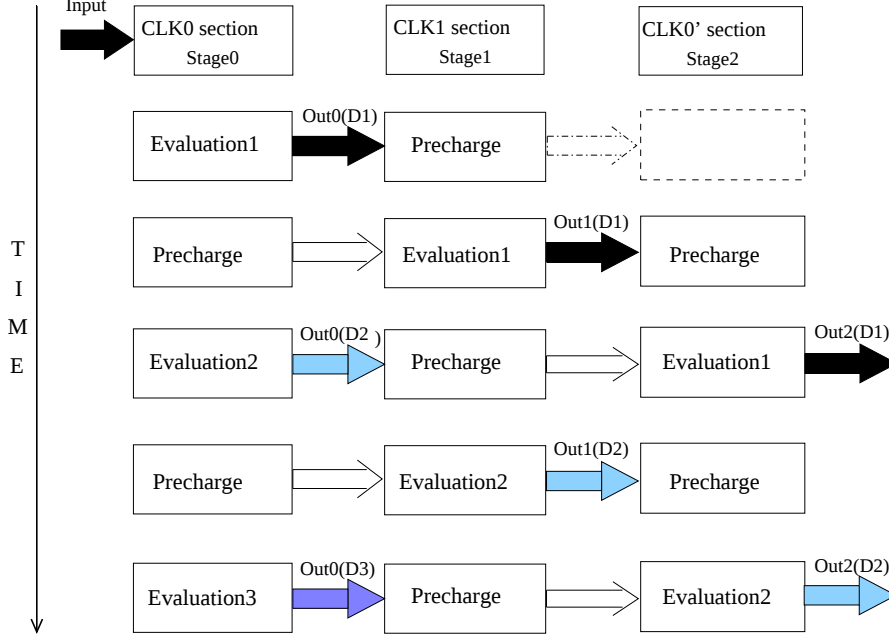


Figure 2.13: A timing diagram of three SC²L-based pipelined stages.

most common way to save power in synchronous circuits by selectively powering down logic blocks when they are not in use [22] [23]. This feature is inherent to the asynchronous approach since unused logic blocks remain in standby mode until data are available at their input lines [22], which makes this approach prone to reduce power in SoCs. Surely the features of asynchronous circuits helps to achieve a better gain in power consumption than in synchronous ones though this gain remains application-dependent due to the overhead circuitry used to implement the handshaking protocols.

A brief list of asynchronous microprocessors resulting mostly from academic research over the last 15 years has been reviewed in [22]. This review shows encouraging results with regard to power consumption reduction. Table 2.1 summarizes the performances achieved by some of these studies.

Processor	Designer	Technology and distinctive feature	Performances
Caltech asynchronous processor (CAP) 16-bit RISC arch. 1980	Alain Martin's group at Caltech	1.6 μ m MOSIS SCMOS	26MIPS, 1.5W at 10V 18MIPS, 225mW at 5V 5MIPS, 10.4mW at 2V
Asynchronous 80C51 8-bit CISC arch. 1995	Eindhoven University of technology	0.5 μ m CMOS	4MIPS, 9mW at 3.3V (The energy consumption is reduced by a factor of 4 in comparison with the synchronous version)
Asynchronous MiniMIPS MIPS R3000 arch. 1997	Alain Martin's group at Caltech	0.6 μ m SCMOS	150MIPS, 1W at 2V 280MIPS, 7W at 3.3V
Asynchronous Processor (ASPRO) 16-bit RISC arch. 1998	Ecole nationale supérieure de Télécom. de Bretagne	0.25 μ m CMOS (Synthesis was partially performed by hand)	24MIPS, 20mW at 1V 140MIPS, 350mW at 2.5V
AMULET3 (Asynch. micro-processeur using low energy and techno.) 32-bit ARM (advanced RISC machine micro-processeur arch.) 2000	University of Manchester.	0.35 μ m technology (Full-custom design for the data-path)	120MIPS, 155mW at 3.3V

Table 2.1: Review of some asynchronous microprocessors resulting from academic research during the last fifteen years

In synchronous circuits, worst case delays proper to critical paths determine the clock-cycle. The non-critical paths are not considered when assessing the speed performance, while asynchronous circuits operate with a variable processing times bounded by a minimum and maximum values that correspond to the shortest and critical paths respectively. Asynchronous designs are generally characterized by an average speed as opposed to the worst-case speed in their synchronous counterparts. However, it was reported in the literature [24] that an analytical model of this average time is difficult to set up because the internal delay dependencies are much more complex in asynchronous modules than in the synchronous versions and delay modeling needs a deep understanding of the temporal behaviour of these. Only a few studies have proposed empirical models, but only for a limited number of asynchronous arithmetic functions. Nonetheless, timing characterization in asynchronous circuits may be brought back to the worst-case.

2.3 Arithmetic adders

This section reviews three well-known architectures among existing adders: the ripple carry adder (RCA), the conditional sum adder (CSA) and carry lookahead adder (CLA).

2.3.1 Ripple carry adder

RCA is a chain structured adder where the carry chain is propagated through n basic logic units called full adder to achieve an addition of two operands $A_{n-1}, A_{n-2} \dots A_0$ and $B_{n-1}, B_{n-2} \dots B_0$ as illustrated in Figure 2.14. A 1-bit full adder is a binary combinatorial logic circuit that accepts two operand bits (A_i and B_i) and a carry-in bit (Cin). If we assume that the estimation of arithmetic circuit's speed versus the number of logic levels is still valid in modern CMOS technologies, the delay profile in n -bit RCA is estimated to be $O(n)$ (i.e. proportional to n). The RCA is surely the slowest adder but it is also the cheaper in terms

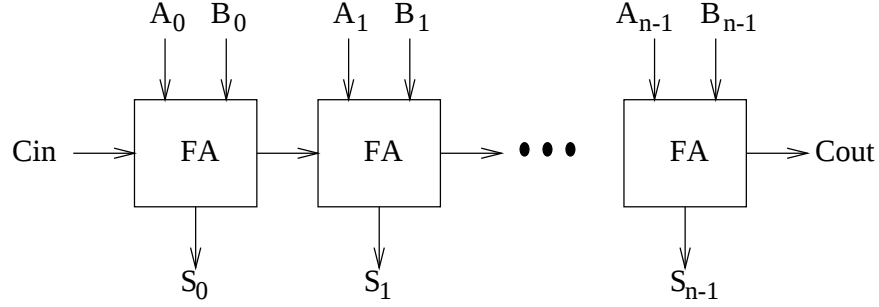


Figure 2.14: Ripple carry adder (RCA).

of area cost and power consumption. The area profile is estimated to $O(n)$.

2.3.2 Conditional sum adder

Generally in conditional sum adder, the incoming n -bits are divided into smaller k -bit groups. In each k -bit group, the addition is splitted into two parallel additions where sum and carry-out are computed both for carry-in=0 and carry-in=1. The real value of carry-in, which is computed by the previous group (denoted $Cout_{LSB}$ in Figure 2.15) then selects the correct output values for sum and carry-out. This implementation is illustrated in Figure 2.15 where CPA depicts a carry propagate adder that can simply implemented with a RCA one. Carry select adder is a variation of the conditional sum adder. This parallel addition scheme features a delay profile of $O(\sqrt{n})$. But this low latency is achieved at the expense of larger area ($O(n * \sqrt{n})$) and higher power consumption than in the RCA version.

2.3.3 Carry lookahead adder

The carry lookahead adder (CLA) is known as the fastest architecture among existing adders. Its structure lies on a judicious implementation

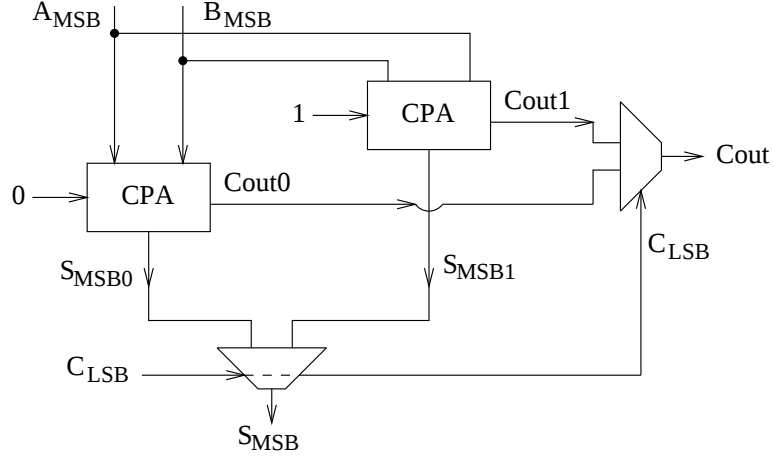


Figure 2.15: Conditional sum adder (CSA).

of the carry chain. This adder exploits the following principle:

- If $A_i = B_i$, the carry-out is independent of the carry-in value. It is then generated locally and equals 0 when $A_i = B_i = 0$ and 1 when $A_i = B_i = 1$.
- If $A_i \neq B_i$, the carry-in is simply propagated to the carry-out ($C_{i+1} = C_i$)

Generation and propagation of the carry have been exploited to speed up the addition in many versions of carry lookahead adders. They are computed as follows:

$$G_i = A_i \cdot B_i \quad (2.3)$$

$$P_i = A_i \oplus B_i \quad (2.4)$$

The Boolean expression for the outgoing carry-out is as follows:

$$C_{i+1} = G_i + P_i \cdot C_i \quad (2.5)$$

The delay profile of the overall addition in carry look ahead generator is evaluated to $O(\log(n))$. The high speed in CLA comes with a high power consumption and a large area. Among the three reviewed adder types, the conditional sum adder is known to be a trade-off between ripple carry and carry lookahead adders in terms of speed and power consumption.

2.3.4 Power aware full adders

An extensive variants of full adders have been investigated by the academic and industrial research communities. The usual performance evaluation are the speed, power consumption and area. However, since mobile and embedded applications have stood out the power consumption at the top of these performance evaluation of circuits and systems, goal of many of these full adder variants has used to be the reduction of transistor count. Some of them achieve this goal at the expense of a possible degradation of the signal being propagated in chain structured implementations. This section reviews some of static binary full adder designs.

2.3.4.1 Static CMOS and pass-transistor logic full adders

The static complementary CMOS full adder [25] (Figure 2.16(a)) might be the most popular since it owns the inherent advantage of static CMOS style in robustness to scaling of both voltage and transistor sizing. However, its transistor count (28) has often been considered in the open literature as being no longer suitable for low power arithmetic circuits. Hence, several full adder cells based mostly on pass-transistor logic have been proposed. Pass-transistor logic was primarily considered because it needs small transistor number to implement a logic function particularly for XOR gate, which is a fundamental cell in the binary full adder.

The complementary pass-transistor logic (CPL) full adder [8] shown in Figure 2.16(b) features a dual-rail structure that provides true signals and theirs complements. A good driving capability is achieved through

2.3 ARITHMETIC ADDERS

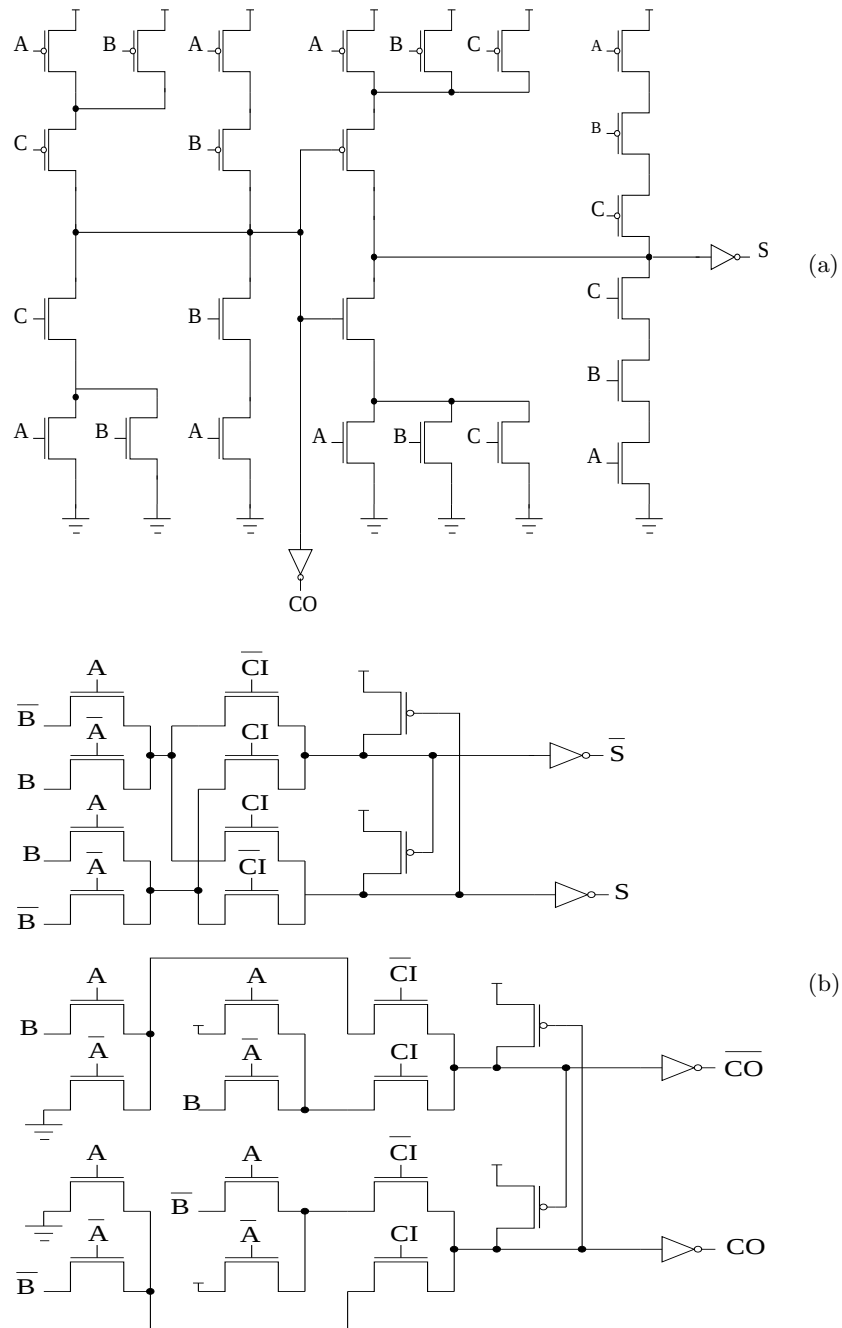


Figure 2.16: Full adder cells (a) Static CMOS. (b) CPL.

a level-restoration and output inverters. Many works have reported the advantage of arithmetic circuits based on the CPL FA in comparison to their counterparts based on the static CMOS FA [9] [4] in terms of delay and power-delay product. However, because it needs two NMOS transistor networks to implement the dual-rail structure, the CPL FA features high transistor number (32) and high number of internal nodes, which leads to a high layout/wiring complexity as opposed to the regular layout topology owned by the static CMOS FA, and a high power consumption [8].

Zhuang et al. [26] have proposed the transmission function full adder (TFA) shown in Figure 2.17(c). Despite its low transistor number (16-transistors), it features a high number of internal nodes, it might show a high wiring complexity in comparison for instance with the static CMOS FA. This subsequently might impact the performances of large arithmetic circuits that need many of such instances.

The transmission gate adder (TGA) [27] uses CMOS transmission gate logic. Similarly to the static CMOS full adder, it requires complementary inputs as shown in Figure 2.17(d) but features a lower transistor number per stack than in the static complementary CMOS FA which benefits to speed and transistor count (20). Though TFA and TGA have few transistors, it was shown in [27] that additional buffers are needed at each output due to their weak driving capability. This subsequently increases the power consumption and area.

Further smaller transistor count full adders have followed. Amongst these, the 14T full adder (Figure 2.18(e)) and the 10T full adder (Figure 2.18(f)) that own their names due to their transistor count.

It was shown in [27] that the speed of the 14T decreases more dramatically with supply voltage scaling than other full adder cells. This is due to the voltage drop problem and the lack of level restoration. Moreover, the 14T cell has shown an operation failure at low voltage. The 10T full adder has demonstrated in [27] further speed degradation and excessive static power dissipation due to the same problems as in the 14T. The 10T cell failed to function at low supply voltage. Indeed, from

2.3 ARITHMETIC ADDERS

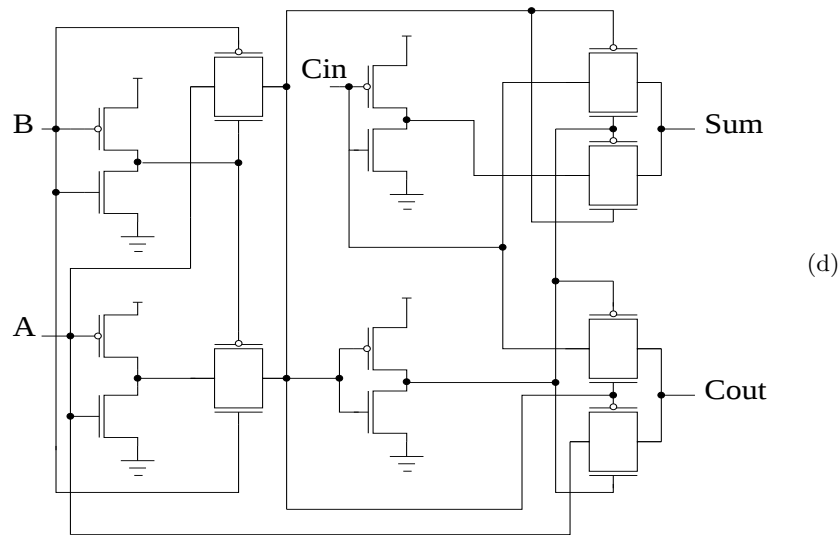
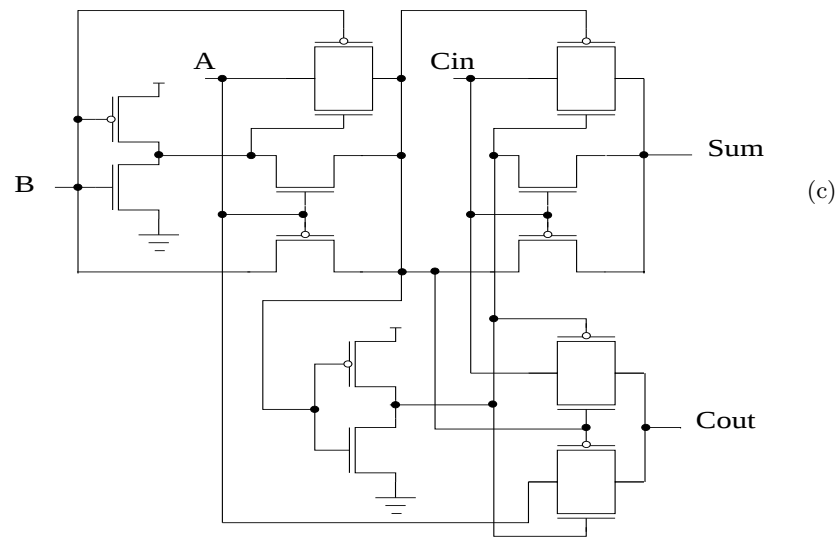


Figure 2.17: Full adder cells (c) TFA. (d) TGA.

simulations [27] under a supply voltage range of $0.8 - 1.2V$ in $0.18\mu m$ CMOS technology, the lowest voltage at which the 10T cell can function at 100MHz is 1.8V. This contrasts with the static CMOS and CPL FA, which operate under supply voltages as below as 0.8V in the same technology and the same operating frequency. Moreover, Chang et al. have demonstrated in [27] that both the 14T and 10T full adders exhibit a high total power consumption due to an excessive static power. The latter originates from the voltage drop problem.

2.3.4.2 Hybrid full adders

Abu-Khater et al. have combined CPL XOR function and transmission gate logic to implement the CPL-TG full adder [28] as shown in Figure 2.19(g). This cell provides a full-swing signal for sum and carry output signals. Moreover, it features a dual-rail structure with slightly lower transistor count than in CPL FA (30 instead of 32 transistors). However, the sum and carry networks can suffer from a high fan-in because the generated XOR/XNOR signals (P and $\bar{P}$ in Figure 2.19(g)) control the gate of a high number of transistors. This may slow down the speed and affect the switching power. It was shown in [28] that the speed of CPL-TG FA outperforms the static CMOS FA speed by a factor of two while consuming the same power. Hence, the CPL-TG shows a gain of 50% in power-delay product. The use of the CPL-TG FA in a 6x6 multiplier has resulted in 18% less power and a gain of 30% in speed in comparison with the static CMOS implementation.

Another hybrid full adder design has been proposed more recently in [27]. This hybrid implementation uses the same pass-transistor circuit as in the 14T transistor to generate the XOR and XNOR functions. However, since this pass-transistor circuit suffers from voltage drop problem that slows down the response at the transitions $01 \rightarrow 00$ and $10 \rightarrow 11$ for AB, authors have modified the sub-module that generates the XOR and XNOR functions. Two series PMOS transistors were added to solve the problem for transition $01 \rightarrow 00$ and two series NMOS transistors were

2.3 ARITHMETIC ADDERS

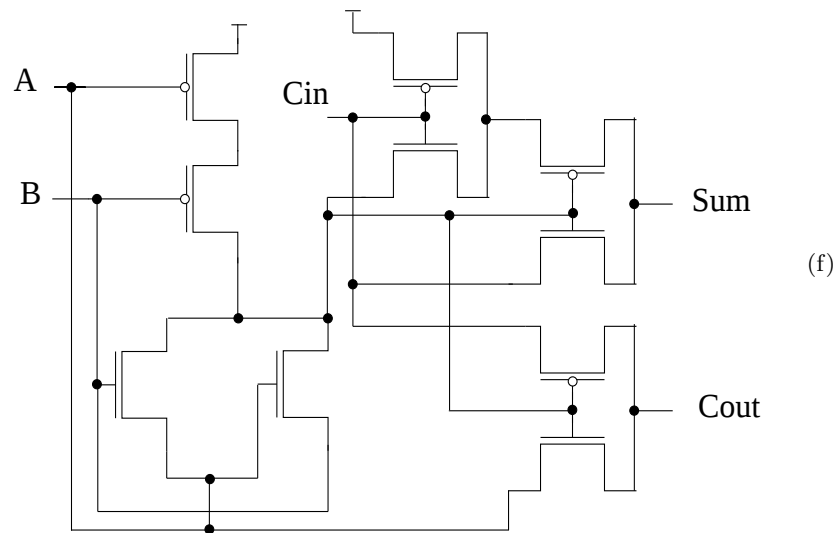
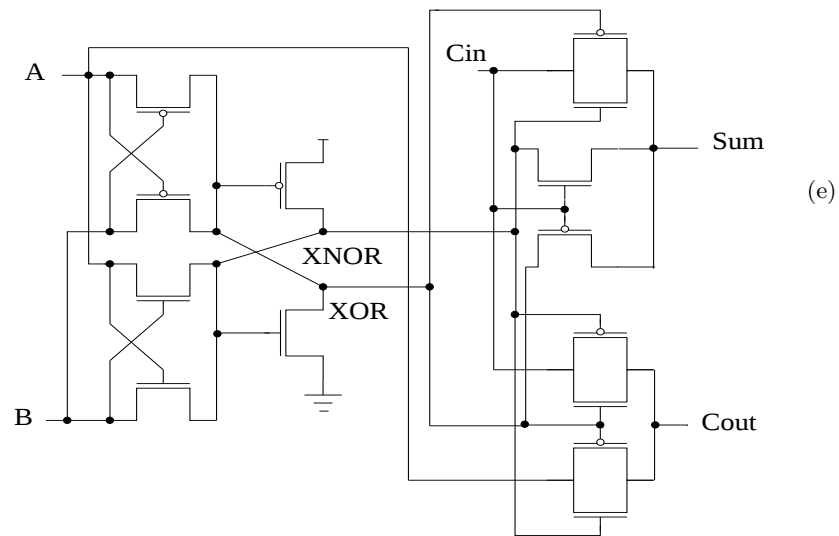


Figure 2.18: Full adder cells (e) 14T. (f) 10T.

added to solve the problem for transition $10 \rightarrow 11$. Besides the XOR-XNOR circuit, authors also used the same 6-pass transistor circuit as that in TFA and 14T to generate the sum output [27]. To enhance the driving capability at the output of the 6-pass transistor circuit, the output is re-generated through an inverter. On the other hand, the carry output is generated through a static complementary CMOS based circuit in order to provide a full-swing signal for carry-out. This hybrid full adder implementation is shown in Figure 2.19(h). The hybrid full adder has shown in [27] a slightly higher speed and almost the same power consumption as its counterpart in static complementary CMOS. Table 2.2 summarizes performance evaluation for full adder cells that we described in this section, resulting from both reviews and our personal estimation.

2.3 ARITHMETIC ADDERS

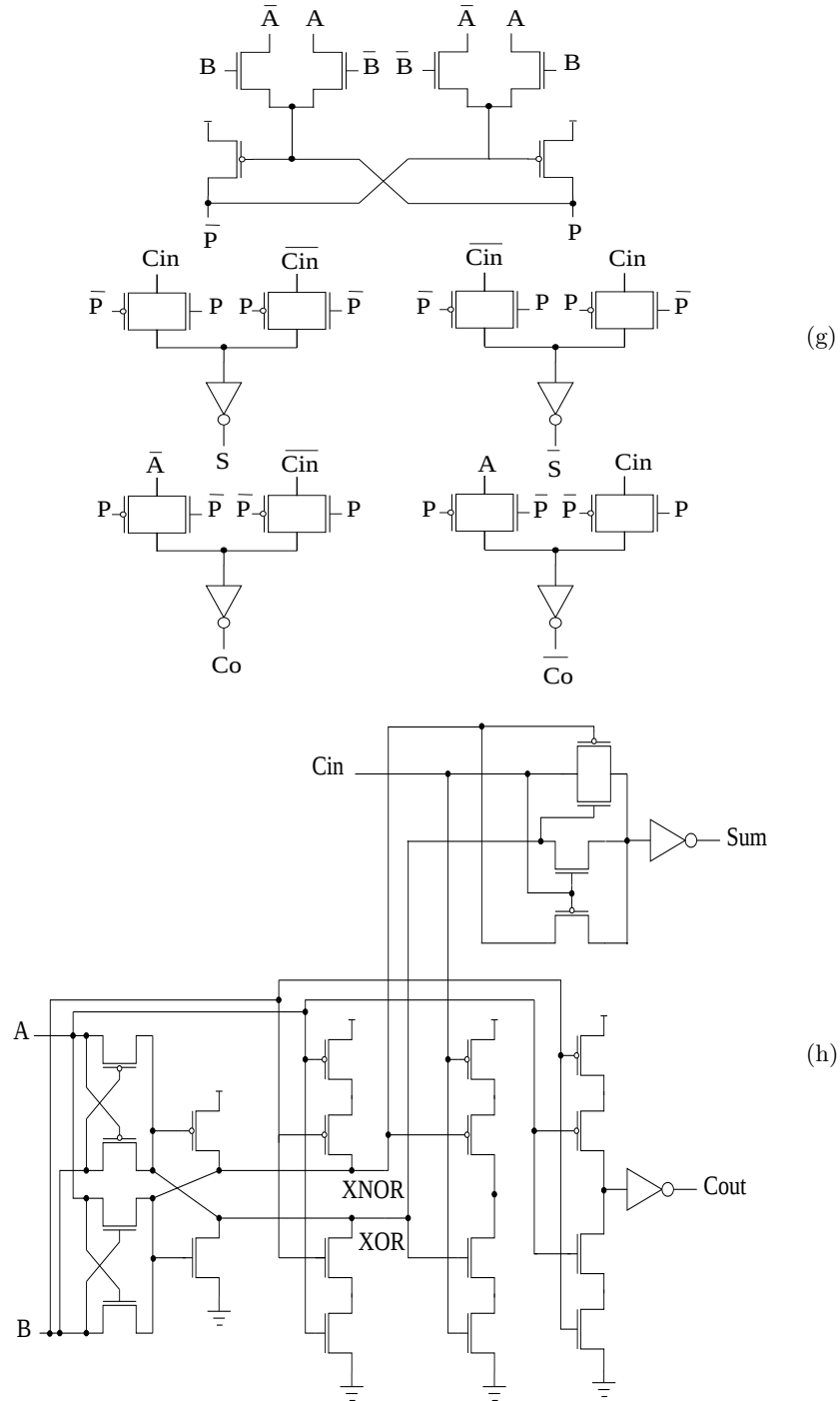


Figure 2.19: Full adder cells (g) CPL-TG. (h) Hybrid full adder [27].

Full adder cell	Power	Delay	Power delay product	Output driving capability	Operation at low voltage	Transistor count
Static CMOS FA	Efficient	Efficient ⁻	Efficient ⁻	Good	Robust	28
CPL FA	Moderate	Efficient ⁺⁺	Efficient ⁺	Good	Robust	32
TFA	Moderate ⁻	Moderate	Moderate ⁻	Weak	Robust with additional buffers	16
TGA	Efficient ⁻	Moderate	Moderate	Weak	Robust with additional buffers	20
14T	Poor	Poor	Poor	Weak	Fails at $V_{dd} < 1V$	14
10T	Poor ⁻⁻	Poor ⁻	Poor ⁻	Weak	operates only at high V_{dd}	10
Hybrid FA [27]	Efficient	Efficient	Efficient	Good	Robust	26
CPL-TG	Moderate ⁻	Efficient	Efficient	Good		30

Table 2.2: Performance evaluation for the state-of-the-art full adder cells.

2.4 Security ICs criteria

Variation in energy consumption with the input data which is often referenced to as “imbalance”, was defined in [29] as follows:

$$d = \frac{|e_1 - e_2|}{e_1 + e_2} \cdot 100\% \quad (2.6)$$

where e_1 and e_2 are the energy consumptions of two different input data. This imbalance was reported to be significant in single-rail circuits due to the data dependency of the quantity of switching. Conversely, dual-rail (i.e. two logic networks for computation of true and inverted output signal) circuits have achieved a smaller imbalance thanks to more regular activity whatever the processed data [29]. Besides the use of dual-rail circuit topology, other methods like the use of asynchronous architectures and dynamic differential logic styles have demonstrated encouraging results with regard to reduction of data-dependent power signature.

2.4.1 Self-timed design to prevent DPA

If self-timed design has been found promising to implement low power embedded systems, it demonstrated a great potential at the hardware level in the cryptography community as well [30] [31]. Indeed, whereas DPA attack has been well studied for synchronous designs [31], in asynchronous designs, there is no global clock to take as timing reference. As each individual circuit in the chip has its local self-timing, and thus operates independently, the operation of individual units is consequently masked. This makes the power analysis cracking much harder for the attacker [31].

The advantage of asynchronous circuits in terms of security can be explained by their power consumption behaviour. The previously reported studies led on asynchronous microprocessors and architectures have shown that the locally distributed nature of the communication protocols allows to reduce activity and hence power consumption. More-

over, this local synchronisation uniformly distributed on chip avoids currents peaks and hence power consumption peaks, as opposed to synchronous circuits where the tasks of individual modules are synchronized by the global clock signal. This results in a maximum activity and power consumption peaks at each clock-event.

Moreover, the current peaks on power supply lines used to be observed in synchronous circuits induce a high electromagnetic emission, which increases eventually the noise due to cross-talk but also makes synchronous circuits vulnerable to power analysis and electromagnetic analysis attacks. These weakness can be alleviated when using asynchronous circuits.

Nonetheless, even though asynchronous design makes the statistical analysis more difficult since there is no timing reference as used in synchronous one to synchronize the observations of the encryption module behaviour, it was demonstrated in [31] that it is not sufficient to prevent power analysis attacks, and dual-rail logic is needed to reduce the data-dependent power signature. A self-timed ARM compatible processor core (SPA) using both an enhanced dual-rail logic and an enhanced functional arithmetic units (adder, multiplier) to achieve a data-independent execution time, has been developped for a smart-card chip [31]. It has shown an improved robustness to side-channel attacks in comparison to the single-rail SPA. However, this was achieved at the cost of an increase in transistor count and hence in power and speed performance.

2.4.2 *Dynamic differential logic to prevent DPA*

Dynamic differential logic styles have been proposed as a solution to balance power at the cell level [32] [33]. To explain their advantage in comparison to more classic logic styles, let us examine the expression of the average dynamic power consumption.

$$P_{dyn} = \alpha_{0 \rightarrow 1} \cdot C_L \cdot V_{dd} \cdot V_{swing} \cdot f \quad (2.7)$$

The value of $\alpha_{0 \rightarrow 1}$ depends on different components, that are the type of logic style, the type of logic function, circuit topology and the sequence of operations [15]. $\alpha_{0 \rightarrow 1}$ is defined as the probability of an output transition $0 \rightarrow 1$. Let us assume a 2-inputs NOR gate with uniformly distributed and random inputs. For a NOR gate implemented in static CMOS, $\alpha_{0 \rightarrow 1} = \frac{3}{16}$. The same gate implemented in a dynamic logic style gives $\alpha_{0 \rightarrow 1} = \frac{3}{4}$, while its implementation in dynamic differential logic gives $\alpha_{0 \rightarrow 1} = 1$. Hence, dynamic differential gates ensure one output transition every cycle and this independently of the data inputs.

Though the imbalance is not completely eliminated when using dynamic differential logic, it can be reduced thanks to the significant reduction of the data-dependent power signature [33], therefore, making DPA analysis more difficult. Nonetheless, this is achieved at the expense of high power consumption due to the 100% switching activity. Reducing the output swing can help to reduce the dynamic power. On the other hand, it was shown in [32] [34] that all dynamic differential logic styles are not equal in terms of robustness against power analysis attacks. Other factors like the total amount of the capacitance at the switching nodes and the circuit topology may make the difference.

To balance the total amount of the capacitance, authors in [34] have proposed Sense Amplifier Based Logic (SABL), which is a dynamic differential logic. The structure of a SABL gate as shown in Figure 2.20(a) consists in a precharge circuit (Q5,Q8), two cross-coupled inverters to provide a stable dual-rail output signal, a differential NMOS tree for logic evaluation, a NMOS transistor (Q1) to enable evaluation and a NMOS transistor (Q2) which is always ON, connected between the differential nodes X and Y. SABL gate operates as follows. When CLK is low (precharge phase), the output nodes (OUT and $\overline{OUT}$) are precharged to V_{dd} while nodes X and Y are precharged to $V_{dd} - V_{tn}$. When CLK goes high (evaluation phase), the precharge circuit switches OFF while the transistor Q1 switches ON. According to the data input values, one current path is created from node X or Y to ground, thereby discharging this node to 0. Let us assume that X is the discharged node. The

NMOS transistor Q3 in the cross-coupled inverter will turn ON (since $\overline{OUT}$ was precharged to V_{dd}) and discharge the node OUT to 0. Transistor Q7 will switch ON and $\overline{OUT}$ will remain then at V_{dd} while Q6 is switched OFF. The node Y in turn is discharged to 0 through transistor Q2. Hence, whatever, the discharging node (X or Y), all internal node capacitances are discharged through Q2. These discharged capacitances are then charged during the precharge phase. As a consequence, the total parasitic capacitance is balanced and is independent of the processed data. Nonetheless, this hypothesis does not hold when the NMOS tree implementing the logic function is asymmetric as it is the case in some logic gates like AND/NAND, because according to which path is switched ON, the total discharged capacitance is not the same. To counter this situation, SABL authors have transformed the logic tree that implement the AND/NAND function as shown in Figure 2.20(b) the transistor Q1 is repositioned in a such way that the node Z is connected to the discharged node (X or Y) instead of remaining floating in case of some input data sets in the original NMOS tree. This transformation does not affect the functionality of the gate since $\overline{A} + \overline{B}$ and $\overline{A}.B + \overline{B}$ are equivalent. As demonstrated by the SABL authors, this technique achieve a balance of the total parasitic capacitance in the AND/NAND gate. Nevertheless, can we talk about a generic gate if the NMOS tree must be transformed for each asymmetric logic function? Is this is not costly in terms of dedicated design-time and/or integration of logic style based library in automated synthesis tools?

MOS current mode logic (MCML) can be an efficient way to prevent power and electromagnetic attacks even it is a static logic family, the power consumption in MCML gates is independent of the input data because it is constant and equals $V_{dd}.I$. DyCML logic that operates in current mode and with a low-swing as well, has been evaluated with regard to its capability to implement secure components in [32]. It has shown the same security margins as SABL while featuring a higher speed and lower power consumption.

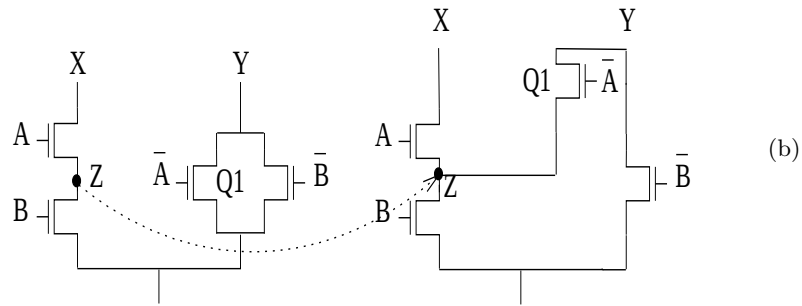
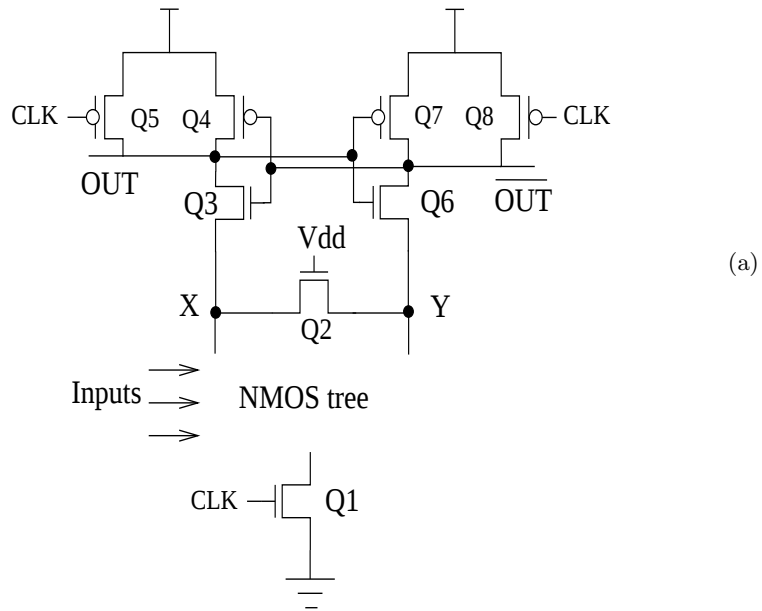


Figure 2.20: (a) SABL gate. (b) Transformation of AND/NAND gate logic tree.

2.5 Conclusion

In this chapter, we made a survey of the state of the art relating to our contribution and the target applications. Previously reported Logic styles and full adders as well as their reported performances in terms of power and speed were presented. We presented also solutions at the logic style level that target security of cryptography components against power analysis attacks.

References

- [1] M. Anis and M. Elmasry, *Multi-Threshold CMOS digital circuits, managing leakage power*, Kluwer Academic Publishers, 2003.
- [2] G. Gerosa, S. Gary, C. Dietz, D. Pham, K. Hoover, J. Alvarez, H. Sanchez, P. Ippolito, T. Ngo, S. Litch, J. Eno, J. Golab, N. Vanderschaaf, and J. Kahle, "A 2.2w, 80MHz superscalar RISC microprocessor," *IEEE J. Solid-State Circuits*, vol. 29, no. 12, pp. 1440–1454, December 1994.
- [3] J.B. Kuo and J-H. Lou, *Low Voltage CMOS VLSI Circuits*, By John Wiley & Sons, 1999.
- [4] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS digital design," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 4, pp. 473–484, April 1992.
- [5] I. Abu-Khater, A. Bellaouar, and M.-I. Elmasry, "Circuit techniques for CMOS low-power high-performance multipliers," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 10, pp. 1535–1546, October 1996.
- [6] Vojin G. Oklobdzija, "Differential and pass-transistor CMOS logic for high-performance systems," in *Proceedings of the 21th International Conference on Microelectronics*, Nis, Yugoslavia, 15-17 September 1997.
- [7] K. Yano, T. Yamanaka, T. Nishida, M. Saito, K. Shimohigashi, and A. Shimizu, "A 3.8-ns CMOS 16×16 -b multiplier using complementary pass-transistor logic," *IEEE Journal of Solid-State Circuits*, vol. 25, no. 2, pp. 388–395, April 1990.
- [8] R. Zimmermann and W. Fichtner, "Low-power logic styles: CMOS versus pass-transistor logic," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 7, pp. 1079–1089, 1997.
- [9] J.M. Quintana, M.J. Avedillo, R. Jiménez, and E. Rodriguez-Villegas, "Low-power logic styles for full-adder circuits," in *Pro-*

ceedings of International Conference on Electronics, Circuits and Systems (ICECS), 2001.

- [10] C. Piguet, V. von Kaenel, J.-M. Masgonty, J.-F. Perotto, and R. Klootsema, “Basic design techniques for both low-power and high-speed Asic’s,” in *Proceedings of EURO ASIC’92 conference*, Paris, France, 1-5 June 1992.
- [11] J.-M. Masgonty, Ph. Mosch, and C. Piguet, “Branch-based digital cell libraries,” in *Proceedings of EURO ASIC’91 conference*, Paris, France, 27-31 Mai 1991.
- [12] L. G. Heller, W. R. Griffin, J. W. Davis, and N. G. Thoma, “Custom and semi-custom design techniques,” in *Proceedings of IEEE Int. Solid-State Circuits Conference (ISSCC), Dig. Tech. Papers*, February 1984, pp. 16–17.
- [13] Pius Ng, P.T. Balsara, and D. Steiss, “Performance of CMOS differential circuits,” *IEEE Journal of Solid-State Circuits*, vol. 31, no. 6, pp. 841–846, June 1996.
- [14] G. A. Ruiz, “Evaluation of three 32-bit CMOS adders in DCVS logic for self-timed circuits,” *IEEE Journal of Solid-State Circuits*, vol. 33, no. 4, pp. 604–613, April 1998.
- [15] M. Renaudin, B. El Hassan, and A. Guyot, “A new asynchronous pipeline scheme: Application to the design of a self-timed ring divider,” *IEEE Journal of Solid-State Circuits*, vol. 31, no. 7, pp. 1001–1012, July 1996.
- [16] T. H. Y. Meng, R. W. Brodersen, and D. G. Messerschmitt, “Automatic synthesis of asynchronous circuits from high-level specifications,” *IEEE Journal of transactions on computer-aided design*, vol. 8, no. 11, pp. 1185–1205, November 1989.
- [17] M. Yamashina and H. Yamada, “An MOS current mode logic (MCML) circuit for low-power GHz processors,” *NEC research and Development*, vol. 36, no. 1, pp. 54–63, January 1995.

2.5 CONCLUSION

- [18] M.W. Allam and M.-I. Elmasry, "Dynamic current mode logic (Dy-CML): A new low-power high-performance logic style," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 3, pp. 550–558, March 2001.
- [19] A. Fahim and M.-I. Elmasry, " SC^2L : A low-power high-performance dynamic differential logic family," in *IEEE Int. Symp. Low Power Electronics Design*, 1999, pp. 88–90.
- [20] A. Fahim and M.-I. Elmasry, "Low-power high-performance arithmetic circuits and architectures," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 1, pp. 90–94, January 2002.
- [21] J.-D. Legat, "Comparaisons des logiques différentielles à faible consommation et à amplitude réduite," in *4me journées d'études Faible Tension Faible Consommation (FTFC)*, Paris, May 14-16 2003, pp. 69–74.
- [22] *Low-power Electronics Design*, Edited by C. Piguet, CRC Press, 2004.
- [23] J. Schutz, "A 3.3V 0.6 μ m BiCMOS superscalar microprocessor," in *Proceedings of ISSCC Dig. Tech. Papers*, February 1994, pp. 202–203.
- [24] G. K. Theodoropoulos, "Modelling and distributed simulation of asynchronous hardware," *Journal of simulation practice and theory*, vol. 7, pp. 741–767, 2000.
- [25] K. Martin, *Digital Integrated Circuit Design*, Oxford University Press, 2000.
- [26] N. Zhuang and H. Wu, "A new design of the CMOS full adder," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 5, pp. 840–844, May 1992.
- [27] C-H. Chang, J. Gu, and M. Zhang, "A review of 0.18 μ m full adder performances for tree structured arithmetic circuits," *IEEE Journal of Transactions On VLSI Systems*, vol. 13, no. 6, pp. 686–694, June 2005.

- [28] I-S. Abu-Khater, A. Bellaouar, and M-I. Elmasry, “Circuit techniques for CMOS low-power high-performance multipliers,” *IEEE Journal of Solid-State Circuits*, vol. 31, no. 10, pp. 1535–1546, October 1996.
- [29] D. Sokolov, J. Murphy, A. Bystrov, and A. Yakovlev, “Design and analysis of dual-rail circuits for security applications,” *IEEE Transactions on Computers*, vol. 54, no. 4, pp. 449–459, April 2005.
- [30] S. Moore, R. Anderson, P. Cunningham, R. Mullins, and G. Taylor, “Improving smart card security using self-timed circuits,” in *Proc. Int. Symp. Asynchronous Circuits and Systems (ASYNC)*, 2002.
- [31] Z. C. Yu, S. B. Furber, and L. A. Plana, “An investigation into the security of self-timed circuits,” in *Proceedings of Int’l Symp. Asynchronous Circuits and Systems (ASYNC)*, 2003, pp. 206–215.
- [32] F. Mace, F-X. Standaert, I. Hassoune, J-D. Legat, and J-J. Quisquater, “A dynamic current mode logic to counteract power analysis attacks,” in *Proceedings of DCIS conference*, Bordeaux, France, 2004, pp. 186–191.
- [33] K. Tiri and I. Verbauwhede, “Securing encryption algorithms against DPA at the logic level: Next generation smart card technology,” in *Proceedings of CHES, Springer-Verlag*, 2003.
- [34] 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 ESSCIRC Conference*, 2002.

CHAPTER 3

CLASS OF LOW SWING CURRENT MODE LOGIC STYLES

3.1 *Introduction*

In this chapter, a new class of low-swing current mode logic [1] is presented. It features a dynamic differential structure and a reduced swing current mode operation and moreover, offers a self-timing scheme. Thanks to its small output swing, the dynamic power is reduced. We describe hereafter the structure and operation of the original version called LSCML but also the progressive optimizations that we carry out on the LSCML to achieve a better power delay product. This was first realized through the self-timing scheme ST2 which has achieved a good power delay product at the expense of an increase of power consumption, IFLSCML has demonstrated a reduction of both power and delay in comparison with LSCML, and finally DDSLL has achieved further reduction of power delay product at the expense of a higher transistor count.

3.2 *LSCML structure and operation*

Figure 3.1 shows the basic structure of the LSCML logic gate as we propose here. It consists in a dynamic current source realized by transistors (Q3,Q1), a precharge circuit (Q6,Q7,Q2), a NMOS tree for logic evaluation, a latch (Q8,Q9) to maintain logic output value after evaluation and a feedback circuit onto the dynamic current source, realized by two PMOS transistors (Q4,Q5). Figure 3.2 shows the output waveforms of a LSCML gate. The basic operation of such LSCML gate is explained as follows. When the clock signal En_i is low (precharge phase, denoted (1) in Figure 3.2), the transistors Q6 and Q7 turn ON and charge the output nodes to V_{dd} , while the transistor Q2 switches ON to discharge the

CHAPTER 3. CLASS OF LOW SWING CURRENT MODE LOGIC STYLES

node ENO to 0. On the other hand, since the transistor Q1 is switched OFF, there is no DC path from V_{dd} to ground. During the evaluation phase (denoted (2) in Figure 3.2), the clock signal En_i goes high, the precharge circuit (Q6, Q7, Q2) is switched OFF while the transistor Q1 is turned ON. Since the node ENO was discharged to 0, the transistor Q3 is switched ON. Therefore, a current path is created from the output nodes to ground. It is to be noticed that in low swing current mode logic, some transistors are fully ON while the others are partially ON. As these two paths have different impedances depending on the data input values, one output node will be discharged faster than the other one. In the latch circuit (Q8, Q9), the transistor whose gate is connected to the node which discharges faster, will turn ON as soon as the voltage at this node is less than $V_{dd} - |V_{tp}|$, and will charge the other output node to V_{dd} . When this condition is fulfilled, one transistor of the feedback circuit (Q4, Q5) turns ON and charges the node ENO to V_{dd} , this will consequently turn OFF the transistor Q3 thereby limiting the amount of voltage drop.

3.3 LSCML gate cascading

The LSCML gates can be used in a self-timed cascading as each gate generates the completion signal for the subsequent logic gate. This signal can be the voltage at node ENO as it does not switch until the evaluation is completed. As its slope is too low because of the insufficient drive capability that results from the low V_{gs} voltage in the feedback transistor, a clock buffering is needed to enhance it.

3.3.1 Clock buffering with simple inverters

The first solution used to regenerate the completion signal was simply two inverters in series. However, this solution has proved to be disadvantageous in terms of both power and delay as it will be shown in section 3.3.2. Thus, other solutions have been investigated.

3.3 LSCML GATE CASCADING

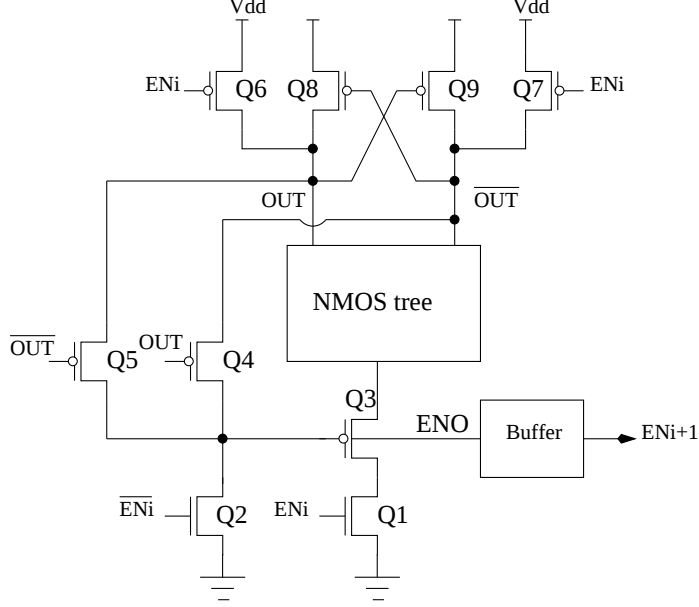


Figure 3.1: LSCML gate.

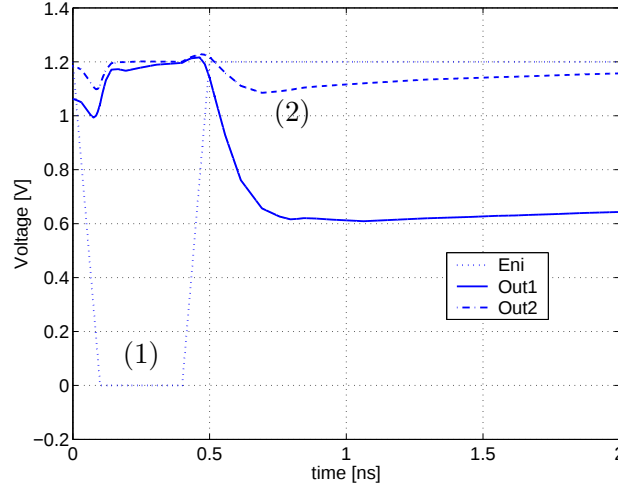


Figure 3.2: LSCML gate output waveforms, where *Out1* and *Out2* depict the voltages at outputs *OUT* and $\overline{OUT}$ nodes of a carry gate in $0.13\mu\text{m}$ PD SOI/CMOS, ST LL (low leakage) MOSFETs ($V_t = 0.36\text{V}$) under $V_{dd} = 1.2\text{V}$ with transistor sizes shown in appendix A and $C_L = 3fF$. It can be seen that during the evaluation phase, one node charges to V_{dd} , while the other discharges to $V_{dd} - V_{swing}$.

3.3.2 Clock buffering with ST1 scheme

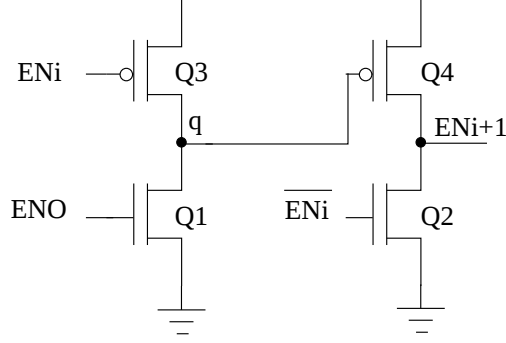
We used the same self-timing buffer as in DyCML [2] to regenerate the completion signal. This buffer is used in the DyCML [2] to convert the voltage on the capacitor which is a small swing signal to a full-swing signal (as described in section 2.2.3). In LSCML logic, this self-timing buffer is used to improve the signal slope and not to convert it to a full-swing signal because the voltage at node ENO is already a full-swing signal. The operation of the self-timing buffer which is depicted in Figure 3.3(a), is explained as follows. When the clock EN_i is low, the transistor Q3 is switched ON and charges the node q to V_{dd} , while the transistor Q1 is turned off as the node ENO is pulled down to 0 during the precharge phase. Meanwhile, the transistor Q2 turns ON and discharges the node EN_{i+1} to 0. When the clock EN_i starts to rise, the transistors Q3 and Q2 switch off. On the other hand, when the voltage at node ENO goes high, the node q discharges to 0 through Q1. This in turn will switch ON the transistor Q4. Therefore, the node EN_{i+1} charges to V_{dd} . The signal at node q is used for the signal complement $\overline{EN_{i+1}}$.

Table 3.1 shows the substantial advantages in terms of power consumption and delay time of the ST1 scheme over the simple clock-buffering with two inverters in series, through 1b and 8b LSCML carry-propagate circuits. This advantage is mainly due to the absence of short-circuit currents in the ST1 self-timing scheme.

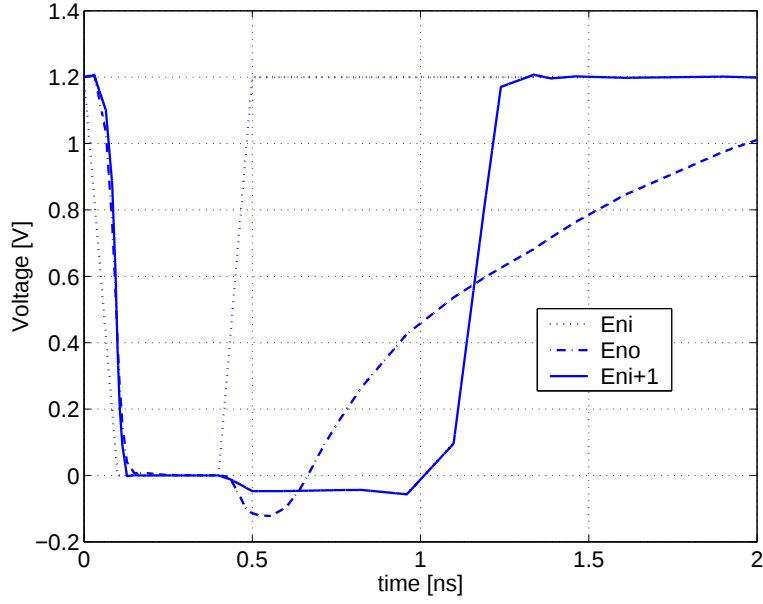
3.3.3 Clock buffering with ST2 scheme

Even though we used the self timing scheme ST1, the generation of the completion signal remains not sufficiently fast. We then tackled the problem at its source by making the self-timing in LSCML independent of the feedback circuit to speed-up cascaded LSCML gates. The solution depicted in Figure 3.4(a) is then used. It consists in a AND/NAND gate conditioned by the clock signal of the “current” logic stage. This

3.3 LSCML GATE CASCADING



(a)



(b)

Figure 3.3: Self-timing buffer (ST1) [2] (a). LSCML (ST1) self-timing scheme waveforms (b), where EN_{i+1} depicts the buffered completion signal obtained from ENO through the ST1 scheme circuit. Obtained in a carry gate in $0.13\mu\text{m}$ PD SOI/CMOS, ST LL (low leakage) MOSFETs ($V_t = 0.36\text{V}$) under $V_{dd} = 1.2\text{V}$, with transistor sizes shown in appendix A and $C_L = 3fF$.

CHAPTER 3. CLASS OF LOW SWING CURRENT MODE LOGIC STYLES

	LSCML Simple clock buffering	LSCML (ST1)
Average power [μ W] at f=100MHz	4	2.24
Delay [ns]	1.23	0.61
PDP [fJ] at f=100MHz	4.92	1.36
Transistor count	29	29

(a)

	LSCML Simple clock buffering	LSCML (ST1)
Average power [μ W] at f=50MHz	20	13.7
Delay [ns]	12.2	5.8
PDP [fJ] at f=50MHz	244	79.46
Transistor count	232	232

(b)

Table 3.1: Clock-buffering circuit comparison in $0.13\mu\text{m}$ PD SOI CMOS technology under $V_{dd} = 1.2V$, $V_{t0} = 0.36V$ and floating body devices. 1b carry-propagate circuit comparison (a). 8b carry-propagate circuit comparison (b).

3.3 LSCML GATE CASCADING

solution will be called self-timing (ST2). In Figure 3.4(a), OUT and $\overline{OUT}$ depict the full-swing outputs of the current logic stage “i”, En_i is the clock signal of the logic stage “i” while En_{i+1} and $\overline{En_{i+1}}$ are the clock signal and its complement required for the subsequent logic stage “i+1”. The full-swing outputs are obtained using the single ended buffer proposed in [2]. The operation of this level converter is described in section 3.4.

The basic operation of the self-timing scheme (ST2) is explained as follows. When En_i is low (precharge phase), the nodes OUT and $\overline{OUT}$ are at V_{dd} . The node En_{i+1} is then discharged to 0. This will in turn switch ON the transistor Q2 and charge the node $\overline{En_{i+1}}$ to V_{dd} . Meanwhile, there is no current path through the transistor Q3. When En_i goes high, two cases can occur. First, OUT is at V_{dd} while $\overline{OUT}$ is at 0. In the second case, OUT is at 0 while $\overline{OUT}$ is at V_{dd} . In both, the node $\overline{En_{i+1}}$ is discharged to 0. This will turn ON the transistor Q1 and charge the node En_{i+1} to V_{dd} . The resulting output signals En_{i+1} and $\overline{En_{i+1}}$ are full-swing and have a quite high slope as it can be seen on Figure 3.4(b). Hence, no buffering is required and the resulting outputs can be used directly as a clock signal for the following LSCML logic stage. This self-timing solution allows a high-speed operation as the completion signal is generated as soon as the evaluation of the “current” logic stage is completed and its delay does not depend on the feedback circuit. However, the drawback of the self-timing (ST2) is that the signals OUT and $\overline{OUT}$ are the full-swing output voltages. Therefore, it requires implementation of the single ended buffers, which increases the power consumption. This is different from the self-timing (ST1) where the completion signal is generated independently of the full-swing output voltage. Hence, implementation of the single ended buffers is not required in this case.

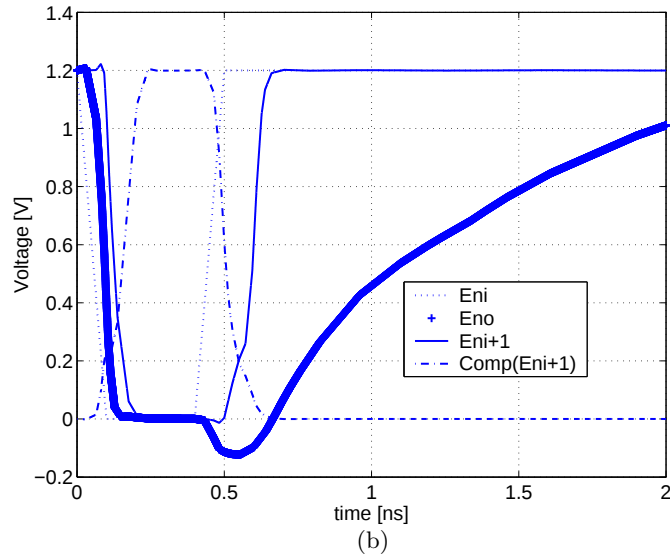
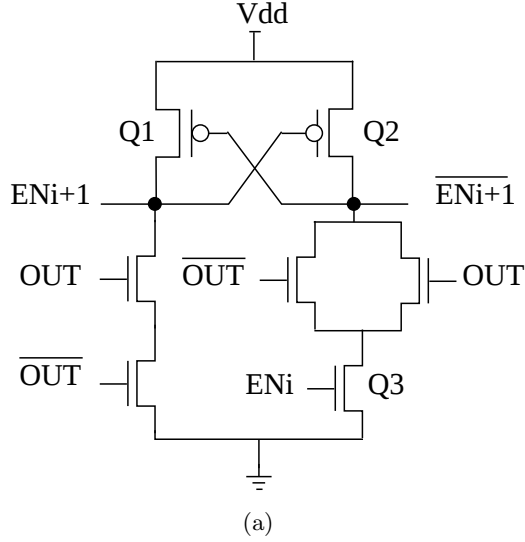


Figure 3.4: LSCML (ST2) self-timing scheme (a). LSCML (ST2) self-timing scheme waveforms (b), where ENO is the completion signal resulting from the feedback circuit in LSCML gate, while $Eni+1$ and $comp(Eni+1)$ depict the output completion signals generated by the ST2 scheme. Therefrom, we can observe the delay time difference between generation of the completion signals Eno and $Eni+1$. Obtained in a carry gate in $0.13\mu m$ PD SOI/CMOS, ST LL (low leakage) MOSFETs ($V_t = 0.36V$) under $V_{dd} = 1.2V$, with transistor sizes shown in appendix A and $C_L = 3fF$.

3.4 *Interfacing with full-swing logic styles*

Interfacing between the LSCML gates and full-swing logic gates is achieved through the single-ended buffer [2] which is a special level converter. It is shown in Figure 3.5(a). The conversion of the small output voltage to a full-swing voltage is described as follows. When the clock is low, the outputs of LSCML circuit are precharged to V_{dd} . The transistor Q2 is then switched off while the transistor Q1 turns ON discharging the node q to 0 and the output is then at V_{dd} . When the clock goes high, the transistor Q1 switches off while the state of the transistor Q2 will depend on the state of the output entering its gate. The transistor Q2 will then either turn ON and charge the node q to V_{dd} , this leads to state 0 at the output, or it will remain switched off and the output will be then at V_{dd} .

Figure 3.5(b) shows the waveforms of conversion of small output swing signal to full swing signal through the single ended buffer. OUT and Comp(OUT) denote the low swing signal, Out1 and Out2 denote the full-swing signal.

3.5 *Functioning behaviour of LSCML circuits*

Even though LSCML circuits have digital applications purpose, their functioning behaviour makes them as sensitive as analog circuits to some parameters variations. The output voltage swing ΔV depends on the current value I drained by transistor Q3 of the circuit in Figure 3.1. As illustrated in Figure 3.6, the output voltage swing increases with the W/L ratio of transistor Q3. When the $W/L(Q3)$ ratio starts to become significantly high (larger than 15 according to Figure 3.6), the junction capacitances of the transistor Q3 subsequently increase and charge sharing occurs between these junction capacitances and the parasitic capacitance at the output. Therefore, the output voltage swing starts to decrease. If a reduction of the current I decreases ΔV and benefits

CHAPTER 3. CLASS OF LOW SWING CURRENT MODE LOGIC STYLES

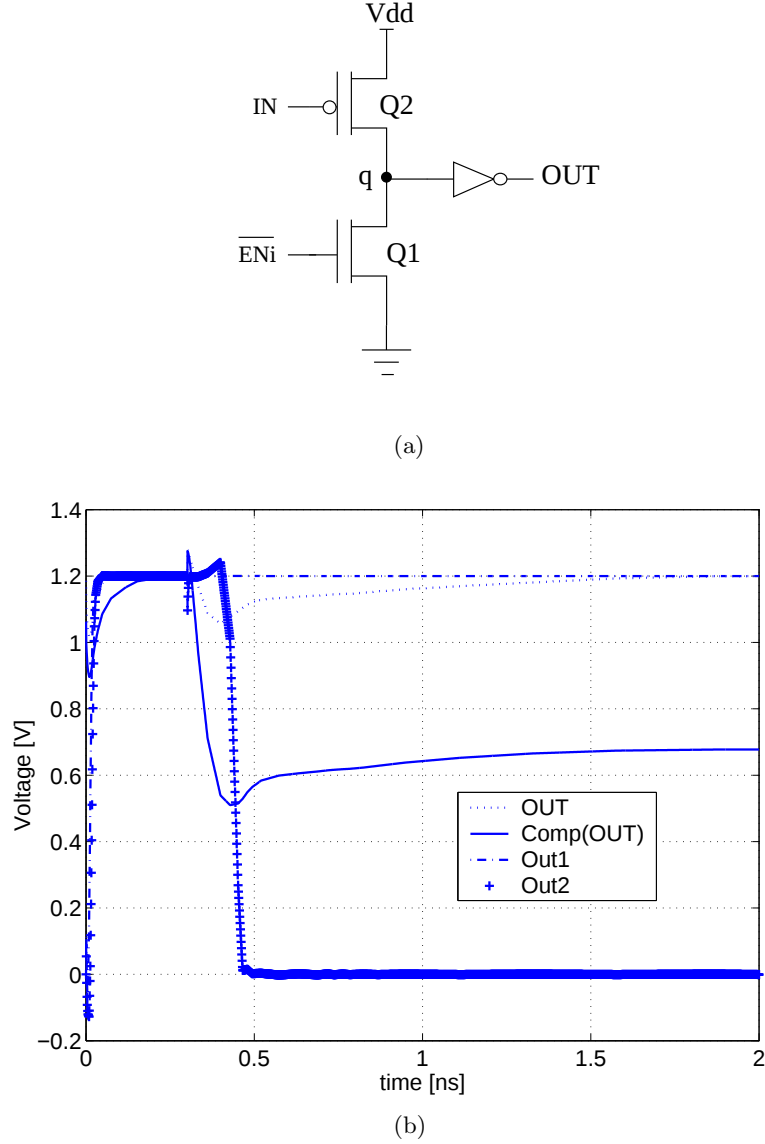


Figure 3.5: Single ended buffer [2] (a). Swing buffering waveforms (b), where (OUT, Comp(OUT)) depict the low swing signal and (Out1, Out2) depict the full swing signal obtained through the single ended buffer. Obtained in a carry gate in $0.13\mu\text{m}$ PD SOI/CMOS, ST LL (low leakage) MOSFETs ($V_t = 0.36\text{V}$) under $V_{dd} = 1.2\text{V}$, with transistor sizes shown in appendix A and $C_L = 3fF$.

3.5 FUNCTIONING BEHAVIOUR OF LSCML CIRCUITS

to the power consumption, it will on the other hand decrease the noise margin of the LSCML circuit and speed performance. The voltage drop across the output is limited by the feedback circuit. The functionality of the latter depends on the threshold voltage of the PMOS transistors in the feedback circuit. Hence, increasing the V_{tp} of those transistors leads to an increase in output voltage swing and delay time of the signal at node ENO. Indeed, the gate voltage of transistor Q3 depends on the current flowing through the PMOS transistor which is switched ON in the feedback circuit. Subsequently, the delay of the signal at node ENO increases when using high V_{tp} . Therefore, the output voltage swing increases also as the voltage drop is limited when the node ENO is charged to V_{dd} .

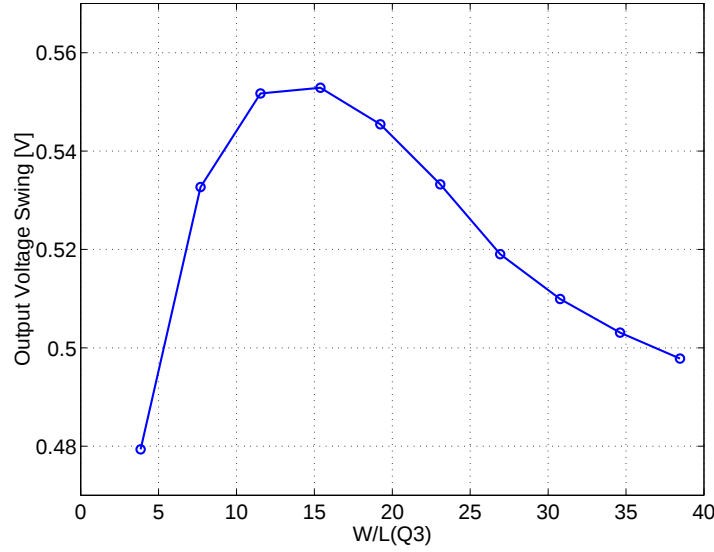


Figure 3.6: Effect of the transistor Q3 sizing on the output voltage swing.

3.6 Modified LSCML gate (IFLSCML)

In the LSCML gates as shown in Figure 3.1, the source capacitances of the PMOS transistors in the feedback circuit increase the parasitic capacitance at the output nodes. This negatively affects both delay and power consumption. To prevent this drawback, the sources of the PMOS transistors in the feedback circuit will be connected to V_{dd} instead of the output nodes as shown in Figure 3.7. Moreover, as the output voltage swing in the LSCML gate is sensitive to the fanout, the improved feedback shown in Figure 3.7 will ensure more stable gate-to-source voltage of the PMOS transistor in the feedback circuit and therefore, a more stable driving capability at node ENO. This results in an improvement of evaluation and completion signal delays. The modified LSCML gate that uses this solution will be denoted IFLSCML (Improved feedback low swing current mode logic).

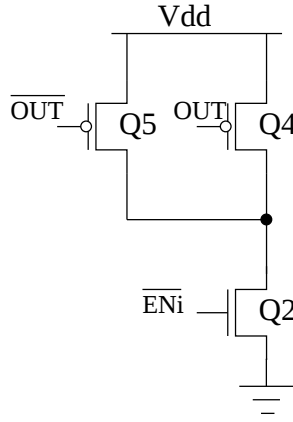


Figure 3.7: Improved feedback circuit in IFLSCML gate.

3.7 Further optimizations of the LSCML

The feedback circuit in LSCML gate as shown in Figure 3.1, limits the performance because it slows down the completion signal. We propose

3.8 STABILITY OF LSCML CLASS OPERATION

Dynamic Differential Swing Limited Logic (DDSLL) to overcome this drawback. Figure 3.8 shows the basic structure of DDSLL gates. Unlike LSCML gates, the precharge circuit is made up of four transistors (Q6,Q7,Q2,Q10), transistors (Q4,Q5,Q11) form the feedback circuit, while a NMOS transistor (Q3) is used in the dynamic current source instead of a PMOS transistor. The basic operation of DDSLL gates is explained as follows. When En_i is low, Q6 and Q7 turn ON and charge the output nodes to V_{dd} , while Q10 switches ON to charge the node $\overline{EN0}$ up to V_{dd} . Meanwhile, Q4 and Q5 are switched OFF and Q2 turns ON to discharge the node s to 0. Q11 is then switched OFF. Since Q1 is turned OFF, there is no DC path from V_{dd} to ground. When En_i goes high, Q1 switches ON and the precharge circuit (Q6,Q7,Q2,Q10) is switched OFF, while Q3 is turned ON because the node $\overline{EN0}$ was charged to V_{dd} . The logic function is then evaluated. Q4 or Q5 will then turn ON and charge the node s to V_{dd} . Q11 switches ON to discharge the node $\overline{EN0}$ to 0. Q3 switches OFF, thereby limiting the amount of voltage drop.

The DDSLL gate cascading is achieved through a simple inverter that takes the signal at node $\overline{EN0}$ as input signal. The resulting complementary signal En_{i+1} is used as a clock signal for the subsequent stage. DDSLL gate output waveforms are shown in Figure 3.9.

Interfacing between DDSLL gates and full-swing logic families can be achieved through the same level converter as depicted in section 3.4.

3.8 *Stability of LSCML class operation*

The impact of fanout, input voltage swing and supply voltage on output swing in the three variants of LSCML is illustrated in Figures 3.10 and 3.11.

From Figure 3.10, it can be seen that the output voltage swing ΔV is almost insensitive to the input voltage swing variation in LSCML and IFLSCML gates. Subsequently LSCML and IFLSCML gate per-

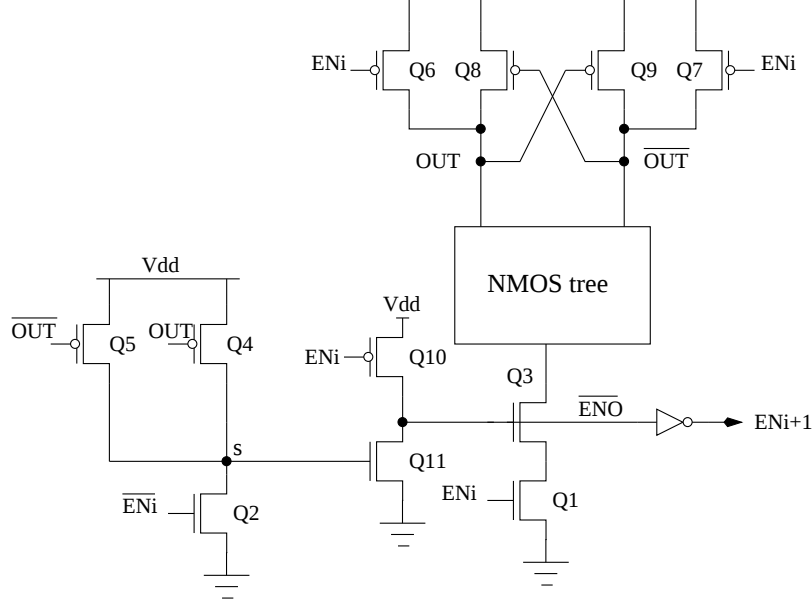
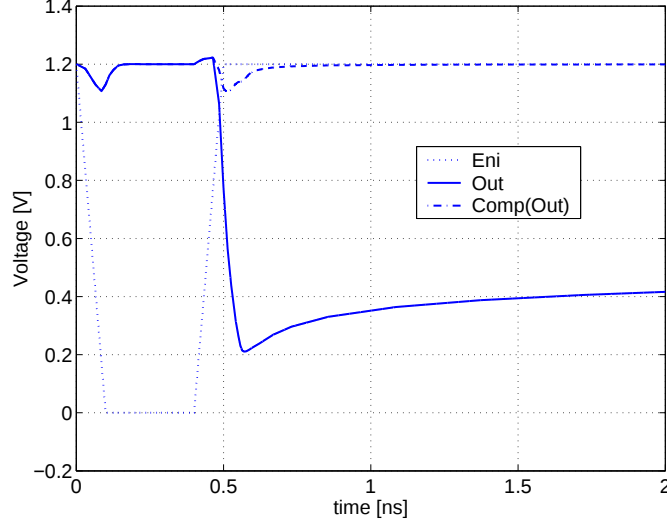


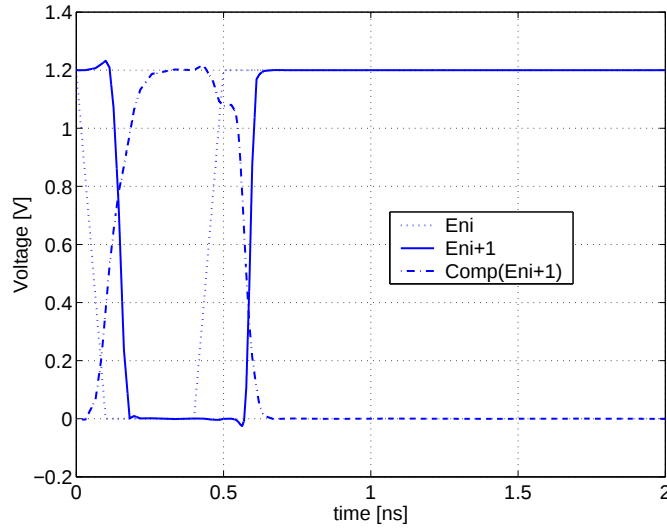
Figure 3.8: DDSLL gate.

formances are almost not affected by an input voltage swing variation. Conversely, ΔV sensitivity to input voltage swing seems to be more obvious in DDSLL. However, as Figure 3.12 shows, a variation of 0.7V on the input voltage swing causes a very little change on the power delay product of a 8b DDSLL RCA. As it can be observed, the output swing in DDSLL is higher in comparison to the one in LSCML and IFLSCML. This is due to two main reasons. First, the DDSLL uses a NMOS transistor instead of a PMOS transistor in the dynamic current source. For the same W/L , the NMOS transistor ensures more driving capability than the PMOS transistor. This speeds up the evaluation and increases the output swing ΔV since the output swing in current mode logic depends on the current value drained by the current source. Secondly, the voltage drop across the output in LSCML and DDSLL is limited by the feedback circuit. This latter is not the same in the two logic variants. In LSCML gates, the voltage drop is stopped as soon as the V_{gs} (i.e. ΔV) of the PMOS transistor in the feedback circuit becomes more than

3.8 STABILITY OF LSCML CLASS OPERATION



(a)



(b)

Figure 3.9: DDSLL gate output waveforms, where Out and Comp(Out) depict the voltages at the outputs nodes OUT and $\overline{\text{OUT}}$ (a). DDSLL self-timing scheme waveforms, where En_{i+1} depicts the generated completion signal through the inverter and Comp($En_i + 1$) depicts its complement generated at node $\overline{ENO}$ (see Figure 3.8) (b). Obtained in an inverter gate in $0.13\mu\text{m}$ PD SOI/CMOS, ST LL (low leakage) MOS-FETs ($V_t = 0.36\text{V}$) under $V_{dd} = 1.2\text{V}$, with transistor sizes shown in appendix A and $C_L = 3fF$.

CHAPTER 3. CLASS OF LOW SWING CURRENT MODE LOGIC STYLES

$|V_{tp}|$. This is not the case in DDSLL gates where the voltage drop is stopped only when Q11 switches ON.

Figure 3.11 compares the impact of V_{dd} on the output voltage swing ΔV in the three variants of LSCML class. The output voltage swing decreases with the supply voltage scaling. However, the impact of V_{dd} on ΔV seems to be a bit less marked in IFLSCML than in LSCML. Though the LSCML class speed performance decreases with supply voltage scaling, their operation remains correct.

Due to the sensitive behaviour of LSCML, IFLSCML and DDSLL gates, circuits based on these logic families should be supplied by a separate V_{dd} if they are combined with full-swing logic on the same design in order to avoid supply noise problems.

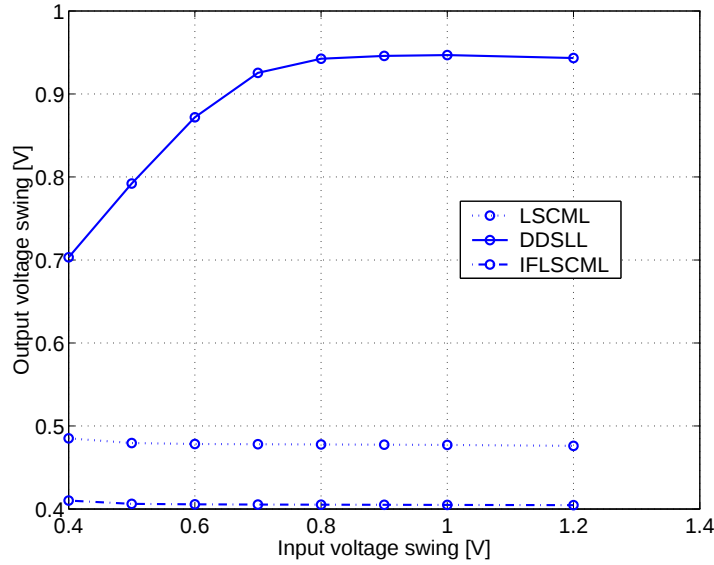


Figure 3.10: Effect of the input voltage swing on the output voltage swing.

3.8 STABILITY OF LSCML CLASS OPERATION

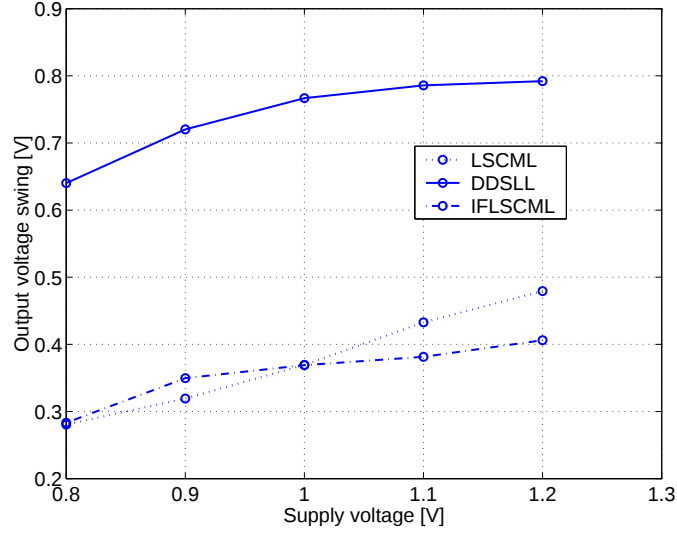


Figure 3.11: Effect of the supply voltage on the output voltage swing.

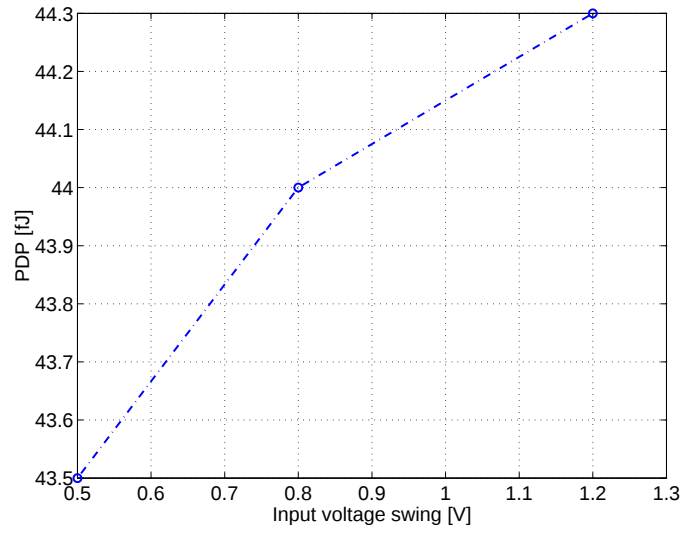


Figure 3.12: PDP of a DDSLL-based 8-bit RCA versus the input voltage swing.

CHAPTER 3. CLASS OF LOW SWING CURRENT MODE LOGIC STYLES

3.9 conclusion

In this chapter, we presented LSCML structure and its basic operation, as well as the numerous optimizations that we carried out to achieve better performance in terms of power-delay product. Evaluation of the original structure and these different optimizations is carried out versus two other valuable dynamic differential logic styles in the next chapter.

References

- [1] I. Hassoune and J-D. Legat, “Low swing current mode logic family,” *Patent number: WO2005112263, Priority number: US20040571383P 20040514*, November 2005.
- [2] M.W. Allam and M.-I. Elmasry, “Dynamic current mode logic (Dy-CML): A new low-power high-performance logic style,” *IEEE Journal of Solid-State Circuits*, vol. 36, no. 3, pp. 550–558, March 2001.

CHAPTER 3. CLASS OF LOW SWING CURRENT MODE LOGIC STYLES

CHAPTER 4

INVESTIGATED APPLICATIONS OF THE LSCML STYLE

4.1 *Introduction*

In this chapter, we carry out a performance evaluation of the LSCML class through electrical simulations of several basic and complex gates. Furthermore, the power consumption behaviour in the different versions of LSCML is assessed through simulation results of the power consumption of a module of the Khazad cipher algorithm (the Khazad S-box). These study has been performed in $0.13\mu\text{m}$ PD (partially depleted) SOI CMOS technology. The applicability of LSCML in implementation of self-timed arithmetic circuits is further shown through simulation results of an 8b RCA and 8b CLA adders. Therefrom, in comparison with other logic styles previously targeting the same objectives, LSCML class further demonstrates an advantageous compromise between power and reduction of the power consumption variation, which gives LSCML a real potential to implement secure devices against power analysis attacks. Indeed, the LSCML(ST1) S-box has shown a power consumption standard deviation more than two times smaller than the one in DyCML and six times smaller than the one in self-timed DDCVSL.

4.2 *Power analysis attacks*

The power consumption behaviour can be used by the attacker to recover the secret information handled by the circuit. Indeed, the attacker can estimate the power consumption of a CMOS circuit at time t by simply predicting the number of transitions occurring at this time. Let us suppose the circuit under attack is realizing the ciphering of data using a key with a length of K bits. The purpose of an attack is to determine the K bits of the key, or at least, a smaller number of bits k . Then, for these

CHAPTER 4. INVESTIGATED APPLICATIONS OF THE LSCML STYLE

k bits, there exist $n = 2^k$ possible values of the targeted key. A prediction of the power consumption can be made for each possible key, which will be or not correlated to the real power consumption behaviour of the circuit using the right key. The importance of this correlation essentially depends on the quality of the power prediction model, the quality of the power consumption measurements and the type of logic style with which the circuit is realized.

To realize an efficient power analysis attack, it will be divided in 3 steps [1]:

- STEP 1 (Prediction): The attacker predicts, for a number m of plaintexts to be encrypted (i.e. the message to be encrypted), and for the n possible values of the targeted key bits, a prediction of the number of transitions occurring in the circuit for each plaintext. This leads to a prediction matrix P of size m x n.
- STEP 2 (Measurements): The attacker measures, for the same m plaintext used during the prediction phase, the power consumption of the circuit realizing the encryption. He thus obtains a measurement vector M of length m.
- STEP 3 (Correlation): The attacker correlates the value present in each column of the prediction matrix P with the measurements present in the measurement vector M. This correlation is achieved using the Pearson coefficient [1]. For each key guess, if the prediction is incorrect, the computation of the correlation gives a small value compared to the correct prediction that produces the highest value of the correlation.

The attack efficiency depends only on the correlation between measurements and theoretical predictions of the power consumption.

4.3 *Figures of merit*

Authors in [2] evaluate robustness against power analysis attack of SABL (reviewed in section 2.4.2) according to the Normalized Energy Devia-

4.3 FIGURES OF MERIT

tion (NED) and Normalized Standard Deviation (NSD) expressed as follows:

$$NED = \frac{\max(P) - \min(P)}{\max(P)} \quad (4.1)$$

$$NSD = \frac{\sigma}{\mu} \quad (4.2)$$

where $\max(P)$ and $\min(P)$ are the maximum and the minimum power consumptions respectively, σ the standard deviation of the power consumption and μ its mean value. For N input sets applicable to the S-box, these parameters are computed as follows:

$$\mu = \frac{\sum_{i=1}^N P_i}{N} \quad (4.3)$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (P_i - \mu)^2}{N - 1}} \quad (4.4)$$

Authors in [3] have investigated the cryptographic relevance of NED and NSD criteria and have demonstrated that they are not optimal statistical parameters to evaluate the practical resistance of a logic style to power analysis attacks and that even though a dedicated logic style having reduced NED and NSD is used to implement the critical parts of a cipher algorithm, a power analysis attack remains theoretically feasible against it. The attack efficiency depends only on correlation between measurements and theoretical predictions of the cipher component power consumption. Thus standard deviations, NSD and NED, have no practical relevance. Nonetheless, as stated in [3] [4], under real conditions, neither the predictions nor the measurements are perfect, and thus generate noise. Therefore, for the same level of noise, the logic style achieving the higher reduction of the power consumption variation will correspond to the more decreased correlation value. Indeed, the reduction of the power consumption variation causes the measurements to have to become more accurate and thus the measurements to become more difficult. The reduction factor of the correlation when using dedi-

CHAPTER 4. INVESTIGATED APPLICATIONS OF THE LSCML STYLE

cated logic style (i.e. dynamic and differential logic style), is difficult to quantify since it strongly depends on measurement setup of the attacker, measurement noise...

The features on which we will particularly focus our interest, are the standard deviation and the mean value of the power consumption since they allow to fairly assess the efficiency of the countermeasure. Indeed, in [3], it has been demonstrated that the DyCML achieves the same performances as SABL in terms of NSD and NED while reducing the standard deviation and the mean power consumption by a factor of two. Furthermore, the DyCML S-box has shown largely better delay and power delay product and lower device count.

4.4 *Khazad S-box*

In order to evaluate the LSCML class in terms of security concept, we focused on a module of a particular cipher algorithm called Khazad [5]. The Khazad S-box is composed of small pseudo-randomly generated 4 bit to 4 bit mini-boxes implementing the P and Q functions [5]. The structure of the Khazad S-box is illustrated in Figure 4.1.

The implementation of the S-boxes in our simulation was done as follows. Each 4 bit to 4 bit mini-box was decomposed into four 4 bit to 1 bit functions and each of these functions was implemented as a single gate. We applied the methodology proposed in [6] to choose the adequate structure for the differential pull down network implementing the function in each gate. This methodology ensures that the power consumption of the implemented functions will be the hardest to predict. Indeed, the implementation of the logic P,Q pull down networks at the transistor level were designed to nearly symmetrize the number of transistors connected to the output nodes as it can be observed in Figure 4.2.

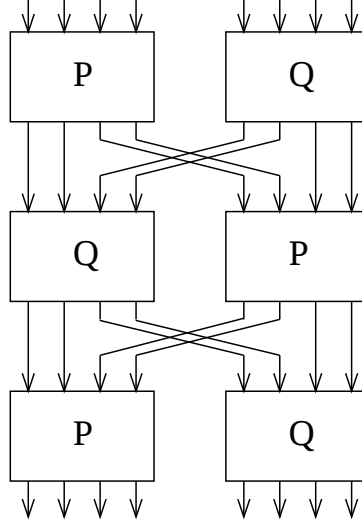


Figure 4.1: Khazad S-box structure.

4.5 Logic styles selected for comparison

As explained in section 2.4.2, the value of $\alpha_{0 \rightarrow 1}$ depends on the type of logic style and the type of logic function. Therefrom, for the same logic function, the power consumption behaviour of standard CMOS, which is a widely used logic style (this logic style will simply be denoted hereafter as CMOS) easily emphasizes its weaknesses because of the clear data dependency of its power consumption. The probability of an output transition from a “0” level to a “1” level, corresponding to the charge of the parasitic capacitance from 0V to V_{dd} , is directly linked to the function implemented within the CMOS gate, and thus to the data handled by the circuit.

To counteract the leakage of information related to the power consumption behaviour of CMOS, it has been proposed to use other logic styles than CMOS. These logic styles are differential (logic styles for which we compute both the result of the logic function f implemented within the gate and its complementary $\bar{f}$) and dynamic (logic styles for which the execution time is divided into two phases and controlled by a specific signal; these phases being a precharge phase consisting in precharging

CHAPTER 4. INVESTIGATED APPLICATIONS OF THE LSCML STYLE

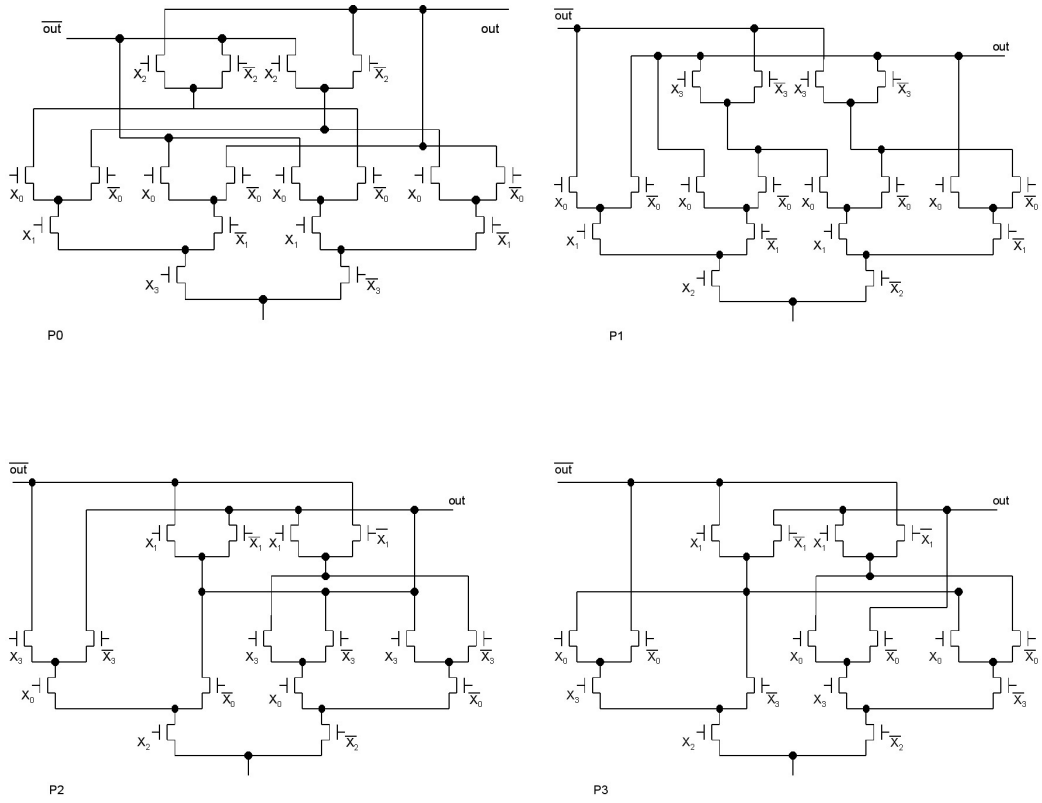


Figure 4.2: NMOS networks logic trees (P-box).

4.5 LOGIC STYLES SELECTED FOR COMPARISON

the outputs of each logic gate and an evaluation phase for which one of the outputs is discharged depending on the inputs). Indeed, as introduced in 2.4, these logic styles have the advantage on CMOS that they ensure 100% switching activity ($\alpha_{0 \rightarrow 1} = 1$) for each input set. Indeed, in dynamic differential logic styles, there is always one output transition during a precharge/evaluation cycle, and this independently of the presence/absence of transitions on the data inputs. We will then say that dynamic and differential logic styles achieve a regular switching activity for the circuit. It is then clearly not possible to correlate the measured power consumption with the number of predicted transitions occurring within the circuit, as each gate realizes a switching for each cycle.

Consequently, logic styles to be preferably considered for their robustness to DPA are these with dynamic and differential structure. Thus, they constitute the focus of this chapter.

In addition, because they ease self-timing implementation, such logic styles could also feature interesting properties with regard to power challenges in the future nanoscaled multi-Gigahertz technologies. In order to meet the temporal and the power consumption specifications, clocking has indeed become one of the most important considerations when designing digital systems.

Nevertheless, it has been shown that all dynamic and differential logic styles are not equal in terms of security against power analysis attacks [2] [3]. Indeed, even if they all realize a regularity of the switching activity and thus make the prediction of the transitions occurring within the circuit not usable, due to the implementation of the logic function by a particular network of transistors, small variations still appear in the power consumption depending on the inputs applied to the gate. These variations are in fact associated to variations of the total load capacitance, given the variation of the contribution, to this capacitance, of the network of transistors [2].

To counteract this phenomenon, it was proposed to use certain logic styles able to uncorrelate the power consumption and the data handled, at both the levels of the switching activity and the influence of the

CHAPTER 4. INVESTIGATED APPLICATIONS OF THE LSCML STYLE

parasitic capacitances. Amongst these, in Sense Amplifier Based Logic (SABL) the whole internal capacitance of the gate is discharged for each input sequence applied to the gate [2]. This makes the power consumption regular at the expense of an increase in the power consumption.

4.5.1 DDCVSL(ST) versus DDCVSL(CD)

In order to evaluate LSCML with regard to security concept and further applications, we choose the DDCVSL because it is a very popular dynamic differential logic in both academic and industrial research communities due to its compactness (i.e. area) and its reported high advantage when compared to many other logic styles [7] in terms of power delay product. DDCSVL has been investigated with two clocking schemes for clock generation in cascaded DDCVSL gates: clock delay (denoted DDCVSL(CD)) and with self-timing (denoted DDCVSL(ST)). The two clocking schemes have been reviewed in chapter 2. The DDCVSL gates and circuits using these two clocking schemes are compared in Tables 4.1, 4.2.

Table 4.1 shows comparison of basic logic gates. Therefrom, it can be seen that the AND/NAND and XOR/XNOR gates implemented with DDCVSL(ST) achieve lower power consumption while those implemented with DDCVSL(CD) have lower delay and power delay product. More complex gates like the FA 1b appear to be more advantageous in terms of power and PDP when implemented with DDCVSL(ST). This trend is reinforced in further complex circuits as it can be observed on the RCA and CLA results shown in Tables 4.2(a) and (b), where both power and delay are largely better in the DDCVSL(ST) circuits. The advantage of the DDCVSL(ST) circuits over the DDCVSL(CD) with respect to both power and delay is mainly due to the sizing constraints for a correct operation of the DDCVSL(CD) implementations. Indeed, as introduced in chapter 2, the clock delay scheme must be sized carefully such that to achieve a higher delay than the implemented gate in order to ensure correct evaluations whatever the circuit among those we implement in this work. This is different from the DDCVSL(ST) where

4.5 LOGIC STYLES SELECTED FOR COMPARISON

the completion circuit can be sized with a minimal W/L ratio since the completion signal is generated only when the evaluation is complete independently of the completion circuit sizing. Transistor sizes of DDCVSL(ST) and DDCVSL(CD) implementations are shown in appendix A.

For fair comparison with the LSCML, we retained the DDCVSL(ST) because of the self-timed operation but also because of its better results in terms of power consumption and delay when compared to the DDCVSL using clock delay scheme.

Logic gates		DDCVSL (CD)	DDCVSL (ST)
NAND/AND	Average power [μ W]	8.65	7.07
	Delay [ps]	44.2	85.6
	PDP [fJ]	0.38	0.6
	Transistor count	17	17
XOR/ $\overline{\text{XOR}}$	Average power [μ W]	8.9	7.27
	Delay [ps]	47	93
	PDP [fJ]	0.42	0.67
	Transistor count	19	19
FA 1b	Average power [μ W]	15.4	11.6
	Delay [ps]	106	123.5
	PDP [fJ]	1.63	1.43
	Transistor count	40	40

Table 4.1: Comparison of two clocking schemes for DDCVSL gates in $0.13\mu\text{m}$ PD SOI CMOS under $V_{dd} = 1.2\text{V}$, $f=500\text{MHz}$, $V_{t0} = 0.36\text{V}$ and floating body devices.

CHAPTER 4. INVESTIGATED APPLICATIONS OF THE LSCML STYLE

Logic family	Average power [μ W]	Delay [ns]	PDP [fJ]	Transistor count
DDCVSL (CD)	40.7	1.9	77.3	320
DDCVSL (ST)	23.2	1.38	32	320

(a) 8b RCA adder.

Logic family	Average power [μ W]	Delay [ns]	PDP [fJ]	Transistor count
DDCVSL (CD)	84.27	1.23	103.65	476
DDCVSL (ST)	49.89	0.53	26.6	476

(b) 8b CLA adder.

Table 4.2: Comparison of two clocking schemes for DDCVSL based adders in $0.13\mu\text{m}$ PD SOI CMOS under $V_{dd} = 1.2\text{V}$, $f=100\text{MHz}$, $V_{t0} = 0.36\text{V}$ and floating body devices.

4.5.2 DyCML versus SABL

As mentionned in section 4.3, it has been shown in [3] that one dynamic and differential style, operating in current mode, i.e. Dynamic Current Mode Logic (DyCML), allows to obtain the same security margins (according to criterions defined by the authors of SABL) while featuring better performances in terms of power consumption, delay and device count than those achieved by SABL. Table 4.3 summarizes simulation results of SABL and DyCML (with output swing of 0.4V) Khazad S-boxes [3].

Since our goal here is to achieve security without impairing speed and power consumption, we choose to compare LSCML class to relevant logic styles. Thus, DyCML was also selected for security properties comparison.

4.6 LSCML BASED LOGIC GATES

	Mean power [μ W]	Standard deviation [μ W]	NED	NSD
SABL	145.65	0.3059	0.0083	0.0021
DyCML Swing=0.4V	73.08	0.1437	0.0080	0.0020

Table 4.3: Comparison of SABL and DyCML Khazad S-boxes in $0.13\mu\text{m}$ PD SOI CMOS under $V_{dd} = 1.2\text{V}$, $f=100\text{MHz}$, adapted from [3].

4.6 LSCML based logic gates

To evaluate speed performance and power consumption of LSCML circuits, many basic logic gates have been considered. NAND/ AND, XOR/ XNOR and full-adder have been implemented with LSCML, DDCVSL(ST) and DyCML logic styles. Comparisons are also carried out on 8b RCA and 8b CLA adders. Simulations have been performed in $0.13\mu\text{m}$ PD SOI CMOS technology under a $V_{dd} = 1.2\text{V}$. Transistor sizes that we used to perform this evaluation are shown in appendix A. The calculated power consumption and the delay are respectively the average power and the worst case delay. The average power that we give here, does not include the power consumption of the single ended buffers in LSCML and DyCML gates except for the LSCML gates using the ST2 self-timing scheme (which needs the implementation of full swing conversion for completion signal generation). The same holds for the output inverters in DDCVSL gates. Simulations of the basic gates were carried out at a frequency of 500MHz . Results are shown in Table 4.4 where the LSCML(ST1) gates are those implemented with the self-timing scheme (ST1) while the LSCML(ST2) gates are those implemented with the self-timing scheme (ST2). Therefrom, it can be seen that gates implemented with DDCVSL(ST) logic show the lowest delay. The speed of DDSLL comes second, followed by LSCML(ST2), DyCML, IFLSCML(ST1) and LSCML(ST1) respectively. With regard to power consumption in AND/ NAND and XOR/ XNOR gates, DDSLL, IFLSCML(ST1) and DDCVSL(ST) show bet-

CHAPTER 4. INVESTIGATED APPLICATIONS OF THE LSCML STYLE

ter results. These are closely followed by the power consumption of LSCML(ST1) and DyCML respectively. Finally, LSCML (ST2) shows the highest power consumption. The lowest power consumption in the 1b FA gate is achieved by IFLSCML(ST1), LSCML(ST1) and DyCML respectively. They are then closely followed by gates implemented with DDSLL and DDCVSL(ST). The 1b FA implemented with LSCML(ST2) show the highest power consumption. The best power delay product (PDP) are featured by gates in DDCVSL(ST) and DDSLL respectively. Come then the PDPs of their counterparts in DyCML and LSCML(ST2). The worst PDPs are achieved by gates implemented in IFLSCML(ST1) and LSCML(ST1) due to the slowness of the completion signal generation.

4.7 8b RCA circuit

Table 4.5 shows delay time and power consumption at 100 MHz of 8b RCA circuits implemented with the considered logic styles. Power consumptions of DDCVSL(ST), DyCML, IFLSCML(ST1), LSCML(ST1) and DDSLL circuits are close to each other with a slight advantage to DDCVSL(ST) and DyCML, while the LSCML(ST2) circuit consumes the most. With regard to delay time, the DDCVSL(ST) circuit is the fastest. However, it can be seen that its delay time is only 19.8% lower than the one of DDSLL circuit. LSCML(ST1) circuit shows the worst delay, however one can see the significant reduction in delay time which can be obtained in LSCML when the self-timing scheme (ST2) is used. DDCVSL(ST) circuit shows the best compromise between power and delay by featuring the lowest power delay product. The latter is 26.4% lower than the PDP of DDSLL circuit, which shows the best power delay product among the considered low-swing logic styles. LSCML(ST1) circuit shows the worst PDP product because of its slowness. Nevertheless, this PDP is significantly improved when using (ST2) solution.

4.7 8B RCA CIRCUIT

Logic gates		DDCVSL (ST)	DyCML	LSCML (ST1)
NAND/AND	Average power [μ W]	7.07	8.2	7.5
	Delay [ps]	85.6	218	523
	PDP [fJ]	0.6	1.78	3.92
	Transistor count	17	25	25
XOR/ $\overline{\text{XOR}}$	Average power [μ W]	7.27	8.22	7.4
	Delay [ps]	93	215	540
	PDP [fJ]	0.67	1.77	3.99
	Transistor count	19	27	27
FA 1b	Average power [μ W]	11.6	10	9.84
	Delay [ps]	123.5	274	684
	PDP [fJ]	1.43	2.74	6.73
	Transistor count	40	54	56

Logic gates		LSCML (ST2)	IFLSCML (ST1)	DDSL
NAND/AND	Average power [μ W]	13.85	7.32	7.01
	Delay [ps]	150	405.4	133
	PDP [fJ]	2.07	2.97	0.93
	Transistor count	28	25	25
XOR/ $\overline{\text{XOR}}$	Average power [μ W]	15	7.09	6.99
	Delay [ps]	160	414.8	135.8
	PDP [fJ]	2.4	2.94	0.95
	Transistor count	30	27	27
FA 1b	Average power [μ W]	18.6	9.35	11.2
	Delay [ps]	200	509.9	160.9
	PDP [fJ]	3.72	4.77	1.8
	Transistor count	59	56	58

Table 4.4: Basic logic gates comparison in 0.13 μ m PD SOI CMOS under $V_{dd} = 1.2V$, $f=500\text{MHz}$, $V_{t0} = 0.36V$ and floating body devices.

CHAPTER 4. INVESTIGATED APPLICATIONS OF THE LSCML STYLE

Logic family	Average power [μ W]	Delay [ns]	PDP [fJ]	Transistor count
DDCVSL (ST)	23.2	1.38	32	320
DyCML	23.3	2.53	58.9	432
LSCML (ST1)	25.2	7.64	192.5	448
LSCML (ST2)	38.2	2.05	78.3	472
IFLSCML (ST1)	25	5.38	134.4	448
DDSSL	25.3	1.72	43.5	464

Table 4.5: 8b RCA comparison in $0.13\mu\text{m}$ PD SOI CMOS under $V_{dd} = 1.2V$, $f=100\text{MHz}$, $V_{t0} = 0.36V$ and floating body devices.

4.8 8b CLA circuit

As introduced in section 2.3.3, generation and propagation of the carry are computed as follows [8]:

$$G_i = A_i \cdot B_i \quad (4.5)$$

$$P_i = A_i \oplus B_i \quad (4.6)$$

The outgoing carry-out is computed as follows [8]:

$$C_{i+1} = G_i + P_i \cdot C_i \quad (4.7)$$

If we substitute $C_i = G_{i-1} + P_{i-1} \cdot C_{i-1}$ in the above equation, we obtain:

$$C_{i+1} = G_i + G_{i-1} \cdot P_i + C_{i-1} \cdot P_{i-1} \cdot P_i \quad (4.8)$$

By carrying out further substitutions, we obtain:

$$C_{i+1} = G_i + G_{i-1} \cdot P_i + G_{i-2} \cdot P_{i-1} \cdot P_i + C_{i-2} \cdot P_{i-2} \cdot P_{i-1} \cdot P_i = \dots \quad (4.9)$$

4.8 8B CLA CIRCUIT

This equation allows then to compute all the outgoing carries in parallel from the two operands $A_{n-1}A_{n-2}...A_0$ and $B_{n-1}B_{n-2}...B_0$ and the incoming carry-in C_{in} . Therefore, this principle avoids the need to wait for the propagation of the correct carry from the stage where it was generated.

For instance in a 4b adder with an incoming carry-in “ C_o ”, the carries are expressed as follows [8]:

$$C_1 = G_0 + C_0P_0 \quad (4.10)$$

$$C_2 = G_1 + G_0P_1 + C_0P_0P_1 \quad (4.11)$$

$$C_3 = G_2 + G_1P_2 + G_0P_1P_2 + C_0P_0P_1P_2 \quad (4.12)$$

$$C_4 = G_3 + G_2P_3 + G_1P_2P_3 + G_0P_1P_2P_3 + C_0P_0P_1P_2P_3 \quad (4.13)$$

For a large value of n, the adder may be splited into equal-sized groups in order to avoid a very large fan-in on the gates [8]. A group size of 4 is commonly used because it is a common factor of most word sizes [8]. To propagate the generated carry from one group to another, a group-generated carry denoted G^* and a group-propagated carry denoted P^* are defined by the following Boolean equations [8]:

$$G^* = G_3 + G_2P_3 + G_1P_2P_3 + G_0P_1P_2P_3 \quad (4.14)$$

$$P^* = P_0P_1P_2P_3 \quad (4.15)$$

P^* and G^* can then be used to generate group carry-in in the same way as for a single-bit carry-in as shown in equation 4.10. An illustration of the carry look ahead generator implementing these equations is shown in Figure 4.3 through an 8b CLA adder. In the latter, there is two groups with outputs G_0^*, P_0^* and G_1^*, P_1^* . The carry outputs C_4 and C_8 are given by:

$$C_4 = G_0^* + C_0P_0^* \quad (4.16)$$

$$C_8 = G_1^* + G_0^*P_1^* + C_0P_0^*P_1^* \quad (4.17)$$

CHAPTER 4. INVESTIGATED APPLICATIONS OF THE LSCML STYLE

Table 4.6 shows 8b CLA circuit comparison. This time, the DyCML

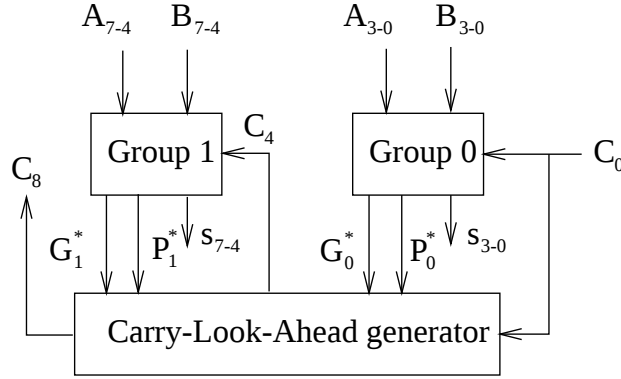


Figure 4.3: Block architecture of the implemented 8b CLA adder (adapted from [8]).

Logic family	Average power [μ W]	Delay [ns]	PDP [fJ]	Transistor count
DDCVSL (ST)	49.89	0.53	26.6	476
DyCML	31.25	2.063	64.47	612
LSCML (ST1)	34.06	3.2	109	672
LSCML (ST2)	64.94	1.66	108	749
IFLSCML (ST1)	34.28	1.35	46.3	672
DDSSL	41.57	0.52	21.6	744

Table 4.6: 8b CLA comparison in $0.13\mu\text{m}$ PD SOI CMOS under $V_{dd} = 1.2V$, $f=100\text{MHz}$, $V_{t0} = 0.36V$ and floating body devices.

circuit shows the lowest power consumption. The power consumption of LSCML(ST1) and IFLSCML(ST1) circuits is slightly higher than the DyCML one. These are followed by DDSSL, DDCVSL(ST) and LSCML(ST2) circuits respectively. Regarding delay time, DDSSL and

DDCVSL(ST) show the lowest values. The DDSLL circuit shows the best compromise between power and delay. The worst PDP is shown by the LSCML(ST1) and LSCML(ST2).

For all these implemented circuits, DDCVSL is surely the most advantageous in terms of the transistor count since it features lower complexity than low swing current mode logic families.

4.9 *Khazad S-box comparisons*

To evaluate LSCML in terms of reduction of data-dependent power signature, the Khazad S-box has been implemented with the different versions of LSCML class, DDCVSL(ST) and DyCML. Simulations were carried out at the schematic transistor level under comparable conditions for the three logic styles. In these, the outputs of the S-box were loaded by the single ended buffers in LSCML and DyCML and by the output inverters in DDCVSL. Simulations were performed at 100MHz, in 0.13 μ m PD SOI CMOS with floating body devices, under a supply voltage of 1.2V. Once the power consumption behaviour relative to each possible input of the S-box was determined for each logic style, we extract their statistical properties, i.e. for each logic style, we extracted the mean power consumption (μ), the power consumption standard deviation (σ), the NED and the NSD parameters.

Table 4.7 shows simulation results for the Khazad S-box. It can be seen that the LSCML class except the LSCML(ST2) shows the most reduced power consumption standard deviations, NED and NSD. The LSCML(ST2) and DDCVSL(ST) show the highest values. The IFLSCML and LSCML(ST1) achieve the lowest power consumption, while the DDCVSL(ST), LSCML(ST2) and DDSLL consume the most. The best power delay product is achieved by DDCVSL(ST) thanks to its lower delay. The S-box implemented with DDCVSL(ST) has also the advantage of using the lowest transistor count. Nonetheless, one can see that the power delay product of the DDSLL S-box is only slightly higher than

CHAPTER 4. INVESTIGATED APPLICATIONS OF THE LSCML STYLE

the DDCVSL(ST) one, while reducing its power consumption standard deviation by almost a factor of four.

Logic family	Min power [μ W]	Max power [μ W]	Mean power [μ W]	Std. dev. power [μ W]	NED	NSD	PDP [fJ]	Transistor count
DDCVSL (ST)	33.240	35.163	34.171	0.4040	0.0547	0.0118	21.68	566
DyCML	27.760	28.657	27.997	0.1566	0.0313	0.0056	59.18	694
LSCML (ST1)	22.675	23.028	22.829	0.0676	0.0153	0.0030	72.40	694
LSCML (ST2)	32.608	34.447	33.462	0.4307	0.0534	0.0129	55.23	719
IFLSCML (ST1)	21.383	21.774	21.553	0.0821	0.0179	0.0038	45.89	694
DDSL	33.160	33.646	33.396	0.0966	0.0144	0.0029	26.16	784

Table 4.7: S-Box comparison in 0.13 μ m PD SOI CMOS under $V_{dd} = 1.2V$, $f=100$ MHz, $V_{t0} = 0.36V$.

4.9.1 *Impact of the output swing variation*

As mentioned before, it has been shown that dynamic and differential logic styles are not equal in terms of resistance to power analysis attacks. The charged/discharged amount of capacitance makes the difference. Tiri et al. have demonstrated in [2] that this amount strongly depends on the switched transistors (i.e. on the processed data) in DCVSL logic which results in power consumption variation, and balancing the total charged/discharged capacitance whatever the input data leads to reduced data-dependent power signature. This was indeed achieved in SABL logic [2].

Full-swing logic (like DCVSL) operates in voltage mode and thus depends on the value of input voltages (0 or V_{dd}) to switch fully ON or OFF the transistors. This will create a current path between the output and one of the power supply lines, and then charge/discharge the total parasitic capacitance.

In low swing logic, the charge needed for charge/discharge of C_L is smaller than in full swing logic. This leads to lower current swing for charge/discharge of C_L in a certain time than in the full-swing configuration. The amount of the power consumption variation in low swing current mode logic appears to be smaller than in full swing logic where the variation of amount of charged/discharged capacitance has a meaningful impact on power behaviour. In low swing logic, the amount of charge transferred from outputs is limited as soon as the condition of voltage drop limitation is fulfilled. This condition depends on which low swing current mode logic is used. The output voltage swing is proportional to the drained current. This latter is independent of the data input values. However, it will be seen hereafter that there is a variation in output voltage swing depending on the kind of implemented logic function and the load capacitance. The amount of this variation depends on which low-swing current mode logic is used. This is illustrated with three logic functions: inverter, AND/NAND and 1b carry gates, implemented with DyCML and LSCML class. The output voltage swing values shown in Tables 4.8(a), (b) and (c) are given for a fixed

data input set. We can see that variation of the output voltage swing in LSCML class versus the load capacitance is lower than that in DyCML. As described before, the S-box is based on mini-boxes P and Q. Each mini box is based on four 4b to 1b logic functions. The implementation at the transistor level of these four functions is not the same (as can be seen on P-functions shown in Figure 4.2). Besides the variation of the amount of parasitic capacitance according to the switched transistors, there is a variation in fan-in of the cascaded gates which contributes to variation of total parasitic capacitance. This variation introduces a variation in the output swing and thus in the power consumption.

As we mentioned before, the output swing variation is more or less important depending on the logic style. The lower the output swing variation, the more reduced the power consumption variation, according to equation 1.1. We believe that this explains the advantage of LSCML class in reduction of the power consumption variation.

4.9.2 *ST2 versus ST1*

Within the S-box, LSCML(ST1), IFLSCML(ST1), DDSLL and DyCML do not need the full swing conversion of the output signal (i.e. the implementation of the single ended buffer of Figure 3.5). Thus, the power consumption of the single ended buffers is not included in the total power consumption as opposed to the total power consumption of LSCML circuits using ST2 scheme. Both ST1 and ST2 circuits shown in Figures 3.3(a) and 3.4(a) respectively, have one transition $0 \rightarrow 1$ at each cycle. Thus, in this sense both ST1 and ST2 circuits are not leaky in terms of power variation due to the amount of switching activity. Nonetheless, the difference between them lies in the fact that the single ended buffer, which power consumption is included in total power of the circuit using the ST2 scheme, leaks information through the single ended buffer power component that originates from the short-circuit current. Indeed, because in dynamic differential logic, there is always a transition $0 \rightarrow 1$ at each cycle, there is consequently a transition $0 \rightarrow 1$

CHAPTER 4. INVESTIGATED APPLICATIONS OF THE LSCML STYLE

	DyCML	LSCML	IFLSCML	DDSL
1fF	0.485V	0.546V	0.463V	0.812V
3fF	0.375V	0.564V	0.477V	0.822V
5fF	0.300V	0.560V	0.480V	0.822V

(a) Output swing values in an inverter gate versus the load capacitance.

	DyCML	LSCML	IFLSCML	DDSL
1fF	0.481V	0.535V	0.456V	0.783V
3fF	0.366V	0.541V	0.458V	0.788V
5fF	0.291V	0.540V	0.460V	0.779V

(b) Output swing values in an AND/NAND gate versus the load capacitance.

	DyCML	LSCML	IFLSCML	DDSL
1fF	0.461V	0.525V	0.459V	0.787V
3fF	0.352V	0.512V	0.461V	0.784V
5fF	0.287V	0.511V	0.452V	0.764V

(c) Output swing values in a 1b carry gate versus the load capacitance.

Table 4.8: Output swing variation in DyCML and LSCML classes.

in one of the two level converter circuits (for OUT and $\overline{OUT}$). However, since the short-circuit current in the single ended buffer depends on the input “IN” entering the gate of Q2 of the circuit shown in Figure 3.5(a), which corresponds to the output voltage, a variation in the output swing produces a variation in the driving capability of the capacitance at node “q”. This may affect the slope of the signal at node “q” and consequently produces a variation in the short circuit current through the output inverter. Thereby producing a variation in the dynamic power. This probably explains the power consumption behaviour in LSCML S-box using the ST2 scheme.

4.9.3 *Impact of the fanout mismatch*

In this section, we investigate the effect on LSCML class due to a possible capacitive mismatch between differential outputs, and in this sense we performed simulations of the Khazad S-Box that take into account a possible capacitive mismatch (which can be introduced by routing interconnections) between the differential outputs. To do so, a capacitor of 5fF was connected to one of the differential outputs of each gate. We then extracted the same parameters as for the “perfect” implementations in the first set of simulations. Simulation results for the LSCML(ST1), IFLSCML(ST1) and DDSLL Khazad S-boxes are shown in Table 4.9 and where we compare LSCML class with the DDCVSL(ST) including the same capacitive mismatch. Therefrom, we can see that though it increases in comparison to that in the “perfect” implementation of the Khazad S-box, the power consumption standard deviation in IFLSCML(ST1) for instance is almost four times smaller than the one in the DDCVSL(ST) based S-box.

4.9.4 *Impact of the output swing value*

We also investigated the effect of the output swing reduction on the power consumption variation. For that sake, DyCML was used as benchmark as its structure allows varying the output logic swing without af-

CHAPTER 4. INVESTIGATED APPLICATIONS OF THE LSCML STYLE

Logic family	Min power [μ W]	Max power [μ W]	Average power [μ W]	Std. dev. power [μ W]
DDCVSL (ST)	36.614	47.947	42.809	2.8812
LSCML (ST1)	24.604	30.116	27.368	1.1899
IFLSCML (ST1)	22.976	26.506	24.672	0.7293
DDSL	35.385	41.221	38.252	1.2393

Table 4.9: S-Box with capacitive mismatch comparison in $0.13\mu\text{m}$ PD SOI CMOS under $V_{dd} = 1.2\text{V}$, $f=100\text{MHz}$, $V_{t0} = 0.36\text{V}$.

fecting its functionality. The power consumption standard deviation as well as other electrical characteristics of implementations of DyCML with different values of output logic swing (0.4V, 0.48V, 0.6V, 0.7V and 0.8V) were extracted. Simulation results are shown in Table 4.10. According to equation 1.1 that gives average dynamic power, lowering the output voltage swing reduces the dynamic power. This behaviour can globally be observed on simulation results given in Table 4.10.

Equation 4.4 gives the standard deviation through the square root of the variance of the power consumption, where P_i corresponds to the individual power consumption extracted by simulation for an input set of the S-box. According to the formula presented in equation 1.1, we can see that a logic swing reduction yields a reduction of P_i values. Then, we can also deduce that reducing the individual power consumption of each input set will have a global effect, a reduction of the variance of the power consumption of the S-box. This was verified by our simulations. As a consequence, the global tendency of the power consumption standard deviation is to decrease with the output voltage swing reduction, as shown in Table 4.10.

4.10 CONCLUSION

Logic family	Min power [μ W]	Max power [μ W]	Mean power [μ W]	Std. dev. power [μ W]	PDP [fJ]
DyCML (Swing=0.4V)	26.988	28.075	27.306	0.1446	47.689
DyCML (Swing=0.48V)	27.760	28.657	27.997	0.1566	59.18
DyCML (Swing=0.6V)	29.588	31.225	29.919	0.1428	51.094
DyCML (Swing=0.7V)	33.002	34.427	33.717	0.2765	81.084
DyCML (Swing=0.8V)	25.243	42.122	39.582	1.3548	162.13

Table 4.10: DyCML S-Box comparison in $0.13\mu\text{m}$ PD SOI CMOS under $V_{dd} = 1.2\text{V}$, $f=100\text{MHz}$, $V_{t0} = 0.36\text{V}$.

4.10 Conclusion

In this chapter, we have investigated the use of LSCML class in different applications through a qualitative comparison with two relevant logic styles. Depending on the application and the kind of the circuit to be implemented, a significant difference can be observed between all aspects of performance that characterize the different logic styles. Simulation results of the investigated circuits, have shown that the 8b RCA is surely faster and cheaper when implemented with DDCVSL(ST) due to its lower power consumption and implementation cost (i.e. transistor count). For the 8b CLA circuit, even though the DDSLL circuit shows the best PDP, the DDCVSL(ST) presents the best compromise between PDP and implementation cost.

Among the investigated dynamic differential self-timed logic styles, the LSCML class has shown the most reduced power consumption variation, but this advantage in terms of security concept comes with increased PDP and area. The different versions of LSCML class (except

CHAPTER 4. INVESTIGATED APPLICATIONS OF THE LSCML STYLE

the LSCML(ST2)) have shown almost similar performances in terms of σ , NED and NSD criteria and thus feature the same security margins against power analysis attacks for general security applications. Consequently, the selection of a version from the LSCML class should then be performed with regard to other electrical characteristics like power consumption, delay, power delay product, or implementation cost. And finally, even their slowness, LSCML(ST1) and IFLSCML(ST1) are probably the most appropriate among the investigated logic styles, for implementation of large low-power self-timed circuits.

References

- [1] E. Brier and C. Clavier, *Optimal statistical power analysis*, IACR e-print archive 2003/152, <http://eprint.iacr.org>, 2003.
- [2] 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 ESSCIRC Conference*, 2002.
- [3] F. Mace, F.-X. Standaert, I. Hassoune, J.-D. Legat, and J.-J. Quisquater, “A dynamic current mode logic to counteract power analysis attacks,” in *Proceedings of DCIS conference*, Bordeaux, France, 2004, pp. 186–191.
- [4] 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,” *Accepted for publication in the Microelectronics Journal*, 2006.
- [5] P. Barreto and V. Rijmen, *The Khazad Legacy-Level Block Cipher*, <http://paginas.terra.com.br/informatica/paulobarreto/Khazad-Page.html>.
- [6] F. Mace, F.-X. Standaert, J.-J. Quisquater, and J.-D. Legat, “A design methodology for secured ics using dynamic current mode logic,” in *15th International Workshop-PATMOS*, Springer-Verlag LNCS 3728, 2005, pp. 550–560.
- [7] Pius Ng, P.T. Balsara, and D. Steiss, “Performance of CMOS differential circuits,” *IEEE Journal of Solid-State Circuits*, vol. 31, no. 6, pp. 841–846, June 1996.
- [8] I. Koren, *Computer Arithmetic Algorithms*, Englewood Cliffs, NJ: Prentice-Hall, 1993.

CHAPTER 4. INVESTIGATED APPLICATIONS OF THE LSCML STYLE

CHAPTER 5

ULPFA: A POWER AWARE HYBRID FULL ADDER

5.1 Introduction

In this chapter, we propose a new structure of a hybrid full adder [1] that we implemented by combining branch-based logic and pass-transistor logic. Moreover, MTCMOS (multi-threshold) circuit technique was applied on the proposed full adder to achieve a trade-off between low power and high performance design. Design with DTMOS (dynamic threshold) devices was also investigated with two threshold voltage values (0.28V and 0.4V) and $V_{dd} = 0.6V$. Evolution of the proposed cell from its original version up to an ultra-low-power cell is also described. A comparison between the proposed full adder, its optimized versions and its counterparts in conventional static CMOS logic and complementary pass logic (CPL), was carried out in a $0.13\mu m$ PD SOI CMOS for a supply voltage $V_{dd} = 1.2V$.

1-bit binary full adder is a binary combinatorial logic circuit that accepts two operand bits (A_i and B_i) and a carry-in bit (Cin). The full adder implements a binary addition through the following Boolean equations:

$$Sout_i = A_i \oplus B_i \oplus Cin \quad (5.1)$$

$$Cout_i = A_i \cdot B_i + Cin \cdot (A_i \oplus B_i) \quad (5.2)$$

In section 2.3.4, we surveyed different variants of previously reported 1-bit full adders. In this chapter, we investigate a new hybrid structure of full adder. Selection of logic style for each computation block (sum and carry) was performed according to our objectives to achieve low power, low implementation cost (i.e. area) and an acceptable delay performance. Among the reviewed logic styles in section 2.2.3, we believe that branch-based logic can achieve a good trade-off between low-power, delay and die area [2] [3]. Thus it is well suited to implement circuits

for portable applications or wireless sensor networks.

5.2 *Branch-Based design*

We introduced in section 2.2.3 the major conditions to be satisfied for low-power design purpose. A cell schematic must be implemented with as few transistors and intra-cells node connections as possible. The branch-based design meets these requirements while ensuring robustness to voltage and device scaling. Moreover, at the layout level, branch-based design diminishes the diffusion capacitance since it eases diffusion sharing. It also leads to very regular and compact layout topologies.

Figure 5.1 illustrates for the same logic function, the equivalent schematics with and without branches as well as their layout implementations.

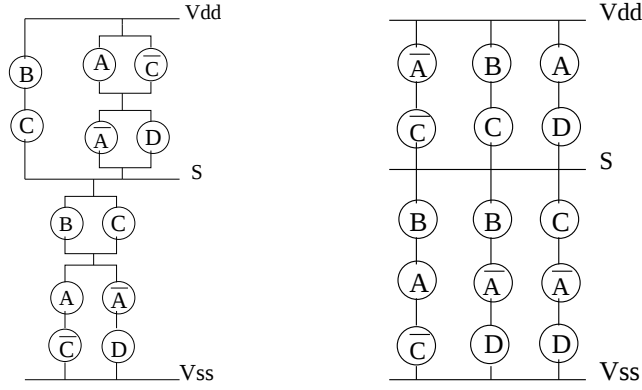
If we consider the PMOS network, which contains the same number of transistors in both schematics, the layout that implements PMOS network with branches contains only four routing connections in comparison with the six routing connections in its counterpart obtained from the schematic without branches.

5.3 *The BBL-PT hybrid full adder*

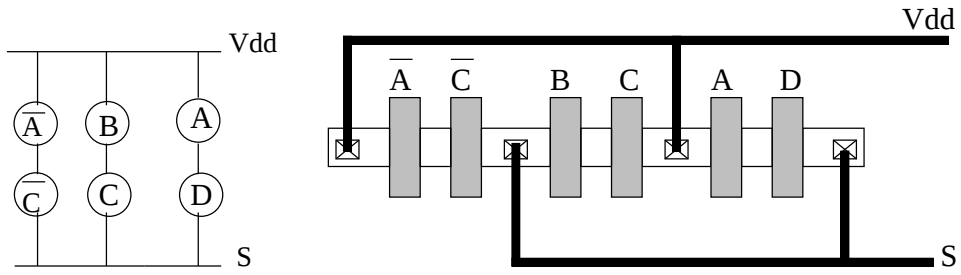
In branch-based design, some constraints are applied on the NMOS and PMOS networks. Indeed, they are only composed of branches, this is a series connection of transistors between the output node and the supply rail. NMOS and PMOS networks are sums of products and are obtained from Karnaugh maps [5]. Simple gates are constructed of branches and more complex gates are composed of simple cells. We use the simplification method given in [5] to implement the carry-block of a full adder with a branch structure. The obtained equations of NMOS and PMOS networks are expressed as follows:

$$C_N = \overline{A}.\overline{Cin} + \overline{B}.\overline{Cin} + \overline{A}.\overline{B} \quad (5.3)$$

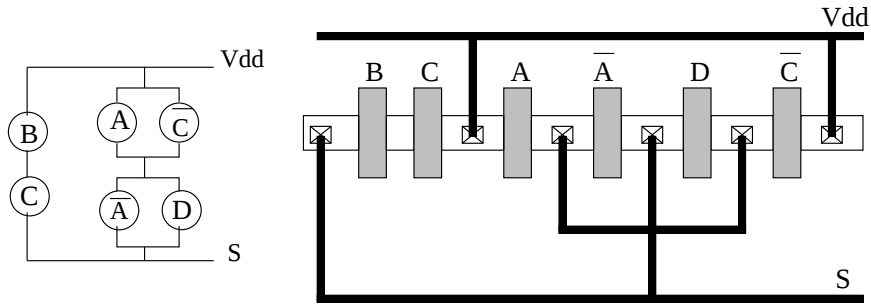
5.3 THE BBL-PT HYBRID FULL ADDER



(a) Schematics implementing logic function $S = \overline{(A \cdot \overline{C} + \overline{A} \cdot D)} \cdot (B + C)$.



(b)



(c)

Figure 5.1: Equivalent schematics with and without branches of logic function 'S'(a). Layout implementation of the PMOS network schematic with branches (b). Layout implementation of the PMOS network schematic without branches (c). (Adapted from [4]).

$$C_P = \overline{A}.\overline{Cin} + \overline{B}.\overline{Cin} + \overline{A}.\overline{B} \quad (5.4)$$

However, to implement the sum-block with only branches is not advantageous. Indeed, with this method, the sum-block needs 24 transistors for 1-bit sum generation and stacks of three devices. Therefore, an implementation with pass-transistors was used for the sum-block (Figure 5.2) particularly as they ease a simple implementation of XOR function with a low transistor number.

The disadvantage of this implementation lies in the resulting weak high output level in pass-transistors used in the sum-block of the proposed full adder. We used the feedback realized by the pull-up PMOS transistor in order to restore the weak logic “1” caused by the pass transistors, and provide sufficient drive to the eventual successive stages. However, the level restoration by the way of this level keeper causes a voltage step at the output node “Sout” during a transition $0 \rightarrow 1$ as it is shown in Figure 5.3. This voltage step is due to threshold voltage drop in pass transistors and the delay needed by the feedback keeper to restore the weak logic level. When a logic “1” is passed through the NMOS network, the node “Sout” tends to be charged to a weak logic “1” (i.e. $V_{dd} - V_{tn}$). When voltage at node “Sout” is $< \frac{V_{dd}}{2}$, the pull-up PMOS is turned OFF and the node “Sout” is charged with an effective drive current that equals the current of the NMOS network. When the voltage at node “Sout” approaches $\frac{V_{dd}}{2}$, the inverter reaches the commutation threshold, the pull-up PMOS turns ON and the effective drive current charging the capacitance at node “Sout” becomes the sum of the current flowing through the NMOS network and the pull-up PMOS current. This increased drive current boosts up the charge of the parasitic capacitance. Since it happens after the response delay of the feedback level restorer, the increase of the effective drive current causes a voltage step on the sum signal.

By choosing this implementation, we break some rules specified in the branch-based logic as BBL design does not implement gates with pass-transistors. Nevertheless, Branch-Based logic in combination with pass-

5.3 THE BBL-PT HYBRID FULL ADDER

gate logic, allows a simple implementation of full adder gate, namely the BBL-PT (Branch-Based Logic and Pass-Transistor) full adder, with only 23 transistors versus the 28-transistors static CMOS full adder or the 32 transistors in CPL. Figure 5.3 shows the output waveforms of the BBL-PT full adder.

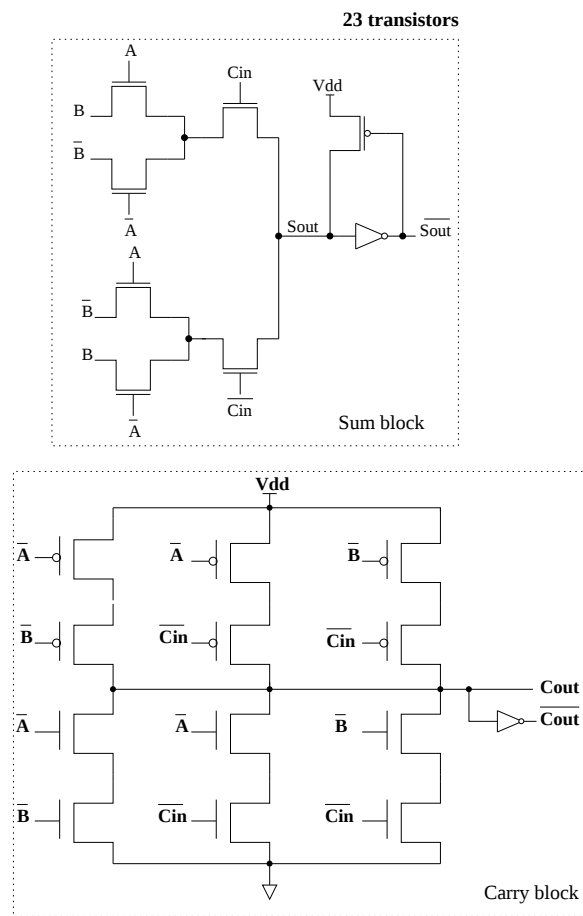


Figure 5.2: Hybrid Full adder (BBL-PT).

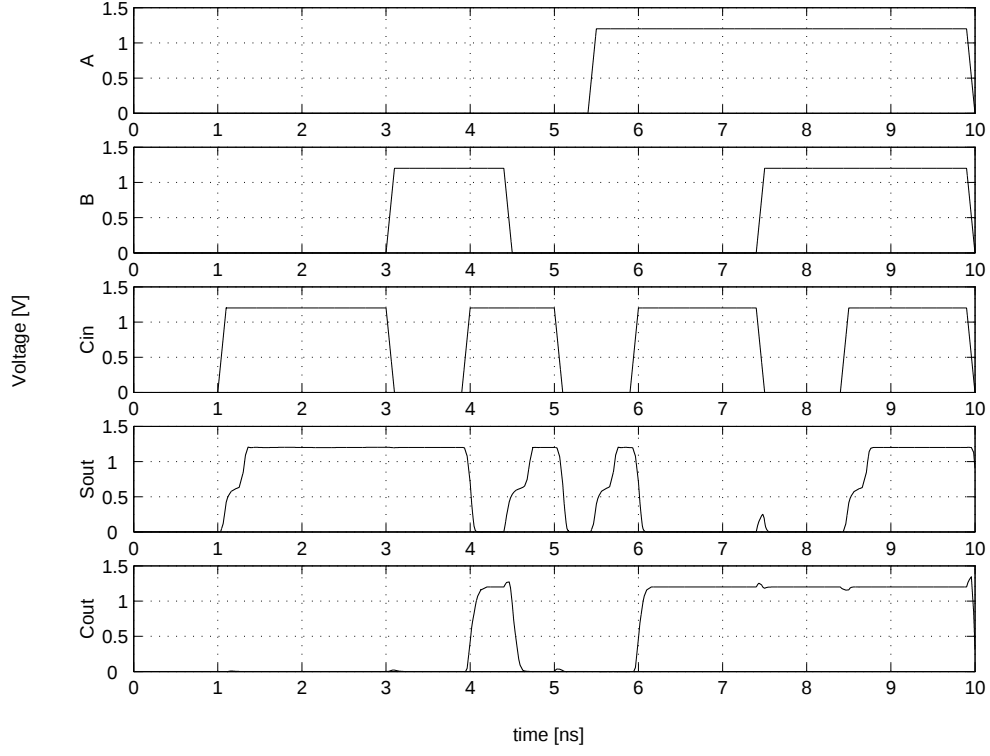


Figure 5.3: BBL-PT full adder output waveforms in $0.13\mu\text{m}$ PD SOI/CMOS, ST LL MOSFETs ($V_t = 0.36\text{V}$), with the transistor sizes shown in appendix B, $V_{dd} = 1.2\text{V}$. One can observe the voltage step that occurs at the output node “Sout” during a transition $0 \rightarrow 1$.

5.4 The BBL-PT FA vs the static CMOS FA

Simulations were carried out at schematic level for the proposed BBL-PT full adder shown in Figure 5.2, and its counterpart in conventional CMOS (denoted CMOS FA hereafter) shown in Figure 2.16(a), in $0.13\mu\text{m}$ PD SOI CMOS and with a supply voltage $V_{dd} = 1.2\text{V}$, floating body devices and $V_t = 0.28\text{V}$. The same $\frac{W}{L}$ ratio was considered for the two full adder gates, except for the pull-up transistor which must have a high ON resistance in order to restore the high logic level without affecting the low logic level at the output node. Thus, a lower $\frac{W}{L}$ ratio was used for the pull-up transistor. Each full-adder is loaded by an inverter. This inverter has a separate supply voltage in order to be able to extract the power consumption of the full adder cell. Simulations were performed at clock frequency of 100MHz. For all possible input sets applicable to the full adder, we extracted the average power and the worst case delay. Simulation results, given in Table 5.1, show that the delay of the BBL-PT full adder is lower than in the CMOS FA. The BBL-PT sum-block needs the true carry signal and its complement. Therefore, it should be useful to evaluate also the $\overline{C_{out}}$ delay.

With regard to the total power consumption, the BBL-PT full adder consumes only 9% more than its counterpart in conventional CMOS, while static power is less in the BBL-PT full adder gate, thanks to the structure in branch and the lower number of transistors.

With a power delay product (PDP) of 0.47fJ, the BBL-PT full adder performs better than the CMOS FA cell, which features a PDP of 0.69fJ. Nevertheless, because of the presence of pass-transistors, the BBL-PT full adder must be used with high $\frac{V_{dd}}{V_t}$ ratio (> 3) because of delay performance impairing in pass-transistor logic for reduced $\frac{V_{dd}}{V_t}$ ratio.

5.5 MTCMOS and DTMOS circuit techniques

As supply voltage reduction remains the most effective approach to reduce the power dissipation, a trade-off is needed to achieve this goal

CHAPTER 5. ULPFA: A POWER AWARE HYBRID FULL ADDER

	Delay (ps)			Total power (μ W)	Static power (nW)
	Cout	Sout	$\overline{Cout}$		
CMOS FA	118	157		4.40	170
BBL-PT FA	51	34	98	4.80	150
BBL-PT FA with MTC- MOS	59	34	130	4.23	0.0185

Table 5.1: Simulations results with $V_{dd} = 1.2\text{V}$, $f = 100\text{MHz}$, fan-out=1, Leti HS (High speed) MOSFETs ($V_t = 0.28\text{V}$) and Leti LL (Low leakage) MOSFETs ($V_t = 0.4\text{V}$) for sleep transistors, floating body configuration.

without a significant speed penalty. Indeed, maintaining a high performance when V_{dd} is reduced, requires an aggressive V_t scaling. However, this increases leakage currents. Circuit techniques like MTCMOS [6] [7] have been proposed to allow a leakage control when lowering supply voltage and when V_t is reduced.

In this circuit technique, high-threshold switch transistors are used in series with the low threshold voltage logic block. In standby mode, high- V_t transistors are turned off, thus suppressing leakage, while in the active mode, the high- V_t transistors are turned on and act as virtual V_{dd} and ground.

The DTMOS technique was proposed [8] for ultra-low supply voltage operation ($\leq 0.6\text{V}$) to improve circuit performance. Because of the body effect, V_t can be changed dynamically by using the DTMOS configuration where the transistor gate is tied to the body. By choosing this device configuration, V_t is lower during the on-state of the DTMOS device, thereby increasing the transistor drive current; while V_t is higher during its off-state, therefore suppressing leakage current.

The potential of these design techniques has been assessed at the cell level through simulation results of the BBL-PT full adder.

5.6 Application to the BBL-PT FA: MTCMOS technique

The MTCMOS circuit technique was applied to the BBL-PT full adder as shown in Figure 5.4(a). A high threshold voltage ($|V_t| = 0.4V$) was used for the switch transistors (T_1 and T_2), while a low threshold voltage ($|V_t| = 0.28V$) was used for the full adder logic block. Due to the unavoidable increase in delay when using MTCMOS technique, and which results from the effective supply voltage reduction, the sleep transistors should generally be sized larger than normal in order to avoid an excessive impact on delay. However, because low-power is our first target in this thesis, the PMOS and NMOS sleep transistors have been sized with the same $\frac{W}{L}$ ratio as for PMOS and NMOS transistors respectively in the logic block.

Simulations were carried out with $V_{dd} = 1.2V$ with floating body devices. The same $\frac{W}{L}$ ratio, the same fanout and input clock frequency as those used for the BBL-PT full adder using one threshold voltage value $V_t = 0.28V$, were considered.

Results are given in the third row of Table 5.1. With regard to the delay performance, there is an increase of around 16% on “Cout” and about 33% on $\overline{\text{Cout}}$ in the BBL-PT with the MTCMOS technique in comparison with its counterpart using only low- V_t ($V_t = 0.28V$) devices. The delay on node “Sout” is still unchanged due to the fact that sleep transistors are in series with the inverter and the pull-up PMOS transistor used to restore the high logic level, in the sum block (illustration is shown in Figure 5.4(b)). On the other hand, since the sleep transistors have a $|V_t| = 0.4V$, with a $\frac{V_{dd}}{V_t} = 3$, their drive capability is just enough and the critical path on the sum block is not affected when the MTCMOS technique is used. With regard to the total power consumption, the full adder gate consumes about 12% less when the MTCMOS circuit technique is used thanks to leakage reduction but also reduced short-circuit current through the inverter. And finally, the static power dissipation is reduced to a negligible value thanks to leakage current suppression in

the standby mode.

5.7 The BBL-PT full adder with DT MOS devices

As the use of DT MOS device (shown in Figure 5.5) is limited to voltage supply up to 0.6V [8] in order to prevent large forward bias diode currents, which increases the static power, simulations were carried out on the BBL-PT full adder with $V_t = 0.28\text{V}$ and $V_{dd} = 0.6\text{V}$ in the same conditions than those given in section 5.4. The full adder gate with DT MOS devices was compared with its counterparts using transistors in floating body configuration and with a third version using the MTCMOS technique (shown in Figure 5.4). A floating body devices configuration was considered in the BBL-PT full adder with the MTCMOS technique. Figure 5.6 compares the id-vg characteristics of Floating body and DT MOS devices in $0.13\mu\text{m}$ PD SOI/CMOS.

From simulations results given in Table 5.2, it appears that there is no gain in delay performance or power consumption when the DT MOS device is used with $V_t = 0.28\text{V}$. Results show that both are impaired in DT MOS configuration making the BBL-PT full adder using the floating body devices more advantageous. The increase of the switching load capacitance due to the overhead of junction capacitance, degrades the performance when DT MOS devices and a low $V_t = 0.28\text{V}$ are used. Thus, the expected gain in performance is not obtained in this case.

However, simulations results given in the next section, show that a gain in performance can be obtained when the DT MOS device is used in combination with a higher V_t .

With regard to the BBL-PT with the MTCMOS technique and floating body devices, it appears that a degradation up to 58% is observed on delay, while a gain of 5% on total power consumption is obtained in comparison with its counterpart using floating body devices and one $V_t = 0.28\text{V}$. As mentioned in section 5.6, the static power dissipation

5.7 THE BBL-PT FULL ADDER WITH DTMOS DEVICES

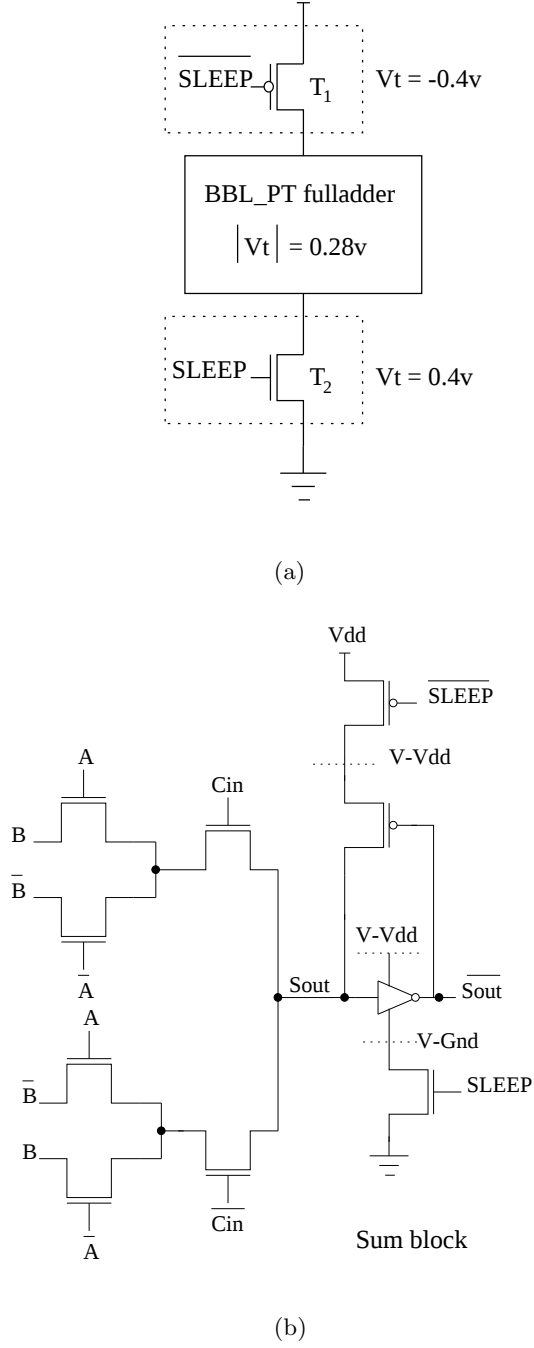


Figure 5.4: The MTCMOS circuit technique (a). The MTCMOS circuit technique applied to the sum-block of the BBL-PT full adder (b).

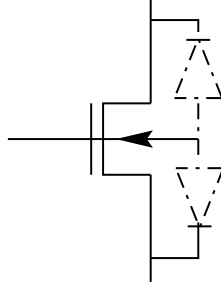


Figure 5.5: DTMOS device.

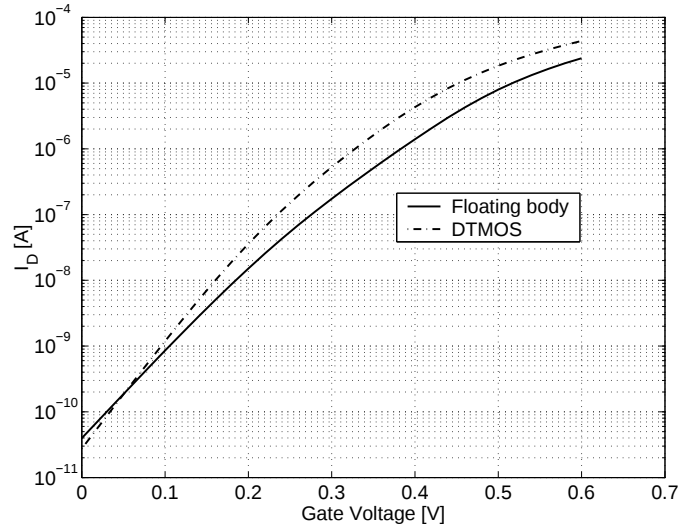


Figure 5.6: Simulated I_d - V_g characteristics of floating body and DTMOS devices in $0.13\mu\text{m}$ PD SOI/CMOS, N-MOSFETs, $W = 0.5\mu\text{m}$, $L = 0.13\mu\text{m}$.

5.7 THE BBL-PT FULL ADDER WITH DTMOS DEVICES

becomes negligible.

It should be reminded that in section 5.6, the delay on node “Sout” was the same in the BBL-PT full adder using MTCMOS technique and its counterpart using one $V_t = 0.28V$, with $V_{dd} = 1.2V$. This time, because of the high output level degradation in the sum block due to the lower $\frac{V_{dd}}{V_t}$ ratio, and moreover, the sleep-transistors which work in this configuration with an unsufficient $\frac{V_{dd}}{V_t}$ ratio (i.e. < 3), the high logic level and the delay on the critical path suffer much from this $\frac{V_{dd}}{V_t}$ scaling. As a consequence, the delay on node “Sout” is as well degraded when the MTCMOS technique is used.

	Delay (ps)			Total power (μW)	Static power (nW)
	Cout	Sout	C_{out}		
BBL-PT FA with DT- MOS dev., $V_t = 0.28V$	170	215	300	1.3	2
BBL-PT FA with floating body dev., $V_t = 0.28V$	120	160	273	1.03	1.56
BBL-PT FA with MTC- MOS, float- ing body dev.	163	185	430	0.95	0.008

Table 5.2: Simulation results with $V_{dd} = 0.6V$, $f=100MHz$ and fan-out=1.

5.8 The BBL-PT FA with DTMOS devices and high V_t

This section discusses a comparison between the BBL-PT full adder using DTMOS devices and its counterpart using floating body devices, when a high V_t ($V_t = 0.4V$) and a supply voltage $V_{dd} = 0.6$ are used. Simulations were carried out in the same conditions as described in section 5.4.

From simulations results given in Table 5.3, it appears that a performance gain up to 36% is obtained when using the DTMOS technique. The most significant gain is observed on the critical path that appears this time in the sum block. DTMOS device might be then a solution for the performance degradation of pass-transistors when an aggressive $\frac{V_{dd}}{V_t}$ scaling is needed.

With regard to the total power dissipation, as it can be observed in Table 5.3, the BBL-PT full adder consumes more when the DTMOS device is used because of the higher junction capacitances and the higher short-circuit current. The static power dissipation is as well much higher in this configuration. This is due to the forward biased diodes associated with DTMOS device. The static power dissipation in the circuit using floating body configuration is mainly due to small leakage current of the devices. The diode leakage in floating body device is negligible because the parasitic diodes are strongly reversely biased.

In this study, $\frac{V_{dd}}{V_t} = 3$ was considered as a threshold that we used to measure the driving capability of devices. In circuits based on long-channel devices, $\frac{V_{dd}}{V_t} = 3$ is the optimal ratio to obtain a minimal power delay product (indeed, the minimal PDP can be obtained from $PDP \propto \frac{V_{dd}^3}{(V_{dd}-V_t)^2}$ under a criterion of $\frac{dPDP}{dV_{dd}} = 0$, therefrom, the minimum is obtained for a ratio of $\frac{V_{dd}}{V_t} = 3$). However, due to the continuous device shrinking, the gate delay in CMOS is expressed by [9]:

$$t_d \propto \frac{C_L \cdot V_{dd}}{W \cdot (V_{dd} - V_{th})^\alpha} \quad (5.5)$$

5.9 ADVANCES IN THE HYBRID FULL ADDER

	Delay (ps)			Total power (μ W)	Static power (nW)
	Cout	Sout	$\overline{Cout}$		
BBL-PT FA with DT- MOS dev.	280	1740	496	0.63	0.6
BBL-PT FA with floating body dev.	327	2700	670	0.43	0.03

Table 5.3: Simulations results with $V_{dd} = 0.6V$, $f=100MHz$, fan-out=1, and Leti LL (Low leakage) MOSFETs ($V_t = 0.4V$).

where $1 < \alpha < 2$ ($\alpha = 2$ for long channel devices). The factor α tends to decrease with the technology scaling. The power supply is decreasing much more rapidly than threshold voltage. If this trend benefits to total power consumption, this however results in a progressive loss in $(V_{dd} - V_{th})$ and subsequently in device performance. Indeed, the gain in performance of deep submicron circuits and systems is achieved thanks to device scaling.

5.9 Advances in the Hybrid full adder

In order to prevent the voltage step that appears in transition $0 \rightarrow 1$ on output sum signal and reduce dynamic power which is impaired by the large drain and gate capacitance on “Sout” node, further optimization has been carried out on the sum-block of the hybrid full adder. We implement hereafter the sum-block with a low-power (LP) XOR, XNOR gates [10] and the ULP diode [11].

5.9.1 Conventional level restorer

When the pull-down network pulls the node “Sout” to ground, the pull-up PMOS transistor in the feedback level restorer (shown in Figure 5.7)

causes contention, i.e. the effective pull-down current is that of the pull-down network minus the current from the pull up transistor. Delay is therefore increased in this case. This problem can be alleviated by weakening the pull up transistor, which must be sized such as voltage at node “Sout” is $< 0.1V_{dd}$ when a “0” logic is transmitted by the pull-down network, but this often entails decreasing $\frac{W}{L}$ ratio. Moreover, a strong restorer (i.e. increasing $\frac{W}{L}$) is required to ensure sufficient noise margins. This creates a tricky dilemma between contention and noise margins.

Furthermore, the large capacitance at node “Sout” due to drain and gate capacitance of the level restorer increases dynamic power and delay in the sum block.

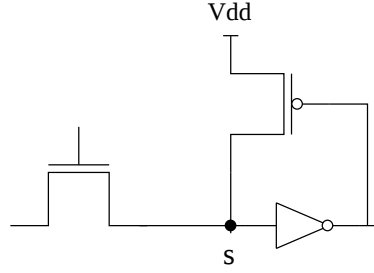


Figure 5.7: Conventional level restoration (for voltage at node “s”) in pass-transistor logic.

5.9.2 ULP diode based level restorer

In [11], a novel ULP diode architecture was proposed with strongly reduced leakage current when compared to a standard diode-connected MOSFET while maintaining similar forward current drive capability. Recently, basic circuits targeting ultra-low power (ULP) applications that exploit the combination of various threshold voltages (MTCMOS process) and the ULP diode concept, have been developed and implemented in SOI [12]. A direct application of the ULP diode was the realization of a memory cell that achieves a reduction of the static and dynamic power consumptions by using transistors in very weak inversion

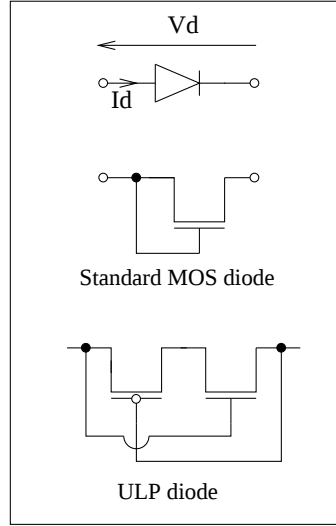
regime [12].

The ULP diode is obtained by the combination of a NMOS and a PMOS transistor as depicted in Figure 5.8(a). Ultra-low leakage is obtained because, when the ULP diode is reversely biased, both transistors operate with negative V_{gs} voltages, leading to strongly reduced leakage current in comparison to standard diode. Furthermore, when increasing the reverse bias voltage, the reverse ULP diode current first increases due to the V_{ds} increase of the transistors. The current reaches a peak value and then strongly decreases with the V_{gs} of the transistors becoming more and more negative. Therefore, this behaviour leads to a negative resistance region as depicted in the simulated I-V characteristic of the ULP diode (ULPD) in $0.13\mu\text{m}$ PD SOI/CMOS (Figure 5.8(b)). Figure 5.9 shows the measured characteristics of standard and ULP diodes in $0.13\mu\text{m}$ PD SOI/CMOS.

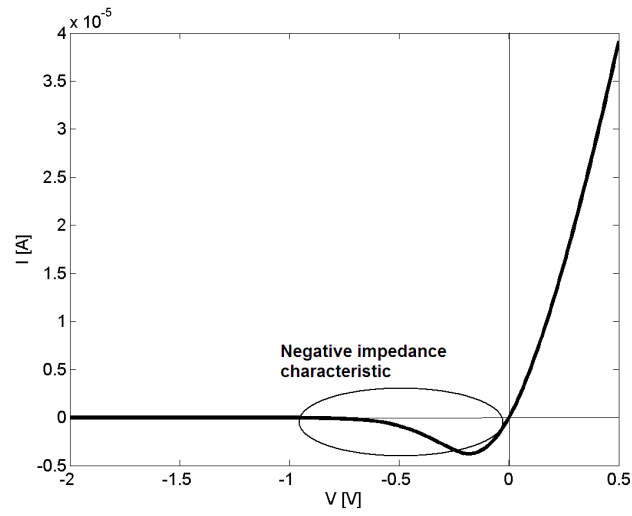
Even though it was first proposed for applications operating in very weak inversion regime, the reverse biased (i.e. $V < 0$) ULP diode can also be used in moderate or strong inversion depending on the threshold voltages of the NMOS and PMOS transistors used in the diode. Subsequently, higher reverse current peaks in the negative resistance region can be reached for NMOS and PMOS transistors with negative and positive V_t respectively as shown in Figure 5.10 (ULP-NMOS and ULP-PMOS transistors). As detailed in [12], the value of this current can be roughly judged at the intersection of N and P Id-Vg curves. Surely, this high current peak comes with higher leakage current than in the ULP diode operating in weak inversion regime, however for correct operation of the level restorer, it is necessary to ensure sufficient level restoration with short delay timing.

The negative resistance region in the ULP diode (ULPD) characteristic is exploited here to restore the weak logic level at the sum output node.

The ULPD based level restorers that we used are shown in Figure 5.11. These depict the low logic level restorer and the high logic level restorer respectively. Indeed, they are both needed to implement the ULPGA based RCA circuit that we describe in section 5.9.7. C_{node} depicts the



(a)



(b)

Figure 5.8: ULP Diode (a). Simulated ULP diode characteristic in $0.13\mu\text{m}$ PD SOI CMOS technology, LETI depletion mode MOSFETs, $V_{tn} = -0.28\text{V}$, $V_{tp} = +0.28\text{V}$ (b).

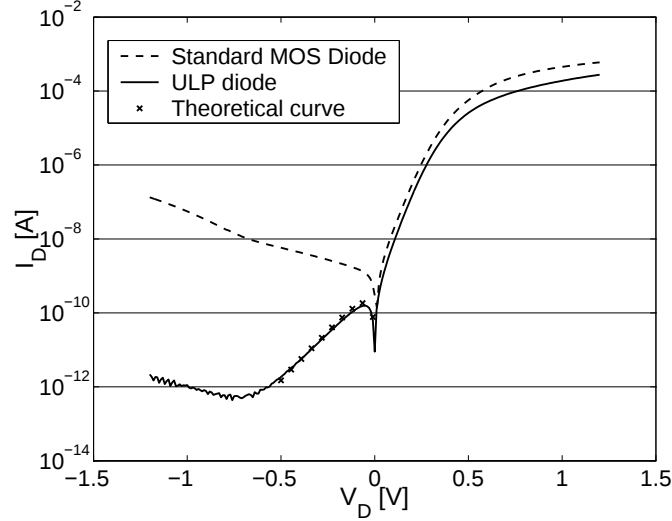


Figure 5.9: Measured characteristics of standard MOS and ULP diodes and theoretical modelling of the ULP diode, (ST MOSFETs, $W = 1\mu m$, $L = 0.13\mu m$). (Adapted from [13]).

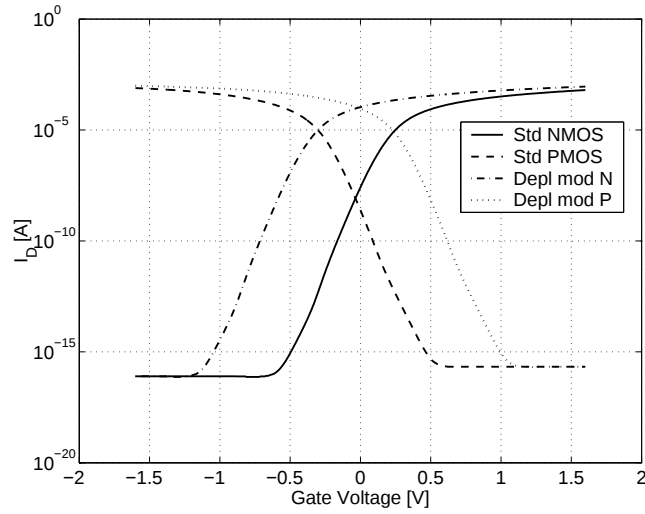


Figure 5.10: Simulated I_d - V_{gs} characteristics of multi-threshold NMOS and PMOS ($W/L=4.6$ for N-type $W/L=12.5$ for P-type, $|V_{ds}|=1.2V$ and floating body devices configuration) in $0.13\mu m$ PD SOI CMOS technology. “Depl mod N” and “Depl mod P” depict depletion mode NMOS and PMOS respectively.

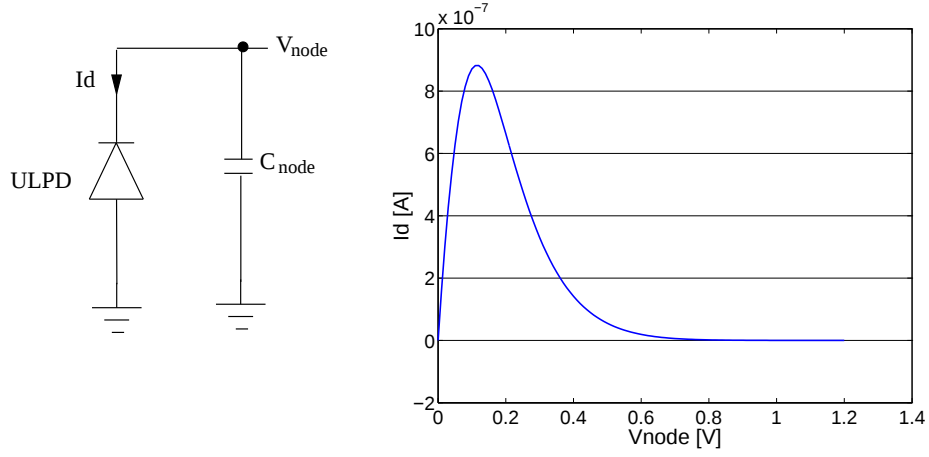
parasitic capacitance at the level restorer node. The operation of the low logic level restorer shown in Figure 5.11(a) is as follows. When the voltage V_{node} is between 0 and $V_{dd}/2$, the ULPD current I_d is positive and discharges C_{node} . For V_{node} voltages comprised between $V_{dd}/2$ and V_{dd} , both NMOS and PMOS transistors are turned OFF (since they have negative V_{gs} values). Thus, no current flows through the ULP diode. Reciprocally in the high logic level restorer shown in Figure 5.11(b), when the voltage V_{node} lies between $V_{dd}/2$ and V_{dd} , the ULPD current peak drives the voltage V_{node} up to V_{dd} .

These Level restorers use the depletion mode MOSFETs whose Id- V_{gs} characteristics are shown in Figure 5.10. Indeed, the current peak obtained with standard MOSFETs (their Id- V_g characteristics are shown in Figure 5.10) as shown in Figure 5.12 does not give satisfactory performance.

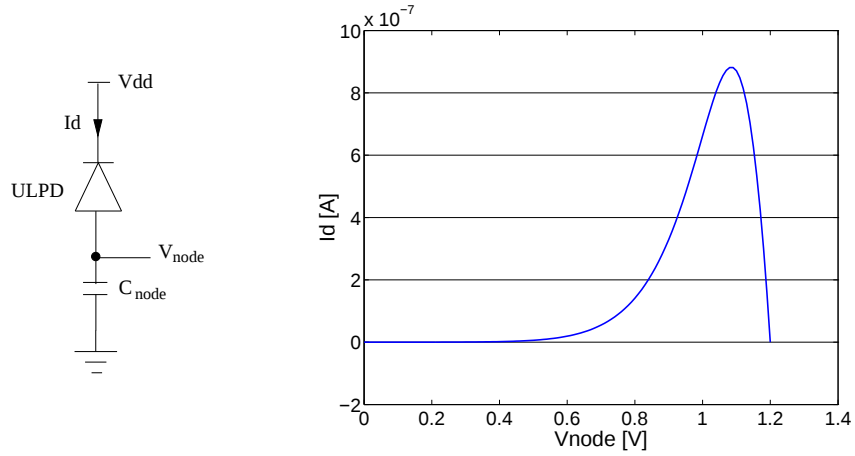
5.9.3 Feasibility in $0.13\mu\text{m}$ PD SOI/CMOS

As stated before, the value of the current peak in the reverse characteristic depends on the V_t of the transistors. In order to have a sufficient current peak for logic level restoration within an acceptable delay, the ULPD must be implemented with negative V_t transistors.

Multiple threshold voltage transistors can be obtained in several ways. The most common one is the use of different channel doping densities. Using different oxide thickness (T_{ox}) also leads to different V_t , but both these techniques complicate the process and thus multiple- V_t are not often available for designers. In $0.13\mu\text{m}$ PD SOI/CMOS process, only intrinsic devices (corresponding to almost zero- V_t transistors) are available. Varying the body or back gate voltage for bulk and fully depleted SOI devices respectively, can also be a solution, but this is not feasible since devices cannot share the same well (for bulk devices) or the same back-gate (for FD SOI devices) in this case. Partially depleted SOI devices are better suited to this technique since the body is isolated and can then be contacted to different biasing. Nonetheless, if multiple- V_t is not feasible for designers in the targeted technology, the level restorer



(a)



(b)

Figure 5.11: The ULP diode based low-logic level restorer and simulated current-voltage characteristic (a). The ULP diode based high-logic level restorer and simulated current-voltage characteristic (b), in $0.13\mu\text{m}$ PD SOI CMOS process, $W = 0.25\mu\text{m}$, $L = 0.13\mu\text{m}$, with depletion mode MOSFETs (whose I_d - V_g characteristics are shown in Figure 5.10).

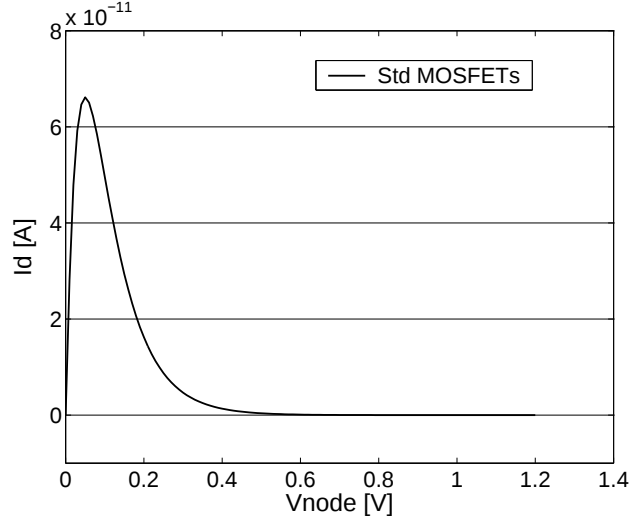


Figure 5.12: Simulated current-voltage characteristic in the ULP diode based low-logic level restorer, in $0.13\mu\text{m}$ PD SOI CMOS process, Standard MOSFETs (whose Id-Vg characteristics are shown in Figure 5.10), $W = 0.25\mu\text{m}$, $L = 0.13\mu\text{m}$.

with a high current peak as required for the optimized structure of the hybrid full adder cannot then be realized. Consequently, we propose two variants of the sum-block. The first one uses the ULPD as level restorer and the second uses inverters as it was proposed by authors of the LP (low power) XOR/XNOR gates [10] in order to restore the weak logic level. These gates are presented in the next section.

5.9.4 LP XOR/XNOR gates

The LP XOR/XNOR gates were proposed in [10]. They are depicted in Figure 5.13(a) and 5.14(a). From the LP XOR gate analysis, it can be seen that the output signal has a good logic level for input signals $(A,B)=(0,1), (1,0), (1,1)$. For $(A,B)=(0,0)$ configuration, each PMOS is switched ON and pass a weak “0” logic (i.e. $|V_{tp}|$). Reciprocally, the XNOR gate shows good logic levels for input signals $(A,B)=(0,0),(0,1),(1,0)$. For $(A,B)=(1,1)$ configuration, each NMOS is switched ON and pass a weak “1” logic (i.e. $V_{dd} - V_{tn}$). In order to

enhance the driving capability at the output nodes, Wang et al. in [10] use a CMOS inverter as shown in Figure 5.15.

Figures 5.13(b) and 5.14(b) show the output waveforms of LP XOR and XNOR gates. “XOR” and “XNOR” signals depict the un-restored logic levels at the output nodes for the mentioned configurations; “XOR-R” and “XNOR-R” depict the restored weak logic levels in XOR and XNOR gate respectively using the ULP-diode based level restorers.

Authors in [14] have carried out a comparison between basic gates implemented with different logic styles or topologies in deep submicron technologies. This study has shown that the LP XOR gate has the lowest leakage power in comparison with XOR gates based on other well-known topologies.

By using the LP XOR/XNOR gates, the sum-block does not need the true input signals and their complements at the same time. Thus, in case of RCA implementation, the sum-block does not need the complementary carry-in signal since we implement it with LP XOR and XNOR gates, as opposed to the sum-block in the BBL-PT FA. This avoids the presence of inverters on the carry-chain in multiple-bit adders and thus leads to a gain in delay performance. The output inverter in the carry-block can then be removed.

5.9.5 *The Ultra Low Power Full adder*

The ULPFA based on LP XOR gate and ULP diode is shown in Figure 5.16(a). The second variant of the hybrid full adder that uses LP XNOR gates and inverters is shown in Figure 5.16(b). The voltage step observed in the $0 \rightarrow 1$ transition on the sum output node of the BBL-PT full adder, is removed when using this two new variants as it can be seen on their respective output waveforms shown in Figures 5.17 and 5.18.

To evaluate the performances of the ULPFA and the hybrid FA versus the BBL-PT FA and two other full adders based on well-known low power static design styles namely the conventional static CMOS (CMOS FA) and the complementary pass logic (CPL FA), reviewed in section 2.3.4.1.

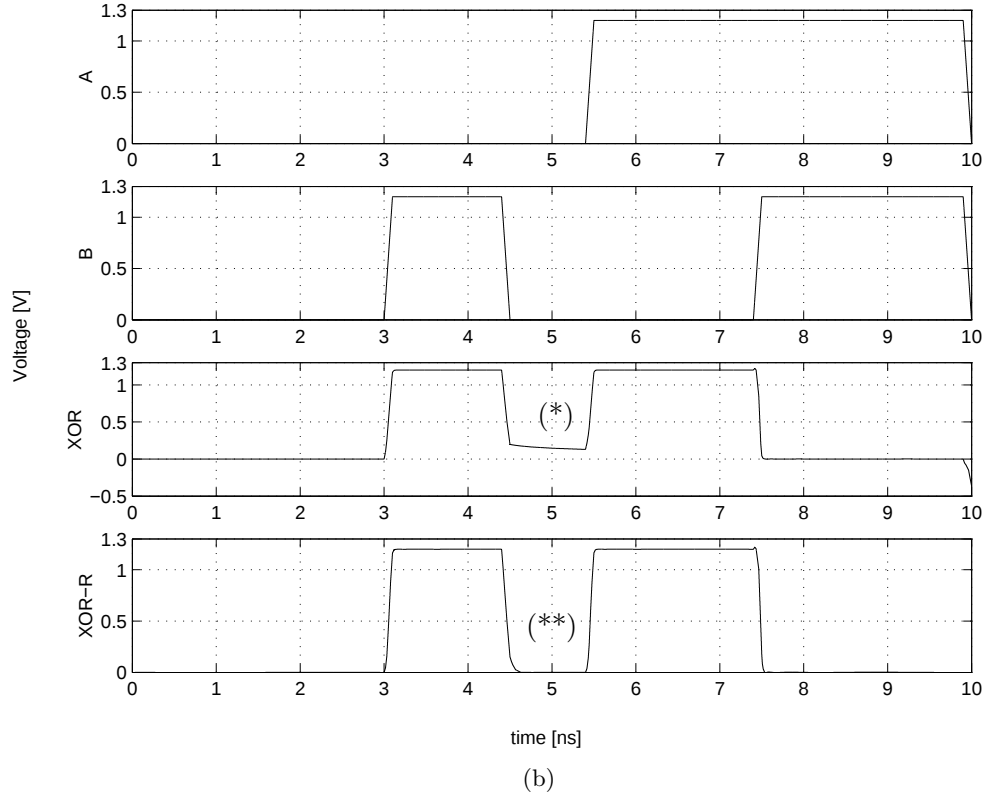
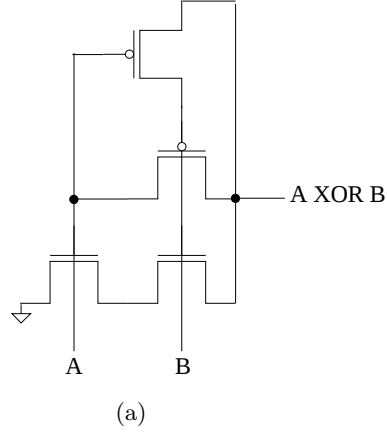


Figure 5.13: LP XOR gate [10] (a). LP XOR output waveforms (b). One can observe the weak “0” logic that occurs on “XOR” signal for $(A,B)=(0,0)$ configuration (denoted $(*)$ on the waveforms), and the good “0” logic obtained in “XOR-R” signal through level restoration (denoted $(**)$ on the waveforms).

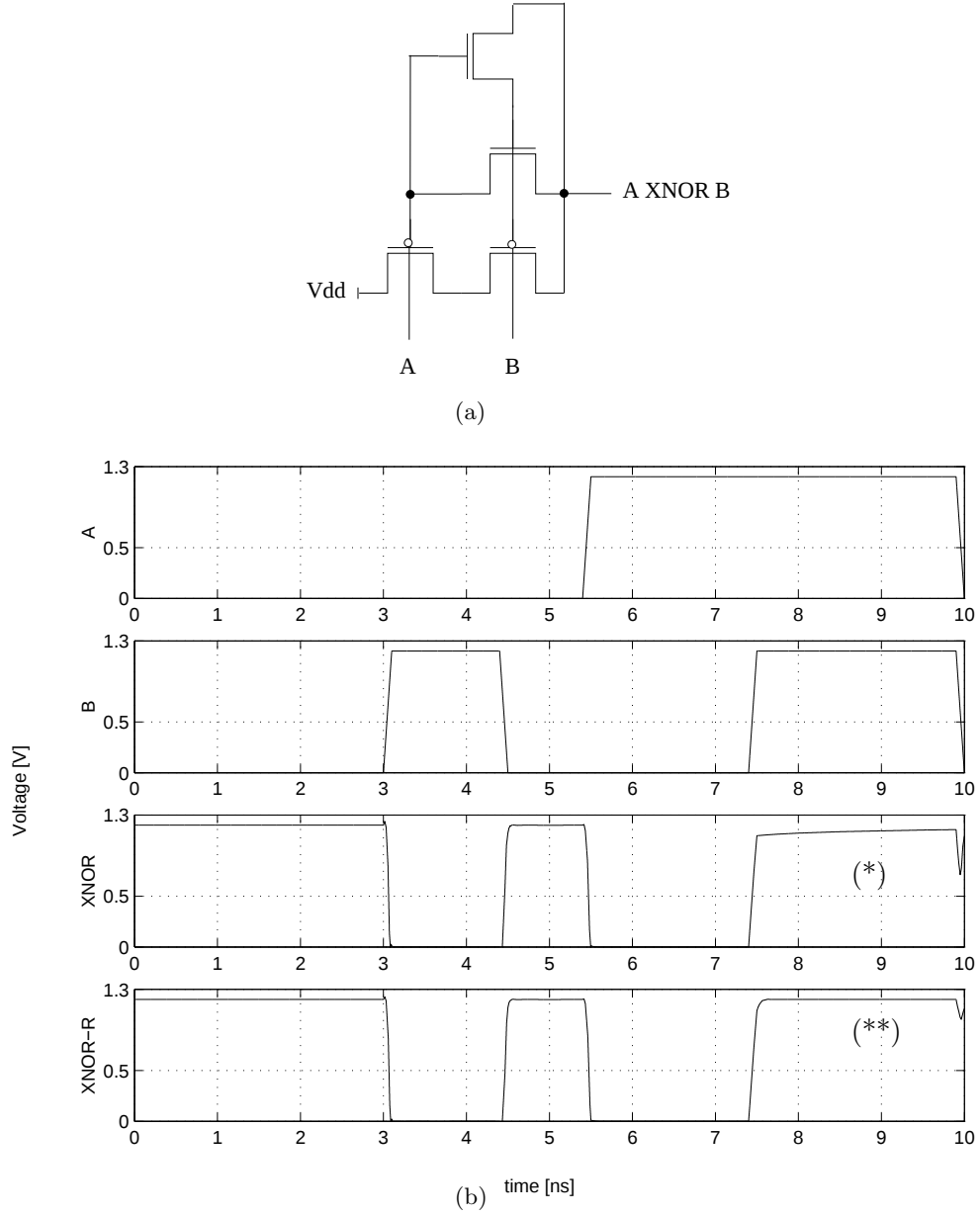


Figure 5.14: LP XNOR gate [10] (a). LP XNOR gate output waveforms (b). One can observe the weak “1” logic that occurs on “XNOR” signal for (A,B)=(1,1) configuration (denoted (*)) on the waveforms), and the good “1” logic obtained in “XNOR-R” signal through level restoration (denoted (**)) on the waveforms).

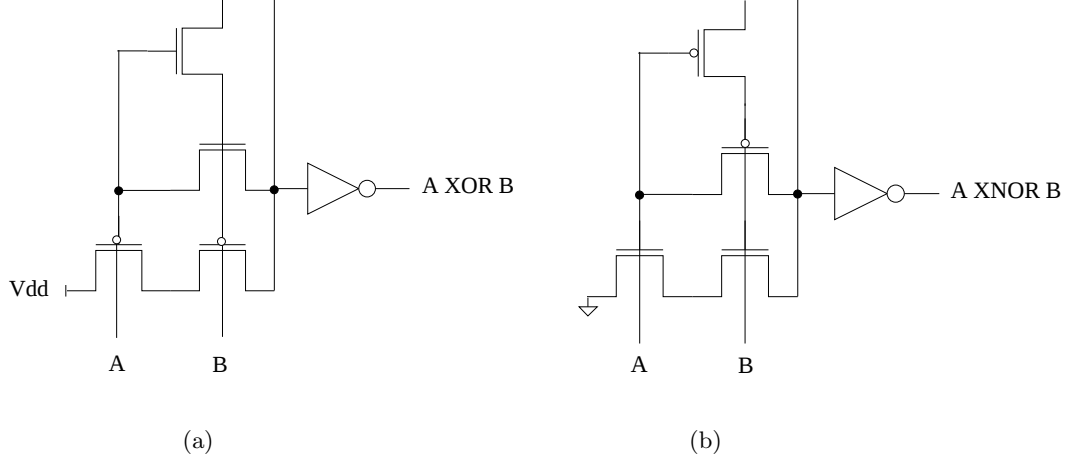


Figure 5.15: The LP XOR and XNOR structures with driving outputs [10]. XOR gate (a). XNOR gate (b).

Simulations were performed in $0.13\mu\text{m}$ PD SOI/CMOS. SPICE simulations were carried out under a $V_{dd}=1.2\text{V}$ and a clock frequency of 100MHz . No fan-out is loading the five full adders. Transistor sizing of these is shown in Appendix B. For all possible input sets applicable to the full adder, we extracted the mean power consumption and the worst case delay. Table 5.4 summarizes simulation results. Therefrom, it can be seen that CPL FA shows a better delay and power delay product (PDP) in comparison to CMOS FA. Conversely, it shows the highest total and static power dissipations due to its higher transistor number. The ULPFA outperforms the four full adders in both delay and power delay product thanks to a limited number of transistors on the paths between power supply lines and the output nodes and a reduced parasitic capacitance at the sum and carry output nodes. Regarding the hybrid FA shown in Figure 5.16(b), it shows a higher delay and PDP than all full adders because of the higher number of transistors on the critical path that occurs in the sum-block. Even it consumes lower total and static power than CPL FA, it appears disadvantageous in terms of power consumption in comparison to the ULPFA, BBL-PT FA and CMOS FA. Because of the threshold loss, the internal node X or Y (see

5.9 ADVANCES IN THE HYBRID FULL ADDER

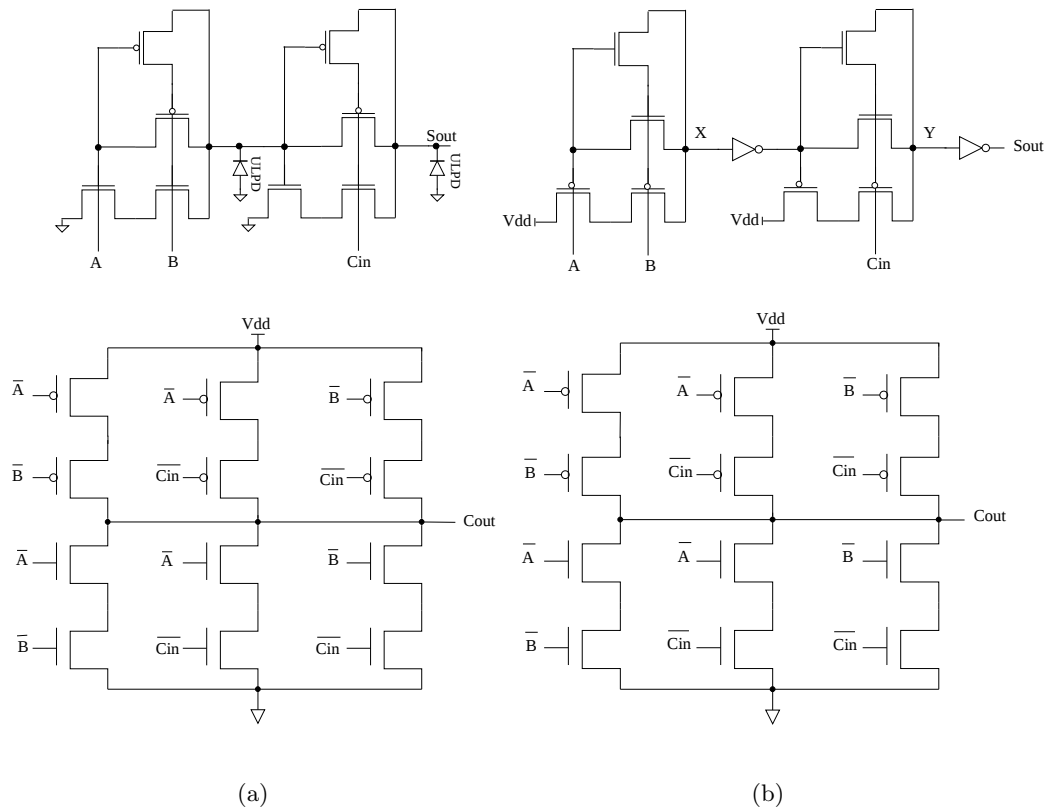


Figure 5.16: The Ultra Low Power Full adder (ULPFA) using the ULP diodes (a). The Hybrid full adder (Hybrid FA) using the LP XNOR gate with output inverters for level restoration (b).

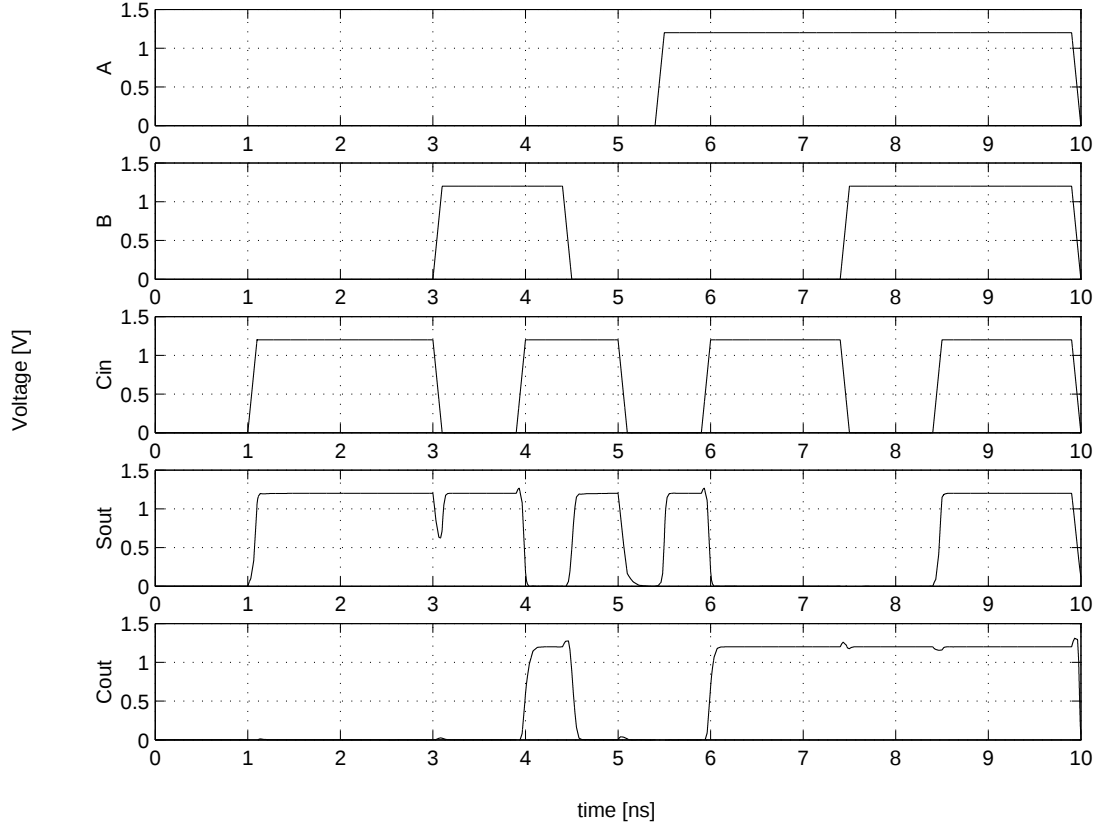


Figure 5.17: The ULPFA output waveforms in $0.13\mu\text{m}$ PD SOI/CMOS, ST LL MOSFETs, with the transistor sizes shown in appendix B, $V_{dd} = 1.2\text{V}$. One can observe that the voltage step which occurs at the output node “Sout” during a transition $0 \rightarrow 1$ in the BBL-PT is removed.

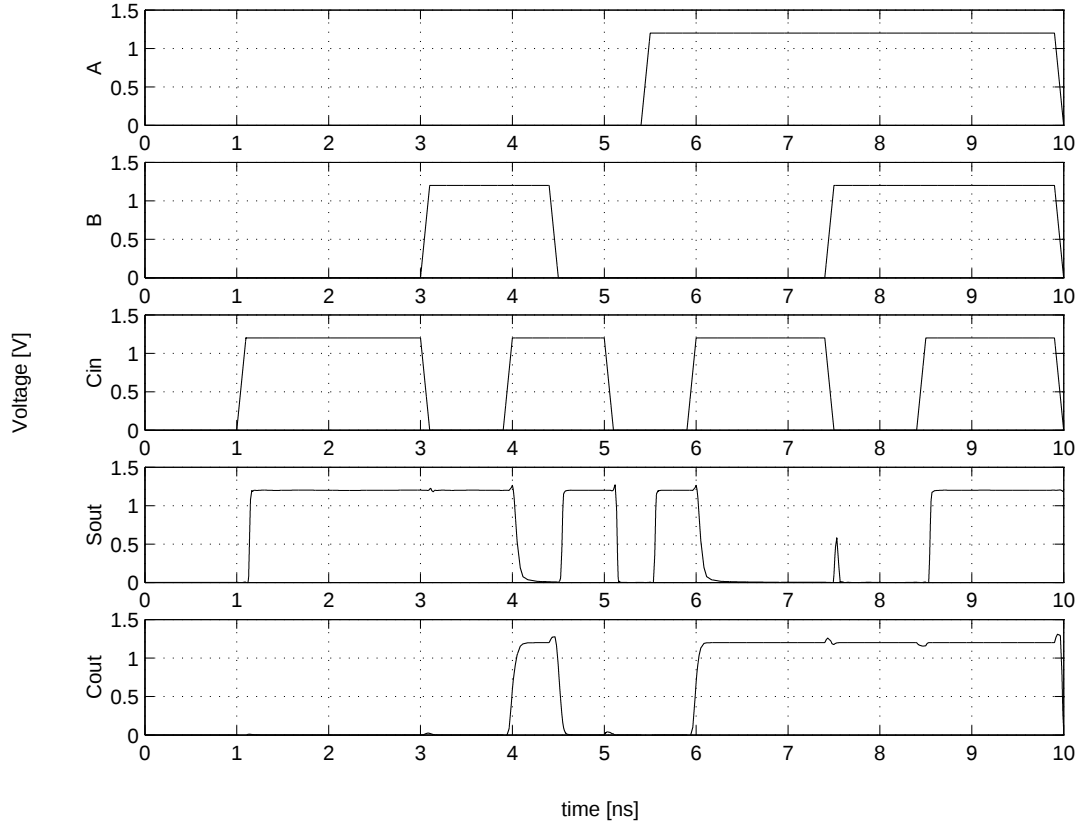


Figure 5.18: Output waveforms of the hybrid full adder using the LP XOR/XNOR and inverters for level restoration, in $0.13\mu\text{m}$ PD SOI/CMOS, ST LL MOSFETs, with the transistor sizes shown in appendix B, $V_{dd} = 1.2\text{V}$. One can observe that the voltage step which occurs at the output node “Sout” during a transition $0 \rightarrow 1$ in the BBL-PT is removed.

Design	Delay (ps)	Total power (μ W)	PDP (Joule)	Static power (nW)	Device count
CMOS FA	124	1.43	1.77E-16	0.8	28
CPL FA	80.26	1.72	1.38E-16	1.2	32
BBL-PT FA	110.3	1.49	1.64E-16	0.66	23
ULPFA	59.5	0.505	0.3E-16	0.2	24
hybrid FA	142	1.57	2.24E-16	0.9	24

Table 5.4: 1-bit Full adder comparison in 0.13 μ m SOI/CMOS under a $V_{dd} = 1.2$ V, $f=100$ MHz, with ST LL MOSFETs $V_t = 0.36$ V (except for transistors in the ULP diode), no fan-out.

Figure 5.16(b)) will be lower than normal by a voltage V_{tn} for input configuration (1,1). This results in a continuous DC paths through the inverter and consequently increases power in the circuit. However, it shows better delay performance when cascaded in a 8-bit RCA circuit as it is shown in section 5.9.7.

The functionality of the ULPFA and hybrid FA has been observed by simulation under supply voltage below 1V. They both operate reliably under voltages as low as 0.6V. However, the delay performance of the sum response dramatically degrades for $V_{dd} < 0.8$ V.

5.9.6 Static power in the ULPFA

R.-X. Gu et al. have introduced in [15] an analytical model of the leakage current for a series of stacked transistors. This analytical model has shown that the more transistors are in stack, the smaller is the leakage current, i.e. in case of one transistor (I_{s1}), two stacked transistors (I_{s2}) and three stacked transistors (I_{s3}) (see Figure 5.19), $I_{s3} < I_{s2} < I_{s1}$. The stack effect contributes then in leakage reduction. Let us consider two stacked devices in the standby mode (i.e. $V_g = 0$) as shown in Figure 5.19. Due to the small drain current, V_{d2} will have a positive

value. The gate-to-source voltage of transistor Q_1 has consequently a negative value. According to equation 5.6 [15], the negative V_{gs} gives a smaller leakage current than the current in a branch containing only one transistor.

$$I_s = I_0 \exp\left(\frac{V_{gs} - V_{th}}{nV_T}\right) \left(1 - \exp\left(\frac{-V_{ds}}{V_T}\right)\right) \quad (5.6)$$

R.-X. Gu et al. have calculated I_{s1} , I_{s2} and I_{s3} for typical deep submicron CMOS technology under supply voltage $V_{dd}=0.9$ to $1.5V$ in [15]. Therefrom, they obtain the following equations:

$$I_{s1} = 1.8 \cdot I_0 \exp(-V_{tho}/nV_T) \cdot \exp(\eta V_{dd}/nV_T) \quad (5.7)$$

$$I_{s2} = 1.8 \cdot I_0 \exp(-V_{tho}/nV_T) \quad (5.8)$$

$$I_{s3} = I_0 \exp(-V_{tho}/nV_T) \quad (5.9)$$

with $V_{th} = V_{tho} - \eta V_{dd}$. η models the drain induced barrier lowering (DIBL) effect.

The leakage current becomes negligible when the number of stacked devices is more than three. The leakage current for transistors in parallel equals the sum of currents through each transistor.

Branch based gates for which the stacked transistors per branch is ≥ 2 can be considered then as standby power aware gates.

If we examine the leakage current in the XOR using the ULPD diode for level restoration, assuming that the leakage current in ULP diode is negligible compared to leakage in a MOSFET transistor, the highest leakage current is given by $2(I_{s1})_{PMOS}$ and occurs for configuration (A,B)=(1,1). The lowest leakage current is given by $(I_{s2})_{NMOS}$ and occurs for configuration (A,B)=(0,0). This is in contrast to gate obtained by cascading a XNOR and an inverter as shown in Figure 5.15(a), having a leakage current given by $2(I_{s1})_{NMOS} + (I_{s1})_{PMOS}$ for configurations (A,B)= (0,0), (0,1), and (1,0), and $(I_{s1})_{PMOS} + (I_{s2})_{PMOS}$ for configuration (A,B)=(1,1).

We can conclude that the combination of standby power aware struc-

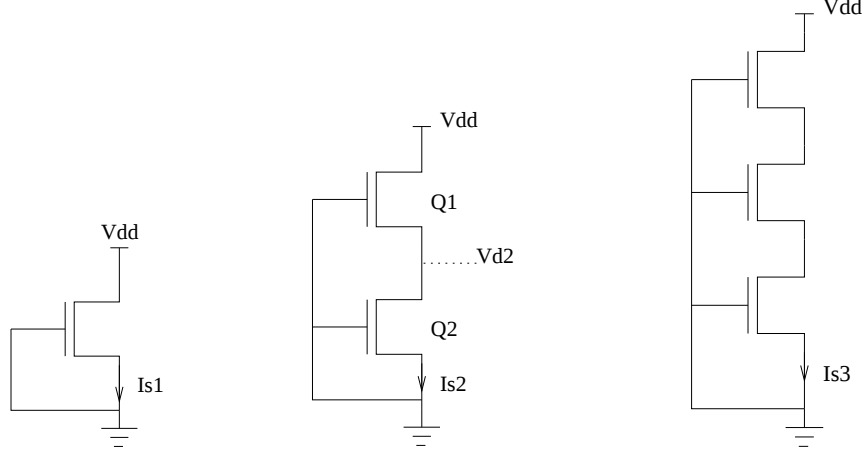


Figure 5.19: Stack of transistors with $V_g = 0$.

tures lead to the much low static power in the ULPFA cell.

5.9.7 ULPFA based 8-bit RCA

Because some cells might show good performances in stand alone operation but fail to perform well when cascaded in larger circuits due to unsufficient driving capability, the five 1-bit full adders were cascaded in 8-bit RCA implementation in order to achieve a fair qualitative comparison. A multiple-bit RCA based on the ULPFA can be implemented by alternating a 1-bit ULPFA with true input signals (the outputs are then $Sout_i$ and $\overline{Cout_i}$) as shown in Figure 5.21(a), and 1-bit ULPFA with complementary input signals. Since the outputs in this configuration are $\overline{Sout_i}$ and $Cout_i$, we can use a LP XNOR gate instead of the second LP XOR gate in order to obtain $Sout_i$ (see Figure 5.21)(b) since:

$$X = A_i \oplus B_i = \overline{A_i} \oplus \overline{B_i} \quad (5.10)$$

$$X \oplus \overline{Cin} = \overline{X \oplus Cin} = \overline{Sout} \quad (5.11)$$

5.9 ADVANCES IN THE HYBRID FULL ADDER

As mentioned in section 5.9.4, the LP XNOR gate delivers a weak logic “1” for (A,B)=(1,1) configuration. To restore the weak logic “1”, a high logic level restorer based on the ULP diode (Figure 5.11(b)) was used. The described gate cascading is illustrated in Figure 5.20. A similar implementation principle of the 8-bit RCA based on the hybrid FA, alternates the cell shown in Figure 5.22(a) and the cell shown in Figure 5.22(b).

8-bit RCA circuits were implemented with the five full adders and

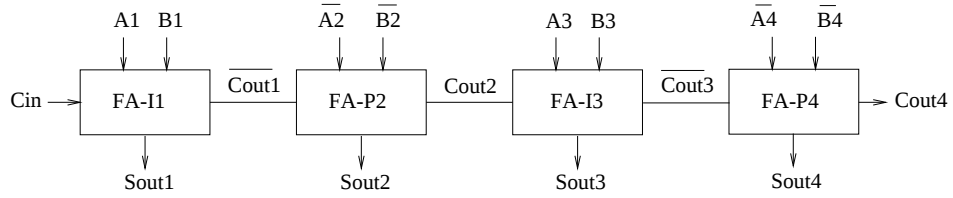


Figure 5.20: A 4-bit RCA based on the ULPGA or the hybrid FA cell.

compared under $V_{dd} = 1.2V$ and $f=100MHz$. Simulation results are summarized in Table 5.5. It can be seen that the CPL 8b RCA shows

Design	Delay (ps)	Total power (μW)	PDP (fJ)	Static power (nW)	Device count
CMOS FA	833	17.5	14.57	6.1	224
CPL FA	514	22	11.3	9.58	256
BBL-PT FA	845	16.78	14.2	4.9	184
ULPGA	706.6	8.69	6.15	2.35	192
Hybrid FA	750.8	17.24	12.94	7.37	192

Table 5.5: 8-bit RCA adder comparison in $0.13\mu m$ SOI CMOS under a $V_{dd} = 1.2V$, $f=100MHz$, with ST LL MOSFETs $V_t = 0.36V$ (except for transistors in the ULP diode).

the best delay performance but consumes the most in both total and

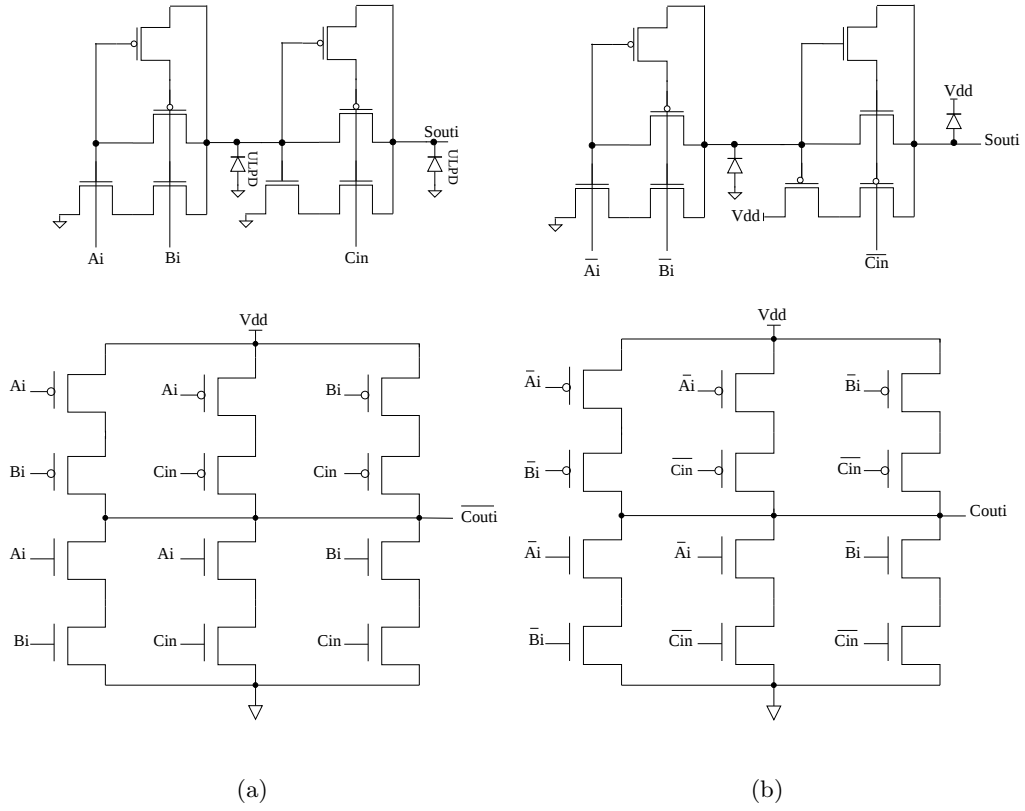


Figure 5.21: ULPFA cell implementing the FA- I_x boxes as shown in Figure 5.20 (a). ULPFA cell implementing the FA- P_x boxes (Figure 5.20)(b).

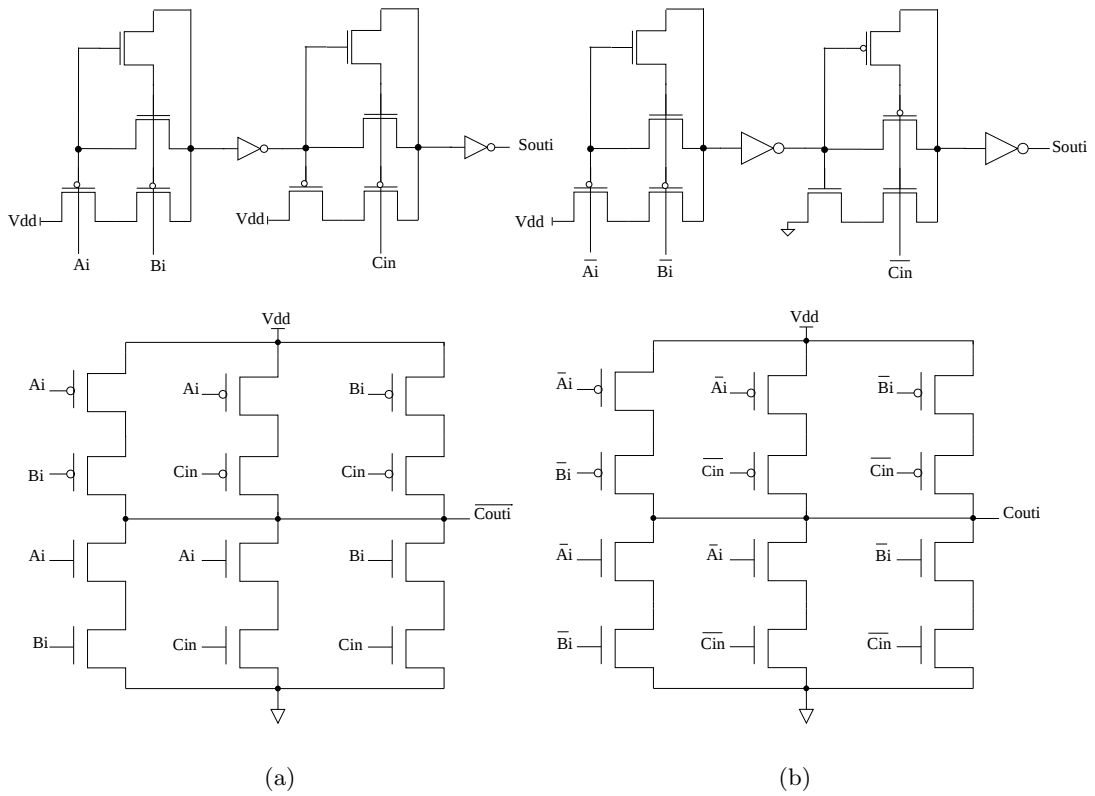


Figure 5.22: Hybrid FA cell implementing the FA- I_x boxes (a). Hybrid FA cell implementing the FA- P_x boxes (b).

static power due to its high transistor number. Thanks to its high speed, the CPL 8b RCA shows better power delay product than the CMOS 8b RCA, the BBL-PT and the hybrid FA based 8b RCA circuits. The BBL-PT 8b RCA shows a slightly higher delay than the CMOS 8b RCA but consumes lower total and static power than the CMOS circuit thanks to its lower device number. Moreover, it features slightly better PDP than the CMOS circuit.

As mentioned before, the 8b RCA circuit based on the hybrid FA shows better delay and power delay product compared to BBL-PT based RCA circuit. The 8b RCA based on the ULPFA cell outperforms the four circuits with regard to power consumption while featuring an acceptable delay performance since it does not use additional inverters, which increase both power and delay. It is informative to mention that it was reported in [16] that the parasitic capacitance in the ULP diode is more than 2 times smaller than in an inverter gate. With a total power almost 2 times smaller than in the BBL-PT circuit and more than 2 times smaller than in CPL circuit, and thanks to its good delay performance, the ULPFA based 8b RCA shows the best power delay product.

5.10 Conclusion

In this chapter, we have presented a new hybrid structure of full adder. Its optimization has been carried out for further lower total and static power. Simulations in deep submicron technology have shown that the ULPFA variant of this full adder achieves largely better power delay product when compared with two other valuable full adders and the two other variants of the hybrid full adder.

Furthermore, a study of MTCMOS and DTMOS techniques has been carried out at the cell level. Simulations have shown that DTMOS technique can be used to improve pass-transistors performance when high V_t devices are used. However, the increase of the switching capacitance makes this technique not suitable for low power applications.

References

- [1] I. Hassoune, A. Neve, J.D Legat, and D. Flandre, “Investigation of low-power low-voltage circuit techniques for a hybrid full-adder cell,” in *Proceedings of PATMOS conference, Springer-Verlag, LNCS 3254*, Santorini, Greece, 15-17 September 2004, pp. 189–197.
- [2] I. Hassoune, A. Neve, J.D Legat, and D. Flandre, “Branch-based design of carry calculation cell for ultra-low power and high-temperature applications,” in *Proceedings of HITEC’04 Conference*, Santa-Fe, USA, 17-20, May 2004.
- [3] A. Neve, *Top-Down optimization for High-performance and low-power adders in deep-submicron SOI*, PhD thesis, UCL, Belgium, January 2004.
- [4] J.-M. Masgonty, Ph. Mosch, and C. Piguet, “Branch-based digital cell libraries,” in *Proceedings of EURO ASIC’91 conference*, Paris, France, 27-31 Mai 1991.
- [5] C.Piguet, A. Stauffer, and J. Zahnd, *Conception des circuits ASIC numériques CMOS*, Dunod, 1990.
- [6] J.B. Kuo and J-H. Lou, *Low Voltage CMOS VLSI Circuits*, By John Wiley & Sons, 1999.
- [7] M. Anis and M. Elmasry, *Multi-Threshold CMOS digital circuits, managing leakage power*, Kluwer Academic Publishers, 2003.
- [8] F. Assaderaghi, D. Sinitsky, S.-A. Parke, J. Bokor, P.-K. Ko, and C. Hu, “Dynamic threshold-voltage MOSFET (DTMOS) for ultra-low voltage VLSI,” *IEEE Transactions on Electron Devices*, vol. 44, no. 3, pp. 414–422, March 1997.
- [9] T. Sakurai and A.-R. Newton, “Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas,” *IEEE J. Solid-State Circuits*, vol. 25, pp. 584–594, April 1990.
- [10] J.-M. Wang, S.-C. Fang, and W.-S. Feng, “New efficient designs for

xor and xnor functions on the transistor level”,” *IEEE Journal of Solid State Circuits*, vol. 29, no. 7, pp. 780–786, July 1994.

- [11] V. Dessard, *SOI specific analog techniques for low-noise, high-temperature or ultra-low power circuits*, PhD thesis, UCL, Belgium, 2001.
- [12] D. Levacq, C. Liber, V. Dessard, and D. Flandre, “Composite ULP diode fabrication, modeling and applications in multi-vth FD SOI CMOS technology,” *Solid State Electronics*, vol. 48, no. 6, pp. 1017–1025, June 2004.
- [13] D. Levacq, *Low leakage SOI CMOS circuits based on the Ultra-Low Power diode concept*, PhD thesis, UCL, Belgium, 2006.
- [14] G. Merrett and B.-M. Al-Hashimi, “Leakage power analysis and comparison of deep submicron logic gates,” in *Proceedings of PAT-MOS conference, Springer-Verlag, LNCS 3254*, Santorini, Greece, 15-17 September 2004, pp. 198–207.
- [15] R. X. Gu and M.-I. Elmasry, “Power dissipation analysis and optimization of deep submicron CMOS digital circuits,” *IEEE Journal of Solid-State Circuits*, vol. 31, no. 5, pp. 707–713, May 1996.
- [16] D. Levacq, V. Dessard, and D. Flandre, *Science and Technology of semi-conductor-on-Insulator structures and devices operating in a harsh environment*, Edited by D. Flandre, A.-N. Nazarov and P.-L.-F. Hemment, NATO Science series, Kluwer Academic Publishers, Series 2: Mathematics, Physics and chemistry, vol. 185, pp.133-144, 2005.

CHAPTER 6

CONCLUSIONS

Since integration technology is approaching the nanoelectronics range, some practical limits are being reached (leakage power, clock distribution). This trend will become difficult to maintain unless new circuit methods, synchronization architectures and new CAD tools are emerging and adopted by industry in order to meet the specifications in high performance and low power integrated circuits and systems.

This thesis investigated a new class of dynamic differential logic dedicated to implementation of low-power self-timed circuits. The applicability of LSCML style in implementation of self-timed arithmetic circuits has been shown through simulation results of an 8b RCA and CLA adders. Furthermore, we demonstrated through simulations of the Khazad S-box module, that LSCML class has a real potential for lowering power consumption variation. Indeed, the LSCML(ST1) S-box has shown a power consumption standard deviation more than two times smaller than the one in DyCML and six times smaller than the one in self-timed DDCVSL. The figures of merit used to measure its efficiency regarding the security concept have no practical relevance since in the context of optimal statistical analysis of the power consumption, the attack efficiency depends only on the correlation between power consumption predictions and practical measurements of the cipher device [1] [2]. A power analysis attack remains feasible against LSCML class. Nonetheless, independently of the accuracy of the model used by the attacker to predict the power consumption, we believe that thanks to the reduced power consumption variation, LSCML based implementations results in a need for more accurate measurements, thus making them much more difficult to realize. Consequently it makes the cracking through a power analysis, much harder than for devices based on static CMOS logic.

Furthermore, thanks to its low swing, the amount of charge needed for charging/discharging parasitic capacitances, is limited and consequently

avoids high current peaks. This might reduce noise due to switching and probably gives an other potential to LSCML circuits in the implementation of advanced low-power and low switching noise circuits for mixed-signal analog-digital applications. Indeed, some current mode logic families have been proposed to reduce switching noise in analog-digital integrated circuits [3]. These logic families work with a constant current source which eliminates current variation and avoids switching noise. However, the DC power is a major drawback of such logic families which makes them not appropriate for low power circuits.

The switching noise on power supply lines, which basically results from an inductive coupling through parasitic inductance in package and power supply distribution network [4], is a $L_M \frac{di}{dt}$ voltage drop (where L_M denotes a mutual inductance) on the power supply lines. Since the switching speed of devices increases, large current swing within a short time will produce high $L_M \frac{di}{dt}$ voltage drop. Furthermore, high $\frac{di}{dt}$ might generate an inductive crosstalk between two circuits (or two lines) placed close to each other, through a mutual inductive coupling. Thus, in high speed design, the high current peak occurring in short time, obviously leads to significant inductive crosstalk and switching noise.

A capacitive crosstalk between two circuits (or signals) placed close to each other, basically results from a capacitive coupling through a mutual capacitance C_M . The injected current through C_M is proportional to $\frac{dv}{dt}$. It thus makes sense that the higher the voltage swing, the higher the capacitive crosstalk.

Therefore, logic families that feature a low output voltage swing and low current peaks can bring important advantages in terms of reduction of crosstalk, switching noise and electromagnetic emissions. Consequently, they might bring a solution for the incoming drawbacks in high-speed nanoscaled integrated circuits. This aspect was not investigated in this thesis. Further research might reveal the potential of LSCML in implementation of low switching noise and low crosstalk applications.

As introduced in chapter 2, the self-timing approach features more robustness to variation of some environmental parameters (supply-voltage,

temperature) than the synchronous one. Since the LSCML class features a self-timed operation, this makes it more prone to resist such parameters variation than logic styles that use the delay scheme for clock signal generation. Indeed, the delay variability with such parameters can produce erroneous evaluations. Furthermore, we reviewed in chapter 2 some works reporting the power saving when using asynchronous architectures. Since LSCML class might be used to implement asynchronous circuits, it is probably interesting to investigate within future research the power saving at the architecture level.

We proposed in chapter 5 a new structure of a hybrid full adder. The 8b RCA circuit based on the optimized structure namely the ULPFA, achieves a total power and a leakage power, which are both reduced by 50% compared to the 8b RCA implemented with conventional CMOS full adder. The reduced dynamic and leakage power obtained in the ULPFA, makes this full adder well suited to implement low power multipliers and further arithmetic circuits. Application of some low power low voltage techniques at the cell level has shown that DTMOS technique offers a little gain in speed and only when high- V_t transistors are used. However, it can be used to improve pass-transistors performance. The increase of the switching capacitance makes this technique not suitable for low power applications.

Further future research on arithmetic binary adders might demonstrate the potential of branch-based design in implementation of low power circuits and moreover compare different adder architectures based on BBL logic through an evaluation of the ULPFA based RCA proposed in chapter 5, the carry-select adder structure proposed in [5] and the branch-based CLA based on the 4b BBL CLA that we synthesize in appendix C. The latter can be used to implement n-bit CLA adders. Simulations have shown a correct functionality of the 4b BBL-based CLA. Further research should evaluate it versus CLA circuits based on other static logic styles.

CHAPTER 6. CONCLUSIONS

Given the technology trend, an aknowledge of the behaviour of the proposed logic style and the ULPFA with respect to voltage and transistor scaling, process and environment parameters variation and compatibility with surrounding circuits is of a great importance.

Generally, dynamic logic are not suitable for environments where temperature, and threshold voltage may vary significantly due to their sensitivity to leakage currents. Low swing dynamic differential logic styles are a restricted version of this type of logic and thus can show the same behaviour in harsh environments or in deep submicron technologies with high leakage currents. Regarding their robustness to voltage scaling, LSCML, IFLSCML and DDSLL can operate reliably at V_{dd} as low as 0.8V. Their behaviour was also observed with a mismatch on (W/L) up to 10% in the differential pair, a negligible variation of electrical characteristics was observed in this case.

The robustness of the ULPFA to voltage scaling has also been discussed. Despite a significant speed degradation of the sum signal, it still operates correctly at voltages as low as 0.6V. Indeed, since the sum block is based on pass-transistors, it suffers from a speed impairing when the ratio $\frac{V_{dd}}{V_t}$ is less than three.

References

- [1] F. Mace, F-X. Standaert, I. Hassoune, J-D. Legat, and J-J. Quisquater, “A dynamic current mode logic to counteract power analysis attacks,” in *Proceedings of DCIS conference*, Bordeaux, France, 2004, pp. 186–191.
- [2] 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,” *Accepted for publication in the Microelectronics Journal*, 2006.
- [3] P. Parra, A. Acosta, and M. Valencia, “Reduction of switching noise in digital CMOS circuits by pin swapping of library cells,” in *Proceedings of PATMOS conference*, 2001.
- [4] S. Zhao and K. Roy, “Estimation of switching noise on power supply lines in deep sub-micron CMOS circuits,” in *13th International conference on VLSI design*, 2000.
- [5] A. Neve, *Top-Down optimization for High-performance and low-power adders in deep-submicron SOI*, PhD thesis, UCL, Belgium, January 2004.

CHAPTER 6. CONCLUSIONS

APPENDIX A LSCML CIRCUITS EVALUATION: TRANSISTOR SIZES

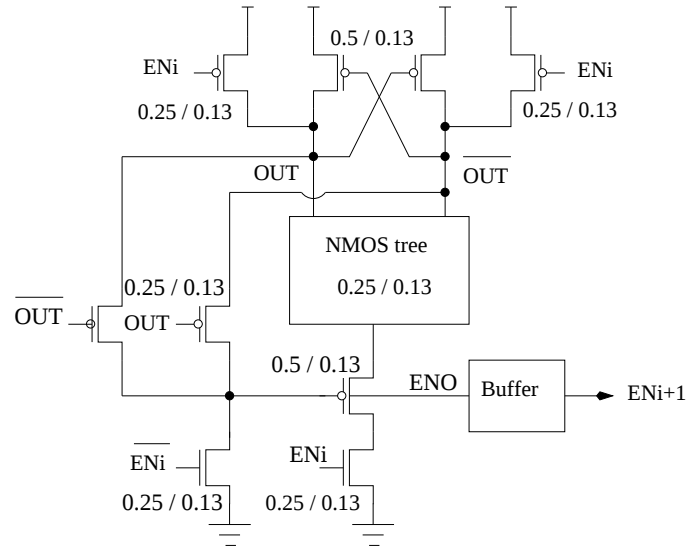


Figure A.1: LSCML gate.

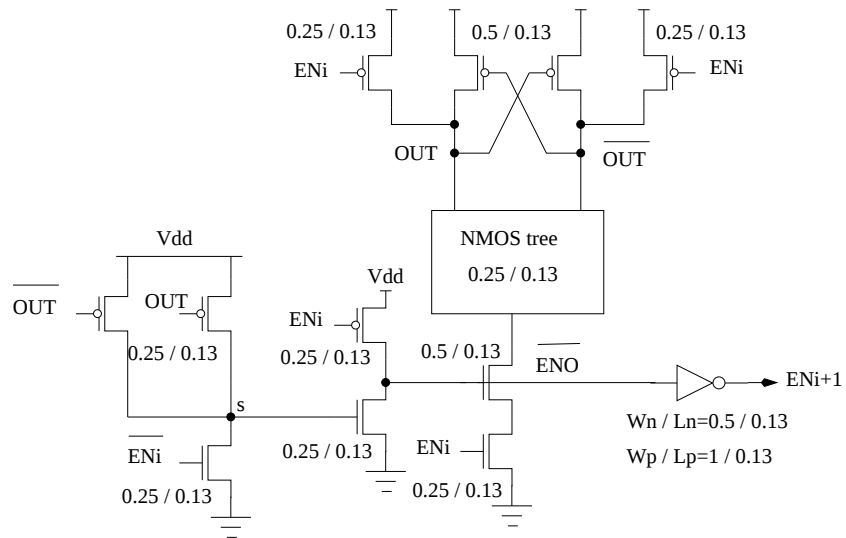


Figure A.2: DDSLL gate.

APPENDIX A. LSCML CIRCUITS EVALUATION: TRANSISTOR SIZES

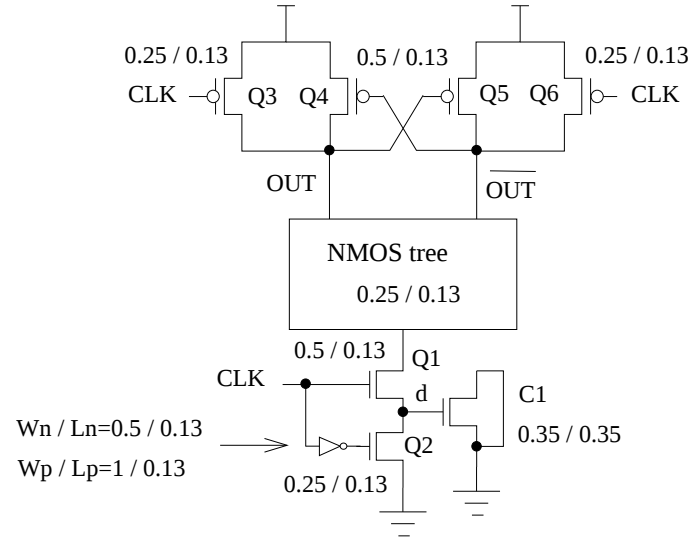


Figure A.3: DyCML gate.

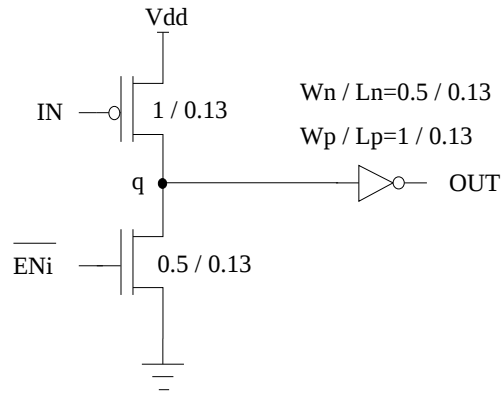


Figure A.4: Single ended buffer.

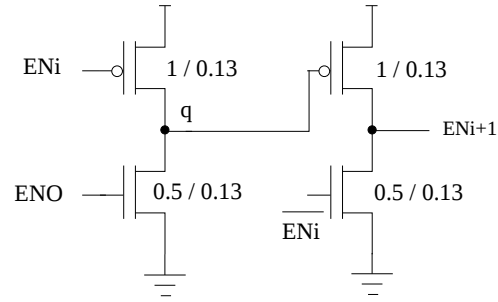


Figure A.5: Self-timing buffer (ST1 scheme in LSCML cascaded gates).

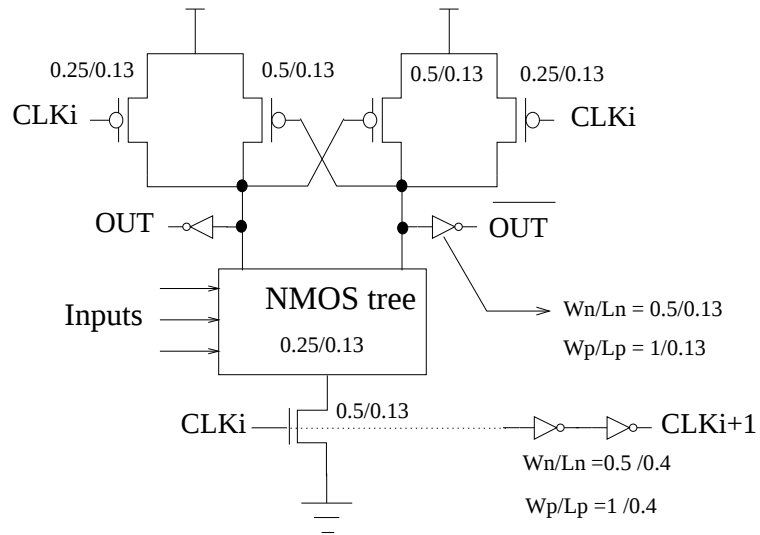


Figure A.6: DDCVSL gate with clock delay scheme (DDCVSL(CD)).

APPENDIX A. LSCML CIRCUITS EVALUATION: TRANSISTOR SIZES

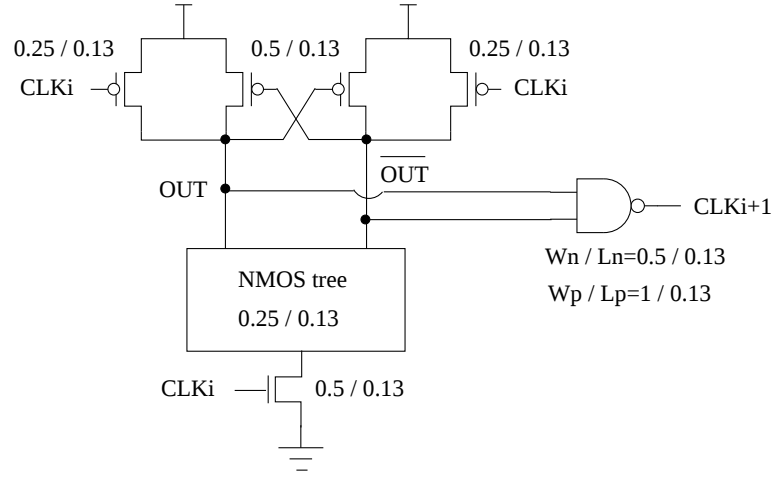


Figure A.7: DDCVSL gate with self-timing scheme (DDCVSL(ST)).

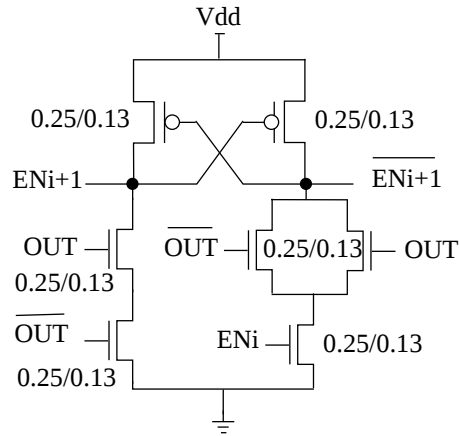
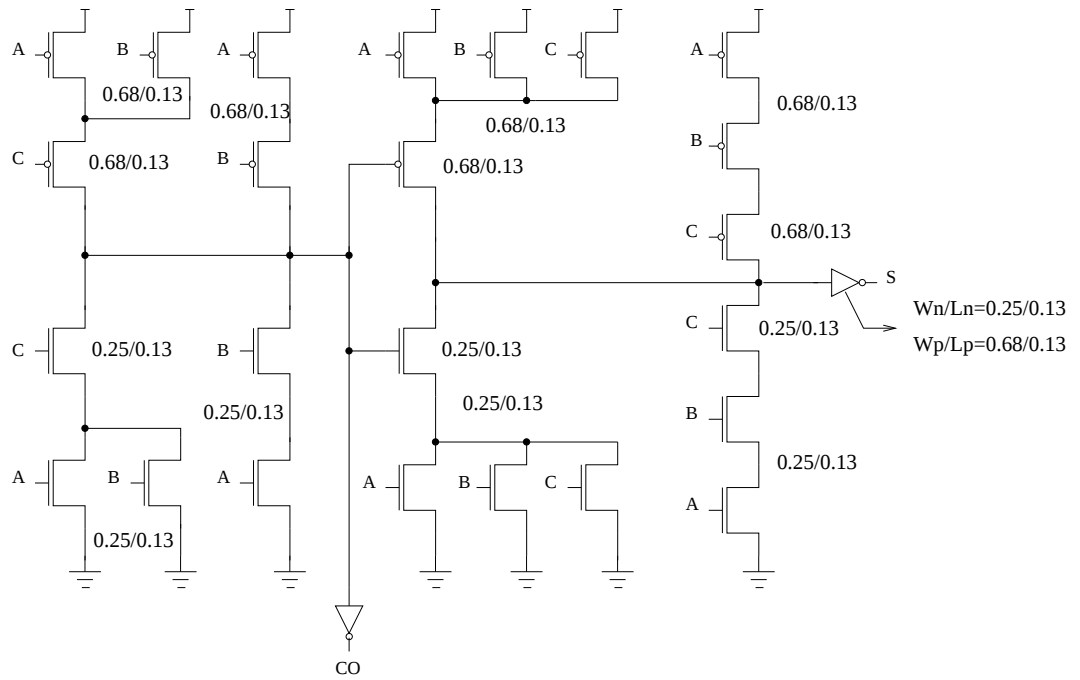


Figure A.8: ST2 self-timing buffer.

APPENDIX B HYBRID FULL-ADDER EVALUATION: TRANSISTOR SIZES



APPENDIX B. HYBRID FULL-ADDER EVALUATION: TRANSISTOR SIZES

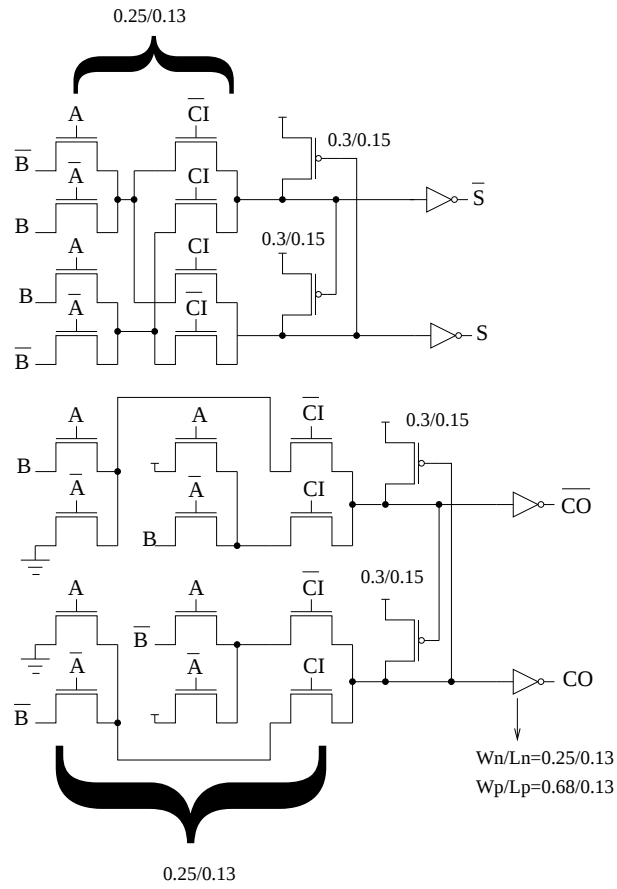


Figure B.2: CPL full adder.

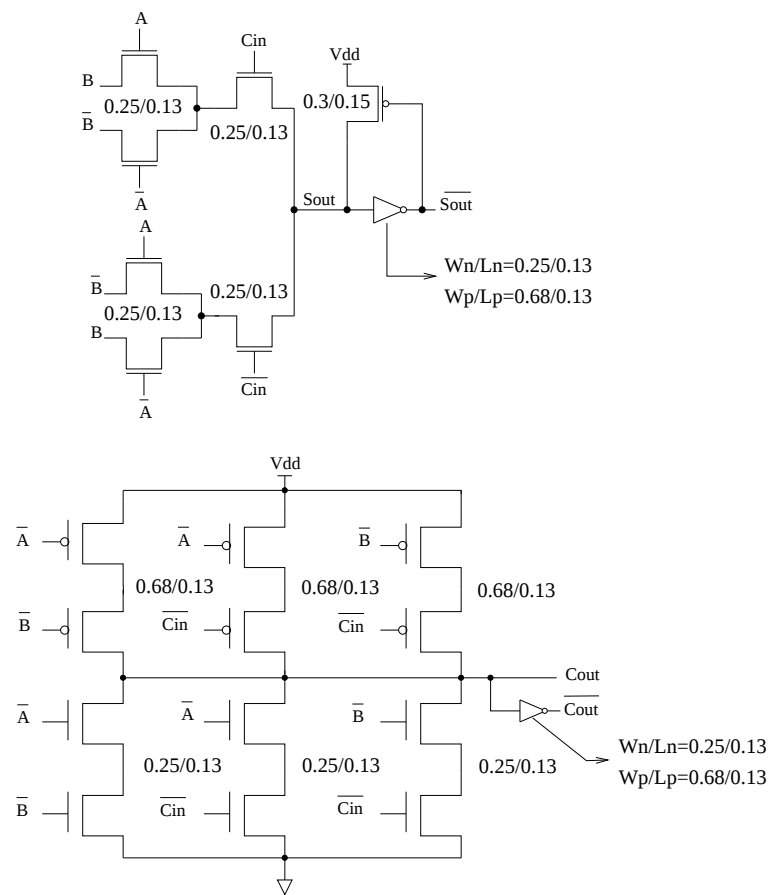


Figure B.3: BBL-PT full adder.

APPENDIX B. HYBRID FULL-ADDER EVALUATION: TRANSISTOR SIZES

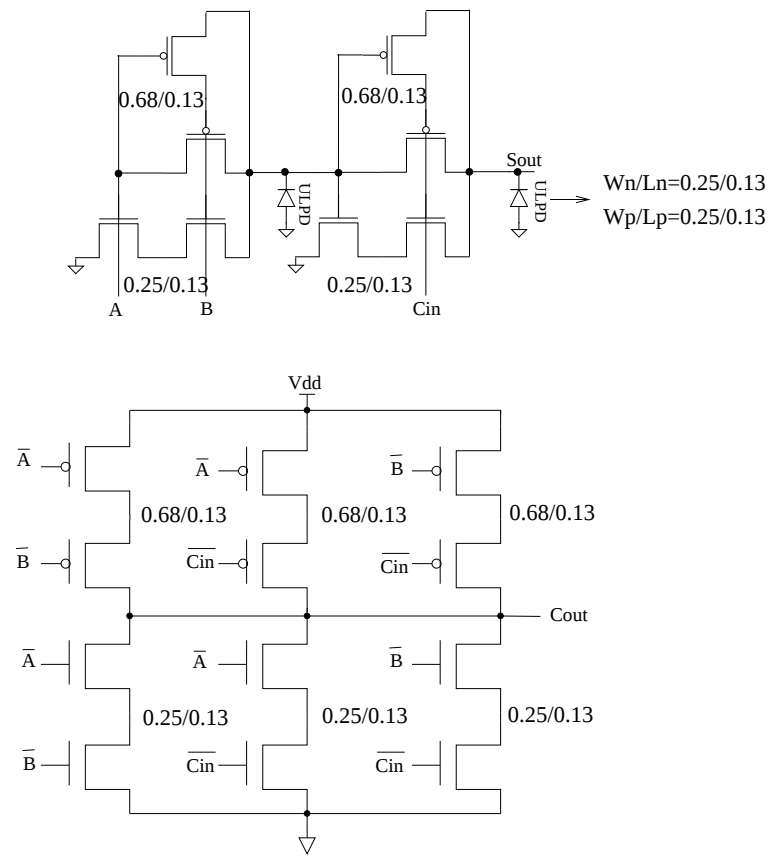


Figure B.4: ULPFA full adder.

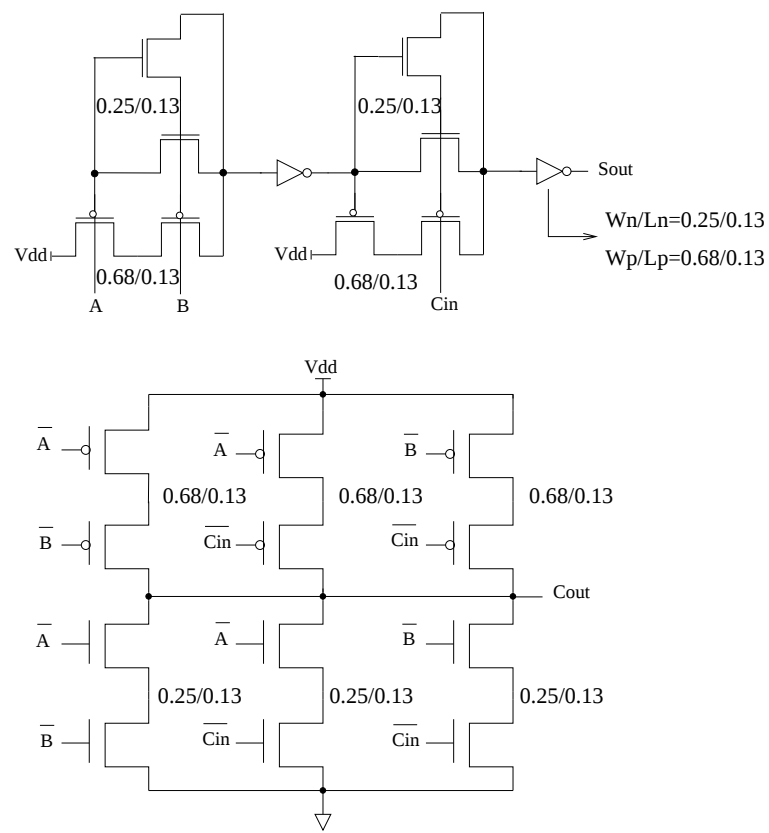


Figure B.5: Hybrid full adder.

APPENDIX B. HYBRID FULL-ADDER EVALUATION: TRANSISTOR SIZES

APPENDIX C

4 BIT BRANCH BASED CLA SYNTHESIS

The equations that give the outgoing carries in the 4b CLA adder having as inputs the two operands $A_3...A_0$ and $B_3...B_0$ and the incoming carry-
in Cin , described in section 4.8, are expressed by:

$$C_0 = G_0 + P_0 Cin \quad (C.1)$$

$$C_1 = G_1 + G_0 P_1 + Cin P_0 P_1 \quad (C.2)$$

$$C_2 = G_2 + G_1 P_2 + G_0 P_1 P_2 + Cin P_0 P_1 P_2 \quad (C.3)$$

The outgoing carry C_3 can be computed as a group carry output as described in section 4.8:

$$C_3 = G_1^* + P_1^* Cin \quad (C.4)$$

where P_i and G_i are respectively the propagation and the generation of the carry. They are given by:

$$P_i = A_i \oplus B_i$$

$$G_i = A_i \cdot B_i$$

P_1^* and G_1^* are respectively the group propagation and generation of the carry.

In order to obtain the P_i functions, the XOR implementation as shown in Figure C.7 can be used (a synthesis of XOR with branch based logic gives the same implementation in branch as in static CMOS), while a static CMOS AND gate can be used to generate the G_i functions.

We synthesized the outgoing carries of the 4b CLA C_0 , C_1 , C_2 and C_3 with branch-based logic. Therefrom, the obtained equations of each logic function and its corresponding schematic are summarized below:

- Carry out C_0

$$C_{0N} = \overline{Cin} \cdot \overline{G_0} + \overline{P_0} \cdot \overline{G_0}$$

$$C_{0P} = \overline{G_0} + \overline{Cin} \cdot \overline{P_0}$$

The corresponding schematic is shown in Figure C.1.

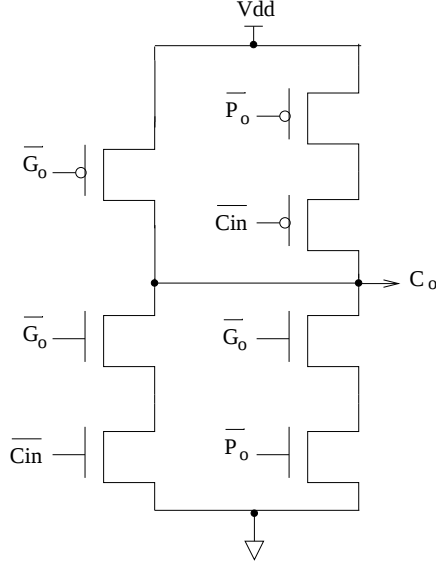


Figure C.1: Branch-based implementation of the carry function C_0 .

- Carry out C_1

$$\begin{aligned} C_{1N} &= \overline{G_1} \cdot \overline{P_1} + \overline{P_0} \cdot \overline{G_0} \cdot \overline{G_1} \cdot Cin + \overline{G_0} \cdot \overline{G_1} \cdot \overline{Cin} \\ C_{1P} &= \overline{G_1} + \overline{G_1} \cdot \overline{P_1} \cdot Cin + \overline{P_0} \cdot \overline{P_1} \cdot \overline{Cin} + \overline{G_0} \cdot \overline{P_1} \end{aligned}$$

Implementation of the C_1 carry out function is shown in Figure C.2. This implementation uses a serial connection of four transistors, while the branch-based design recommends a series connection with a maximum of three transistors per branch. We believe that delay of series-connected MOSFETs in short channel devices is not as bad as in long channel devices. Indeed, let us consider the equivalent RC model (see Figure C.3) of four series-connected MOSFETs, where each transistor is modeled as a resistor in series with an ideal

switch. C_1 , C_2 and C_3 depict the internal node capacitances due to source/drain regions and C_{load} is the load capacitance at the output node.

The propagation delay can be computed with the Elmore delay

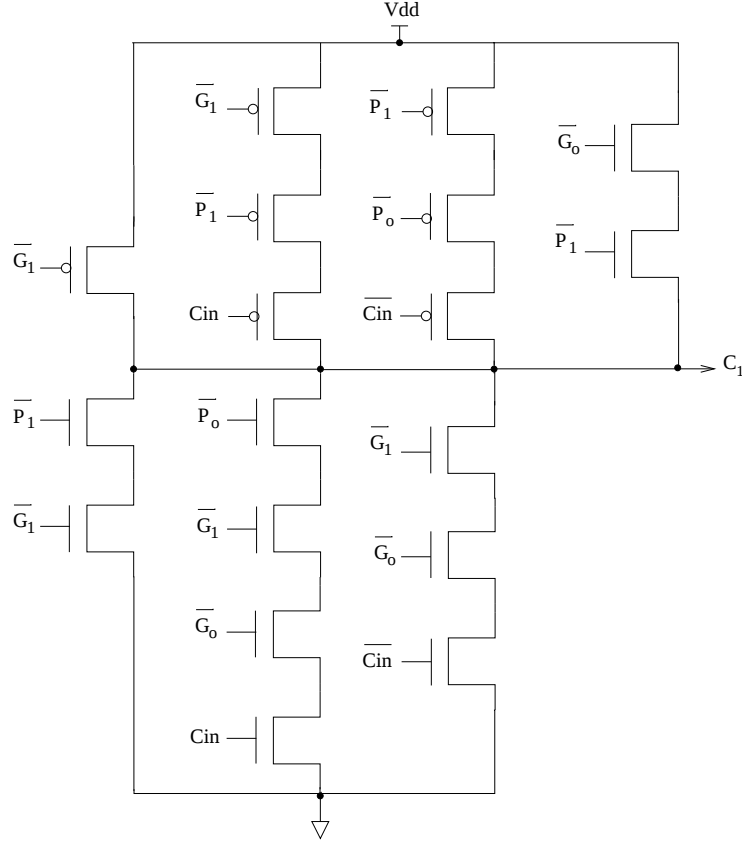


Figure C.2: Branch-based implementation of the carry function C_1 .

model:

$$t_d = 0.69 \cdot (R_1 C_1 + (R_1 + R_2) C_2 + (R_1 + R_2 + R_3) C_3 + (R_1 + R_2 + R_3 + R_4) C_{load})$$

Assuming that the four transistors have equal sizes (i.e. $R_1 = R_2 = R_3 = R_4 = R$ and $C_1 = C_2 = C_3 = C$), expression of t_d can be simplified to:

$$t_d = 0.69 \cdot R \cdot (6C + 4C_{load})$$

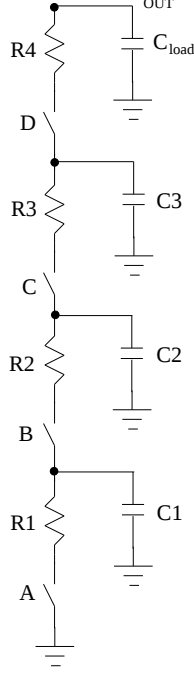


Figure C.3: RC equivalent model of four series-connected MOSFETs.

Since for nano-devices, R and C are being decreasing, we can then assume that the delay of four series-connected MOSFETs should not be so bad in comparison to the delay of three devices (maximum) per stack recommended in branch-based logic, in deep submicron technologies.

- Carry out C_2

Regarding the carry out C_2 , we firstly decomposed the equation C.3 into two components:

$$C_2 = C'_2 + C''_2 \quad (\text{C.5})$$

where the components C'_2 and C''_2 are:

$$\begin{aligned} C'_2 &= G_2 + G_1 \cdot P_2 + G_0 \cdot P_1 \cdot P_2 \\ C''_2 &= C_{in} \cdot P_0 \cdot P_1 \cdot P_2 \end{aligned}$$

C'_2 can be synthesized in the same way as for C_1 function. This gives

the schematic shown in Figure C.4.

C_2'' can simply be implemented with a static CMOS AND, while

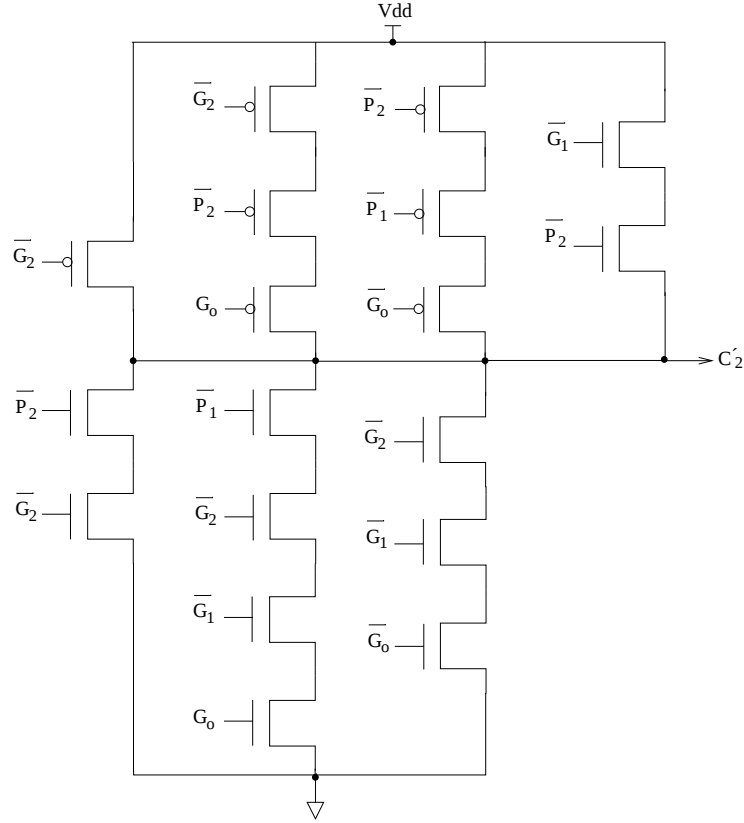


Figure C.4: Branch-based implementation of the component C_2' of the carry function C_2 .

the complete function C_2 can be obtained by an “ORing” of C_2' and C_2'' with a static CMOS gate.

- Carry out C_3

According to equation C.4, the implementation of the carry out C_3 requires G_1^* and P_1^* functions. These are synthesized as follows:

- G_1^* synthesis

Firstly, we remind the expression of G_1^* , which is given by:

$$G_1^* = G_3 + G_2 \cdot P_3 + G_1 \cdot P_2 \cdot P_3 + G_0 \cdot P_1 \cdot P_2 \cdot P_3 \quad (\text{C.6})$$

G_1^* can be decomposed into two components: $G_1^* = G_1'^* + G_1''^*$, where:

$$\begin{aligned} G_1'^* &= G_3 + G_2 \cdot P_3 + G_1 \cdot P_2 \cdot P_3 \\ G_1''^* &= G_0 \cdot P_1 \cdot P_2 \cdot P_3 \end{aligned}$$

$G_1'^*$ can be synthesized in the same way as for the carry out C_1 . This gives the schematic shown in Figure C.5. The $G_1''^*$ function can simply be implemented by a static CMOS AND. Finally, the complete function G_1^* can be implemented by “ORing” $G_1'^*$ and $G_1''^*$ through a static CMOS gate.

– P_1^* function

The group propagation of the carry P_1^* given in equation C.7, can also be implemented by a static CMOS AND gate.

$$P_1^* = P_0 \cdot P_1 \cdot P_2 \cdot P_3 \quad (\text{C.7})$$

Now since the functions P_1^* and G_1^* are implemented, the carry out C_3 can be obtained. This is achieved through the same synthesis as for carry out C_0 . This gives the schematic shown in Figure C.6.

The sum functions can be obtained through the XOR gate shown in Figure C.7 since they are given by:

$$S_i = P_i \oplus C_{i-1} \quad (\text{C.8})$$

Simulations of the 4b BBL CLA as described here, have shown a correct functionality. It can be used as a basic element to implement n-bit CLA adders.

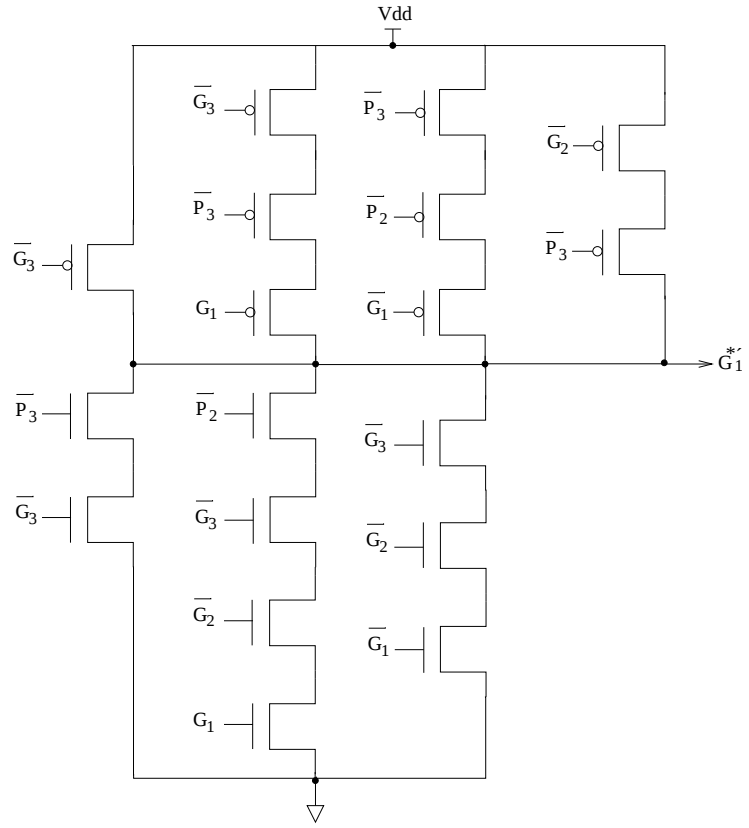


Figure C.5: Branch-based implementation of the component G'_1 of the group function G_1 .

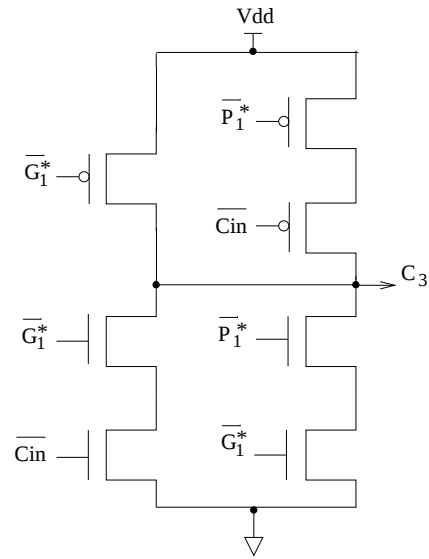


Figure C.6: Branch-based implementation of the carry function C_3 .

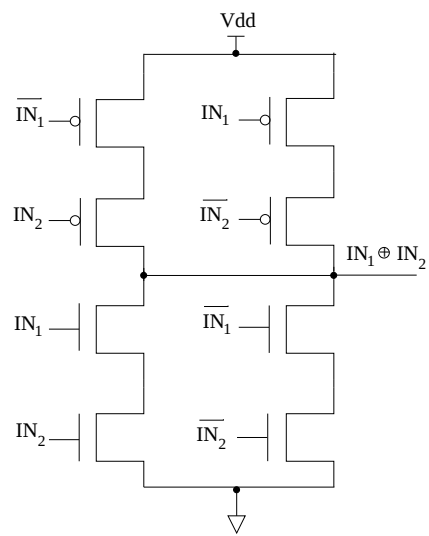


Figure C.7: XOR gate.

