



# **Thèse**

**présenté pour obtenir le grade de docteur**

**de l'Ecole Nationale Supérieure des Télécommunications**

**Spécialité: Electronique et Communications**

**Présenté par :**

**Andrés David García García**

**Titre :**

**Etude sur l'estimation et l'optimisation de la consommation de puissance des circuits logiques programmables du type FPGA**

**Directeurs de thèse :**

**Wayne P. Burleson et Jean Luc Danger**

**Jury :**

Jean Didier LEGAT	Rapporteur
Amara AMARA	Rapporteur
Christian PIGUET	Examineur
Russell TESSIER	Examineur
Jean Michel VUILLAMY	Examineur
Jacky RENAUX	Examineur

## Acknowledgements

This manuscript represents not only the end of my studies and the beginning of my professional life as a future research-professor; it is the end also of the five years that I spent in France learning about others cultures. Many persons make it possible and I want to thank all of them.

First of all, I would like to thank Nicolas Demassieux and Arnaud Galisson who accepted my candidature and give me the opportunity to continue my studies in France.

I would also like to thank the staffs of the electronics department, especially all the members of the digital systems design group. Lirida Naviner de Barros who has spent some time to review my articles, Willy Gane for all the technical support; and specially thanks to Jean Leroux les Jardins who show me more about French culture. I extend my thanks to the other Ph.D. students for their encouragement, support and friendship.

I would like to acknowledge all the members of the committee, Jean Didier Legat, Amara Amara, Christian Piguet, Russel Tessier, Jean Michel Vuillamy and Jacky Renaux, for their interest and patience for reading this manuscript and their remarks during the oral presentation. I extend this acknowledge to my English teacher Mr. Peter Weyer Brown who has reviewed the five chapters of this manuscript.

I am grateful to my advisor Wayne Burleson, associated professor of the UMASS for having proposed this subject, for advice me and encourage me all the time even if sometimes it was hard for him to help me by e-mail.

I am especially grateful to my co-advisor Jean Luc Danger, head of the digital systems design group, for his encouragement, his patience and support during all the phases of this work. This work would not have finished without his critics, his corrections and his help.

Thank to my family, to my brother who has encouraged me to continue my studies, to my sisters and friends for their support. Specially thanks to my friend Luis Gonzalez for his friendship and his encouragement.

I would like to thank most of all to my beloved wife, Edith, who has been my support and my motivation, without her love, encouragement and patience, I would not have accomplished this project.

Finally, few years ago, this project was only a dream of two persons. This dream now is becoming true. I want to dedicate this thesis to my parents, to the memory of my father and the courage of my mother.

Thanks all.

This work was supported by the Mexican council of science and technology CONACyT (Consejo Nacional de Ciencia y Tecnología); the French foundation SFERE (Société Française d'Exportation de Ressources Educatives) that belongs to the French Secretary of Foreign Affairs; the French society ARECOM (Association pour la Recherche et l'Enseignement en Communications); and the ITESM (Instituto Tecnológico y de Estudios Superiores de Monterrey).

Ce travail de thèse de docteur est la fin d'un rêve qui a été partagé par plusieurs personnes dont je tiens à remercier énormément, sans leur présence dans ma vie, cette réussite n'aurait jamais arrivée. Je tiens à leur dire que c'est aussi sa réussite, encore merci d'avoir être avec moi.

A mon frère Eduardo,

A mes sœurs Claudia et Lilia,

A mes parents José de Jesús et Lilia Griselda,

A ma très chère et aimée femme Edith.



# Résumé

## ***Introduction***

Diminuer la puissance consommée par les circuits intégrés devient une des priorités majeures des concepteurs de systèmes électroniques numériques. Ce facteur est d'autant plus important que le marché des applications portables est en plein essor (exemples : ordinateurs portables, téléphones mobiles, calculatrices, etc.). Les dernières technologies électroniques offrent des transistors de plus en plus petits et nécessitent une tension d'alimentation de plus en plus faible, ce qui va tout à fait dans le sens de la réduction de la consommation. Le problème vient du fait que les applications sont de plus en plus gourmandes en puissance de calcul et en fonctionnalités, engendrant des tailles de circuit et des fréquences de fonctionnement très élevées. Malgré les progrès technologiques, la tendance générale est donc vers une augmentation de la puissance consommée des circuits. Afin de réduire la consommation, des méthodes au niveau algorithmique et architectural des systèmes doivent être abordées.

Les circuits du type FPGA ("Field Programmable Gate Array") sont de plus en plus utilisés pour le prototypage ou la fabrication en petites séries de systèmes électroniques numériques. Leur principal avantage est avant tout leur programmabilité. Les dernières familles de FPGA ont atteint des niveaux de complexité et performances tels qu'il devient possible de valider rapidement un système électronique numérique intégrant quelques millions de portes et fonctionnant à quelques dizaines de MHz. Pour donner un exemple, les plus gros circuits FPGA en septembre 2000 intégraient plus de deux millions de portes logiques équivalentes et plus de 420 kilo bits de mémoire embarquée.

Les fabricants de ces circuits n'ont pas porté leurs efforts pour améliorer la consommation de puissance de leurs composants. Ceux-ci sont généralement perçus comme des éléments "consommants", ce qui est en sorte le prix à payer pour leur vertu de "programmabilité". Les équations fournies par les constructeurs pour calculer la consommation de puissance ne sont que des estimations comprenant des facteurs difficiles à déterminer.

Le but de ce travail a été d'élaborer un modèle de consommation de puissance propre aux FPGAs et de proposer des méthodes d'optimisation au niveau circuit et système. Dans un premier temps des mesures exhaustives ont permis de construire un modèle précis de distribution de la consommation de puissance. Les familles des circuits FPGA utilisées afin de réaliser cette étude sont la FLEX10K Altera™ et la XC4000E Xilinx™.

Dans un deuxième temps une méthode d'optimisation de la consommation a été proposée. Cette méthode, classique par son approche visant à diminuer la tension d'alimentation, s'est révélée particulièrement efficace pour les FPGAs qui ont un comportement en puissance qui dépend non seulement du carré mais aussi du cube de la tension d'alimentation. Elle a montré qu'il était possible de réduire la consommation tout en gardant un très bon niveau de performances sans augmenter notablement le nombre de portes. Cette technique repose sur l'implémentation des architectures du type "pipeline" avec une faible tension d'alimentation, elle a été mise en marche avec l'utilisation de deux sources d'alimentation, une source d'alimentation fixe pour les cellules d'entrée-sortie, et une source d'alimentation variable pour "sous alimenter" la logique interne du circuit. Celui-ci permettra d'utiliser les cellules d'entrée-sortie comme interface entre la logique interne "sous alimentée" et des autres composants externes fonctionnant à une tension d'alimentation "normale".

### ***Le travail effectué***

La consommation de puissance d'un FLEX10K100 et d'un XC4010E a été mesurée en utilisant une méthodologie de test permettant d'obtenir la consommation de chacun des sous éléments internes.

Cette méthodologie consiste à mesurer le courant d'un FPGA en utilisant des architectures très simples avec un taux d'activité connu, une fréquence d'horloge constante et la tension d'alimentation fixée à 5 Volts. Lorsque l'on mesure la consommation d'un élément, la consommation des autres reste constante.

Cette technique "incrémentale" permet d'identifier la contribution de courant, autrement dit, la puissance consommée pour chacun des sous éléments. Nous avons identifié cinq éléments principaux :

1. Les éléments d'interconnexion internes.
2. Les cellules logiques.
3. Les cellules d'entrée/sortie.
4. L'arbre d'horloge.
5. Les cellules de mémoire embarquée ou distribuée.

Enfin, une table de distribution comprenant les cinq éléments a été construite. Ce modèle de distribution de la consommation de puissance pour les familles Flex10K et XC4000E a été validé avec une précision variant entre 92 et 98 %.

Le modèle obtenu, appelé "modèle ENST", a été comparé avec les modèles proposés par les fabricants des FPGA ainsi qu'avec un modèle proposé par le groupe de recherche sur les applications sans fils de l'université de Californie à Berkeley. Les résultats montrent qu'une grande partie de la consommation est dissipée par les cellules logiques et les ressources d'interconnexion. Enfin, le travail de modélisation a mis en évidence la forte consommation des FPGAs ainsi que les imprécisions des modèles proposés par les fabricants des FPGA. La suite du travail porte principalement sur les méthodes d'optimisation de la consommation. Basée sur une méthode architecturale proposée par Chandrakasan (1992) [23] pour réduire la consommation de puissance des circuits ASICs, une technique d'optimisation de la consommation de puissance est proposée dans le chapitre 4. Dans son travail, Chandrakasan a montré que l'on peut réduire la consommation de puissance des circuits CMOS en utilisant des architectures du type pipeline avec une faible tension d'alimentation. Cette technique est avantageuse avec les FPGA grâce à deux facteurs :

1. D'abord, les architectures FPGA ne sont pas construites à base de circuits CMOS purs. Les FPGA comprennent différents types d'éléments tels que des portes CMOS, "Pass-Transistors" et cellules de mémoire RAM. Ceux-ci font que le comportement de la consommation de puissance des FPGA obéit à une équation polynomiale de 3<sup>ème</sup> degré.
2. Une façon efficace de diminuer la consommation de n'importe quel circuit est de réduire la tension d'alimentation, cette technique devient particulièrement intéressante pour les FPGA car l'estimation de la consommation par rapport à la tension d'alimentation est un polynôme de degré plus élevé que pour les circuits entièrement CMOS.

Chaque cellule logique FPGA comprend au moins une bascule programmable du type D. Pour compenser la baisse de la tension d'alimentation qui augmente les temps de propagation, des étages de "pipeline" peuvent être rajoutés. L'utilisation des nouvelles bascules D, lorsque l'on rajoute des barrières supplémentaires de pipeline, ajoute peu de cellules logiques et de lignes d'interconnexion. Donc, la consommation due aux cellules d'entrée-sortie et aux ressources d'interconnexion ne va pas augmenter, seulement le pourcentage de la consommation correspondant aux cellules logiques va changer.

Cette technique est vérifiée en utilisant deux sources de tension, une source de tension fixe à 5 volts ou 3.3 volts pour les cellules d'entrée-sortie et une source variable pour la logique interne (ou "core"). La tension d'alimentation du "core" prend des valeurs à partir du 5 volts ou 3,3 volts, jusqu'à la valeur de tension minimale supportée par le circuit. Ensuite, des barrières de pipeline seront ajoutées à l'architecture afin de retrouver le niveau de performance originale (c'est-à-dire, quand le circuit était alimenté à 5 volts et sans aucun étage supplémentaire de pipeline).

## ***Conclusions***

Dans ce travail, un modèle de distribution de la consommation de puissance propre aux circuits FPGA est proposé. Ce modèle montre la distribution de la consommation à l'intérieur du circuit ainsi que le pourcentage de puissance consommée par chaque élément interne du FPGA. Les résultats obtenus par ce modèle permettent d'acquérir une connaissance très utile sur la consommation de puissance des FPGA ainsi que d'estimer d'une façon plus précise la consommation de ces circuits.

Les mesures réalisées ainsi que les analyses effectuées lors de l'intégration de l'état de l'art ont montré que, d'une façon contraire à un circuit CMOS pur, le comportement de la consommation de puissance des FPGA peut être représenté par une équation polynomiale de degré 3.

Ces résultats ont motivé la conception d'une technique pour la réduction de la consommation de puissance basée sur l'implémentation des architectures du type pipeline avec une faible tension d'alimentation. Cette technique permet de réduire la consommation de puissance de plus de 75%.

Finalement, cette technique avec l'utilisation de deux sources de tension (l'une pour les cellules de E/S et l'autre pour la logique interne) permet de réduire d'une façon efficace la consommation de puissance des FPGA. L'utilisation de deux sources de tension permet l'échange des signaux logiques entre le "core" sous alimenté et les éléments externes. Les résultats des mesures en utilisant une double source d'alimentation et des architectures "pipeline" ont montré qu'il est possible de réduire d'une façon efficace la consommation de puissance d'un circuit FPGA commercial. L'intérêt de cette technique est aussi de ne perdre ni le niveau de performance original de l'application, ni la possibilité d'échanger des signaux logiques avec des autres composants.

### ***Suite du travail***

Les résultats présentés dans ce travail ont été obtenus en utilisant des circuits à 5 volts et à 3.3 volts. Les familles plus récentes des FPGA utilisent des sources d'alimentation à 2.5 volts et même à 1.8 volts. L'un des axes de recherche à développer pourrait porter sur la réalisation des mesures exhaustives de courant en utilisant les nouvelles familles Apex™ et Virtex™ afin de construire un modèle de distribution de la consommation de puissance.

Ensuite, la technique proposée dans ce travail pour la faible consommation devrait être applicable en utilisant les dernières familles des FPGA. Bien que ces familles utilisent déjà une faible tension d'alimentation.

Les résultats pourraient être très intéressants quand la tension d'alimentation ( $V_{DD}$ ) se rapproche du double de la tension de seuil ( $V_T$ ). Ce qui pourrait avoir en conséquence une réduction dramatique des courants de court circuit.

Finalement, des accords de coopération scientifique avec les principaux fabricants des circuits FPGA (Altera™ et Xilinx™) pourraient être mis en place afin de réaliser des simulations avec les modèles SPICE pour raffiner le modèle de consommation et proposer des améliorations aux architectures FPGA pour la faible consommation.



## **Abstract:**

The use of PLDs and specifically Field Programmable Gate Arrays (FPGAs) has been increasing largely due to the ability to rapidly develop prototypes with reduced development times and costs. In addition, they have opened the possibility of new dynamically reconfigurable applications. Meanwhile, power consumption and dissipation are becoming an increasingly important factor in VLSI and system design because of the increase in portable battery-powered applications.

Although some support for power modelling is provided by FPGA vendors, in this work we present a power consumption model for FPGAs based on measurements. This model permits us to optimize power consumption on FPGAs existing architectures, as well as helping direct the design of new power-sensitive FPGA architectures. Based on this model, an architectural technique coupled with the use of low supply voltages is proposed to reduce power consumption in commercial FPGAs while keeping an acceptable performance level. Results explained by our model and our measurements refute several commonly held beliefs about the consumption of power in FPGA architectures and also lend insight into possible optimization and architectures for future power-sensitive FPGA architectures.



# Table of Contents

<b>ACKNOWLEDGEMENTS .....</b>	<b>0</b>
<b>RÉSUMÉ .....</b>	<b>5</b>
INTRODUCTION .....	5
LE TRAVAIL EFFECTUÉ.....	6
CONCLUSIONS.....	8
SUITE DU TRAVAIL.....	9
<b>ABSTRACT: .....</b>	<b>11</b>
<b>TABLE OF CONTENTS .....</b>	<b>13</b>
<b>LIST OF ABBREVIATIONS AND SYMBOLS .....</b>	<b>17</b>
<b>LIST OF FIGURES.....</b>	<b>21</b>
<b>LIST OF TABLES.....</b>	<b>25</b>
<b>CHAPTER I. INTRODUCTION.....</b>	<b>27</b>
1.1 REDUCING POWER CONSUMPTION: A GROWING CHALLENGE.....	28
1.2 FIELD PROGRAMMABLE GATE ARRAYS.....	30
1.2.1 <i>Introduction</i> .....	30
1.2.2 <i>Programmable Logic Devices (PLDs)</i> .....	30
1.2.3 <i>FPGA architecture</i> .....	33
1.2.4 <i>FPGA and Application-Specific Integrated Circuit (ASIC)</i> .....	34
1.3 POWER CONSUMPTION IN FPGAS.....	35
1.4 DISSERTATION OUTLINE.....	36
<b>CHAPTER II. STATE OF ART .....</b>	<b>39</b>
2.1 COMMERCIALY AVAILABLE FPGAS.....	39
2.1.1 <i>Introduction</i> .....	39
2.1.2 <i>FPGA basis</i> .....	39
2.1.3 <i>Currently available FPGAs Technology</i> .....	42
2.2 POWER CONSUMPTION MODEL OF MOS-BASED CIRCUITS.....	59
2.2.1 <i>Introduction</i> .....	59
2.2.2 <i>Power Consumption of Complementary CMOS</i> .....	62
2.2.3 <i>Power Consumption of Pass-Transistor Structures</i> .....	68

2.2.4	<i>Power Consumption of SRAM</i> .....	73
2.2.5	<i>Power Consumption of Input/Output Circuits</i> .....	74
2.2.6	<i>Power Consumption in Clock Circuits</i> .....	78
2.3	POWER CONSUMPTION IN SRAM-BASED FPGAS .....	79
2.4	RELATED WORKS ON POWER CONSUMPTION IN FPGAS .....	80
2.5	SUMMARY .....	82
<b>CHAPTER III. POWER MODELING ON FPGAS</b> .....		<b>85</b>
3.1	INTRODUCTION .....	85
3.2	POWER CONSUMPTION MODEL FROM VENDORS .....	85
3.2.1	<i>Altera™</i> .....	86
3.2.2	<i>Atmel™</i> .....	88
3.2.3	<i>Xilinx™</i> .....	90
3.2.4	<i>Summary</i> .....	91
3.3	INCREMENTAL METHODOLOGY OF POWER MEASUREMENT .....	92
3.3.1	<i>Introduction</i> .....	92
3.3.2	<i>Test Platform</i> .....	94
3.3.3	<i>Measurement Methodology</i> .....	95
3.4	MEASUREMENTS AND RESULTS .....	100
3.4.1	<i>Interconnect resources</i> .....	100
3.4.2	<i>Logic Cells</i> .....	102
3.4.3	<i>I/O Cells</i> .....	103
3.4.4	<i>Clock Tree</i> .....	104
3.4.5	<i>Memory Cells</i> .....	106
3.5	POWER CONSUMPTION MODEL OF COMMERCIAL FPGAS .....	107
3.5.1	<i>Power Consumption Model based on measurements</i> .....	107
3.5.2	<i>Other Power Consumption Models</i> .....	110
3.5.3	<i>Comparisons between all Power Consumption Models</i> .....	113
3.6	SUMMARY .....	115
<b>CHAPTER IV. OPTIMIZING POWER CONSUMPTION IN FPGAS</b> .....		<b>117</b>
4.1	INTRODUCTION .....	117
4.2	FREQUENCY VS POWER CONSUMPTION .....	118
4.3	POWER OPTIMIZATION .....	121
4.3.1	<i>Results from ASICs</i> .....	122
4.3.2	<i>Dynamic power distribution in FPGAs</i> .....	122
4.3.3	<i>Architectural Optimization in FPGAs</i> .....	124
4.4	RESULTS .....	125
4.4.1	<i>Pipeline Coupled with Single Supply Voltage</i> .....	125
4.4.2	<i>Pipeline Coupled with Double Supply voltage</i> .....	129

4.5 SUMMARY.....	135
<b>CHAPTER V. CONCLUSIONS.....</b>	<b>137</b>
5.1 CONCLUSIONS.....	137
5.2 CONTRIBUTIONS .....	139
5.3 FUTURE WORK .....	140
<b>GLOSSARY .....</b>	<b>143</b>
<b>REFERENCES.....</b>	<b>147</b>
INTERNET:.....	151
<b>APPENDIX A: POWER ESTIMATION AND OPTIMIZATION .....</b>	<b>153</b>
POWER ESTIMATION : .....	153
POWER OPTIMIZATION : .....	155
A) CIRCUIT 1.....	155
B) CIRCUIT 2.....	160
POWER BEHAVIOR : .....	161
<b>APPENDIX B: SPICE MODEL OF A PASS-TRANSISTOR STRUCTURE .....</b>	<b>163</b>
A) SIMPLE CMOS INVERTER WITH PASS-TRANSISTOR.....	163
B) THREE CMOS INVERTERS.....	164
C) FIVE CMOS INVERTERS.....	164



## List of Abbreviations and Symbols

**A** : Ampere

**AC** : Alternating Current

**ASIC** : Application-Specific Integrated Circuit

**CAD** : Computer-Aided Design

**C<sub>d</sub>** : Depletion layer capacitance

**C<sub>L</sub>** : Load Capacitance

**CLB** : Configurable Logic Block (Xilinx™)

**CLK** : Clock

**CMOS** : Complementary Metal-Oxide Semiconductor

**CPLD** : Complex Programmable Logic Device

**C<sub>OX</sub>** : Gate oxide capacitance per unit area

**DC** : Direct Current

**DFF** : D-type Flip-Flop

**DPRAM** : Dual-Port Random-Access Memory

**DRAM** : Dynamic Random-Access Memory

**DSP** : Digital Signal Processing

**EAB** : Embedded Array Block (Altera™)

**ε<sub>o</sub>** : Oxide permittivity

**ENST** : Ecole Nationale Supérieure des Télécommunications

**EPLD** : Erasable Programmable Logic Device

**EPROM** : Erasable-Programmable Read-Only Memory

**F** : Frequency

**FIFO** : First-In First-Out

**FLEX** : Flexible Logic Element MatriX (Altera™)

**FPGA** : Field Programmable Gate Array (Xilinx™)

**IC** : Integrated Circuit

**I<sub>CL</sub>** : Load Capacitance Current

**I<sub>DP</sub>** : Direct-Paths Current

**I<sub>DS</sub>** : Drain to Source Current

**I<sub>Leak</sub>** : Leakage Current

**I<sub>SC</sub>** : Short-Circuit Current

**I/O** : Input/Output

**IOB** : Input/Output Block

**GaAs** : Gallium-Arsenide

**GND** : Ground

**HDL** : Hardware Description Language

**LAB** : Logic Array Block (Altera™)

**LB** : Logic Block

**LC** : Logic Cell (in general)

**LCELL** : Logic Cell (Altera™)

**LE** : Logic Element (Altera™)

**L<sub>eff</sub>** : The effective channel length

**LIFO** : Last-In First-Out

**LUT** : Look-Up Table

**mA** :  $1 \times 10^{-3}$  Amperes

**MOS** : Metal-Oxide Semiconductor

**mV** :  $1 \times 10^{-3}$  Volts

**mW** :  $1 \times 10^{-3}$  Watts

**NiCd** : Nickel-Cadmium

**Ni-MH** : Nickel-Metal Hydride

**NVL** : Non-Volatile Memory

**OMO** : Oxide-Nitride-Oxide

**OSC** : Oscillator

**P** : Power Consumption

**PAL** : Programmable Array Logic

**PLA** : Programmable Logic Array

**PLD** : Programmable Logic Device

**PLL** : Phase-Locked Loop

**Poly-Si** : Polysilicon

**PROM** : Programmable Read-Only memory

**PSM** : Programmable Switch Block (Xilinx™)

**q** : Electron charge

**RAM** : Random-Access Memory

**RE** : Read Enable

**ROM** : Read-Only Memory

**RST** : Reset

**S** : Sub-threshold swing parameter

**SRAM** : Static Random-Access memory

**T°C** : Temperature

**TTL** : Transistor-Transistor Logic

**t<sub>ox</sub>** : Gate oxide thickness

**τ** : The average of the rising and falling edges times (or propagation delay).

**μ** : Mobility of the electrons in the MOS transistor channel.

**V** : Volts

**VCC** : Supply Voltage

**VDD** : Drain to Drain Supply voltage

**V<sub>DS</sub>** : Drain to Source Voltage

**V<sub>GS</sub>** : Gate to Source Voltage

**V<sub>IN</sub>** : Input Voltage

**VHDL** : VHSIC Hardware Description Language

**VLSI** : Very Large Scalable Integration

**V<sub>OUT</sub>** : Output Voltage

**V<sub>T</sub>** : Threshold Voltage

**W** : Energy in Watts

**WE** : Write Enable

**W<sub>eff</sub>** : The effective channel width

## List of Figures

FIGURE 1.2.1. THE PROM ARCHITECTURE.....	31
FIGURE 1.2.2. FPLA ARCHITECTURE .....	31
FIGURE 1.2.3. PAL ARCHITECTURE .....	32
FIGURE 1.2.4. THE MACROCELL .....	33
FIGURE 1.2.5. CPLD ARCHITECTURE.....	33
FIGURE 1.2.6. FPGA INTERNAL ARCHITECTURE.....	34
FIGURE 1.2.7. ASICs VERSUS FPGAS .....	35
FIGURE 2.1.1. THE ANTI-FUSE SWITCH .....	43
FIGURE 2.1.2. THE AMORPHOUS-SILICON ANTI-FUSE.....	43
FIGURE 2.1.3. THE EPROM TRANSISTOR .....	44
FIGURE 2.1.4. THE EPROM CELL .....	45
FIGURE 2.1.5. SRAM CELLS WITH (A) PASS-TRANSISTOR, (B) TRANSMISSION GATE, (C) MULTIPLEXER .	47
FIGURE 2.1.6. CMOS SRAM CELL.....	47
FIGURE 2.1.7. HRL SRAM CELL.....	48
FIGURE 2.1.8. A 2-INPUT LUT .....	49
FIGURE 2.1.9. LOGIC CELL WITH A 4-INPUT LUT DEVELOPED BY ALTERA™(COURTESY OF ALTERA™ CO)	51
FIGURE 2.1.10 CONFIGURABLE LOGIC BLOCK DEVELOPED BY XILINX™(COURTESY OF XILINX™ CO) ....	51
FIGURE 2.1.11. MULTIPLEXER-BASED LOGIC CELL.....	51
FIGURE 2.1.12. MULTIPLEXER AND BASIC GATES LCELL PROPOSED BY ATMEL™(COURTESY OF ATMEL™ CORPORATION) .....	52
FIGURE 2.1.13. ACTEL™ ACT1 INTERCONNECT ARCHITECTURE (ROW-BASED).....	53
FIGURE 2.1.14. XC4000E/XL/XV INTERCONNECT ARCHITECTURE (SEGMENTED-BASED) (COURTESY OF XILINX™ CORPORATION).....	54
FIGURE 2.1.15 XC4000 SERIES INTERCONNECT RESOURCES (COURTESY OF XILINX™ CO).....	54
FIGURE 2.1.16. HIERARCHICAL INTERCONNECT (COURTESY OF ALTERA™ CORPORATION) .....	55
FIGURE 2.1.17. PROGRAMMABLE I/O CELL OF A FLEX 10K (COURTESY OF ALTERA™ CO).....	56
FIGURE 2.1.18. I/O CELL WITH TWO PROGRAMMABLE DFF (COURTESY OF ALTERA™ CO).....	57
FIGURE 2.1.19. EMBEDDED MEMORY (A) BLOCK. (B) DISTRIBUTED CELLS.....	58
FIGURE 2.2.1 STANDARD CMOS INVERTER .....	59
FIGURE 2.2.2. DC TRANSFER CHARACTERISTICS OF A CMOS INVERTER. ....	60
FIGURE 2.2.3. INPUT VOLTAGE AND SHORT-CIRCUIT CURRENT.....	66
FIGURE 2.2.4. THE NMOS PASS-TRANSISTOR .....	68
FIGURE 2.2.5. CHARGING CIRCUIT .....	69
FIGURE 2.2.6. DISCHARGE CIRCUIT.....	70
FIGURE 2.2.7 PASS-TRANSISTOR STRUCTURE .....	71

FIGURE 2.2.8. LEAKAGE CURRENT AND DIRECT-PATH CURRENT .....	71
FIGURE 2.2.9. LOAD CAPACITANCE AND SHORT-CIRCUIT CURRENTS .....	72
FIGURE 2.2.10. CURRENT OF A PASS-TRANSISTOR STRUCTURE .....	73
FIGURE 2.2.11. STATIC RAM ARCHITECTURE .....	73
FIGURE 2.2.12. TTL INPUT BUFFER .....	75
FIGURE 2.2.13. TRI-STATE OUTPUT BUFFER.....	76
FIGURE 2.2.14. BI-DIRECTIONAL INPUT CIRCUIT.....	77
FIGURE 2.2.15 CLOCK DISTRIBUTION.....	79
FIGURE 3.3.1. AN EXAMPLE OF THE POWER DISTRIBUTION INSIDE A FPGA.....	93
FIGURE 3.3.2. TEST PLATFORM .....	94
FIGURE 3.3.3. MEASUREMENT OF THE INTERCONNECT POWER CONSUMPTION.....	96
FIGURE 3.3.4. POWER CONSUMPTION OF LUTS.....	97
FIGURE 3.3.5. FLIP-FLOPS AND CLOCK TREE.....	97
FIGURE 3.3.6. POWER CONSUMPTION OF DFFS .....	98
FIGURE 3.3.7. POWER CONSUMPTION OF OUTPUT PADS.....	98
FIGURE 3.3.8. POWER CONSUMPTION OF INPUT PADS .....	99
FIGURE 3.3.9. POWER CONSUMPTION THE CLOCK TREE .....	99
FIGURE 3.3.10. POWER CONSUMPTION OF MEMORY CELLS .....	100
FIGURE 3.4.1. CLOCK TREE OF A FLEX10K DEVICE.....	105
FIGURE 3.4.2. CLOCK TREE OF A XC4000E.....	106
FIGURE 3.5.1. FLEX10K100 POWER DISTRIBUTION.....	108
FIGURE 3.5.2 FLEX 10K100 POWER DISTRIBUTION WITH TOG = 12.5%.....	109
FIGURE 3.5.3. XC4000E POWER DISTRIBUTION.....	109
FIGURE 3.5.4. XC4000E POWER DISTRIBUTION WITH TOG = 12.5% .....	110
FIGURE 3.5.5. POWER CONSUMPTION OF A XC4000A FPGA.....	111
FIGURE 3.5.6. INTERCONNECT POWER BREAKDOWN.....	111
FIGURE 3.5.7. POWER CONSUMPTION MODEL FROM ALTERA.....	112
FIGURE 3.5.8. POWER DISTRIBUTION OF AN AT40K DEVICE.....	112
FIGURE 3.5.9. POWER BREAKDOWN OF A XC4010E DEVICE .....	113
FIGURE 3.5.10. POWER ESTIMATED USING ALTERA™ MODEL AND ENST MODEL VS MEASUREMENT ....	114
FIGURE 4.2.1. THE RING OSCILLATOR.....	118
FIGURE 4.2.2. RING OSCILLATOR: INTERNAL FREQUENCY .....	119
FIGURE 4.2.3. RING OSCILLATOR: POWER CONSUMPTION .....	119
FIGURE 4.2.4. RING OSCILLATOR: INTERNAL FREQUENCY .....	120
FIGURE 4.2.5. RING OSCILLATOR: POWER CONSUMPTION .....	121
FIGURE 4.3.1. DYNAMIC POWER DISTRIBUTION .....	123
FIGURE 4.3.2. CIRCUIT 1. PIPELINE 1.....	124
FIGURE 4.3.3. CIRCUIT 1. PIPELINE 2.....	124
FIGURE 4.3.4. CIRCUIT 1. PIPELINE 3.....	125
FIGURE 4.3.5. CIRCUIT 2.....	125

FIGURE 4.4.1. POWER CONSUMPTION OF CIRCUIT 1 .....	126
FIGURE 4.4.2. MAXIMUM CLOCK FREQUENCY OF CIRCUIT 2 .....	128
FIGURE 4.4.3. POWER CONSUMPTION OF CIRCUIT 2 .....	128
FIGURE 4.4.4. DOUBLE SUPPLY VOLTAGE.....	130
FIGURE 4.4.5 INTERNAL POWER CONSUMPTION OF CIRCUIT 1 .....	131
FIGURE 4.4.6. I/O POWER CONSUMPTION OF CIRCUIT 1 .....	132
FIGURE 4.4.7. GLOBAL POWER CONSUMPTION OF CIRCUIT 1 .....	132
FIGURE A1. PIPELINE 1 ( $V_{CCIO} = 5$ VOLTS).....	155
FIGURE A2. PIPELINE 1 ( $V_{CCIO} = 3.3$ VOLTS) .....	155
FIGURE A3. PIPELINE 2 ( $V_{CCIO} = 5$ VOLTS).....	156
FIGURE A4. PIPELINE 2 ( $V_{CCIO} = 3.3$ VOLTS).....	156
FIGURE A5. PIPELINE 3 ( $V_{CCIO} = 5$ VOLTS).....	157
FIGURE A6. PIPELINE 3 ( $V_{CCIO} = 3.3$ VOLTS).....	157
FIGURE A7. INTERNAL POWER CONSUMPTION OF CIRCUIT 1 .....	158
FIGURE A8 I/O CELLS POWER CONSUMPTION OF CIRCUIT 1 .....	158
FIGURE A9. INTERNAL POWER CONSUMPTION OF CIRCUIT 1 .....	159
FIGURE A10 I/O POWER CONSUMPTION OF CIRCUIT 1 .....	159
FIGURE A11. POWER OPTIMIZATION IN CIRCUIT 2 .....	160
FIGURE A12. POWER CONSUMPTION OF CIRCUIT 2 .....	160
FIGURE A13. POWER CONSUMPTION OF CIRCUIT 1 .....	162
FIGURE A14. POWER BEHAVIOR OF CIRCUIT 1.....	162
FIGURE B1. PASS-TRANSISTOR STRUCTURE.....	165



## List of Tables

TABLE 1.1.1. POWER DISSIPATION OF MICROPROCESSORS (FROM [10]) .....	29
TABLE 1.1.2. TECHNOLOGICAL EVOLUTION (SOURCE SEMICONDUCTOR INDUSTRY ASSOCIATION 1998)..	29
TABLE 2.1.1 COMMERCIAL AVAILABLE FPGA ARCHITECTURES .....	42
TABLE 3.2.1. AT40K™ INTERNAL CURRENT PARAMETERS (SOURCE: AT40K DATA BOOK).....	89
TABLE 3.4.1. FLEX10K100 INTERCONNECT .....	101
TABLE 3.4.2. XC4010E INTERCONNECT.....	102
TABLE 3.4.3. LOOK-UP TABLES .....	103
TABLE 3.4.4. DFFS.....	103
TABLE 3.4.5. OUTPUTS .....	104
TABLE 3.4.6 INPUTS .....	104
TABLE 3.4.7 FLEX 10K100.....	105
TABLE 3.4.8. XC4010E.....	106
TABLE 3.4.9 MEMORY CELLS IN ALTERA AND XILINX .....	107
TABLE 3.4.10 2K*1MEMORY IN ALTERA AND XILINX .....	107
TABLE 3.5.1. POWER CONSUMPTION MODELS .....	115
TABLE 4.3.1. RESULTS FROM [23] .....	122
TABLE 4.4.1. MINIMUM SUPPLY VOLTAGE OF CIRCUIT 1.....	126
TABLE 4.4.2. CIRCUIT 1.....	127
TABLE 4.4.3. POWER OPTIMIZATION OF CIRCUIT 1 .....	127
TABLE 4.4.4. POWER CONSUMPTION BEHAVIOR OF CIRCUIT 2.....	129
TABLE 4.4.5. POWER OPTIMIZATION IN CIRCUIT 2 .....	129
TABLE 4.4.6. CIRCUIT 1: $V_{CCIO} = 5$ VOLTS .....	132
TABLE 4.4.7. POWER OPTIMIZATION OF CIRCUIT 1 WHEN $V_{CCIO} = 5$ VOLTS.....	133
TABLE 4.4.8. CIRCUIT 1: $V_{CCIO} = 3.3$ VOLTS .....	133
TABLE 4.4.9. POWER OPTIMIZATION OF CIRCUIT 1 WHEN $V_{CCIO} = 3.3$ VOLTS.....	133
TABLE 4.4.10. CIRCUIT 2: $V_{CCIO} = 5$ VOLTS .....	134
TABLE 4.4.11. POWER OPTIMIZATION OF CIRCUIT 2 WHEN $V_{CCIO} = 5$ VOLTS.....	134
TABLE 4.4.12. CIRCUIT 2: $V_{CCIO} = 3.3$ VOLTS .....	134
TABLE 4.4.13. POWER OPTIMIZATION OF CIRCUIT 2 WHEN $V_{CCIO} = 3.3$ VOLTS.....	134
TABLE A1. HALF FAST TRACK.....	153
TABLE A2. FULL FAST TRACK.....	153
TABLE A3. COLUMNS.....	153
TABLE A4. LOGIC CELLS.....	153
TABLE A5. OUTPUTS.....	154
TABLE A6. INPUTS .....	154
TABLE A7. D-TYPE FLIP FLOP.....	154

TABLE A6. POWER CONSUMPTION OF CIRCUIT 1. .... 161

## Chapter I. Introduction

In the late 90's, power consumption has become an important issue because of the rapid growth of personal wireless communications, battery-powered devices and portable applications such as digital cellular telephones, pocket calculators, notebook and laptop computers. Recently, many power optimization techniques have been proposed at various levels of abstraction, such as at circuit, logic, architectural and system level. The optimization of power consumption is an important design constraint that should be considered with other constraints such as speed and hardware cost.

The use of FPGA is increasing because it satisfies the high speed of system and hardware cost constraints. FPGAs implementation allows the building of rapid prototypes reducing development times and board area. However, it is not evident that FPGAs could satisfy low-power consumption constraint. Compared to ASICs, FPGAs are generally perceived as non low-power consumption devices, whose only advantage is programmability.

If we consider the points described in the last two paragraphs, a study of power consumption in FPGAs will allow the reduction of power consumption in FPGA based systems. Therefore it will result in power optimization techniques at circuit, logic, and architectural level.

### **1.1 Reducing Power Consumption: a growing challenge**

During the last two decades, integrated circuits designers have focused their efforts on increasing the clock frequency and gate density of systems at the expense of an increase in power. Speed and size are important constraints in determining system performance and cost, but power is critical when considering power supply design, reliability, thermal issues and battery life. Usually, speed and density are used as the performance metric of ICs, and power consumption is an afterthought.

Power consumption becomes significant because of the development of portable wireless systems and it influences a great number of design decisions, such as packaging, and cooling requirements. Power is particularly important since conventional battery technology only provides in general 20 W/h of energy for each pound of weight and the voltage is around 1.2 volts. Improvements in battery technology are being made, but it is unlikely that a dramatic solution to the power problems will be forthcoming (i.e. only 40% improvement in battery performance has been obtained over the last five years). Even with the advance of the battery technology such as Nickel-Metal Hydride (Ni-MH) which provides larger energy density characteristics (almost 30 W/h per pound), the lifetime of the battery is low. The development of low-power techniques is important in portable applications [37].

Power Consumption is not only important for wireless applications, more than 50% of the Energy Consumption of schools, enterprises and governmental offices comes from PCs. Even when power is available in non portable applications, the issue of low-power design is critical because of the difficulty in providing adequate cooling which might either add significant cost to the system or provide a limit on the amount of functionality that can be provided.

It is clear also that power is an important constraint for high-performance systems. With large integration density and improved speed of operation, systems with high clock frequency are emerging. These systems are based on high-speed products such as microprocessors.

The cost associated with packaging, cooling and fans required by these systems is increasing significantly.

Table 1.1 shows the power consumption of various microprocessors that operate in a range of 66 to 300 MHz. These data show that power consumption becomes too excessive at higher frequencies.

Processor	Clock (MHz)	Technology ( $\mu\text{m}$ )	VDD (Volts)	Power Peak (Watts)
Intel Pentium	66	0,80	5,00	16
DEC Alpha 21064	200	0,75	3,30	30
DEC Alpha 21164	300	0,50	3,30	50
Power PC 620	133	0,50	3,30	30
MIPS R10000	200	0,50	3,30	30
UltraSparc	167	0,45	3,30	30

TABLE 1.1.1. POWER DISSIPATION OF MICROPROCESSORS (FROM [10])

Another issue related to power consumption is reliability. An excessive increment of power dissipation can reduce the performance of the circuit. It could also provoke some failure mechanisms such as silicon interconnect fatigue, package related failure, electrical parameter shift, electromigration, and junction fatigue.

Reliability problems coupled with power consumption issues, when scaling down to 0.5 $\mu\text{m}$ , have driven the electronics industry to adopt lower supply voltages. New standards for ICs operating voltage such as 3.3 volts, 2.5 volts and 1.8 volts are adopted. The effect of lowering the supply voltage delivers impressive results in terms of power consumption. Nevertheless, since size, density, frequency and the number of I/O per package are increasing drastically, power dissipation increases also. Table 1.2 shows the evolution of ICs technology and the increment of power consumption.

	1995	1998	2001	2004	2007	2010
Technology ( $\mu\text{m}$ )	0,35	0,25	0,18	0,13	0,1	0,07
DRAM size bits	64 M	256 M	1 G	4 G	16 G	64 G
Transistors per $\mu\text{P}$	12 M	28 M	64 M	150 M	350 M	800 M
gates ASIC	5 M	14 M	26 M	50 M	210 M	430 M
Frequency (MHz)	300	450	600	800	1000	1100
Metal layer	5	5	6	6	7	8
Supply (Volts)	3,3	2,5	1,8	1,5	1,2	0,9
Power (Watts)	80	100	120	140	160	180

TABLE 1.1.2. TECHNOLOGICAL EVOLUTION (SOURCE SEMICONDUCTOR INDUSTRY ASSOCIATION 1998)

We must consider that most recent processors can work at 1GHz.

## **1.2 Field Programmable Gate Arrays**

### 1.2.1 Introduction

In the seventies and eighties, TTL series 54/74 logic circuits were the main stay of digital logic design for implementing combinatorial and sequential logic including discrete logic gates, specific Boolean transfer functions, and memory elements, as well as counters, shift registers, and arithmetic circuits. TTL based systems were represented by big boards with many packages, thus high power dissipation and low speed, and high costs. In the eighties, gate arrays appeared. These circuits allowed the designers to customize and improve performance, density and power consumption. Programmable logic was only just invented during this period.

Several advantages are provided by programmable logic. Clearly, fewer devices are used. The design used only a portion of the device, so one Programmable Logic device (PLD) device could easily replace 10 or more TTL devices, which make the PLD implementation more cost-effective. Moreover, PLD implementation provides flexibility and system reconfiguration.

### 1.2.2 Programmable Logic Devices (PLDs)

In general, a Programmable Logic Device (PLD) is a type of integrated circuit that can be configured by the end user for a particular implementation. Since these kind of circuits are programmed "in the field", they can also called Field Programmable Logic Devices (FPLDs). One of the first PLDs that was currently used is the Programmable Read-Only Memory (PROM). It consists of an array of memory cells that can be programmed using bit patterns of zeros and ones. Figure 1.1 illustrates the internal architecture of a PROM device and its functionality. In this case, the address bits correspond to the input variables and the data stored in the ROM correspond to the truth table defined by the logic function.

A more appropriate PLD device that was developed as a follow-up of the PROM, was the Programmable Logic Array (PLA), also called Field Programmable Logic device (FPLA).

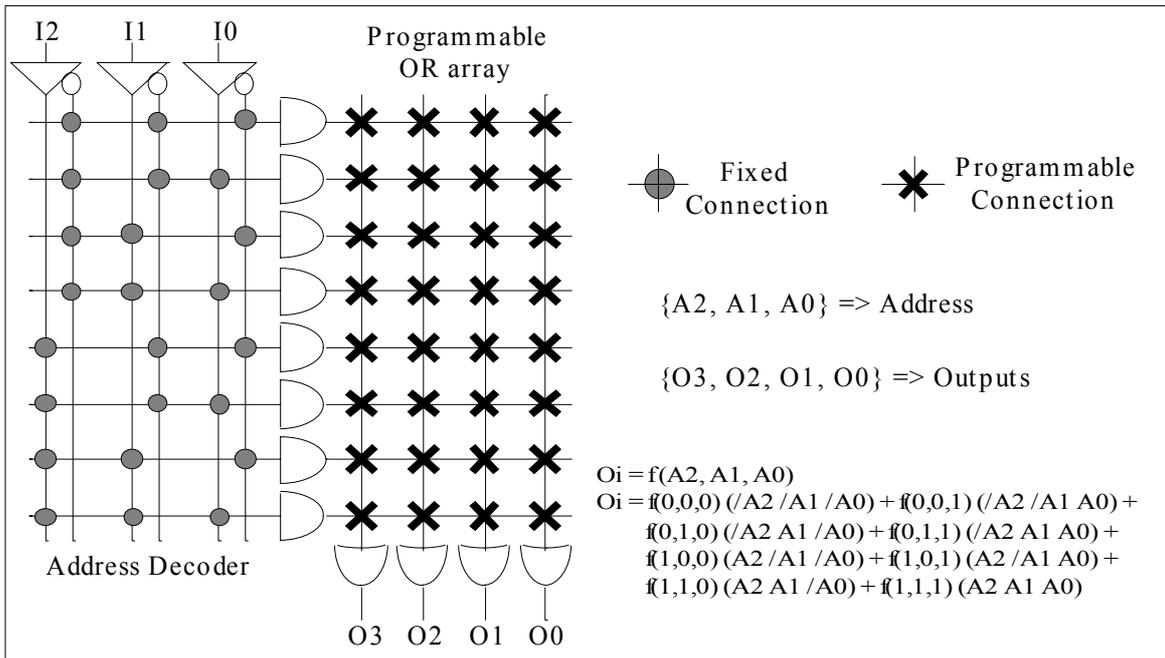


FIGURE 1.2.1. THE PROM ARCHITECTURE

The FPLA consist of an array of AND gates and an array of OR gates, both arrays are programmable. Figure 1.2 illustrates the internal architecture of a FPLA.

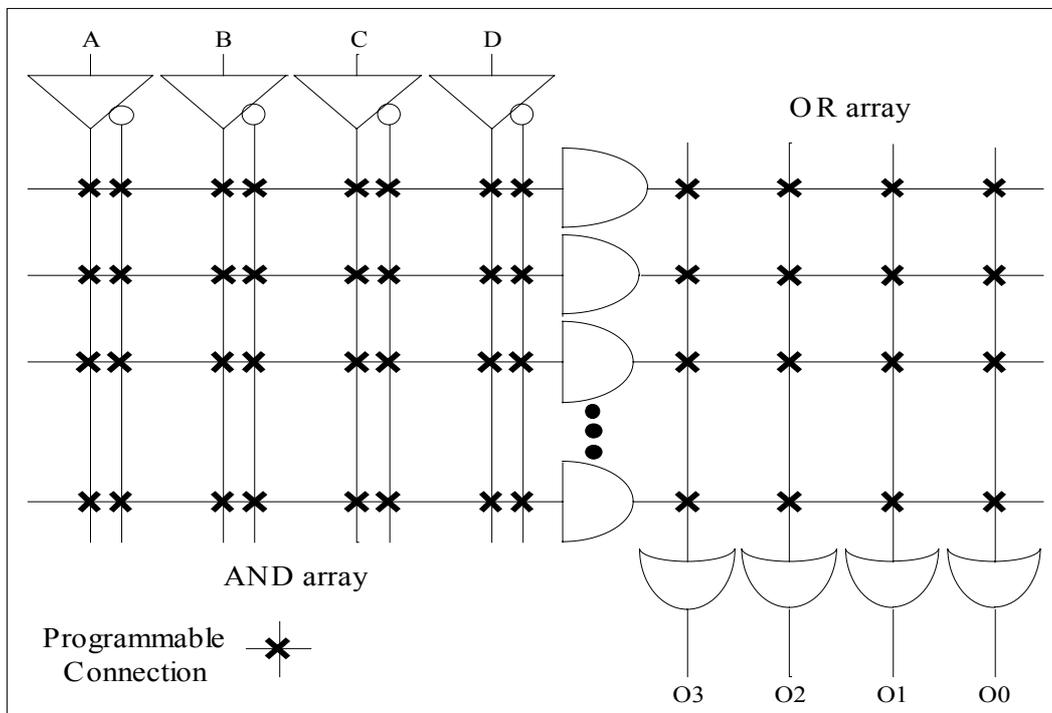


FIGURE 1.2.2. FPLA ARCHITECTURE

Another PLD that was developed based on the PLA architecture was the Programmable

Array Logic (PAL).

A PAL consists of a programmable array of AND gates and a fixed array of OR gates called MacroCell (MCELL). The next figure illustrates the standard gate symbol for a Boolean function ( $F = A/B + /BC + /AB/C$ ) and the equivalent PAL logic diagram:

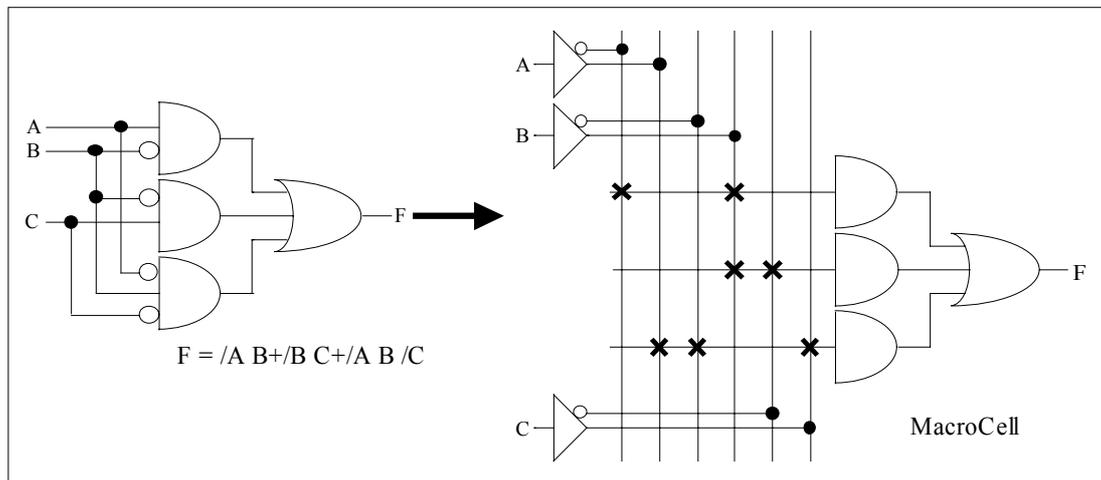


FIGURE 1.2.3. PAL ARCHITECTURE

The single lines extending from the AND gates are used to represent several inputs. The vertical lines represent the signals A, B, and C. Each of the vertical wires is connected to an input signal or its complement. The appropriate connections are made to establish connections to the inputs of the three AND gates, and the OR gate sums all of the product terms.

Recent PLDs include PAL architectures, programmable macrocells, and variable product-term distribution. A macrocell contains the equivalent of 20 ASIC-gates. Each MacroCell in a PLD may be individually configured by programming the state of configuration bits. The programmable macrocell also allows feedback for use as input to the logic array.

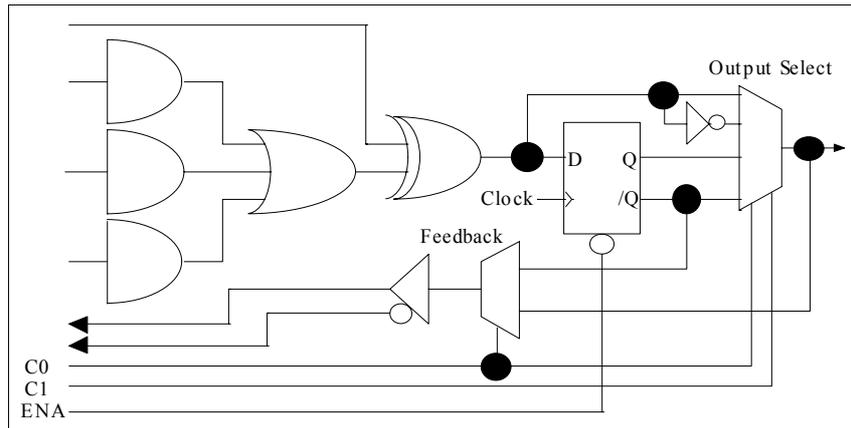


FIGURE 1.2.4. THE MACROCELL

More complex or Erasable PLDs (CPLDs or EPLDs) extend the concept of the PLD to a higher level of integration to improve system performance. EPLDs contain multiple logic blocks, each logic block communicate with another using signals routed via a single programmable interconnect. This architecture makes more efficient use of the available silicon die area, leading to better performance and reduced cost.

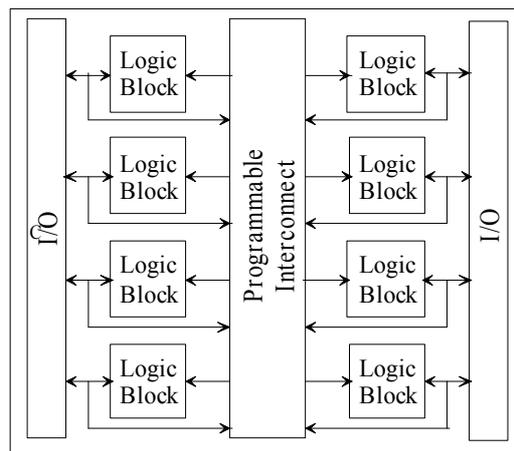


FIGURE 1.2.5. CPLD ARCHITECTURE

### 1.2.3 FPGA architecture

A Field Programmable Gate Array (FPGA) is a type of programmable device which consists of an array of uncommitted logic elements that can be interconnected in a general way. FPGAs were introduced to the market in 1985 by Xilinx Corporation. Since then, many different FPGAs have been developed by a number of companies like Altera, Atmel, Actel, Lucent Technologies, etc. Actually, FPGAs can contain the equivalent of several 10,000 to 2,000,000 logic gates with programmable interconnect and more than 420 kbits of embedded memory. It permits the implementation of customized logic functions. FPGAs can implement any arbitrary Boolean equation

(combinatorial) or registered (sequential) function with built-in logic structures called logic cells (LC) or logic blocks (LB). Like EPROM devices, LCs can implement logic or Boolean functions using a memory to store the logic function. LCs inputs are used to address a memory containing the truth table of the function. This memory is called "*Look-Up Table*" (LUT).

Like a semi-custom gate array, a FPGA consists of a two-dimensional array of logic and memory blocks that can be connected by general interconnection resources. The interconnect comprises segments of wire, where the segment may be of various lengths.

Present in the interconnect are programmable switches that serve to connect the logic blocks to the wire segments, or one wire segment to another. These routing wires also connect to I/O's blocks. FPGA architectures differ from vendor to vendor, but in general it could be described using the following figure:

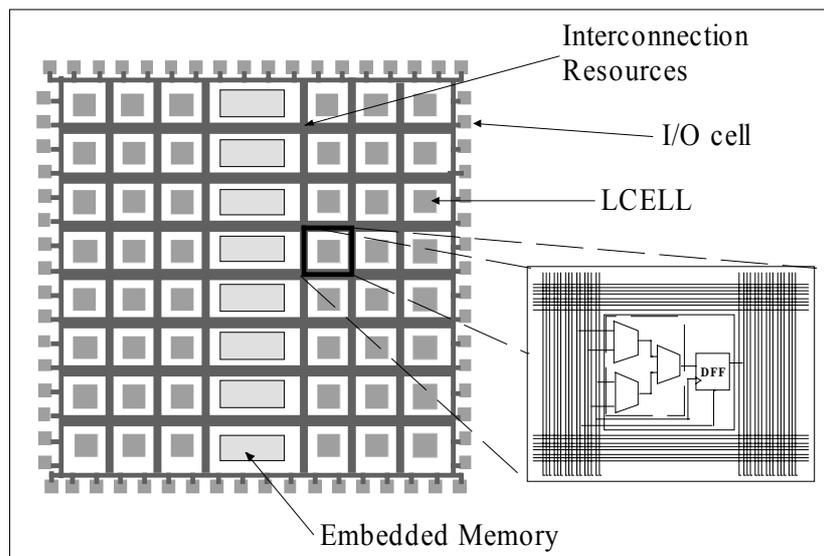


FIGURE 1.2.6. FPGA INTERNAL ARCHITECTURE

#### 1.2.4 FPGA and Application-Specific Integrated Circuit (ASIC)

Before a description of FPGAs, It is important to understand the differences between ASICs and FPGAs.

An ASIC is a custom monolithic IC that is customized on all mask layers. There are two types of custom IC:

3. The Standard cell IC. This device is customized on all mask levels using a cell

library that embodies pre-characterized circuit structures and it is designed using a silicon compiler.

4. The Full custom IC. This device is at least partially "hand-crafted". Handcrafting refers to custom layout and connection work that is accomplished without the aid of a silicon compiler or standard cells.

A FPGA is a PLD that offers fully flexible interconnects, fully flexible logic arrays, and requires functional placement and routing.

We will define an ASIC as a full-custom device that could be customized using a silicon compiler or that it could be handcrafted. The most significant difference between ASICs and FPGAs is that ASICs are mask-programmed devices and FPGAs are interconnect-programmed devices, it permits FPGAs to be reconfigured and re-used.

Recently the use of FPGAs has become popular because it allows the reduction of development times, the fabrication of rapid prototypes and hence decrease costs, especially for low-volume parts. Figure 1.6 illustrates the advantages when using FPGAs.

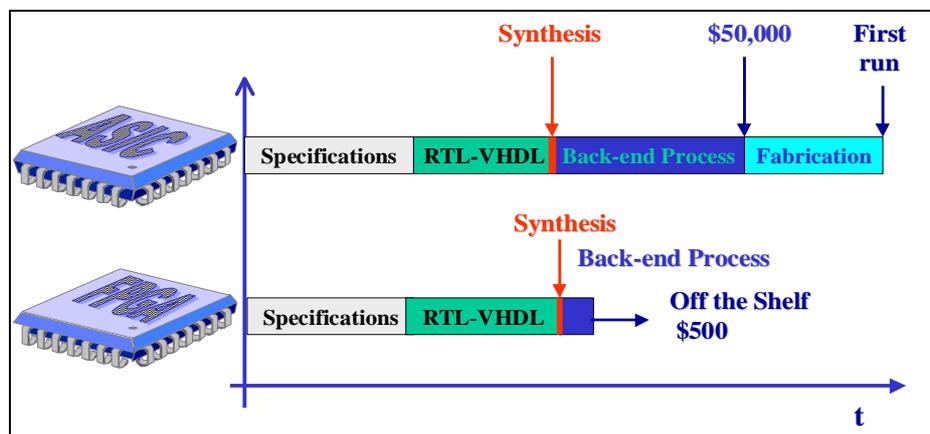


FIGURE 1.2.7. ASICS VERSUS FPGAS

Another advantage when using FPGAs is that the back-end process is reduced. We can verify the architecture functionality by programming a device directly.

### 1.3 Power Consumption in FPGAs

As we said in section 1.2, recently the use of FPGAs has increased for many reasons. And in section 1.1 we emphasize the need for low power systems. If we consider the impact of both factors, it becomes clear that power consumption is an important constraint in developing FPGA based systems.

The potential for low power design with FPGAs is substantial because of the low level of control by the designer because FPGA CAD tools allow the user to exploit program and design architecture methods to decrease power consumption (as we show in further sections). FPGA designers can optimize directly speed, and area.

Since these constraints are closely related to power consumption, a first approach would be the use of architectural techniques to save power consumption while maintaining high speed and low surface.

FPGA CAD tools allows the designer to visualize and optimize the internal resources used by their application. These tools are designed to optimize speed and surface, but they can also be used (or optimized) to reduce power consumption in FPGAs.

Unfortunately FPGA Vendors provide very little support for both estimation and optimization of power consumption. This fact is due to the current applications of FPGAs as well as the difficulties of power estimation for arbitrary circuit configuration. Vendors only provide a set of equations which estimate power dissipation as a function of logic blocks used, I/Os blocks used and a rough factor for logic transitions.

The main goal of this research program is to estimate and optimize power consumption on FPGA-based systems. The research includes the modeling of power consumption in FPGAs; the optimization of power consumption with techniques at the architectural and system level, and the verifications of these models based on actual measurements.

#### ***1.4 Dissertation outline***

In this document we present a study of power consumption in FPGAs. It can be divided into the following parts:

- In chapter 2 we present a brief description of the different FPGAs technologies, it includes internal architectures and programming technologies; an overview of power consumption in MOS-based circuits and a resume of the most important related works.
- In chapter 3 we describe the incremental measurement methodology and the power consumption model for both fine-grained (Xilinx™) and coarse-grained (Altera™) FPGAs based on measurements. This model is compared at the end of this chapter with models proposed by vendors and by another model based on measurements proposed by the University of California Berkeley.
- Based on this model, in chapter 4 we propose some power optimization techniques using commercial FPGAs. As mentioned below, some techniques from the circuit level to the architectural level will be improved to save power in FPGAs while maintaining a good level of performance.
- And finally, our conclusions are presented in Chapter 5. This chapter is concluded by proposing some research axes that can be explored by using this dissertation as a start point.



## **Chapter II. State of Art**

### ***2.1 Commercially Available FPGAs***

#### **2.1.1 Introduction**

As described in chapter 1, a FPGA consists of an array of logic blocs interconnected by general interconnection resources. FPGAs are composed of three fundamental components: Logic Blocks, I/O blocks and programmable routing. More recent FPGAs contain also embedded memory cells and Phase-Locked Loop (PLL) blocks.

This section provides a description of the different FPGA architectures, which support a study on these devices for power consumption. In the following sub-sections, the basis technologies used to make FPGAs programmable, as well as the different types of logic blocks, I/O elements, and interconnect elements will be described. We cover most of programming technologies and SRAM based FPGA architectures in the market.

#### **2.1.2 FPGA basis**

##### ***2.1.2.1 FPGA Architectures***

The variety of FPGA architectures can be as larger as the number of FPGA families in the market. This is caused by the considerable differences in internal architecture, such as the size and structure of logic blocks, and the structure (or complexity) of the interconnect resources.

Some authors like A. Sharma (1998) [74] consider that FPGA architectures can be classified in two different ways: Based on the size and flexibility (or granularity) of the Logic Cell (LC), and based on the routing (or interconnect) architecture. Other authors like V. Betz and J. Rose (1999) [12] consider that FPGAs can be classified based only in their routing architecture. In the following sub-sections, we will explain both types of classifications.

#### *2.1.2.2 Logic Blocks*

The structure and content of a LC (or logic block) can be designed in many different styles and granularities. Some FPGA logic cells are as simple as 2-input Nand gates. Other blocks have more complex structure, such as a Multiplexer or Look-Up Tables (LUT). In some FPGAs, a logic block (LB) corresponds to an entire PAL-like structure. There exist a myriad of possibilities for defining the logic block as a more complex circuit, consisting of several sub-circuits and having more than one output. Most logic cells contain D-type Flip-Flops in order to implement sequential circuits.

Logic blocks can often be classified by their granularity. The granularity of a LC can be defined in different ways, such as the number of transistors, the number of Boolean operations that can be realized by the LB, or the number of inputs and outputs of the block. According to A. Sharma (1998) [78], logic blocks can be classified into two categories:

5. The fine-grain logic blocks architecture consists of few transistors with programmable interconnect resources. The major advantage of this architecture is the high LB utilization achievable. On the other hand, they require many wires segments and programmable switches resulting in additional chip area and an increase of timing delay and power consumption.
6. Coarse-grain architectures are based on the ability of a multiplexer that connects each of its inputs to a constant or to a signal to implement different logic functions.

The most important advantage of Multiplexer-based LBs is that they provide a high degree of functionality with a relative small number of transistors. However, this advantage is achieved at the cost of larger number of input requirements that can place a high demand on routing resources. An example of this architecture is ACTEL™ FPGA Logic Block.

### 2.1.2.3 Interconnect Resources

The routing architecture of an FPGA is the way in which the programmable switches and the wires segments are placed into the circuit to allow the programmable interconnection between the logic and I/O blocks. An FPGA routing architecture typically includes wire segments of varying lengths and interconnection blocks.

The number of wires employed in a FPGA affects the density achieved by the device. If the number of used wires is inadequate, only a small number of logic blocks is achieved. On the other hand, an excessive number of wiring segments can increase the die size and results in poor silicon utilization efficiency.

Routing architectures of FPGAs have to accomplish two constraints: routability and speed. The routability is the capability of the FPGA to accommodate all nets for a typical application even if the number of wire has to be predefined for blank (or unprogrammed) FPGA configuration.

According to A. Sharma (1998) [78], FPGAs are classified into three basic architectural groups based on the logic blocks size, functionality and, the structure of the interconnect resources:

1. Row-based FPGA. This architecture consists basically of coarse-grain logic blocks organized in rows, which are divided horizontally by programmable routing channels. The programmable routing contains wire segments of different lengths. An example of this architecture is ACTEL™ FPGAs.
2. Symmetrical FPGAs. Based on large grain logic blocks that are also called Configurable Logic Blocks (CLBs), this architecture is a matrix of CLBs with horizontal and vertical routing channels. This architecture can be also visualized as a net of programmable wires for direct (or neighborhood) connections, along with general purpose and long lines. Xilinx™, Lucent™ and Atmel™ devices can be considered in this classification.
3. The cellular architecture consists of two-dimensional symmetrical arrays of Logic Cells with a hierarchical interconnect structure. LCs can be connected directly about each other by using a low level of interconnect (or local interconnect). The longer connections use a high level of interconnect (formed by long wires) to reach one section from another, or one LCELL to an I/O cell. Examples of this architecture are Motorola™ and Altera™ FPGAs.

V. Betz and J. Rose (1999) [12] consider also three classifications: Xilinx™, Lucent™ and Vantis™ FPGAs are considered as island-style devices; the Actel™ architecture are row-based; and Altera™ devices are classified as hierarchical FPGAs.

#### 2.1.2.4 Classes of commercial FPGAs

In general there are two kinds of FPGA architectures: fine-grained and coarse-grained. Fine-grained devices consist of a large number of relatively simple logic blocks. The logic block usually contains either a two-input logic function or a 4-to-1 multiplexer and a D-type flip-flop (i.e. Actel™, Atmel™). Coarse-grained architectures consist of large logic blocks containing two or more look-up tables (LUT) and two or more DFFs. In general, these architectures are based on 4-input LUTs (i.e. Altera™, Lucent™, Vantis™ and Xilinx™). The following table summarizes the classification of some FPGA architectures:

Architecture	Anti-fuse	Flash	EPROM	SRAM
Coarse-grained	QuickLogic (pASIC)	Cypress (Delta)	Cypress (Ultra)	Altera (Flex, Apex) Atmel (AT40k) Lucent (Orca) Vantis (VF1) Xilinx (XC3000, XC4000, Spartan, Virtex)
Fine-grained	Actel (ACT)	Actel (ProAsic) GateField	GateField (GF260)	Actel (SPGA) Atmel (AT6K)

TABLE 2.1.1 COMMERCIAL AVAILABLE FPGA ARCHITECTURES

### 2.1.3 Currently available FPGAs Technology

#### 2.1.3.1 Programming Technology

In this section we expose the different programming technologies. FPGAs consist of two layers: a programmable layer that contains programmable elements, such as low-resistance and low-capacitance interconnect switches; and a logic layer which contains logic blocks, I/O elements and interconnect. The programming elements are used to implement the programmable connections among all the internal logic elements. Several different programming technologies are used to implement the programmable switches in FPGAs. In the following section we will describe some of the most currently used technologies.

a) Anti-fuse

The anti-fuse switch or amorphous silicon composition is basically a two-terminal device with an unprogrammed state presenting a very high resistance between its terminals. When a high voltage (from 11 to 21 volts) is applied between both terminals, the anti-fuse is blown to create a low-resistive and permanent link.

The anti-fuse developed by Actel™ consists of three layers. The top layer is a conductor made of polysilicon. The middle layer consists of an oxide-nitride-oxide (ONO) chemical composition used as insulator. The bottom layer is a conductor of negatively doped diffusion. Unprogrammed, the ONO anti-fuse insulates the top layer of metal from the bottom layer.

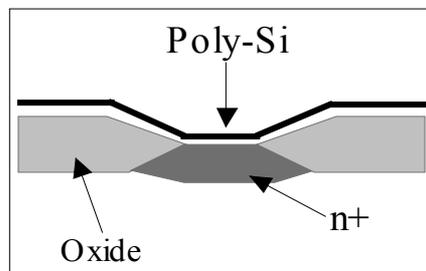


FIGURE 2.1.1. THE ANTI-FUSE SWITCH

When the anti-fuse is programmed, a current of about 5 mA is passing through the device. This procedure generates enough heat in the dielectric to cause it to melt and form a conductive link between the poly-Si and the n+ diffusion. Both, the bottom layer and top layer of the anti-fuse are connected to metal wires. When the anti-fuse is programmed, a very low resistance connection is formed between the two metals wires.

Figure 2.1.2 shows the Amorphous-silicon anti-fuse used by QuickLogic™. In this case, a "via" is placed in the space between two layers of metal. This element is called "vialink".

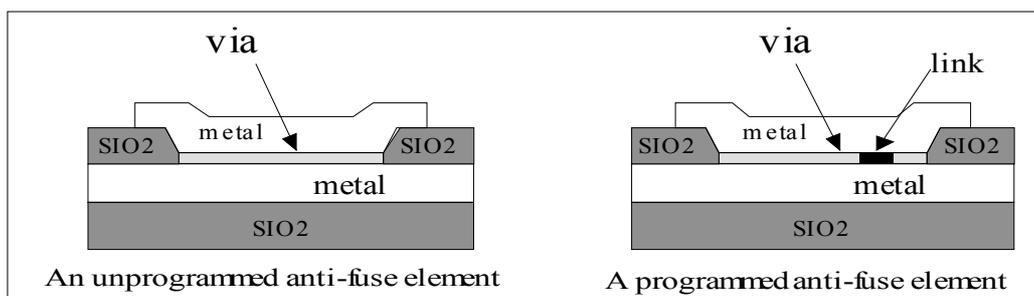


FIGURE 2.1.2. THE AMORPHOUS-SILICON ANTI-FUSE

In this type of element, the two layers of metal are separated by amorphous (uncrystallized) silicon, which provides electrical insulation. A programming pulse of 10 V to 12 V with a specific duration applied across the "via" creates a bi-directional conductive link. Once programmed, the anti-fuse element cannot be reprogrammed.

*b) EPROM*

EPROM programming technology is used in Altera™, Atmel™, Cypress™ and Xilinx™ FPGAs. This technology is the same as that used in EPROM memories. Figure 2.1.3 shows the EPROM transistor. Based on the NMOS transistor, the EPROM transistor contains two gates: a floating gate and a select gate. The floating gate is positioned between the selected gate and the transistor's channel. It is also called "floating" because it is not electrically connected to any circuit.

During the unprogrammed state, no charge is stored on the floating gate and the EPROM transistor can be turned ON by applying a voltage in the selected gate (like a NMOS). When the transistor is programmed by causing a large current to flow between the drain and source, a charge is trapped under the floating gate. This charge forces the transistor to be permanently turning OFF.

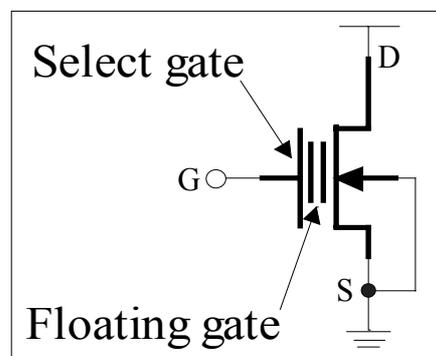


FIGURE 2.1.3. THE EPROM TRANSISTOR

To program the floating gate transistor, a large voltage (16 to 20 V) is applied between the drain and source terminals. Simultaneously, a large voltage (about 25 V) is applied to the select gate. In the absence of any charge on the floating gate, the device behaves as a regular n-channel enhancement MOSFET. An n-type inversion layer (channel) is created at the wafer surface as a result of the large positive voltage applied to the select gate. Because of the large positive voltage at the drain, the channel has a tapered shape.

The drain-to-source voltage accelerates electrons through the channel. The large positive voltage on the select gate establishes an electric field in the insulating oxide. This electric field attracts the electrons and accelerates them toward the floating gate. In this way the floating gate is charged, and the charge that accumulates on it becomes trapped. This charge will cause electrons to be repelled from surface of the substrate. This implies that to form a channel, the positive voltage that has to be applied to the selected gate will have to be greater than the required when floating gate is not charged. In this state, called programmed state, the cell is said to be storing a '0'.

An EPROM transistor can be re-programmed by first removing the trapped charge from the floating gate. Exposing the gate to ultraviolet light exits the trapped electrons to the point where they can pass through the gate oxide into substrate.

The EPROM transistor in a FPGA is used as a pull-down device for logic block inputs. This arrangement, illustrated in figure 2.1.4, has one wire called "word line" which is connected to the select gate of the EPROM transistor.

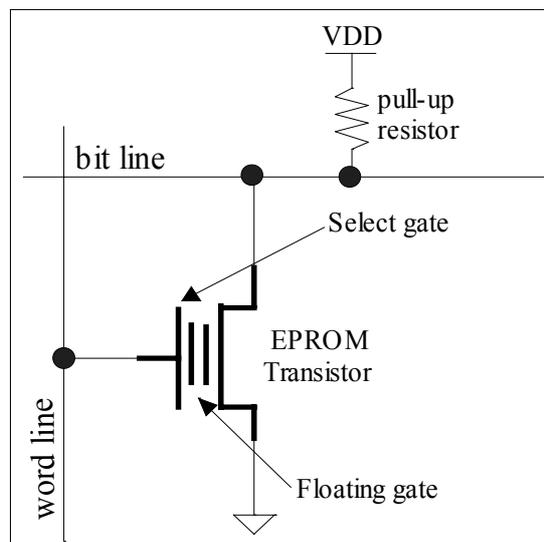


FIGURE 2.1.4. THE EPROM CELL

As long as the transistor has not been programmed into the OFF state, the word line can cause the "bit line", which is connected to a logic block input, to be pulled to logic zero. Since a pull-up resistor is present on the bit line, this scheme allows the EPROM transistor to not only implement connections but also to realize wired-AND logic functions. A disadvantage of this approach is that the resistor consumes static power.

The EEPROM approach is similar to the EPROM technology except that EEPROM transistor can be re-programmed in-circuit. The disadvantage of using EEPROM is that they consume twice the chip area as EPROM transistor and they require multiple voltage sources, which might not otherwise be required.

#### c) Flash Memory

The flash memory is a type of non-volatile memory (NVM). Like EPROMs, NVLs can retain information even when their power supply is removed. Flash memory has distinct advantage over EPROM in that certain types of flash memory can be erased and re-programmed, and in some cases, with no special voltages needed. Flash memory devices are also lower cost and available in higher densities than EPROM.

A flash memory cell is like a conventional transistor with an extra gate (like EPROM cell). Between the source and drain, and the control gate, there is a second gate called "floating gate" that serves as a charge storage mechanism. When a sufficiently large voltage goes across the source and the control gate, electrons tunnel through the oxide layer and accumulate in the floating gate. This process is called "channel hot electron injection". This extra-negative charge in the floating gate reaches the threshold voltage writing a zero in the cell.

To erase the cell, the control gate must be connected to ground, and the programming voltage must be applied to the source. It removes electrons from the floating gate and turns the cell back to a one.

#### d) Static RAM

The most popular technology used to build programmable logic is Static RAM (SRAM). These cells are used to control the elements used to configure programmable wires and logic cells such as pass transistors, full transmission gates, multiplexers or Tri-state buffers. In the case of the pass-transistor approach, the SRAM cell controls whether the pass-gate is on or off. When off, the pass-gate presents a very high resistance between the two wires to which it is attached, when the pass-gate is turned on, it forms a relatively low resistance connection between the two wires.



data retention or standby current of this cell can be as low as  $10^{-15}$  A.

Another memory cell configuration is shown in figure 2.1.7 In this case, high-resistance polysilicon loads replace the PMOS pull-up devices. The area of this cell could be about 40% smaller than the CMOS six-transistor memory cell. This cell is also called High Resistive Load (HRL) memory cell.

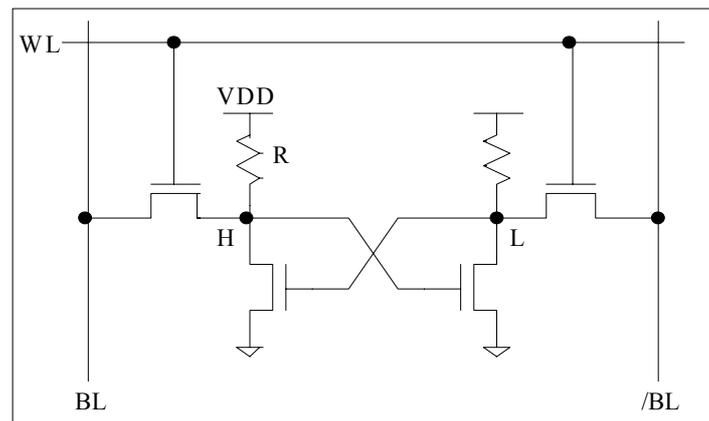


FIGURE 2.1.7. HRL SRAM CELL

The high-state storage node (H) can be pulled down with time due of two sources of leakage currents: the leakage currents flowing through the drain junction, and the sub-threshold current. The voltage drops across the poly-Si resistor R prevents regular cell operation. In several SRAMs based on HRL cells, the total standby current is set to 1  $\mu$ A per chip at room temperature with typical values of resistance equal to  $5 \times 10^{12}$ . The resistance current is limited to  $10^{-13}$  A. This current should be larger than the total leakage current of the storage node to improve the data retention margin. We must notice that the high-level node voltages of all poly-Si load memory cells are  $(V_{DD} - V_T)$  after write cycle. These nodes need a time of several ms to charge up to  $V_{DD}$ . The cell stability is drastically degraded when  $V_{DD}$  is 3V or less.

A SRAM cell is re-programmable, unlike anti-fuse elements, which are physically altered when programmed, SRAM cells are volatile, however, meaning that the states of memory cells are lost when power is not applied. SRAM-based FPGAs must be programmed each time the circuit is powered up.

Compared with other programming technologies described in this section, the chip area required by SRAM approach is relatively large. This is because of the number of

transistors needed for each SRAM cell, as well as the additional transistor for the pass-gates or multiplexers.

The major advantage of this technology is that it allows FPGAs to be reconfigured very quickly and it can be produced using a standard CMOS process technology. That represents an advantage for low power design compared with EPROM and Anti-fuse since SRAM cells can migrate to the next process generation, and both, logic and interconnect benefit from scaling to smaller geometries.

### 2.1.3.2 Logic Blocks Architecture

Logic blocks (or cells) have a great influence in the speed and area efficiency. There are a large number of possibilities for the design of a logic block. In this dissertation, some of the possibilities explored by the FPGA vendors are presented.

#### a) Look-Up Table based Logic Cell

Most recent FPGAs are based on Look-up Table (LUT) logic cells. A  $k$ -input LUT requires  $2^k$  memory cells and a  $2^k$ -input multiplexer to implement any Boolean function of  $k$ -inputs (as mentioned by V. Betz, J. Rose, and A. Marquardt (1999) [12]). Figure 2.1.8 illustrates a 2-input LUT. In this case, 4 SRAM cells are needed to store the truth table of the desired function.

Since LUT-based logic cells are used, research from vendors and from the FPGA research group of the University of Toronto [12] have shown that LUTs with 4 inputs lead to FPGAs with the highest area-efficiency.

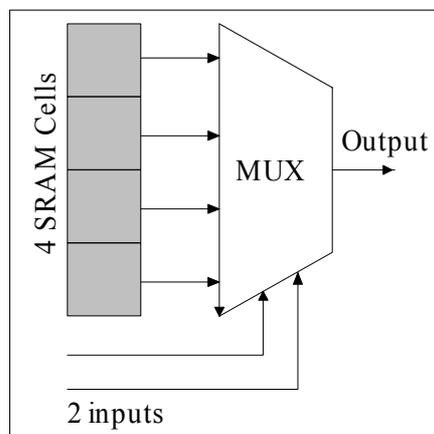


FIGURE 2.1.8. A 2-INPUT LUT

The LUT-based cell developed by Altera™ in Flex™ and Apex™ devices consist, in general, of three elements: a 4-input LUT, a programmable DFF (D-type Flip Flop) and programmable resources that permits the logic element to implement cascade chain and carry chain operations. Figure 2.1.9 shows a Flex 10K Logic Cell.

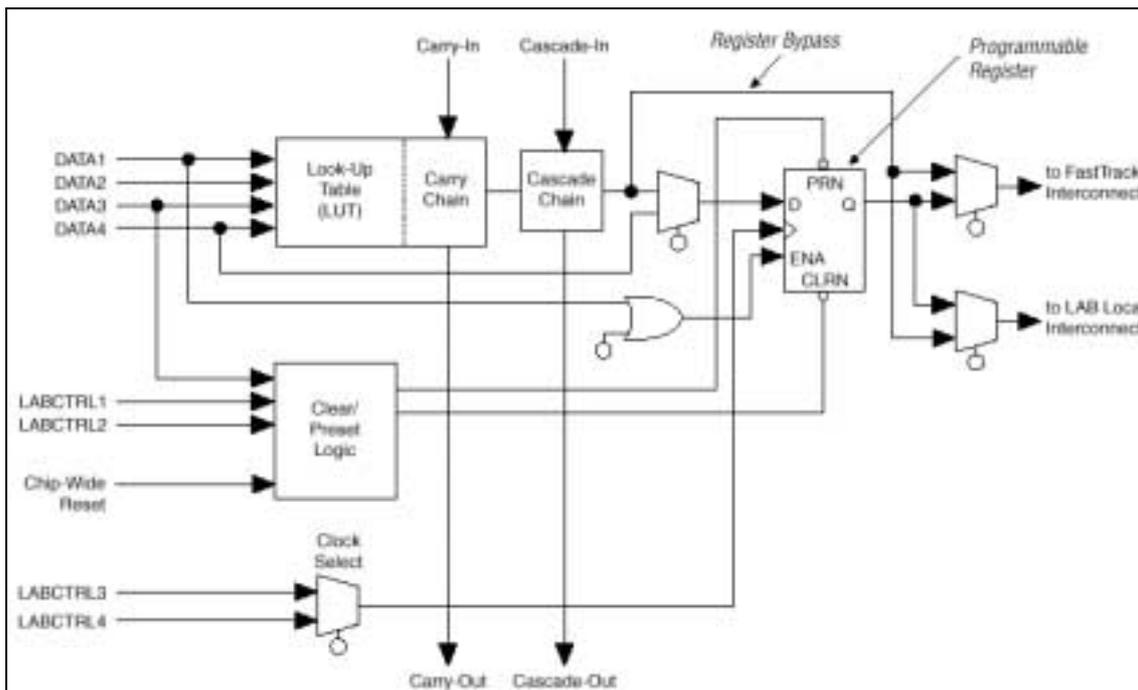


FIGURE 2.1.9. LOGIC CELL WITH A 4-INPUT LUT DEVELOPED BY ALTERA™(COURTESY OF ALTERA™ CO)

Xilinx™ devices contain LUT-based logic blocks. These blocks, called Configurable Logic Block (CLB), are formed by two 4-input LUTs, one 3-input LUT that can be used as multiplexor, and two DFFs. This structure permits to build Boolean functions with 8 inputs. The CLB can be also configured as a 16 x 2 or a 32 x 1 memory cell.

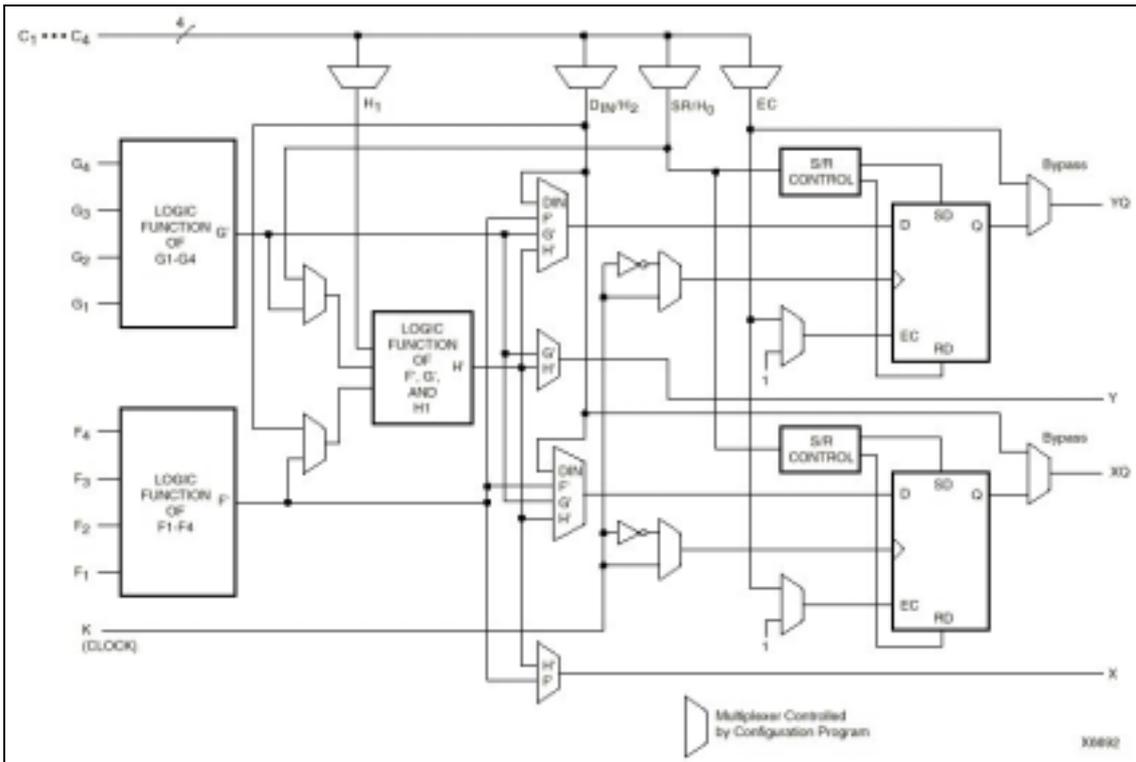


FIGURE 2.1.10 CONFIGURABLE LOGIC BLOCK DEVELOPED BY XILINX™ (COURTESY OF XILINX™ CO)

*b) Multiplexer-based*

The Multiplexer-based logic block developed by Actel™ is presented in figure 2.1.11 It contains three N-input Multiplexer controlled by a certain number of gates (and, or).

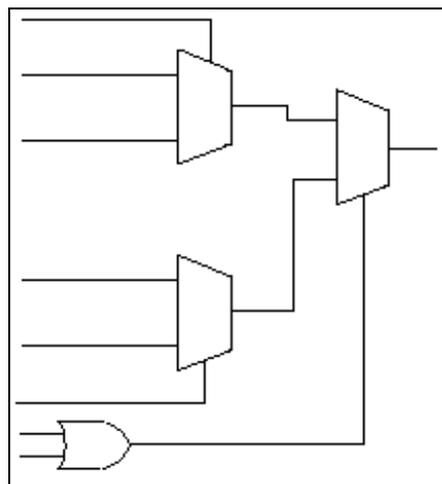


FIGURE 2.1.11. MULTIPLEXER-BASED LOGIC CELL

These modules can implement any of several hundred functions of the inputs. Larger functions can be built by cascading logic cells.

The advantage of this structure is that it requires a small area, it allows the circuit area to be reduced.

*c) Multiplexer and basic Gates or Symmetrical Cell*

Figure 2.1.12 shows the logic block developed by Atmel™. This type of logic block is similar to the Multiplexer-based. It contains four multiplexers named X and Y (two input multiplexers and two output multiplexers), a DFF, and two 8-bit LUTs.

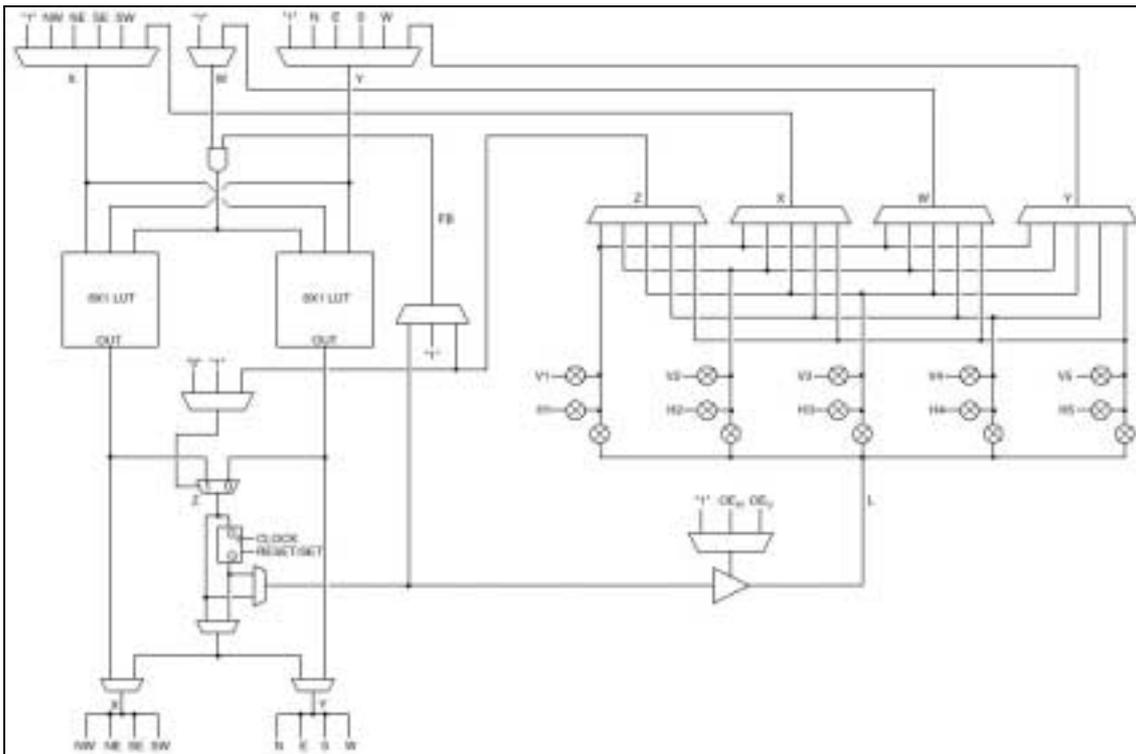


FIGURE 2.1.12. MULTIPLEXER AND BASIC GATES LCELL PROPOSED BY ATMEL™ (COURTESY OF ATMEL™ CORPORATION)

This logic block has 8 inputs and 8 outputs, two pairs for each cardinal side. The logic cell can be accessed by the four cardinal points; this represents an advantage for routing.

**2.1.2.2 Interconnections**

In general, commercial FPGAs can be classified in three groups based on their routing architecture (as we exposed in 2.1.2): Row-based, segmented channel routing (or symmetrical FPGAs), and hierarchical routing.

*a) Row-based FPGAs*

A row-based FPGA proposed by Actel™ consists of rows of logic blocks that are separated by horizontal routing channels.

These channels are formed by wires of various lengths and separated by routing switches. Adjacent wires can be connected to form longer segments where necessary. Dedicated vertical segments are used to connect the inputs and outputs of logic blocks to the interconnect resources via the routing switches.

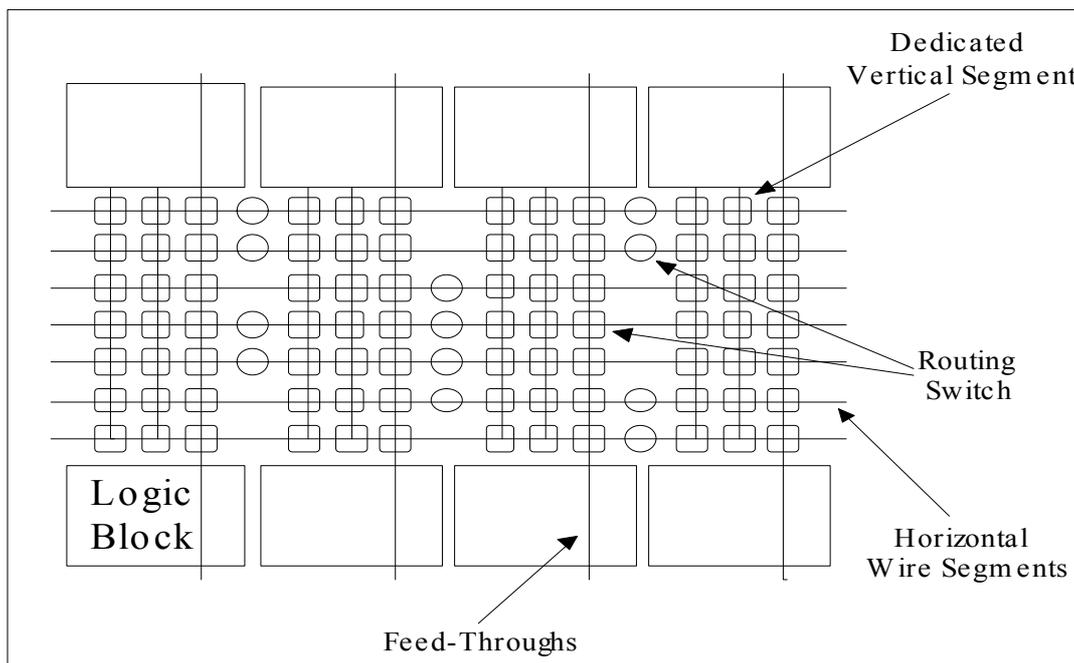


FIGURE 2.1.13. ACTEL™ ACT1 INTERCONNECT ARCHITECTURE (ROW-BASED)

There are also some vertical wires called "feed-throughs" that are used to connect one routing channel to another.

*b) Segment-based FPGAs*

Figure 2.1.14 shows a segment-based (also called island style) architecture implemented in Xilinx™ devices. In this case, the logic cells are surrounded by routing segments. Input or output pins of the LCELL can be connected to some or all of the wiring segments in the channel adjacent to it via a programmable connector switch.

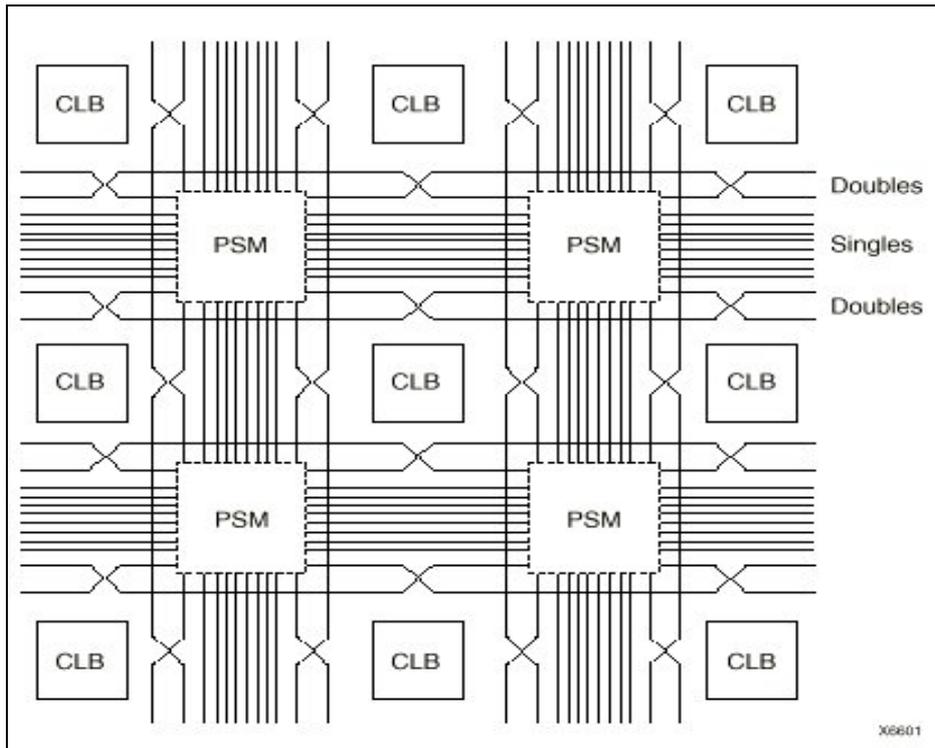


FIGURE 2.1.14. XC4000E/XL/XV INTERCONNECT ARCHITECTURE (SEGMENTED-BASED) (COURTESY OF XILINX™ CORPORATION)

Programmable routing switches allows a segment wire to be connected to another to form longer segments or to connect a horizontal segment with a vertical segment and vice-versa. Long wires that traverse the entire device are dedicated to distribute some important signals like clock, reset, enable, etc. The following figure shows the interconnect resources of the XC4000 series:

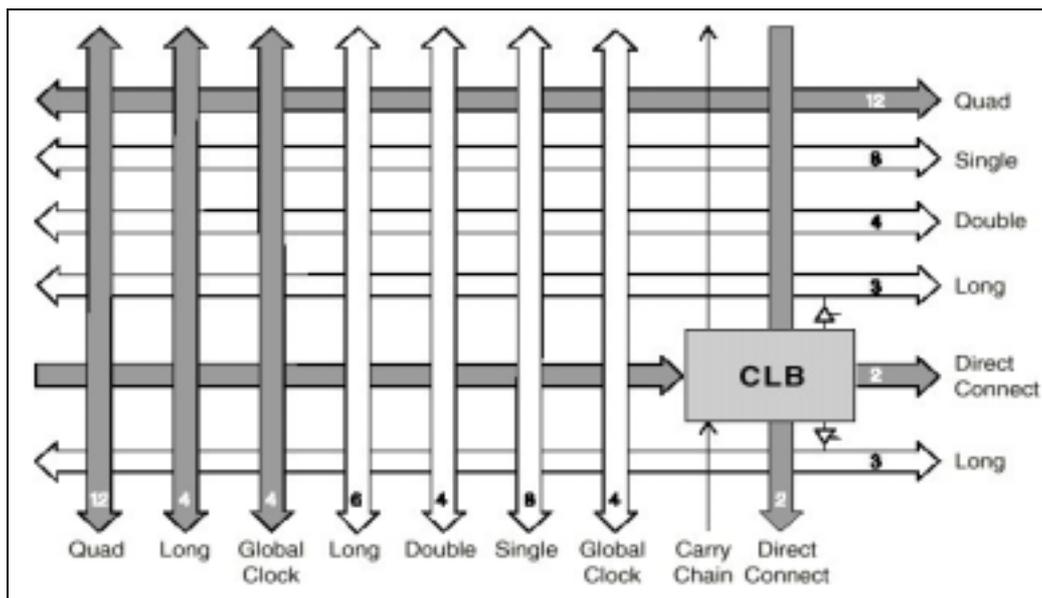


FIGURE 2.1.15 XC4000 SERIES INTERCONNECT RESOURCES (COURTESY OF XILINX™ CO)

c) Hierarchical Routing

Figure 2.1.16 shows the interconnect architecture proposed by Altera™. In this case, the FPGA is organized in long blocks called LABs (Logic Array Block) containing eight logic cells (LCELL). Each LCELL can communicate with the other LCELL of the same LAB by using local wires. LABs can be connected to adjacent blocks by using direct connections.

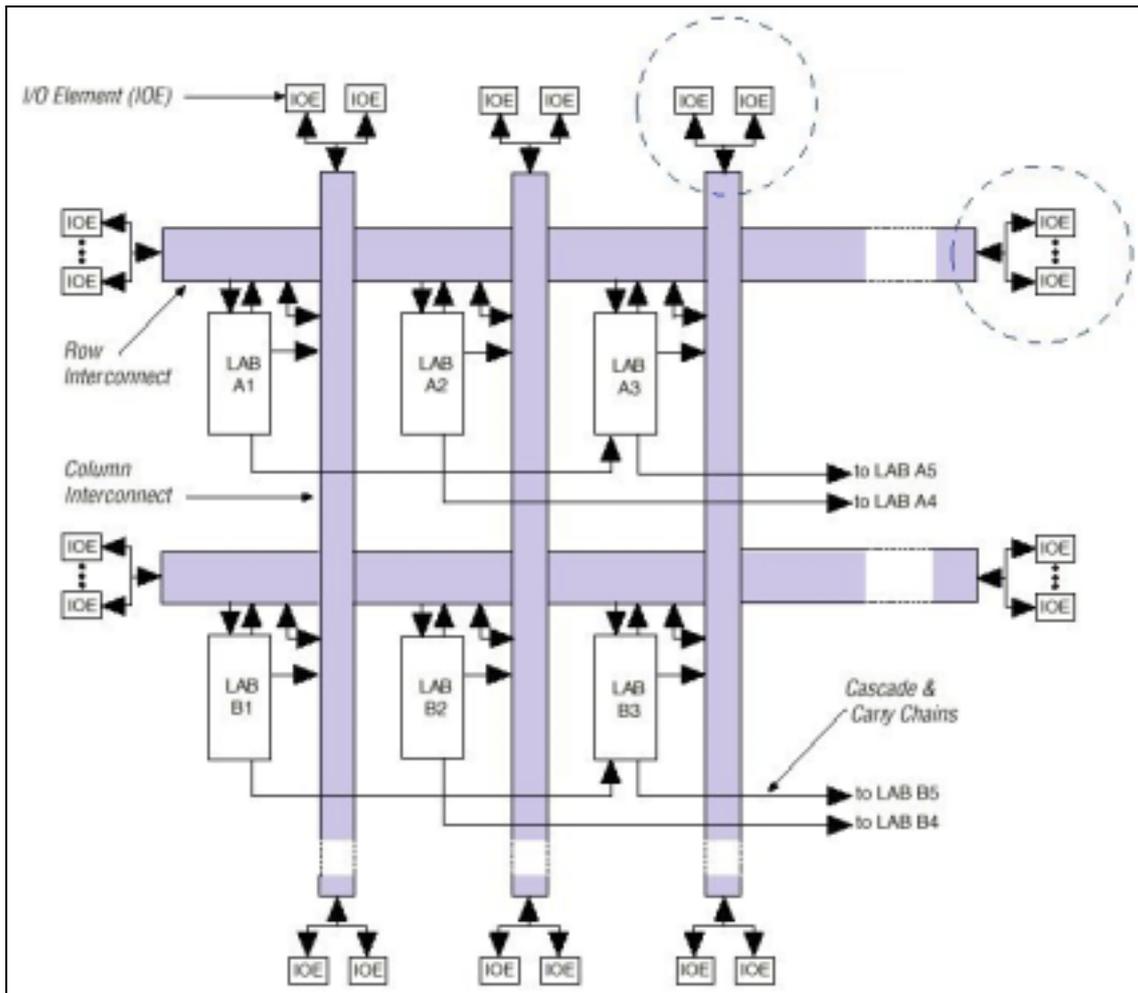


FIGURE 2.1.16. HIERARCHICAL INTERCONNECT (COURTESY OF ALTERA™ CORPORATION)

LCELLS can reach other cells or I/O cells by using long lines, called "fast tracks", that traverse the entire device. According to figure 2.1.16, we can identify three levels of interconnect resources: local (or LAB) interconnect, direct-paths (cascade and carry chain) and fast tracks.

### 2.1.3.4 I/O Structures

There exist different I/O block structures. Most of them consist of DFF-Multiplexer arrays with a slew rate control. The I/O blocks can be configured as Input, Output or bi-directional. I/O units are directly connected to the routing resources of the device. Figure 2.1.17 shows an I/O cell used in Altera™ devices.

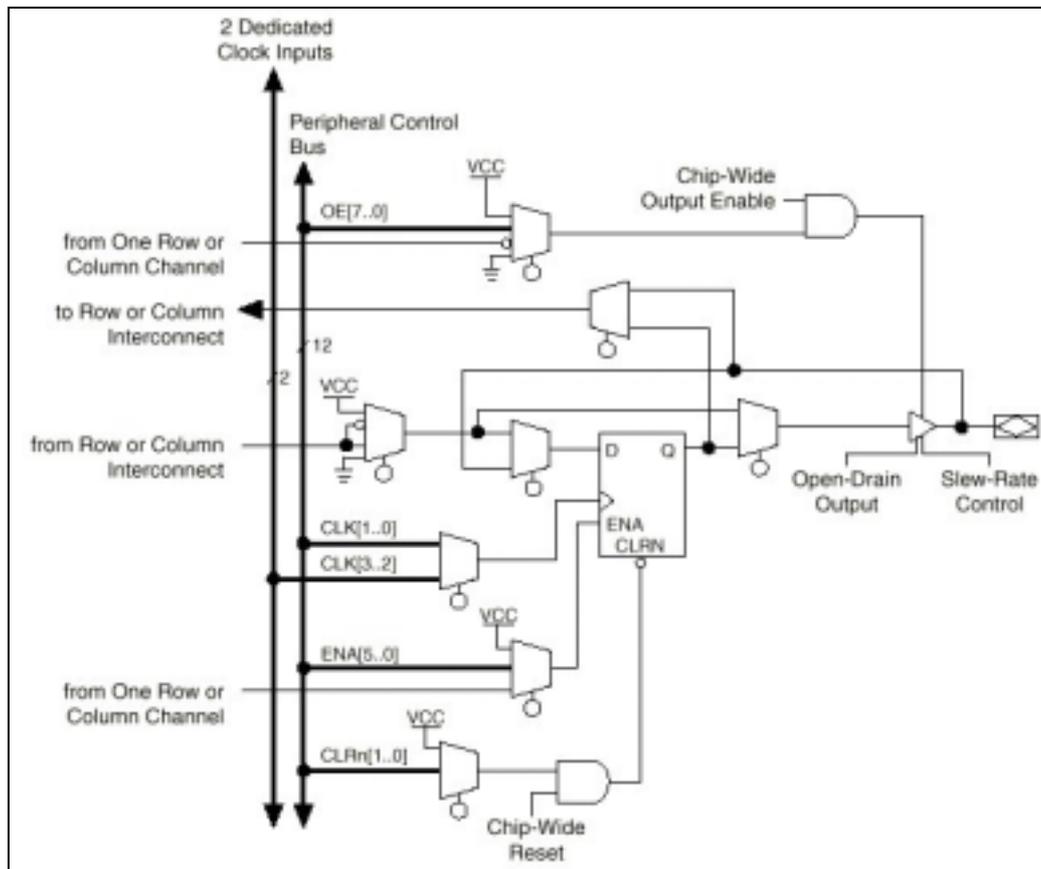


FIGURE 2.1.17. PROGRAMMABLE I/O CELL OF A FLEX 10K (COURTESY OF ALTERA™ CO)

Some FPGAs have "dedicated inputs"; these inputs normally are used for Clocks, Reset, and Enable signals and are directly connected to the internal registers of device. Each I/O units can be configured individually as Input, Output or Bi-directional.

There exist I/O elements that use two DFF registers, one for Input and another one for Output. This reduce control signals when use I/O as Bi-directional representing also an advantage for routing. Figure 2.1.18 shows the I/O cell architecture proposed by Xilinx™.

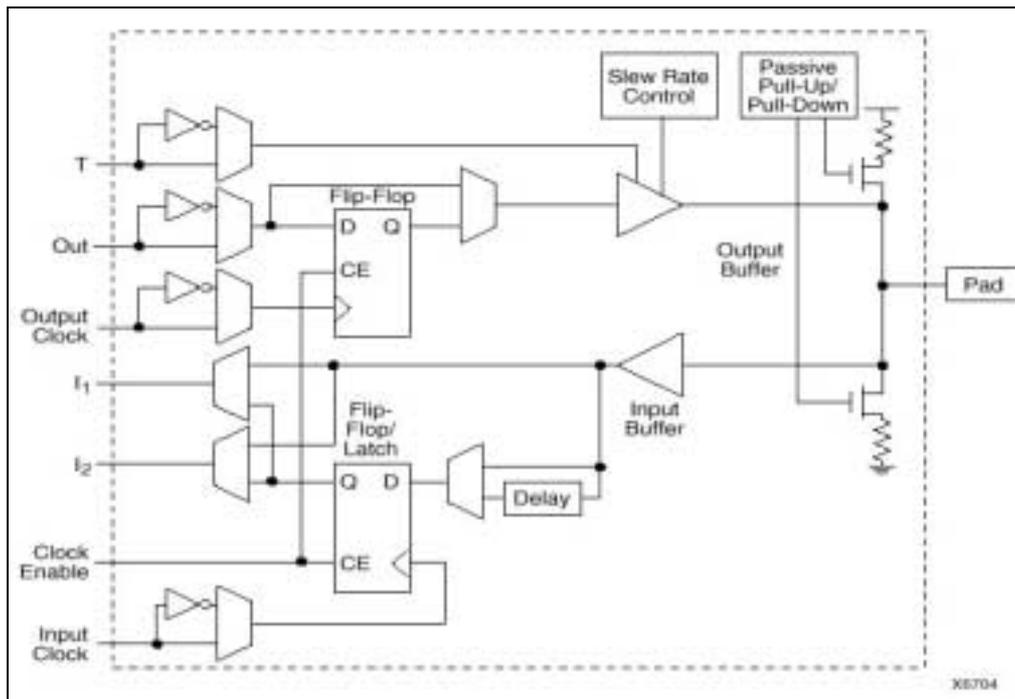


FIGURE 2.1.18. I/O CELL WITH TWO PROGRAMMABLE DFF (COURTESY OF ALTERA™ CO)

The two DFFs provide the ability to register both inputs and outputs. In some cases, a pull-up transistor is added before the output buffer to provide a logical "1" to tri-stated I/O pins.

### 2.1.3.5 Other Resources

Recent FPGAs contain other structures such as embedded memory cells, programmable Phase Lock-Loop (PLL) and, in some cases, thermal sensors. Embedded memory cells can be used to build large memory blocks or to implement logic operations. PLLs are used in FPGAs for several reasons; one of the most important is that they can be used to generate internal clock signals. It is very useful when implementing parallel architectures.

#### a) Embedded RAM Cells

Embedded memory cells can be placed inside the FPGA in big blocks and at the center of the device such as Flex 10K devices, or in small blocks distributed in the whole device as used in the AT40K family. In the last case, specific routing wires and a logic cell are dedicated to optimize the memory access.

Some commercial FPGAs use logic blocks to build memory (i.e. Xilinx™ devices). In this kind of components, there are no embedded cells, and memory blocks are built at expenses of a reduction of the available number of logic blocks.

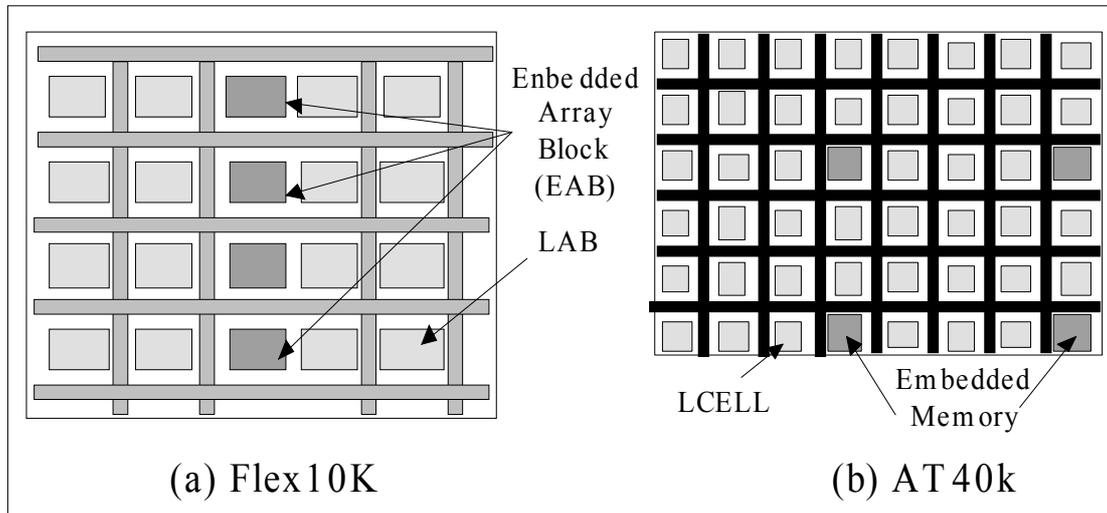


FIGURE 2.1.19. EMBEDDED MEMORY (A) BLOCK. (B) DISTRIBUTED CELLS

Embedded cells can be used to build SRAM, Dual-Port RAM, FIFO (First-in, First-out), LIFO (Last-in, First-Out), and other memory structures like CAM. They can be used to build big logic blocks such as state machines or long logic tables.

*b) Phase Look-loop*

PLLs are normally used to generate internal clock signals from an external clock signal. They allow us to copy the global clock and change its phase. The clock phase can be adjusted by 90° increments for phase shifting of 90°, 180°, and 270°. PLLs are also used to generate two or more internal clock signals with different frequencies by multiplying or dividing the clock frequency.

The use of synchronous PLLs to generate the internal clock allows us to reduce the clock delay and skew within a device. This reduction minimizes clock-to-output and setup times while maintaining zero hold times. Internal PLLs can also be used to create an external clock signal to other devices.

## 2.2 Power Consumption Model of MOS-based Circuits

### 2.2.1 Introduction

Most of the models used to explain the power consumption behavior of ICs are based on the equations derived from the analysis of the CMOS inverter. In order to understand its functionality and to introduce the equations and terms used in further sections, an overview of the CMOS inverter is presented from A. Bellaouar, M. I. Elmasry (1995) [10].

#### 2.2.1.1 The CMOS Inverter

The following figure shows the basic complementary CMOS inverter:

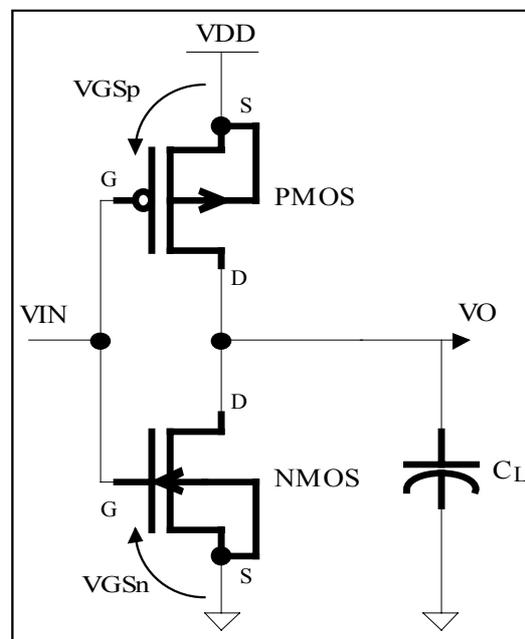


FIGURE 2.2.1 STANDARD CMOS INVERTER

When  $V_{IN}=V_{DD}$ ,  $V_{GSn}=V_{IN}=V_{DD}$  and  $V_{GSp}=V_{IN}-V_{DD}=0$ . In this case,  $V_{GSn}>V_{Tn}$ , and  $|V_{GSp}|<|V_{Tp}|$ . The NMOS is ON and the PMOS is OFF. The NMOS device provides a current path to GROUND (GND), and  $V_O=0$ . When the PMOS is OFF and the  $V_{DS}$  of NMOS device is equal to zero. The DC current from  $V_{DD}$  to GND is controlled by the sub-threshold current of the PMOS device. If the  $V_{Tp}$  (extrapolated threshold voltage) is low enough, the sub-threshold current can be considered negligible, on the other hand, if  $V_{Tp}$  is high, the sub-threshold current is not negligible. In this case, the output voltage is not exactly at zero and can have values of tens of mV.

When  $V_{IN}$  is low (0 volts)  $V_{GSn} < V_{Tn}$  and  $|V_{GSp}| > |V_{Tp}|$ . The PMOS device is ON and the NMOS transistor is OFF. The output voltage is  $V_O = V_{DD}$ . The following figure shows the DC transfer characteristic of a CMOS inverter with the different regions of operation.

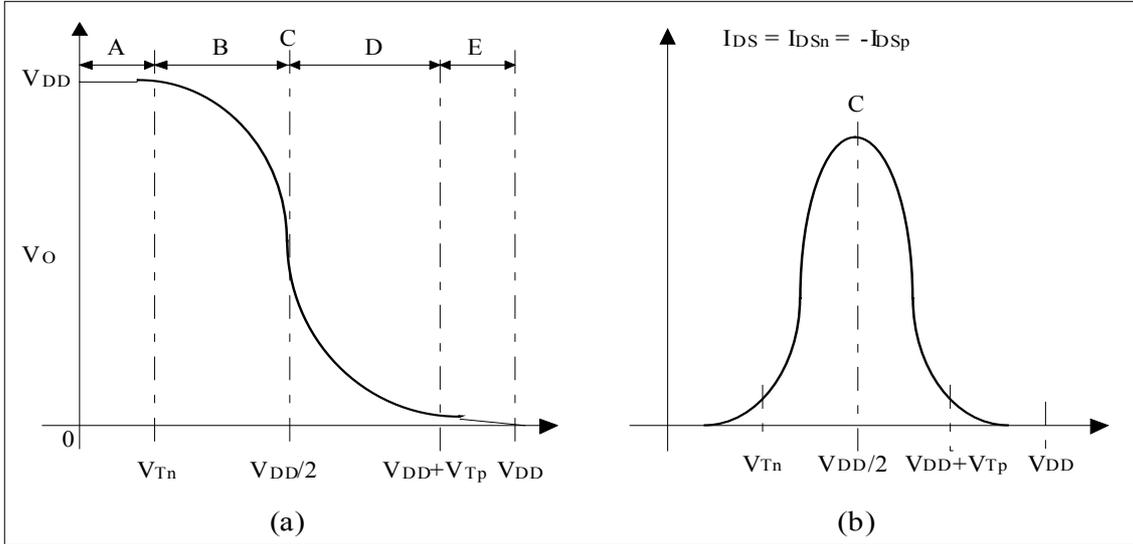


FIGURE 2.2.2. DC TRANSFER CHARACTERISTICS OF A CMOS INVERTER.  
(A) VOLTAGE AND (B) CURRENT

We can notice that the curve is divided into five regions of operation that can be described as follows:

**Region A:** When  $0 \leq V_{IN} < V_{Tn}$ . In this case, the NMOS device is operating in the sub-threshold region and the current is considered zero. The PMOS is in the linear region, and the current flowing through this device is also considered zero. Thus,  $V_O = V_{DD}$ .

**Region B:** When  $V_{Tn} < V_{IN} < V_{INV}$ .  $V_{INV}$  is defined as the input voltage at which the gain of the inverter is maximum and is also defined as the gate threshold voltage. In this case, the NMOS device is operating in the saturation region and the PMOS is operating in the linear region. Since the current in both devices is the same, we have  $I_{DSp} = I_{DSn}$ . The current flowing through the PMOS device is given by:

$$I_{DSp} = -\beta_p \left[ (V_{IN} - V_{DD} - V_{Tn})(V_O - V_{DD}) - \frac{1}{2}(V_O - V_{DD})^2 \right] \quad [2.2.1]$$

Where:

$$\beta_p = k_p \frac{W_{eff}}{L_{eff}} \quad [2.2.2]$$

$$V_{GSp} = V_{IN} - V_{DD} \quad [2.2.3]$$

And:

$$V_{DSp} = V_O - V_{DD} \quad [2.2.4]$$

$W_{\text{eff}}$  is the effective channel width,  $L_{\text{eff}}$  is the effective channel length, and  $K_p$  is a process-dependent parameter that is defined as  $K_p = \mu C_{\text{ox}}$ , where  $\mu$  is the mobility of electrons in the channel of the MOS transistor.  $C_{\text{ox}}$  is the gate oxide capacitance per unit area which is given by  $C_{\text{ox}} = \epsilon_0 / t_{\text{ox}}$ , where  $\epsilon_0$  is the oxide permittivity and  $t_{\text{ox}}$  is the gate oxide thickness.

The saturation current of the NMOS device is:

$$I_{DSn} = \beta_n \frac{(V_{IN} - V_{Tn})^2}{2} \quad [2.2.5]$$

Where:

$$\beta_n = k_n \frac{W_{\text{eff}}}{L_{\text{eff}}} \quad [2.2.6]$$

And:

$$V_{GSp} = V_{IN} \quad [2.2.7]$$

Using equations [2.2.1, and 2.2.5], we can obtain an expression that represents the output voltage (figure 2.2.2 (a)):

$$V_O = (V_{IN} - V_{Tp}) - \sqrt{(V_{IN} - V_{Tp})^2 - 2 \left( V_{IN} - \frac{V_{DD}}{2} - V_{Tp} \right) V_{DD} - \frac{\beta_n}{\beta_p} (V_{IN} - V_{Tn})^2} \quad [2.2.8]$$

**Region C:** When  $V_{IN} = V_{INV}$ . In this case, both, NMOS and PMOS devices are in the saturation region. The PMOS current is given by:

$$I_{DSP} = -\beta_p \frac{(V_{IN} - V_{Tp})^2}{2} \quad [2.2.8]$$

The current flowing through the NMOS device is given by the equation [2.2.5]. By equalizing both equations [2.2.5 and 2.2.8] we can obtain the expression that represents the  $V_{INV}$ :

$$V_{INV} = \frac{V_{DD} + V_{Tp} + V_{Tn} \sqrt{\beta}}{1 + \sqrt{\beta}} \quad [2.2.9]$$

Were  $\beta = \frac{\beta_n}{\beta_p}$  and  $V_{Tn}=V_{Tp}$ . If we consider  $\beta_n=\beta_p$  in a CMOS process defined by:

$$\frac{k_n}{k_p} = \frac{\mu_n}{\mu_p} \approx 2 - 3 \quad [2.2.10]$$

If we consider the following dimension ratio:

$$\left\{ \frac{W_{eff}}{L_{eff}} \right\}_p = 2.5 \left\{ \frac{W_{eff}}{L_{eff}} \right\}_n \quad [2.2.11]$$

We obtain:

$$V_{IN} = V_{INV} = \frac{V_{DD}}{2} \quad [2.2.12]$$

This kind of inverter is called "symmetrical gate". Nevertheless, the output voltage is not necessary equal to  $V_{DD}/2$  and is given by  $V_{IN} - V_{Tn} < V_O < V_{IN} + V_{Tp}$ .

**Region D:** When  $V_{INV} < V_{IN} < (V_{DD} + V_{Tp})$ . In this case, the NMOS is in the linear region and the PMOS is in the saturation region. If we consider the same conditions from Region B, we can obtain:

$$V_O = (V_{IN} - V_{Tn}) - \sqrt{(V_{in} - V_{Tn})^2 - \frac{\beta_p}{\beta_n} (V_{IN} - V_{DD} - V_{Tp})^2} \quad [2.2.13]$$

**Region E:** When  $(V_{DD} + V_{Tp}) < V_{IN} < V_{DD}$ . In this case, the NMOS is ON and the PMOS is operating into the sub-threshold region. If we assume that the current flowing through this device is almost zero, then  $V_O=0$ .

From figure 2 (b), we can notice that when  $V_{IN}=V_{INV}$ , the DC power dissipation is maximal. It is also called short circuit power consumption.

## 2.2.2 Power Consumption of Complementary CMOS

The power consumed by the CMOS inverter, and by all CMOS circuits, can be divided into three components [10, 59 & 81]:

1. The Static Power Consumption caused by the leakage current  $I_{leak}$  and other static current  $I_{ST}$  due to the value of the input voltage.
2. The Dynamic Power Consumption caused by the charge and discharge of the total output capacitance  $C_L$ .

3. The Dynamic Power Consumption caused by the short-circuit current  $I_{SC}$  during the switching transient (also called short-circuit Power Consumption).

### 2.2.2.1 Static Power

There are two sources of static power in a complementary CMOS inverter: the leakage currents; and current drawn from the supply due to the input voltage. The total static power consumption can be expressed by the following equation:

$$P_{STAT} = P_{leak} + P_{dp} \quad [2.2.14]$$

The leakage currents are caused by the parasitic diodes in a CMOS inverter. When the input voltage is not changing, the parasitic diodes are not conducting. According to [10] the current in a diode is given by:

$$I_d = I_s \left( e^{\frac{qV_d}{nkT}} - 1 \right) \quad [2.2.15]$$

Where  $n$  is the emission coefficient of the diode (sometimes  $n=1$ ) and  $V_d$  is the applied voltage to the diode. The total power consumption caused by the leakage currents is:

$$P_{leak} = \sum_i I_{di} V_{DD} \quad [2.2.16]$$

A typical value of  $I_d$  is 1 fA per device. In a pure CMOS circuit containing a million of devices, the total  $P_{leak}$  would be equals to 0.01  $\mu$ W. The power dissipation due to the leakage currents could be neglected. Anyway in circuits containing memory cells, this power consumption could be more important.

The second component of the static power is a function of the input voltage. Assume that the input of the pull-down NMOS is at a voltage  $0 < V_{IN} < V_T$ . In this case, the current is given by the sub-threshold expression:

$$I_{dp} = I_o \frac{W_{eff}}{W_o} 10^{\frac{(V_{IN}-V_T)}{S}} \quad [2.2.17]$$

Where  $V_T$  is the constant-current threshold voltage,  $I_o$  and  $W_o$  are the drain current and the gate width to define  $V_T$ , and  $S$  is the sub-threshold swing parameter. The current  $I_o$  is related to  $V_{DS}$  by:

$$I_o = I_o' \left( 1 - e^{-V_{DS}/V_t} \right) \quad [2.2.18]$$

According to [10], the sub-threshold swing is given by:

$$S \approx 2.3V_t \left( 1 + \frac{C_d}{C_{ox}} \right) \text{ V/decade} \quad [2.2.19]$$

Where  $C_d$  is the depletion-layer capacitance of the source/drain junctions. According to A. Bellaouar, M. I. Elmasry (1995) [10],  $S$  has a theoretical minimum limit of 60 mV/decade. When  $V_{IN} > V_T$ , the current can be expressed as follows:

$$I_{dp} = \frac{W}{LC_{ox}} (V_{IN} - V_T)^{1.5} \quad [2.2.20]$$

Where  $C_{ox}$  is the gate oxide capacitance,  $L$  and  $W$  are the average width and large of the device. The power dissipation caused by the direct-paths currents is:

$$P_{dp} = I_{Dmean} V_{DD} \quad [2.2.21]$$

For a CMOS circuit with more than a million of transistors, This source of static power consumption could be important. Static power consumption increases with temperature. Even if CMOS circuits have been designed to consume energy only during switching, in recent low-power applications with CMOS, the  $V_T$  is becoming low and the static power due to direct-pats current is becoming important.

### 2.2.2.2 Dynamic Power Caused by Load Capacitance

This source of power consumption is due to the currents needed to charge and discharge the effective load capacitance  $C_L$  of figure 2.2.1. Let assume a step input so neither the N and P devices are on simultaneously. The average dynamic power  $P_d$  required to charge and discharge the  $C_L$  during a clock period  $T$  is:

$$P_d = \frac{1}{T} \int_0^T i_o(t) v_o(t) dt \quad [2.2.22]$$

The output current needed to charge  $C_L$  is given by:

$$i_o = i_p = C_L \frac{dv_o}{dt} \quad [2.2.23]$$

And the current flowing through the NMOS during the discharge phase:

$$i_o = i_n = -C_L \frac{dv_o}{dt} \quad [2.2.24]$$

The equation 2.2.22 becomes:

$$P_d = \frac{1}{T} \left[ \int_0^{V_{DD}} C_L v_o dv_o - \int_{V_{DD}}^0 C_L v_o dv_o \right] \quad [2.2.25]$$

The dynamic power dissipation can be expressed as:

$$P_d = \frac{C_L V_{DD}^2}{T} = C_L V_{DD}^2 F \quad [2.2.26]$$

Where  $F$  is the operation frequency of the circuit.

From equation [2.2.26], we can notice that the dynamic power consumption is proportional to  $F$  and  $V_{DD}$ . If the supply voltage is reduced, power consumption will be reduced by a quadratic factor. This equation is only valid for the CMOS-Inverter, but it can be used to determine an equivalent expression for a complex circuit.

### 2.2.2.3 Dynamic Power Caused by Short-Circuit Currents

Even if there are no load capacitance on the output and the parasitic capacitance are negligible, the CMOS-Inverter would still dissipate switching energy. If the Input voltage changes slowly, both the P and N devices are ON. An excess power is dissipated due to the short-circuit current. This current depends on the rising and falling times of the input voltage. If we assume that the falling and rising times are equivalent, the power consumed by the short-circuit current is:

$$P_{SC} = I_{mean} V_{DD} \quad [2.2.27]$$

Where  $I_{mean}$  is estimated using the following figure:

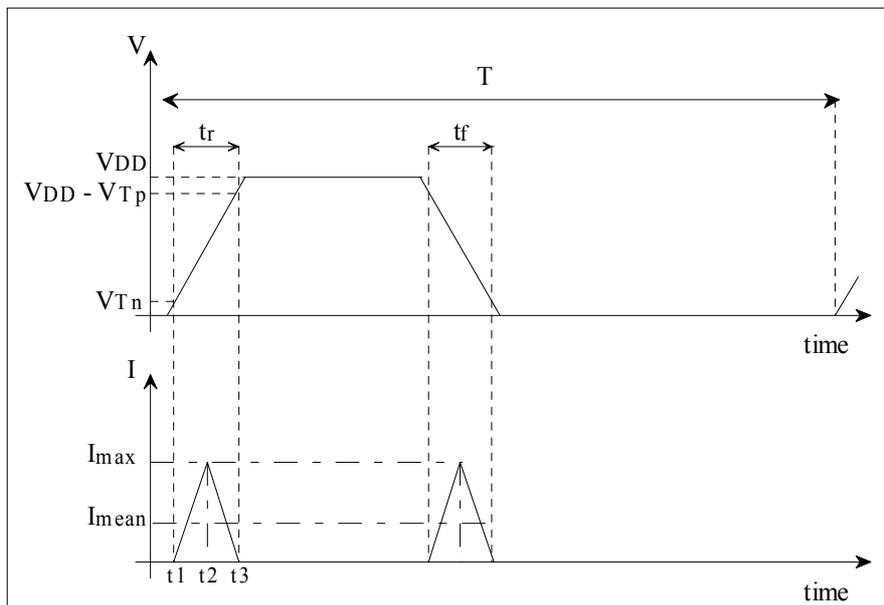


FIGURE 2.2.3. INPUT VOLTAGE AND SHORT-CIRCUIT CURRENT

We assume also that the CMOS-Inverter has symmetrical devices, which means that:  $\beta_n = \beta_p = \beta$  And  $V_{Tn} = V_{Tp} = V_T$ . We assume also that the rising time is equals to the falling time of the input signal ( $\tau_r = \tau_f = \tau$ ) The mean short-circuit current is given by:

$$I_{\text{mean}} = 2 \times \frac{1}{T} \left[ \int_{t_1}^{t_2} i(t) dt + \int_{t_2}^{t_3} i(t) dt \right] \quad [2.2.28]$$

Due to the symmetry, we have:

$$I_{\text{mean}} = \frac{4}{T} \left[ \int_{t_1}^{t_2} i(t) dt \right] \quad [2.2.29]$$

Since the NMOS is operating in the saturation region, the above equation becomes:

$$I_{\text{mean}} = \frac{4}{T} \left[ \int_{t_1}^{t_2} \frac{\beta}{2} (V_{\text{IN}}(t) - V_T)^2 dt \right] \quad [2.2.30]$$

The input voltage is:

$$V_{\text{IN}}(t) = \frac{V_{\text{DD}}}{\tau} t \quad [2.2.31]$$

It can be derived from figure 1 that  $t_1 = \frac{V_T}{V_{\text{DD}}} \tau$  and  $t_2 = \frac{\tau}{2}$ . Then the integral leads to:

$$P_{\text{SC}} = \frac{\beta}{12} (V_{\text{DD}} - 2V_T)^3 \tau F \quad [2.2.32]$$

This equation (used in [10, 63, 80]) shows that the short-circuit power consumption is also proportional to the frequency. For a circuit with several logic gates, when the rising and falling times are equivalent, the short-circuit power dissipation could be less than 20% of the total power consumption

## 2.2.3 Power Consumption of Pass-Transistor Structures

### 2.2.3.1 The NMOS Pass-Transistor

The NMOS pass-transistor is the most basic gating unit. The following figure illustrates a CMOS inverter preceded by a NMOS pass-transistor and the voltage transfer waveform.

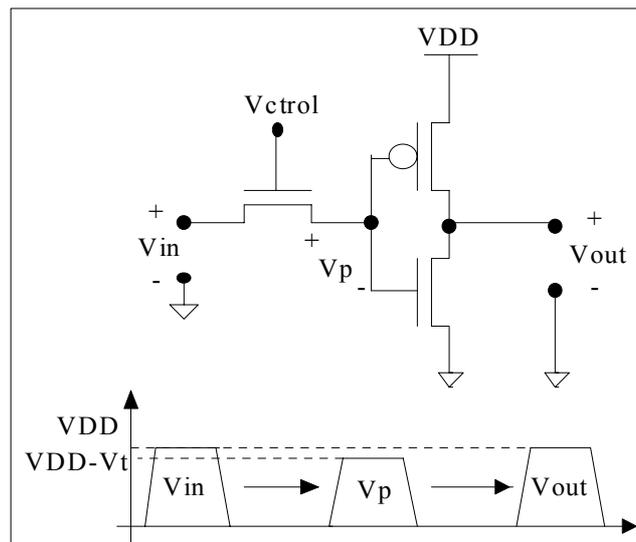


FIGURE 2.2.4. THE NMOS PASS-TRANSISTOR

The data transfer takes place only when  $V_{ctrl}$  is set high. The operational characteristics of the pass-transistor are based on the concept of charge transfer by flowing currents. The NMOS pass-transistor is currently used for controlling logic paths in synchronous logic circuits. The following analysis is based on the contribution of J. P. Uyemura (1988) [81].

#### a) Logic '1' transfer

The following figure shows a charge circuit using the NMOS pass-transistor (a) and the waveform of the transferred voltage  $V_{out}$  (b).

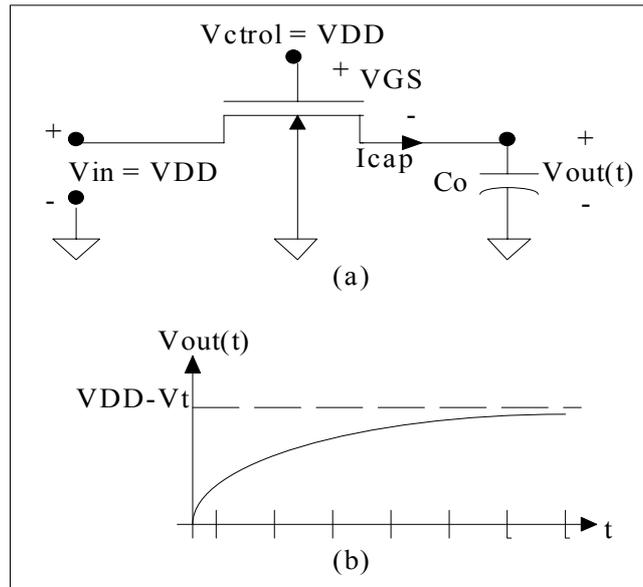


FIGURE 2.2.5. CHARGING CIRCUIT

Let  $C_o$  be the equivalent capacitance of the circuit that is driven by the pass-transistor. The initial conditions are  $V_{out}(0) = 0$  volts. The input value of the pass-transistor is  $V_{in} = V_{DD}$  and  $V_{ctrl}$  is also equals to  $V_{DD}$ . These conditions allow a current  $I_{cap}$  to flow.

To analyze this circuit, we must consider that:  $V_{GS} = V_{DD} - V_{out} = V_{DS}$ . The voltage  $V_{out}$  will vary from 0 volts to  $(V_{DD} - V_T)$ . In this case, the NMOS is saturated, and the current  $I_{cap} = I_{DRAIN}$  is:

$$I_{DRAIN} = \frac{\beta}{2} (V_{DD} - V_{out} - V_T)^2 \quad [2.2.33]$$

Where  $V_T$  is the sub-threshold voltage, and  $\beta$  is the transconductance parameter of the NMOS defined by equation [2.2.6]. It should be notice (from figure 2.2.5) that the low to high time ( $t_r$ ) is very low because the NMOS is always saturated. In this case, when  $V_{GD}$  decreases  $V_{OUT}$  increases, it reduces the current flow level ( $I_{cap} = dV_{OUT}/dt$ ).

*b) Logic '0' Transfer*

Now, let us consider the transfer of a logic '0' level through the pass-transistor that is illustrated in figure 2.2.6.

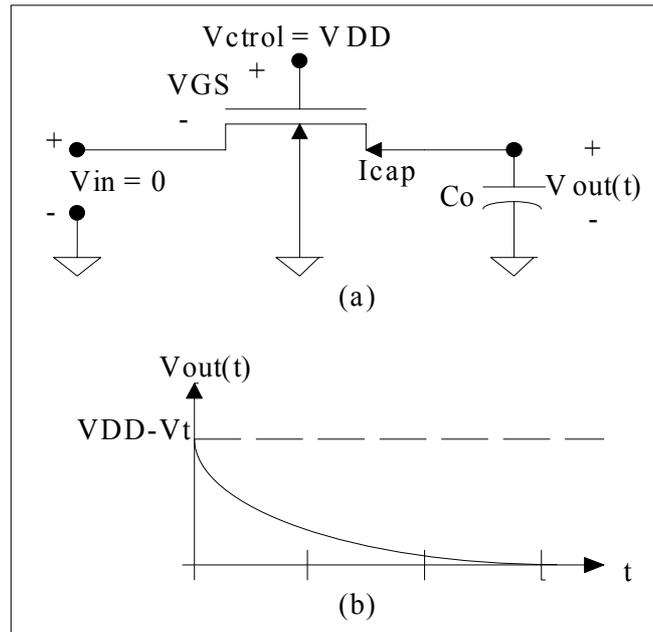


FIGURE 2.2.6. DISCHARGE CIRCUIT

To analyze this circuit, note that:  $V_{GS} = V_{DD} - V_{IN} = V_{DD}$  and  $V_{DS} = V_{out}(t)$ . Since  $V_{GS}$  is at the highest voltage of the system,  $V_{GS} > V_{DS}$  is always true, and the NMOS is in the non-saturation region. The  $I_{CAP}$  is given by:

$$I_{CAP} = \frac{\beta}{2} [2(V_{DD} - V_T)V_{out} - V_{out}^2] \quad [2.2.34]$$

Where the output voltage  $V_{out}$  can take values from  $V_{DD} - V_T$  to 0 volts. It should be noticed (from figure 2.2.6) that the falling edge time ( $t_f$ ) is very fast. It means that  $C_{OUT}$  discharges much faster than it charges. According to Uyemura (1988) [81] the rising edge time is 6.11 times higher than the falling edge time.

### 2.2.3.2 Static Power Consumption

The following figure shows a typical Pass-Transistor structure. NMOS Pass-Transistor device is used to enable or disable the connection between the input of the CMOS logic circuit and a precedent circuit.

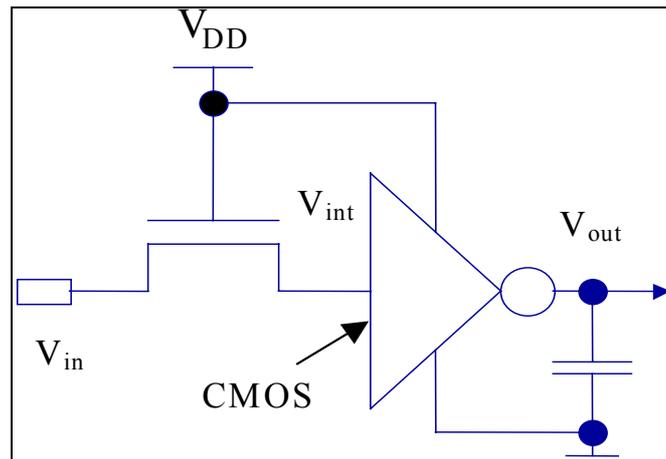


FIGURE 2.2.7 PASS-TRANSISTOR STRUCTURE

As well as the CMOS inverter, in this case there are two sources of Static Power:

1. Static power caused by junction leakage currents (as we explained in 2.2.2.1), and
2. Static power caused by direct-paths currents.

Since  $V_{int}$  takes values from 0 volts to  $V_{DD} - V_T$ , the PMOS device of CMOS inverter can not reach the cut-off region allowing an increase of the direct-path current. Figure 2.2.8 illustrates the voltage transfer and the current  $I_{stat}$  associated to this phenomena.

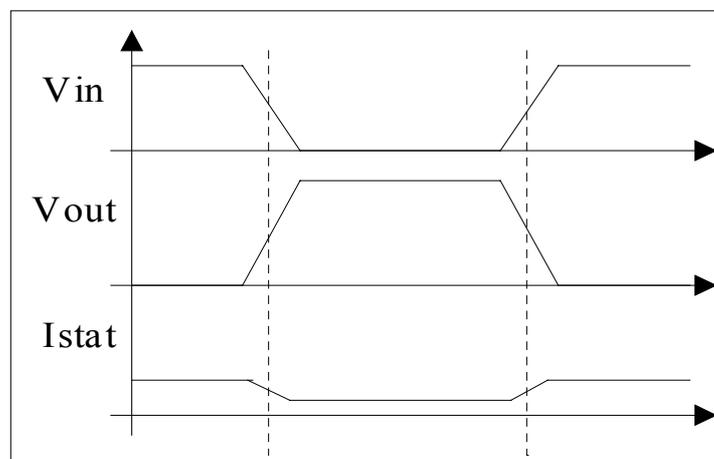


FIGURE 2.2.8. LEAKAGE CURRENT AND DIRECT-PATH CURRENT

Short circuit currents becomes important since the falling and rising edges times are more important. Figure 2.2.9 illustrates the ideal case of  $t_r = t_f$ . In section 2.2.3.1, we have shown that  $t_r$  is much higher than  $t_f$ . The  $I_{SC}$  is much more important during the low to high transitions.

### 2.2.3.2 Dynamic Power Consumption

Dynamic Power in pass-transistor structures is caused by the voltage transfer through the NMOS transistor. The dynamic current that traverses the transistor during transitions is the average value of the  $I_{DRAIN}$  from equation [2.2.33] and [2.2.34]. The dynamic power consumed by the pass-transistor can be expressed as follows:

$$P_{DYN} = V_{DD} I_{D\_MEAN} \quad [2.2.35]$$

Figure 2.2.9 shows the current consumed during transitions. We can identify here the two sources of dynamic power consumption:

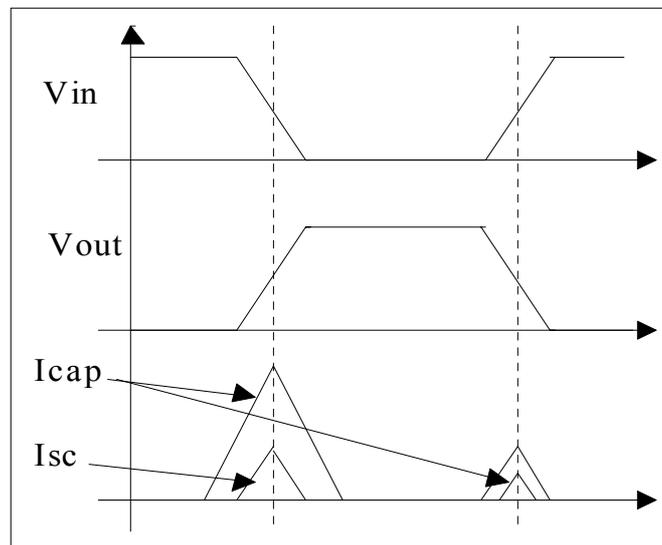


FIGURE 2.2.9. LOAD CAPACITANCE AND SHORT-CIRCUIT CURRENTS

The current consumed by a pass-transistor structure is illustrated in figure 2.2.10.

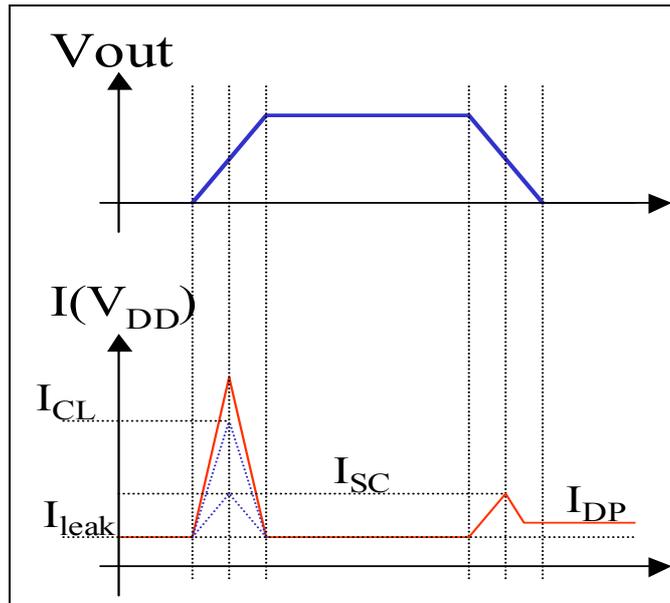


FIGURE 2.2.10. CURRENT OF A PASS-TRANSISTOR STRUCTURE

#### 2.2.4 Power Consumption of SRAM

Different sources of power consumption can be identified using the SRAM architecture of figure 2.2.11. The total power consumption can be divided in two components: The active power consumption and the standby power consumption.

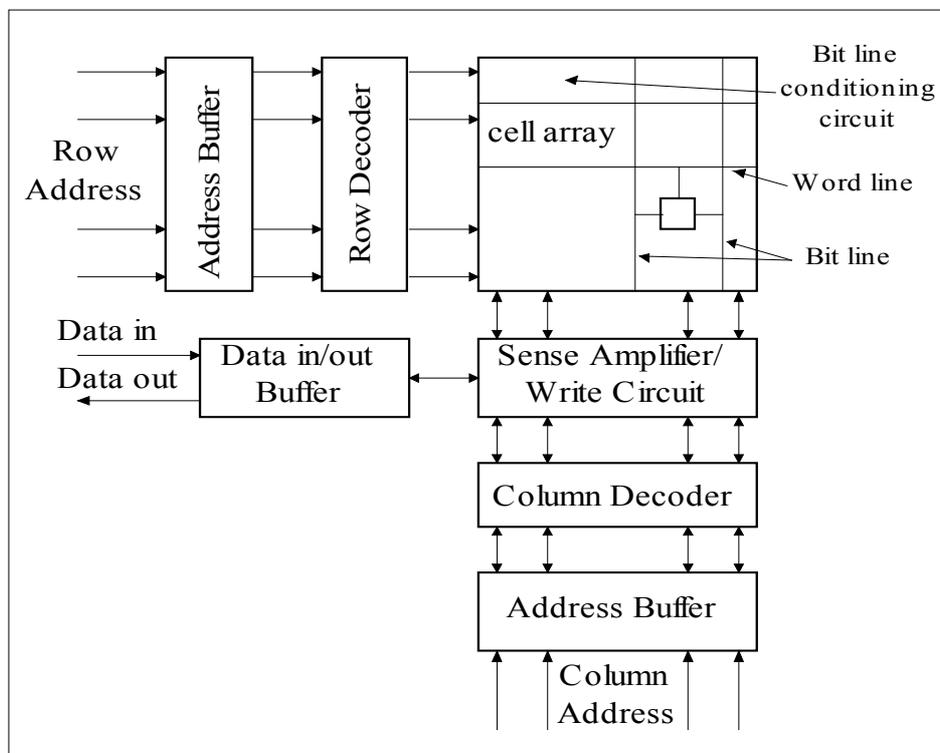


FIGURE 2.2.11. STATIC RAM ARCHITECTURE

The active power is the sum of the power dissipated by the following components:

- The decoders.
- The memory array. If  $m$  is the number of memory cells connected to the same word-line, the active power of the memory array in read mode can be expressed as:

$$P_{\text{mem-array}} = mP_{\text{act}} + (n - 1)mP_{\text{leak}} + mI_{\text{DC}} \Delta t F V_{\text{DD}} \quad [2.2.36]$$

Where  $P_{\text{act}}$  is the power dissipated in active mode when selecting the  $m$  cells and  $P_{\text{leak}}$  is the data retention (standby power) of the unselected memory cells in the  $m \times n$  array. The third term is due to the DC current,  $I_{\text{DC}}$ , during the read operation.  $\Delta t$  is the activation time of the DC consuming parts and  $F$  is the operating frequency ( $F = 1/t_{\text{RC}}$ ).

- Sense amplifiers, they are dominated mainly by DC current.
- Remaining periphery such as input/output buffer, write circuitry, etc.

The standby power (or retention current) of an SRAM has a mayor contribution from the memory cells in the array if the sense amplifiers are disabled in this mode. It can be given by:

$$P_{\text{standby}} = mnP_{\text{leak}} \quad [2.2.37]$$

## 2.2.5 Power Consumption of Input/Output Circuits

The Input/Output (I/O) circuits permit the on-chip logic circuitry to communicate to the external world. The I/O circuits are important in the limitation of speed and power consumption of the entire circuit. There are several kinds of I/O circuits, such as input and output buffers, clock distribution, clock buffering and low-swing I/O. In this section, an overview of I/O circuits based on [10 and 81] is presented.

### 2.2.5.1 Input Circuits

In order to distribute an input signal to the whole circuit, an input buffer is needed. Input buffers are normally formed by at least one inverter. In this section, the power consumption behavior of a TTL to CMOS input circuit is exposed. The power consumed by this circuit is divided in two components: Static and dynamic.

a) Static Power Dissipation

The CMOS input buffer is used to translate the TTL (Transistor-Transistor Logic) or the Low-Voltage TTL levels to CMOS levels. The TTL levels are defined as follows: 0.8 volts for the low-level input maximum, and 2.0 volts for the high-level input minimum.

The inverters that form the input buffer are designed by setting their W/L ratio such that the switching point of the buffer is near 1.4 volts (middle of  $V_{IL}$  and  $V_{IH}$ ). However, since the TTL voltage swing is limited to 1.2 volts, the input buffer is always dissipating DC power. The circuit depicted on the following figure is an example of a 2-stage input buffer.

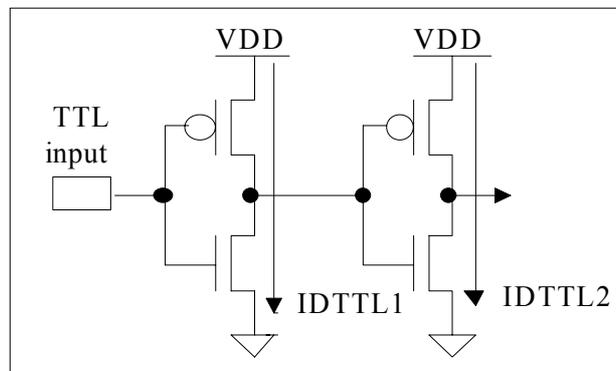


FIGURE 2.2.12. TTL INPUT BUFFER

If the first inverter can not be able to transfer the TTL level, the second one will dissipate some DC power. The static power consumption of this buffer is:

$$P_{TTL} = V_{DD} I_{DTTL} \quad [2.2.38]$$

Where,

$$I_{DTTL} = I_{DTTL1} + I_{DTTL2} \quad [2.2.39]$$

$I_{DTTL}$  is the average current flowing through both inverters when the input is at low and high levels. When the number of a TTL input pads is large, the DC current of the input buffers becomes important.

*b) Dynamic Power Consumption*

For a circuit that contains several I/O pads, the total dynamic power consumption of all input pads can be expressed as follows:

$$P_{\text{inputs}} = \alpha N_i E_{ii} F \quad [2.2.40]$$

Where  $\alpha$  is the switching activity,  $N_i$  is the number of input pads,  $E_{ii}$  is the internal energy of the input pad in Watt/Hz and  $F$  is the operational frequency of the circuit.

*2.2.5.2 Output circuits*

The output buffer must have the ability to drive an important load capacitance (fan-out) maintaining adequate rise and fall times. Normally, an inverter chain that can handle the large capacitance formed by the pad, the package wiring and the off-chip load forms the output circuit. The following figure shows a tri-state output buffer.

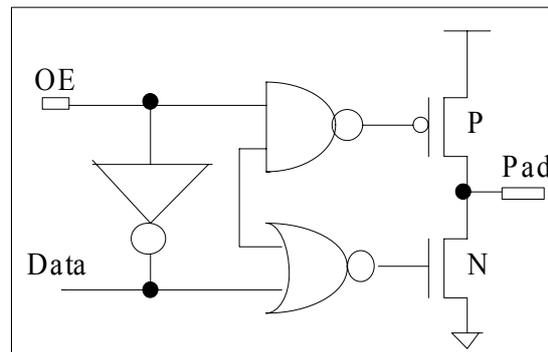


FIGURE 2.2.13. TRI-STATE OUTPUT BUFFER

When the Output Enable signal (OE) is high, the output data is the same that the input data. When OE is low, the pad is set to high impedance (Z). Both the NMOS and the PMOS are cutoff. Figure 2.2.14 illustrates a bi-directional output circuit.

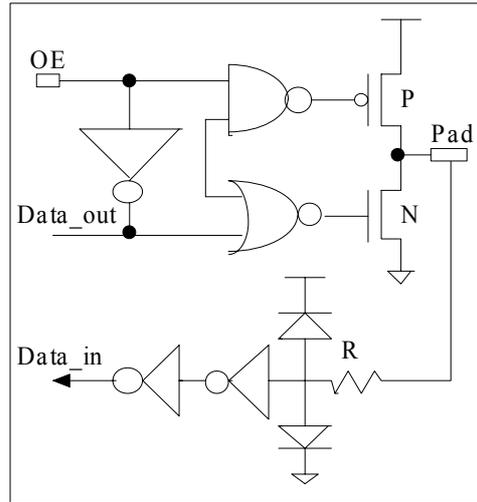


FIGURE 2.2.14. BI-DIRECTIONAL INPUT CIRCUIT

*a) Power Consumption of output circuits*

The power consumed by the output pads can be divided into two components: static and dynamic. The static power consumption is caused by the junction leakage currents of the transistors that form the buffers and by the sub-threshold current from the input voltage. When  $V_T$  is small, the DC power dissipation becomes important due to the sub-threshold currents. The static power consumption for the output pads when driving a CMOS TTL load is given by:

$$P_{\text{static}} = N_O V_{DD} (I_{DS\_mean} + I_{\text{leak}}) \quad [2.2.41]$$

Where  $N_O$  is the number of output pads,  $I_{DS\_mean}$  is the average sub-threshold current when the input is low and high, and  $I_{\text{leak}}$  is the current caused by junctions. If the output pad has to drive a bipolar TTL charge, the output buffer is forced to sink significant amounts of currents (due to the bipolar input transistor). The static power consumed by one output buffer driving a bipolar input pad is:

$$P_{\text{Stat\_TTL}} = V_{OL} I_{OL} \quad [2.2.42]$$

Where  $I_{OL}$  is the current sunk by the output buffer and is equals to the sum of the current from all the bipolar inputs.  $V_{OL}$  is the minimal output voltage when the output data is '0' ( $V_{OL} = 0.4$  volts).

The dynamic power consumed by the output circuits can be expressed by the following equation:

$$P_{DYN} = \alpha(N_O E_{iO} + N_O C_O V_{DD}^2)F \quad [2.2.43]$$

Where  $E_{iO}$  is the internal switching energy of the output pad,  $C_O$  is the average output load capacitance,  $N_O$  is the number of output pads,  $F$  is the clock frequency and  $\alpha$  is the average switching rate of all output pads.

### 2.2.6 Power Consumption in Clock Circuits

The current way to distribute the clock signal on-chip is using input buffers that have the ability to drive high internal load with fast fall/rise times.

Example: Consider a 3.3-volt micro-controller working at 200 MHz, with an internal load for the clock driver equals to 3.2 nF. In this case, the rise/fall times should be equal to 0.5 nS ( $T_{clock} = 5$  nS); according to A. Bellaouar, M. I. Elmasry (1995) [10], the average transient current would be:

$$I_{AVE} = C \frac{\Delta V}{\Delta t} = \frac{3.2 \times 10^{-9} \times 3.3}{0.5 \times 10^{-9}} = 21 \text{ A} \quad [2.2.44]$$

And the dynamic power consumed only by this clock circuit is:

$$P_{dyn} = C V_{DD}^2 F \approx 7 \text{ W} \quad [2.2.45]$$

This is a good example to illustrate the importance of the clock circuit in terms of power. An architectural strategy should be used to distribute the clock signal to the whole circuit with minimum clock skew and low-power consumption.

The following figure shows two examples of clock distribution circuit. The circuit described in (a) consists of cascade inverters; in this case, the last buffer can drive a very high load and feed all the internal blocks of the circuit. The second option (b) consists of a tree of buffers that accomplishes the clock distribution.

In this case, identical buffers have to be used at each level and each buffer has to see the same load capacitance. The buffers of the last level drive the functional elements directly.

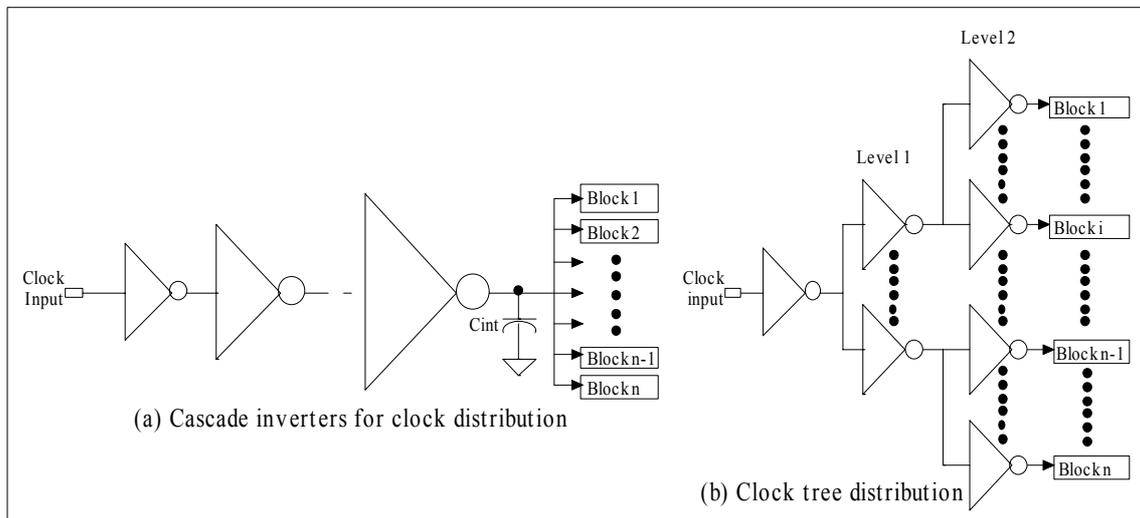


FIGURE 2.2.15 CLOCK DISTRIBUTION

Several techniques can be used to reduce the power dissipation of the clock buffer: The equivalent load capacitance of the clock buffer can be reduced by using low capacitance clock routing lines; the use of low-swing drivers at the top level of the clock tree (b).

### 2.3 Power Consumption in SRAM-based FPGAs.

Recent FPGA architectures are formed by different types of technologies and elements. Logic Elements are formed by LUTs and DFFs. LUTs can be constructed using SRAM cells, and DFF (even if there are different ways to build a DFF) is normally a CMOS device. Embedded Memory Cells must be constructed using SRAM cells. Finally, the interconnect resources must be programmed using SRAM cells that controls pass-transistors. Pass-transistors are used like switch to enable or disable all the internal elements of a FPGA.

If we consider all these elements, the SRAM-based FPGAs are formed by three different technologies: SRAM, pure CMOS and Pass-Transistors. It means that the power consumption modeling in FPGAs becomes more complex than pure CMOS [23, 59].

Power in FPGAs is not only a function of  $V_{DD}^2$ . Since they contain a lot of pass-transistors, short-circuit currents are not negligible in these devices. According to equation 2.2.32, power consumption must be also a function of  $V_{DD}^3$ .

Finally, the direct-paths current in pass-transistor structures becomes important, this factor added to the static current dissipated for all the SRAM cells used in a FPGA, makes that the static power consumed by a FPGA must be considered and it is not negligible.

The Total power consumption of a FPGA can be represented using the following equation:

$$P_{FPGA} = \alpha V_{DD}^3 + \beta V_{DD}^2 + \delta V_{DD} \quad [2.2.46]$$

In this equation, we can assume that  $\alpha$  is the element that corresponds to the dynamic power consumption caused by short-circuit currents ( $V_{DD} - 2V_T$ ), and by a portion of the static power caused by direct-paths currents.  $\beta$  corresponds to the dynamic power caused by the charge and discharge of the load capacitance and  $\delta$  corresponds to the static power consumption.

#### ***2.4 Related Works on Power Consumption in FPGAs***

Power consumption systems based on FPGAs can be optimized at different levels of abstraction such as circuit level, architecture (or algorithm) level and system level. Several works at the different levels have been developed. Some of them propose new FPGA architectures since FPGA is not a truly low power technology and others propose the use of architectural techniques to save power in commercial FPGAs.

Most recent works propose the use of partial reconfiguration to reduce the power consumed by the programmable logic in digital systems. The Power dissipated by the programmable logic in systems based on FPGAs is hard to estimate because it depends on the internal resources used, the complexity of the input vectors and the internal

switching activity.

These factors can produce an unforeseen power overhead that drastically increases the package temperature. Chip temperature can also be produced by an inadequate implementation of the configuration process. E. Boemo and S. Lopez-Buedo (1997) [15] propose the use of ring oscillators placed at the 4 corners of the FPGA to sense its package temperature. This FPGA-oriented temperature-monitoring scheme permits the user to verify the temperature of a FPGA in order to detect and correct several problems. This technique allows us to obtain sensitivities of 17KHz per °C and 77 MHz per °C.

Power in FPGAs can be optimized at the architectural level according to E. Boemo, G. Gonzalez de Rivera, S. Lopez-Buedo, J. M. Meneses (1998) [16]. In this work, the authors have shown that the use of wave pipeline architectures allows the power consumption of a FPGA to be reduced by about 25% to 40%. They have also shown that power can be reduced by almost 45% when improving block partitioning.

S. R. Park, W. Burleson (1999) [51 & 52] propose the use of partial reconfiguration to manage the power consumption of the programmable logic in DSP systems based on FPGA such as motion estimation in MPEG encoders. The main goal of this work is to avoid unnecessary computation by monitoring the input vectors. Identical function blocks can be duplicated, or suppressed, as needed using partial reconfiguration. This technique allows power to be reduced since the internal resources of the FPGA are customized by the input vectors.

Most of the research works on low power in FPGAs have been developed at the circuit level. L. Carloni, P. Chong, E. Kusse (1997) [22] have proposed a new fine grain pass transistor FPGA. This architecture (similar to the Xilinx™ architecture) uses a 0.6 μm CMOS process with 3 levels of metal and NMOS pass transistor to reduce internal capacitances. In this work, the authors have proved that this type of architecture based on pass-transistors can support a supply voltage near  $2V_T$ . Another FPGA architecture is proposed by A. Tisserand, P. Marchal and C. Piguet (1999) [78]. This novel architecture, called Field Programmable On-line oPerators (FPOP), is formed by a

hierarchical structure of three levels.

The first level is formed by analog to digital (A/D) and digital to digital (D/D) converters that permit the input vectors to be encoded into a redundant code, which is needed to the on-line arithmetic. The second level corresponds to the on-line arithmetic computation formed by a matrix of linear and non-linear operators with memory cells. The last level is formed by A/D and D/D converters that can deliver the results in analog or digital manner. The FPOP architecture, based on a serial-pipelined structure, allows power consumption to be reduced while maintaining an acceptable time delay.

Finally, Varghese George, Hui Zhang, Jan Rabaey (1999) [83] proposes a novel FPGA architecture based on Xilinx™ architectures. In this work, the authors also propose a power distribution model of FPGAs. This model shows that most of the power dissipated in a FPGA is caused by the interconnect. Based on this, the authors propose some modifications to the interconnect architecture by incorporating nearest neighbor connections, symmetrical mesh architecture and hierarchical connectivity. They also optimized the logic block structure by using four 3-input LUTs with multiplexers and DFFs. This structure can implement any 5-input Boolean operation as well as 2-bit arithmetic operations. This architecture allows the time delay to be reduced, and thus power consumption is optimized.

## **2.5 Summary**

In the first section of this chapter, we have described the different internal architectures of commercial FPGAs in a general way. Since this work is not dedicated to optimize power consumption of a specific FPGA family, any commercial architecture was used to explain the complexity of FPGA architectures. From section 2.1, we can identify the internal elements that form all FPGA architectures:

- Logic Elements or Logic Blocks. These elements are basically formed by 4-input and 3-input Look-Up Tables with a programmable D-type flip-flop.
- Input/Output cells. These cells can be programmed as input, output or bi-directional. Each I/O cell contains at least one programmable D-type flip-flop.
- Interconnect resources, formed by different types of wires with different lengths, and by programmable interconnect switches.

- The embedded memory cells. These cells can be used to build RAM, ROM, DPRAM, FIFO or LIFO blocks.

Section 2.2 is an overview of the power consumption in MOS-based circuitry. First at all, we have explained the CMOS-inverter, since this component is the base of most of the CMOS power consumption model. We explained also the power consumption in pass-transistor structures, SRAM, I/O circuits and clock circuits. This study allows us to identify another important element of FPGAs that could be important to estimate power consumption: the clock tree.

Based on all theoretical elements, and the specific FPGA architectures formed by CMOS logic and pass-transistors, the power consumption behavior in FPGAs is different to other circuits (like micro-controllers, DSPs or Asics). Equation [2.2.46] summarizes all the theoretical aspects presented in this chapter. Exhaustive measurements and test have been realized to validate this assumption. Finally, section 2.4 presents the most important related works on power consumption in FPGAs. In the following chapter we will present the different power consumption models proportioned by vendors. Then, we will explain the incremental methodology used to obtain our model, and finally we will realize some comparisons between all these models.



## **Chapter III. Power Modeling on FPGAs**

### ***3.1 Introduction***

As we explained in chapter 2. FPGAs are not pure CMOS devices; it makes the power modeling more complex. In order to obtain a more accurate model, a measurement methodology called "incremental" has been used giving many exhaustive measurements. This methodology allows us to obtain the contribution of power consumption from each internal sub-element inside a FPGA. The model is represented by a distribution graphic of the global power consumption.

In this chapter we will describe the incremental methodology used to obtain a power consumption model of commercial FPGAs. This model brings more accurate results than models proposed by vendors, as we will show in section 3.5. Finally, this model is used to inspire some ideas about power optimization at circuit and architectural levels.

### ***3.2 Power Consumption model from Vendors***

In this section we present some power consumption models for commercial FPGAs. These models are proposed by some FPGA vendors to estimate the power consumed by their products. Some models consist of a set of equations based on the number of Logic Cells and I/O cells. For the most recent FPGA families (Virtex™ and Apex™), producers only delivers a tool to estimate power consumption of these devices and the set of equations is not available.

### 3.2.1 Altera™

#### 3.2.1.1 The Flex™ 10K family

The power consumption model proposed by ALTERA™ is based on the following equation:

$$P_{TOT} = P_{INT} + P_{IO} \quad [3.2.1]$$

Where  $P_{INT}$  is the power consumed by the internal elements and  $P_{IO}$  is the power dissipated by the I/O cells when the device is programmed. The internal power consumption is calculated as follows:

$$P_{INT} = (I_{CC\_STANDBY} + I_{CC\_ACTIVE})V_{CC} \quad [3.2.2]$$

Where  $I_{CC\_STANDBY}$  is the current consumed by the internal resources when clock signal is not changing and it has a fixed value of 500  $\mu$ Amps.  $I_{CC\_ACTIVE}$  is the current consumed when the circuit is working, The expression that describes this current is:

$$I_{CC\_ACTIVE} = K F_{MAX} N \text{tog}_{LC} 10^{-6} \text{ Amps.} \quad [3.2.3]$$

Where K is a constant that can take different values depending on the device size (from 90 to 104),  $F_{MAX}$  is the maximum operation frequency in MHz, N is the number of used (or programmable) logic cells and  $\text{tog}_{LC}$  is the average toggling or switching rate. The toggling rate proposed by this vendor is equal to 12.5% (derived from a 16-bit counter behavior).

The power consumed by the I/O cells is divided in two components:

$$P_{IO} = P_{AC\_OUT} + P_{DC\_OUT} \quad [3.2.4]$$

Where  $P_{AC\_OUT}$  is the power consumed by frequently switching outputs and  $P_{DC\_OUT}$  is

the power dissipated by steady-state outputs. The equation that describes  $P_{DC\_OUT}$  is:

$$P_{DC\_OUT} = \sum_{n=1}^d P_{DCn} \quad [3.2.5]$$

Where  $d$  is the number of DC outputs and  $P_{DCn}$  is the DC output power dissipated by the output  $n$ .  $P_{DCn}$  can take several values, depending on the load type.

$P_{AC\_OUT}$  depends on the load capacitance of each output and the frequency at which each output switches, as shown in the following equation:

$$P_{AC\_OUT} = \sum_{n=1}^a C_n V_n f_n V_{CC} \quad [3.2.6]$$

Where  $a$  is the number of outputs,  $C_n$  is the load capacitance of output  $n$  (in pF),  $V_n$  is the voltage swing of output  $n$ , and  $f_n$  is the switching frequency of output  $n$ . This element is computed using the following equation:

$$f_n = (0.5) F_{MAX} \text{tog}_{IO} \quad [3.2.7]$$

Where  $\text{tog}_{IO}$  is the average toggling rate of all outputs and  $F_{MAX}$  is the maximum frequency. The following equation is proposed to compute the total power consumed by frequently switched outputs:

$$P_{AC\_OUT} = (0.5) \text{OUT} C_{AVE} V_O F_{MAX} \text{tog}_{IO} V_{CC} \quad [3.2.8]$$

Where **OUT** is the number of outputs,  $V_O$  is the average voltage swing (when  $V_{CCIO} = 5$  volts,  $V_O = 3.8$  volts), and  $C_{AVE}$  is the average load capacitance (in pF).

### 3.2.1.2 APEX™ 20K Family

Altera™ propose for this last FPGA family a power estimator described in Java™ and that allows us to estimate the power consumption of all APEX™ devices. It can be used

on-line at the following Internet address:

[http://www.altera.com/html/products/power\\_calc.html](http://www.altera.com/html/products/power_calc.html).

This power calculator permits an estimation of the power consumed by the internal resources (DFFs and LUT), the embedded memory cells, and the I/O cells. This tool uses an average toggling rate equals to 12.5% (suggested by vendor).

### 3.2.2 Atmel™

#### 3.2.2.1 The AT40K™ family

The power consumption model proposed for the AT40K family contains two components: power consumption from internal resources and power consumption from I/O cells. Power in AT40K devices can be expressed as follows:

$$P_{AT40K} = P_{INT} + P_{I/O} \quad [3.2.9]$$

The power consumed by the internal resources of an AT40K™ depends on the number of logic cells, the interconnect resources and the frequency of operation used by the design. The following equation is proposed to estimate power consumption of these devices:

$$P_{INT} = (I_{CC} + I_{INT})V_{DD} \quad [3.2.10]$$

Where  $I_{CC}$  is the static current and is normally fixed to 0.25  $\mu$ Amps, and  $I_{INT}$  is the dynamic current consumed by the circuit and it can be divided in four components:

$$I_{INT} = I_{INT\_CLK} + I_{INT\_CORE} + I_{INT\_LOCAL} + I_{INT\_REP} \quad [3.2.11]$$

Where  $I_{INT\_CLK}$  is the current consumed by the clock network that is given by:

$$I_{INT\_CLK} = K_{CLK} F N_{CLK} \quad [3.2.12]$$

$I_{INT\_CORE}$  is the current from the core cells given by:

$$I_{INT\_CORE} = K_{CORE} F N_{CORE} \quad [3.2.13]$$

The current consumed by the local bus driver  $I_{INT\_LOCAL}$  is:

$$I_{INT\_LOCAL} = K_{LOCAL} F N_{LOCAL} \quad [3.2.14]$$

And  $I_{INT\_REP}$  is the current consumed by repeaters:

$$I_{INT\_REP} = K_{REP} F N_{REP} \quad [3.2.15]$$

The parameters used in the set of equations proposed are defined as follows:

- $K_{CLK}$  is the average current consumption of a core DFF toggling in  $\mu\text{Amps}/\text{MHz}/\text{driver}$ .
- $K_{CORE}$  is the average current consumption of a core driver in  $\mu\text{Amps}/\text{MHz}/\text{driver}$ .
- $K_{LOCAL}$  is the average current consumption of a local bus driver in  $\mu\text{Amps}/\text{MHz}/\text{driver}$ .
- $K_{REP}$  is the average current consumption of a repeater driver in  $\mu\text{Amps}/\text{MHz}/\text{driver}$ .
- $F$  is the frequency of operation.
- $N_{CLK}$  is the average number of core DFF.
- $N_{CORE}$  is the average number of core cell drivers.
- $N_{LOCAL}$  is the average number of core-cell local bus drivers.
- $N_{REP}$  is the average number of repeater drivers.

Table 3.1 contains the values of the internal current consumption parameters. These values were computed using a toggling rate equal to 25 % (specified by vendor).

Parameter	$V_{DD}$ (Volts)	Value [ $\mu\text{Amps}/\text{driver}$ ]
$K_{CLK}$	5.0	1.9
	3.3	1.4
$K_{CORE}$	5.0	6.0
	3.3	4.3
$K_{LOCAL}$	5.0	1.5
	3.3	1.1
$K_{REP}$	5.0	2.0
	3.3	1.4

TABLE 3.2.1. AT40K™ INTERNAL CURRENT PARAMETERS (SOURCE: AT40K DATA BOOK)

Like the FLEX family, the power dissipated by the I/O cells of the AT40K family is caused by two types of charge, a pull-up or pull-down resistor having DC dissipation, and a Load Capacitance producing AC dissipation. The power consumption of I/O cells is described using the following equation:

$$P_{I/O} = P_{AC\_OUT} + P_{DC\_OUT} \quad [3.2.16]$$

The power consumed by the AC outputs is computed using the following equation:

$$P_{AC\_OUT} = (0.5) C_{AVE} V_{DD}^2 F \text{tog} N_{AC} \quad [3.2.17]$$

Where  $C_{AVE}$  is the average load capacitance,  $F$  is the clock frequency,  $\text{tog}$  is the average switching rate and  $N_{AC}$  is the number of outputs. The power consumed by the DC outputs is calculated using the following equation:

$$P_{DC\_OUT} = \frac{V_{DD}^2}{20K} N_{DC} \quad [3.2.18]$$

Where  $20K$  represents the load resistance (assuming that the output is connected to a  $20K$  Ohm pull-up resistor), and  $N_{DC}$  is the number of DC outputs.

### 3.2.3 Xilinx™

#### 3.2.3.1 The XC4000X/XV/E™ Families

The power consumption model proposed by Xilinx™ for the XC4000™ series considers three sources of power consumption as described in the following equation:

$$P_{TOT} = P_{STAT} + P_{INT} + P_{IO} \quad [3.2.19]$$

Where  $P_{STAT}$  is the power dissipated by an inactive device connected to the power supply,  $P_{INT}$  is the power caused by the internal nodes switching, and  $P_{IO}$  is the power dissipated by output pads due to load capacitance. The static current given by this vendor is equal to 4 mAmps (i.e. for a 3.3 volt device,  $P_{STAT} = 13.2$  mW).

The equation that describes the internal power consumption for this family is derived based on data taken from a 16-bit counter. The power consumed by internal resources is estimated using the following equation:

$$P_{INT} = V_{CC} K_P F_{MAX} N_{LC} tog_{LC} \quad [3.2.20]$$

Where  $N_{LC}$  is the number of logic cells used by the applications,  $F_{MAX}$  is the clock frequency,  $tog_{LC}$  is the average toggling rate, and  $K_P$  is a constant that depends on family (it can takes values from 28 to 72 in units of  $\mu\text{Amps/MHz}$ ).

Equation [3.2.19] is used to calculate the power dissipated by the output pads.

$$P_{OUT} = (0.5)C_{OUT\_ave} F_{MAX} tog_{OUT} N_{OUT} V_{swing}^2 \quad [3.2.21]$$

Where  $C_{OUT\_ave}$  is the average load capacitance,  $F_{MAX}$  is the maximum clock frequency,  $tog_{OUT}$  is the average toggling rate of the outputs,  $N_{OUT}$  is the number of outputs and  $V_{swing}$  is the output swing voltage.

### 3.2.3.2 The Virtex™ Family

Like the Altera™ APEX™ family, the power consumption of VIRTEX™ devices can be calculated using a tool that allows to estimate the power consumed by Configurable Logic Blocks (LUT, DFF and CLB configured as RAM cell), embedded RAM cells, PLLs blocks and I/O cells. This tool can be used on-line at the following Internet address:

<http://support.xilinx.com/cgi-bin/powerweb.pl>.

### 3.2.4 Summary

In general, the power consumption models proposed by vendors are based on a set of equations that contains several factors such as the number of logic cells, the number of outputs, the average switching rate, the clock frequency and a K factor that depends on the density of the device. Some of these factors are easy to know (like the number of Logic Cells) but others are hard to obtain (like the toggling rate).

All these models ignore the influence of I/O cells configured as inputs, long lines, and other interconnect resources. Moreover, All vendors suggest a typical value of toggling rate of 12.5% or 25 % derived from the behavior of a 16-bit or an 8-bit counter.

Models proposed by vendors only provide a rough estimation of the power that could be consumed by typical applications in FPGAs, but they should not be used to specify the characteristics of a system based on FPGAs nor to design the power supply.

### **3.3 Incremental Methodology of Power Measurement**

#### 3.3.1 Introduction

As mentioned in precedent sections, power consumption in FPGAs depends on several factors, such as clock frequency, supply voltage, number of internal resources used by the application, load capacitance of each output pad, room temperature and switching (or toggling) rate.

Inside the FPGA, the power consumption can be related to a node capacitance. Each internal capacitance increments the propagation time and it consumes also. It means that the propagation delay permits the identification of the internal elements that consume (the elements containing an associated propagation delay can be considered as elements of the power consumption model).

Based on this, the following internal sub-elements of a FPGA an be identified:

- The Logic Cells, that are formed by LUTs and Registers.
- The Interconnect Resources.
- The I/O cells.
- The Clock buffer (Global distribution lines)
- The Embedded or distributed Memory

All these internal elements are contained in the netlist. The netlist is the binary code that allows the FPGA to be configured. The internal resources that have to be used by a specific application are defined in its netlist. The load capacitance of each internal node

is thus contained in the netlist.

If we consider all the factors described below, the power consumed by FPGAs can be described as a function of all these factors:

$$P_{\text{FPGA}} = V_{\text{DD}} F \text{tog}_{\text{AVE}} g(T, \text{netlist}, K) \quad [3.3.1]$$

Where  $V_{\text{DD}}$  is the supply voltage,  $F$  is the clock frequency,  $\text{tog}_{\text{AVE}}$  is the average toggling rate of the circuit, and  $g$  is a function of the room temperature, the netlist and a  $K$  factor which represents the technology process.

The toggling rate is the most complicated factor to determine because its value is hard to compute (because of glitches), and even an approximation is hard to obtain because we need information from each internal node.

Our work only analyzes the power consumption from the netlist. Power supply, room temperature, clock frequency, and technology ( $K$ ) will be considered constant. We propose to use the simplest designs with a fixed toggling rate, and change only the number of internal resources in the design in order to obtain the power consumed by each internal element. Figure 3.3.1 illustrates the power distribution inside the FPGA.

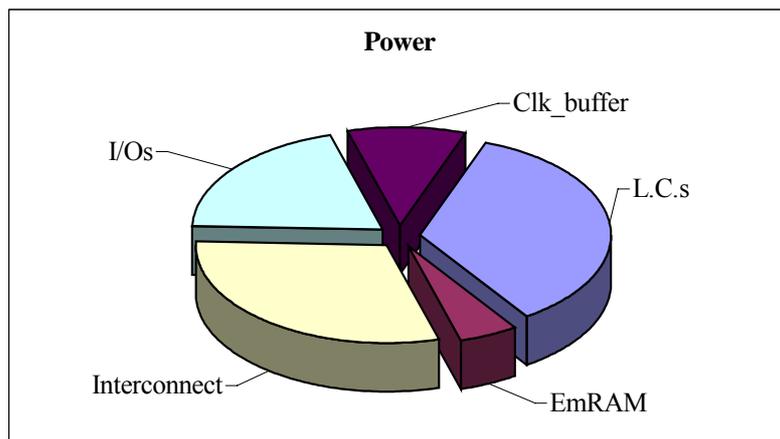


FIGURE 3.3.1. AN EXAMPLE OF THE POWER DISTRIBUTION INSIDE A FPGA

The idea is to increase the number of one internal element while keeping constant the others. This methodology (called incremental) allows a more accurate value of the power consumed by each internal element to be obtained. From equation [3.3.1], the

power consumption of the netlist can be decomposed as follows:

$$P_{\text{FPGA}} = V_{\text{DD}} F \log_{\text{AVE}} [\alpha_{\text{LC}} + \beta_{\text{interconnect}} + \delta_{\text{Clk}} + \epsilon_{\text{I/O}} + \varphi_{\text{Memory}}] g(T^{\circ}\text{C}, \text{K}) \quad [3.3.2]$$

Where  $\alpha_{\text{LC}}$  is the current consumed by the Logic Cells (Logic Elements or CLBs);  $\beta_{\text{interconnect}}$  is the current consumed by the interconnect;  $\delta_{\text{Clk}}$  is the current consumed by the clock tree;  $\epsilon_{\text{I/O}}$  is the current consumed by the I/O cells; and  $\varphi_{\text{Memory}}$  is the current consumed by the memory cells, all of them represented in units of mA/MHz. According to section 2.3 and the equations from section 2.2, all these constants depend on  $V_{\text{DD}}$  and  $V_{\text{T}}$ . For the purpose of this study,  $V_{\text{DD}}$  is always constant.

### 3.3.2 Test Platform

The following figure shows the measurement platform used:

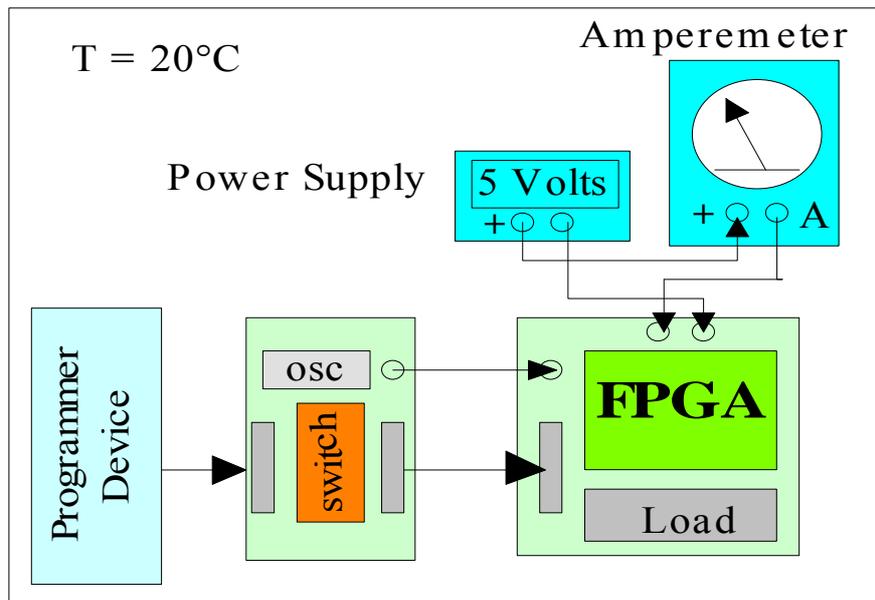


FIGURE 3.3.2. TEST PLATFORM

The devices used in this work are the Altera™ Flex™10K100-BGA-504, the Xilinx™ and XC4003E-PC-84.

Two boards have been used to isolate the FPGA. The first board contains only the FPGA and the load capacitance. The second board is used to connect and disconnect the programmer device and to generate the clock signal.

A digital amperemeter with a high resolution is used to observe the increment of current when we increase the number of internal resources.

### 3.3.3 Measurement Methodology

The methodology used to obtain the current consumed by each internal sub-element of a FPGA can be described as follows:

1. First, we start with a simple design containing a minimum number of internal elements, the toggling rate is fixed and room temperature is considered constant.
2. Then, one of the internal resources (i.e. wires) is increased while keeping the others constant. The difference in the current consumption is computed to estimate the power consumed by this element. This procedure is followed using the floorplan editor from vendors in order to control perfectly the place and route of the netlist.
3. The interconnect must be the first internal element to be measured because it always appears when using internal logic or I/Os. It is clear that if the power consumed by the interconnect is estimated first, then the power consumed by the others must be easily estimated.
4. Finally, the procedure is repeated by each internal element, in some cases, precedent results help to deduce the power consumed by some sub-elements.

#### 3.3.3.1 Interconnect Resources

This feature considers all the internal wires and all the interconnect levels. It can be illustrated using the following equation:

$$\beta_{\text{Interconnect}} = \sum_{i=1}^n N_i \beta_i \quad [3.3.1]$$

Where  $\beta_i$  represents one of the hierarchical levels of the interconnect (i.e. in Flex10K

family: LAB interconnect, Fast Tracks and Columns), and  $N_i$  is the number of interconnect resources used by the application. The following figure shows one of the designs used to estimate the power consumed by the interconnect resources:

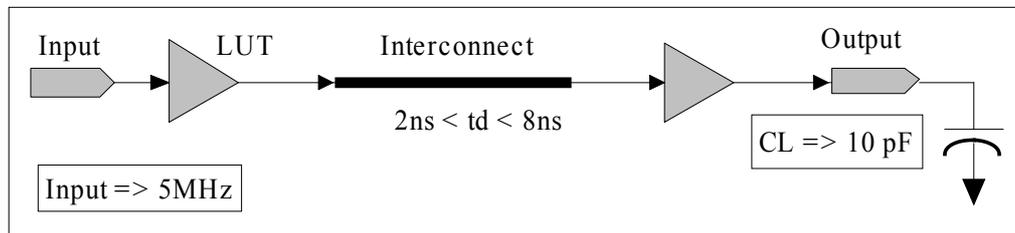


FIGURE 3.3.3. MEASUREMENT OF THE INTERCONNECT POWER CONSUMPTION

In this case, the number of Inputs, Outputs and Logic Cells used are constants. First of all, we measured the current consumed by two adjacent LUT and we considered this result as the reference (offset) value.

Using the Floorplan Editor from MAX+PLUSII™ and the EPIC™ Editor from FOUNDATION™, we changed the distance between both LUTs to increase the number of interconnect resources.

### 3.3.3.2 Logic Cells

Logic cells are formed by two components: Look-Up Tables and DFFs as shown in the following equation:

$$\alpha_{LC} = N_{LUT} \alpha_{LUT} + N_{DFF} \alpha_{DFF} \quad [3.3.2]$$

Where  $N_{LUT}$  and  $N_{DFF}$  are the number of LUTs and DFFs used by the applications.

#### a) Look-Up Tables

Figure 3.3.4 shows the design used to estimate the power contribution of LUTs. In this case, only the number of LUTs of the chain is incremented. Power supply, Input Frequency and the number of I/Os are constants.

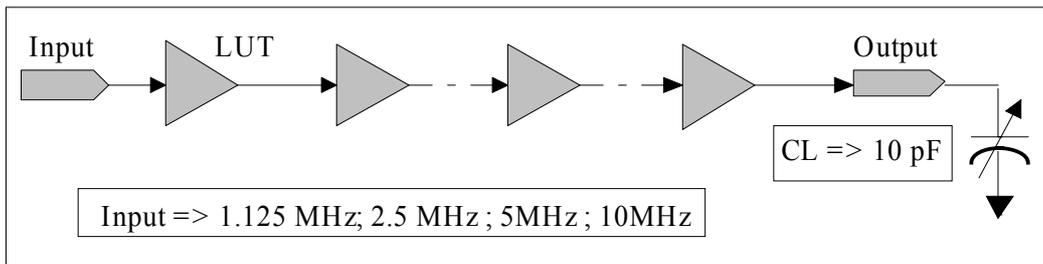


FIGURE 3.3.4. POWER CONSUMPTION OF LUTS

In this case, the offset value corresponds to the power consumed by a little chain (a single input, one LUT and one output with a fixed charge). We increased only the number of LUTs and we registered the increment of current. Then, using the results from sub-section 3.3.3.1, we estimated the current consumed by the interconnect to obtain the power consumption of a single LUT.

*b) D-type Flip-Flops*

Using the design described in figure 3.3.5, we increased the number of DFFs. The input toggling rate was fixed to 12.5%, 25%, and 50 % (referred to the Clock signal) in order to obtain more accurate results. This test permits us to estimate the power consumed by an active DFF. (Note: in order to obtain the power consumed by a single DFF, the power consumption of the clock tree has to be measured before).

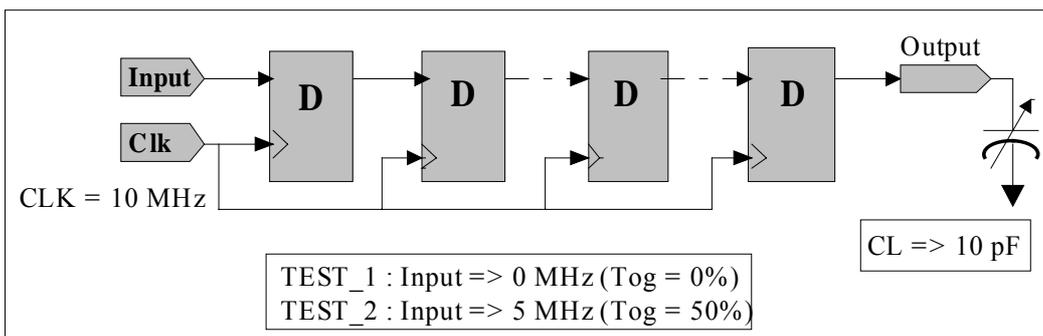


FIGURE 3.3.5. FLIP-FLOPS AND CLOCK TREE

The next figure shows another test design; in this case we use a T flip-flop to generate an internal signal with a toggling rate equals to 50%.

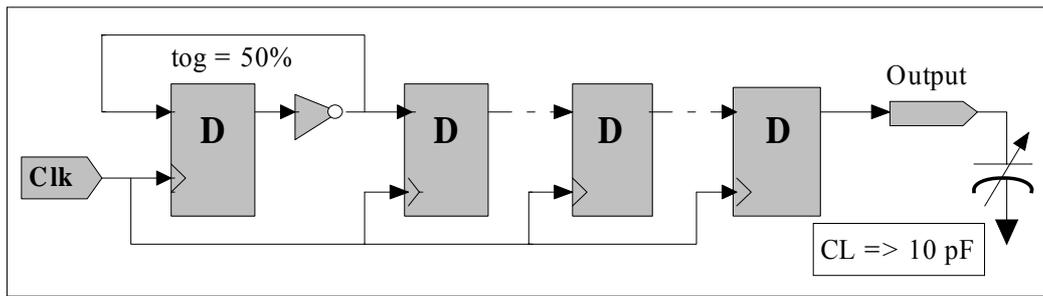


FIGURE 3.3.6. POWER CONSUMPTION OF DFFS

Clock frequency is fixed as well as the Voltage of the power supply. The number of DFF was incremented to estimate the power consumption of this element. In this case the load capacitance was also kept constant

### 3.3.3.3 I/O Cells

The following equation shows the current consumed by I/O cells:

$$\epsilon_{I/O} = N_i \epsilon_{Input} t + N_j \epsilon_{Output} \quad [3.3.3]$$

Where  $N_i$  and  $N_j$  is the number of Input and Output used by the application.

#### a) Outputs

Using a chain with a fixed number of LUTs, we tested the I/O cells configured as Output. In this case, we increased the number of outputs. The power consumed by the interconnect resources was estimated using the results obtained in 3.3.3.1.

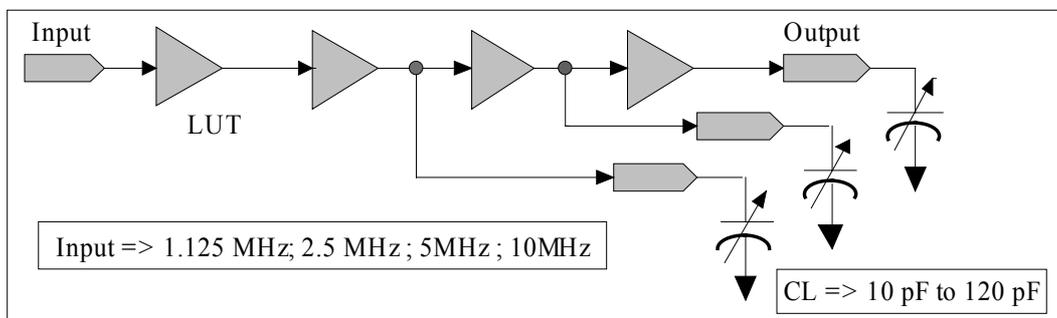


FIGURE 3.3.7. POWER CONSUMPTION OF OUTPUT PADS

The load capacitance has been changed for the same exercise to verify its influence and to verify our results.

b) Inputs

Using the same design used in the last sub-section, we increased the number of Inputs to estimate the influence of an I/O cell configured as Input. The power consumed by the interconnect was also estimated using the results obtained before.

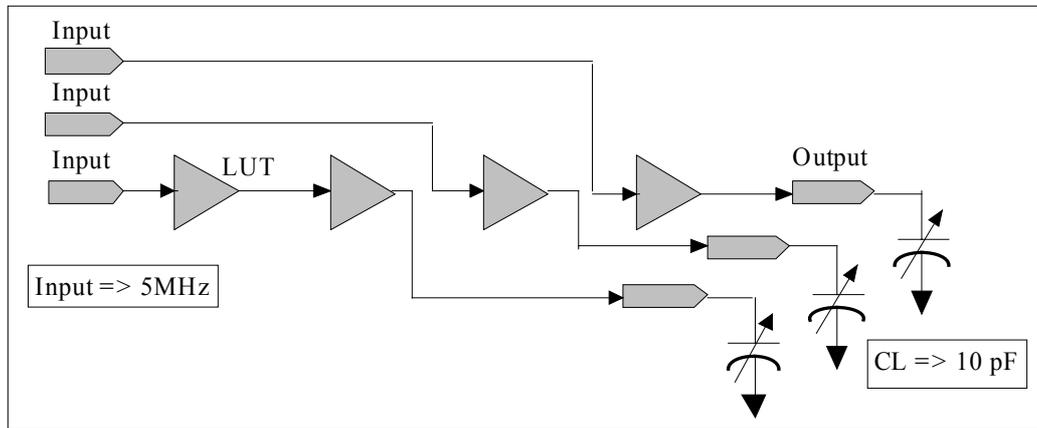


FIGURE 3.3.8. POWER CONSUMPTION OF INPUT PADS

In this case, the number of LUTs and outputs is constant and we only increased the number of I/Os configured as Input. The load capacitance was also kept constant.

3.3.3.4 Clock tree

The design of figure 3.3.5 was used to obtain the power consumption of the Clock tree. The first test consists on an increment of DFFs with an input-toggling rate equal to zero as shown in figure 3.3.9. In this case only the Clock tree will consume.

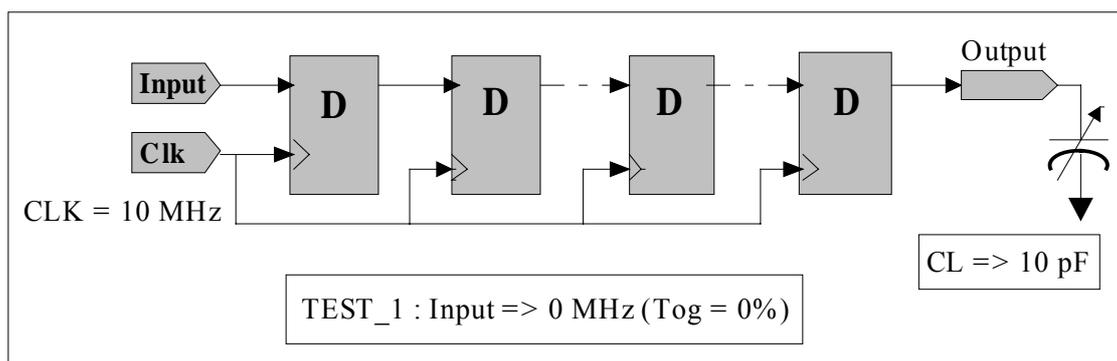


FIGURE 3.3.9. POWER CONSUMPTION THE CLOCK TREE

Like precedent tests, the number of I/Os and the CL were kept constant. The result expected for this test can be described as follows: An initial value (offset) that corresponds to the power consumed by the clock buffer itself must be obtained followed by the increments that correspond to the use of interconnect resources to distribute the

clock signal.

### 3.3.3.5 Memory

As mentioned, we use the simplest memory structures in order to measure their power contribution, in this case, we increased the size of the memory (increasing the size of the address bus) and keeping constant the size of the word.

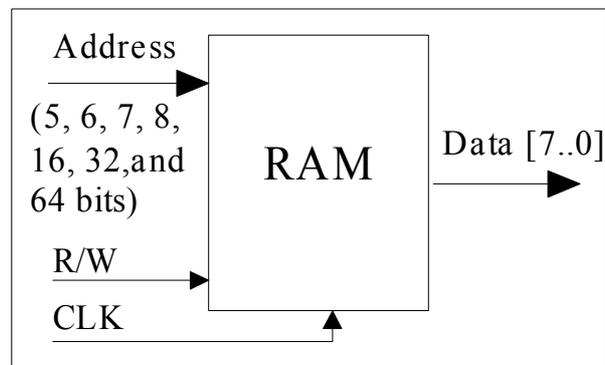


FIGURE 3.3.10. POWER CONSUMPTION OF MEMORY CELLS

For the Flex™10K device, we also increase the number of EAB™. In XC4000 device, we increase the number of CLB configured as memory (64 CLBs correspond to 1 EAB).

## 3.4 Measurements and Results

Using the incremental methodology described in section 3.3, several numerical results have been obtained from measurements. These results correspond to each constant defined in equations 3.3.2 and 3.3.3. Some of these constants have been decomposed in order to obtain a value for each internal sub-element (i.e. a L.E. is decomposed in LUTs and DFFs). The Static current measured for the Flex device is from 1.5 to 2.4 mA; Static current of the XC4000E devices is between 7.2 and 7.6 mA.

### 3.4.1 Interconnect resources

The following equation represents the power consumed by the interconnect of a FPGA:

$$P_{interconnect} = V_{DD} * F * \beta_{interconnect} \quad [3.4.1]$$

Where  $\beta_{interconnect}$  contains all the interconnect resources of the FPGA and it depends on the internal architecture.

a) *Flex 10K devices*

In this case,  $\beta$  can be decomposed for a Flex10K device as follows:

$$\beta_{Flex} = \beta_a N_{half\_FastTrack} + \beta_b N_{full\_FastTrack} + \beta_c N_{columns} + \beta_d N_{LAB} \quad [3.4.2]$$

Where N is the number of used resources (half Fast Track, full Fast Track, Column and LAB interconnect), and  $\beta_a$ ,  $\beta_b$ ,  $\beta_c$ , and  $\beta_d$  are constant in mA/MHz that correspond to each sub-element.

For a Flex10K100 device, the results obtained are presented in table 3.4.1.

ELEMENT	P (mW/MHz)
HALF F.T.	0,115
FULL F.T.	0,19
COLUMN	0,18

TABLE 3.4.1. FLEX10K100 INTERCONNECT

Measurement results have shown that the power consumed by the internal wires of the LAB is negligible.

b) *XC4000E*

For a XC4000E device, the equation that corresponds to the power consumed by the interconnect is decomposed as follows:

$$\beta_{XC4000E} = \beta_a N_{Direct\_paths} + \beta_b N_{Single} + \beta_c N_{Double} + \beta_d N_{Long} + \beta_e N_{Global} + \beta_f N_{PMS} \quad [3.4.3]$$

Where N represents the number of resources contained in the netlist and

$\beta_a, \beta_b, \beta_c, \beta_d, \beta_e, \beta_f$  are constants in mA/MHz that correspond to each sub-element.

Table 3.4.2 shows the results obtained using the XC4010E device.

ELEMENT	P (mW/MHz)
DIRECT PATHS	0,09
SINGLE	0,07
DOUBLE	0,08
LONG LINES	0,4
GLOBAL	0,35
SWITCH BOX	0,01

TABLE 3.4.2. XC4010E INTERCONNECT

Power consumed by the interconnect of the XC4010E device is superior to the power consumed by the Fast Track interconnect because Xilinx devices are larger than Altera devices. Long lines from Xilinx are bigger than Fast tracks from Altera.

### 3.4.2 Logic Cells

Inspired by equation 3.4.1, we can represent the power consumed by logic cells as follows:

$$P_{\text{lut}} = V_{\text{DD}} \alpha_{\text{lcell}} \quad [3.4.4]$$

Where  $\alpha_{\text{lcell}}$  can be decomposed in two components:

$$\alpha_{\text{lcell}} = \alpha_a N_{\text{lut}} + \alpha_b N_{\text{DFF}} \quad [3.4.5]$$

Where N is the number of LUTs or DFF used in the design and  $\alpha_a, \alpha_b$  are constant in mA/MHz.

#### a) Look-Up Tables

For a XC4010E device that contains two types of LUT (4-input LUT and 3-input LUT),

the equation 3.4.6 can be expressed as follows:

$$\alpha_{\text{cell}} = \alpha_a N_{4\_input\_lut} + \alpha_b N_{3\_input\_lut} + \alpha_c N_{\text{DFF}} \quad [3.4.6]$$

Table 3.4.3 shows the results obtained for both devices:

DEVICE	ELEMENT	P (mW/MHz)
FLEX10K	4-INPUT LUT	0,15
XC4000E	4-INPUT LUT	0,10
XC4000E	3-INPUT LUT	0,075

TABLE 3.4.3. LOOK-UP TABLES

The Power consumption of LUTs in XC4000E devices is lower than Altera's LUTs. On the other hand, the CLB in a Xilinx device does not permit a real direct connection between two LUT.

#### b) D-type Flip Flops

The following table shows the results obtained for the DFFs using both Flex10K and XC4000E devices:

DEVICE	P (mW/MHz)
FLEX10K	0,12
XC4000E	0,21

TABLE 3.4.4. DFFS

### 3.4.3 I/O Cells

The power consumption of I/O cells configured as input or as outputs can be represented as follows:

$$\varepsilon_{\text{I/O}} = \varepsilon_a N_{\text{input}} + \varepsilon_b N_{\text{outputs}} \quad [3.4.7]$$

Where N is the number of Inputs or Outputs used in the design and  $\varepsilon_a, \varepsilon_b$  are constants in mA/MHz.

#### a) Outputs

The power consumed by the outputs depends on the load capacitance value. The following table shows the results obtained for both Altera and Xilinx devices:

DEVICE	P (mW/MHz*pF)	I (mA/MHz*pF)
FLEX10K	0,065	0,0130
XC4000E	0,028	0,0056

TABLE 3.4.5. OUTPUTS

In order to obtain more accurate values, measurements with different values of  $C_L$  were taken.

#### b) Inputs

Table 3.4.6 presents the results obtained from the increment of inputs.

DEVICE	P (mW/MHz)
FLEX10K	0,456
XC4000E	0,225

TABLE 3.4.6 INPUTS

If we compare tables 3.4.5 and 3.4.6, we can find an approximate value of the Input Capacitance.

### 3.4.4 Clock Tree

The equation that corresponds to the Clock tree contains basically three elements: The clock buffer and the interconnect that serve to distribute the clock signal; and the multiplexers that permits a selection of the appropriated clock signal.

#### a) Flex 10K

The power consumed by the clock tree in a Flex 10K can be expressed as follows:

$$\delta_{\text{clk}} = \delta_a N_{\text{clk\_buffer}} + \delta_c N_{\text{LAB}} + \delta_c N_{\text{LE}} \quad [3.4.8]$$

The following figure represents the measurements obtained when using a DFF chain with an input signal equal to zero:

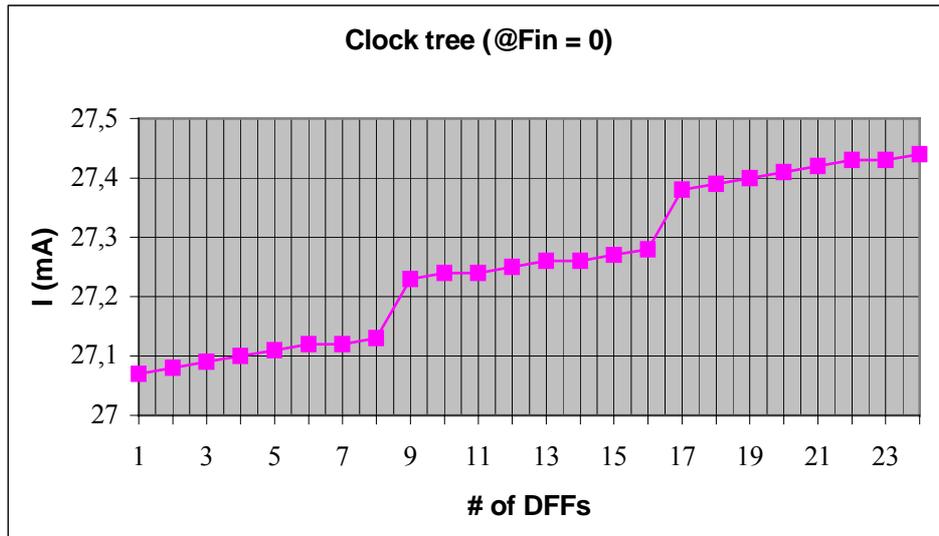


FIGURE 3.4.1. CLOCK TREE OF A FLEX10K DEVICE

As shown in figure 3.4.1, the increments are caused by the two kinds of multiplexers, little increments represent the use of the multiplexer inside the Logic Element and big increments represent the use of an entire LAB (Each lab contains 8 L.E.s). The following table shows the numerical values of these sub-elements:

FEATURE	ELEMENT	P (mW/MHz)
CLOCK	Clk_buffer	12,00
	Mux_LAB	0,05
	Mux_LCELL	0,005

TABLE 3.4.7 FLEX 10K100

b) XC4000E

The following equation corresponds to the power contribution of the clock tree in a XC4010E:

$$\delta_{\text{Clk}} = \delta_a N_{\text{clk\_buffer}} + \delta_b N_{\text{CLB\_mux}} + \delta_c N_{\text{interconnect}} \quad [3.4.9]$$

Figure 3.4.2 illustrates the measurement results using a XC4010E device:

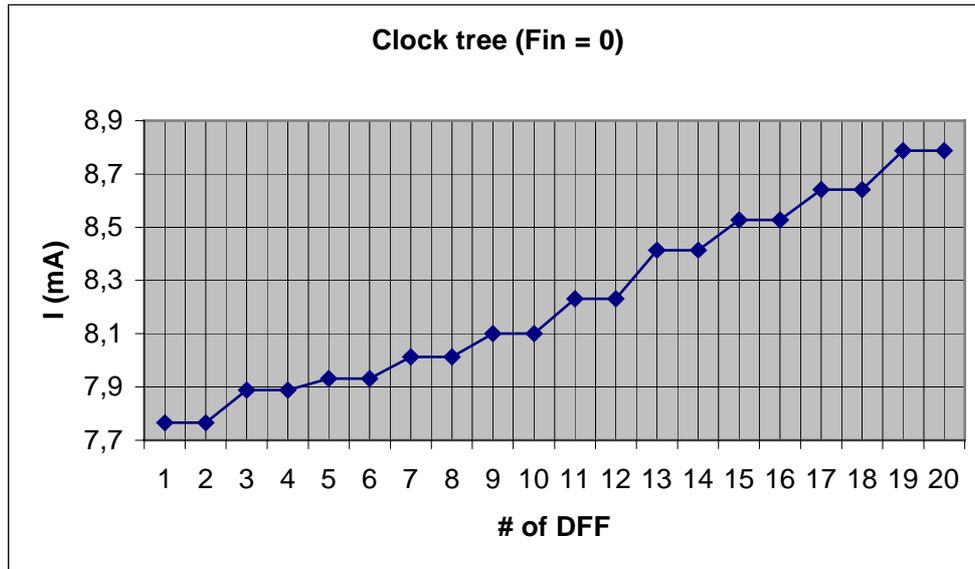


FIGURE 3.4.2. CLOCK TREE OF A XC4000E

Figure 3.4.2 represents the increment of current caused by the increment of the DFFs used. In this case, the increment is found after a couple of DFFs because inside a CLB there are two DFFs that use the same clock signal. This device uses the interconnect resources to provide the clock signal to the CLBs. The next table shows the results obtained using a XC4010E:

FEATURE	ELEMENT	P (mW/MHz)
CLOCK	Clk_buffer	4
	CLB_mux	0,1

TABLE 3.4.8. XC4010E

Notice that in this case, power consumed by the XC4010E clock tree is lower than the power consumption of the Flex10K100 clock tree

### 3.4.5 Memory Cells

The following equation represents the power contribution of the Altera's EABs:

$$\varphi_{\text{memory}} = \varphi_a N_{\text{EAB}} \quad [3.4.10]$$

Where N is the number of EABs used and  $\varphi_a$  is a constant in mA/MHz. The XC4010E device doesn't contain embedded memory blocks, it uses its own CLBs to build memory. The equation that corresponds to this device is expressed as follows:

$$\varphi_{\text{memory}} = \varphi_a N_{\text{CLB\_configured\_as\_memory}} \quad [3.4.11]$$

Table 3.4.9 shows the results obtained using Altera's EABs and Xilinx's CLBs as memory blocks. Each EAB is decomposed in 8 ECELLS.

DEVICE	P (mW/MHz)
1 ECELL	0,18
1 CLB	0,19

TABLE 3.4.9 MEMORY CELLS IN ALTERA AND XILINX

An Altera's EAB can be used to build a 2K\*1 memory. On the other hand, a CLB configured as memory can serve to build a 16\*1 memory. In order to compare both resources, we implemented a 2k\*1 memory block. The next table shows our results when reading and writing every clock cycle:

DEVICE	MEMORY SIZE	RESOURCES	P (MW/MHz)
FLEX10K	2K*1	1 EAB	1,44
XC4000E	2K*1	64 CLBs	12,16

TABLE 3.4.10 2K\*1 MEMORY IN ALTERA AND XILINX

Memory implemented in XC4000 devices represents some disadvantages compared with the use of EABs in Flex10K devices. The use of embedded memory cells is useful to reduce power. On the other hand, the implementation of memory using CLBs increases power consumption and it also sacrifices logic resources. This is a classical trade-off between flexibility and efficiency.

### 3.5 Power Consumption Model of commercial FPGAs

#### 3.5.1 Power Consumption Model based on measurements

Figures 3.5.1 and 3.5.2 contain the power consumption distribution of the five elements forming a FPGA (Logic Cells, interconnect resources, clock tree, I/O cells and Memory cells). This representation of the power distribution inside a FPGA will be useful to understand the behavior of this kind of device and to identify the elements that have to be optimized.

The following pie chart represents the power distribution of a hypothetical case in a Flex10K100 based on the results obtained from measurements.

For this example, the internal activity rate average is equal to 50%,  $C_L$  is equal to 12 pF, and it uses 80 percent of all internal resources. Power consumption is distributed as follows:

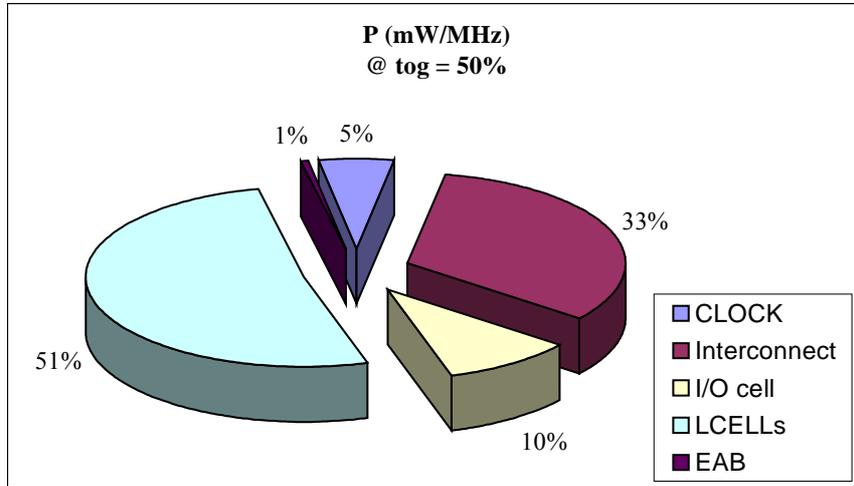


FIGURE 3.5.1. FLEX10K100 POWER DISTRIBUTION

In order to compare our results with other power consumption models, the following set of assumptions is considered to build the pie chart of Figure 3.5.2:

- A global toggling rate equals 12.5%.
- 80% of internal resources are used (including LUTs, DFFs and interconnect resources).
- 80% of the embedded memory cells are used.
- 40% of the I/Os are configured as Input and another 40% are configured as output.
- All outputs drive an equivalent load capacitance of 50 pF.

The following figure represents the power distribution of a Flex 10K100 under the set of considerations described before:

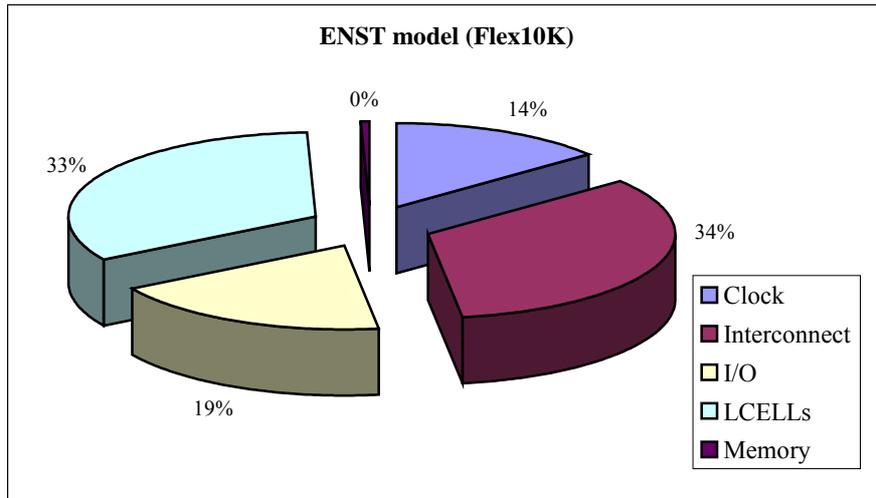


FIGURE 3.5.2 FLEX 10K100 POWER DISTRIBUTION WITH TOG = 12.5%

It has to be notice that it is almost impossible for a design based on FPGAs to have an activity rate superior to 40% (tog = 50 % means that the design is totally sequential and all the signals change each clock cycle). On the other hand, the activity rate of these circuits must be higher than 10%.

In all the distribution graphics presented, we can observe that most of the total power consumption of a Flex10K comes from the LCELL (LUT + DFF). The second most important element in power consumption of these devices is the interconnect resources.

The following figure shows the power breakdown of a XC4000E, in this case, the Memory blocks have a larger contribution of power. The XC4000 family uses the same CLB to generate a Memory cell.

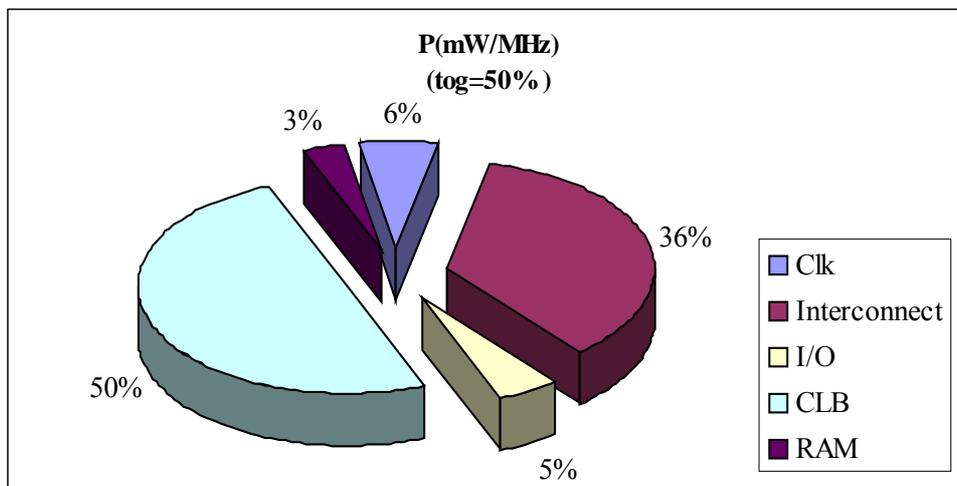


FIGURE 3.5.3. XC4000E POWER DISTRIBUTION

In this case, 25% of CLBs were configured as RAM blocks (320\*1) and the Load Capacitance was fixed to 12 pF. The next figure shows the power distribution of a XC4003E under the set of considerations used to compare the ENST model with the others.

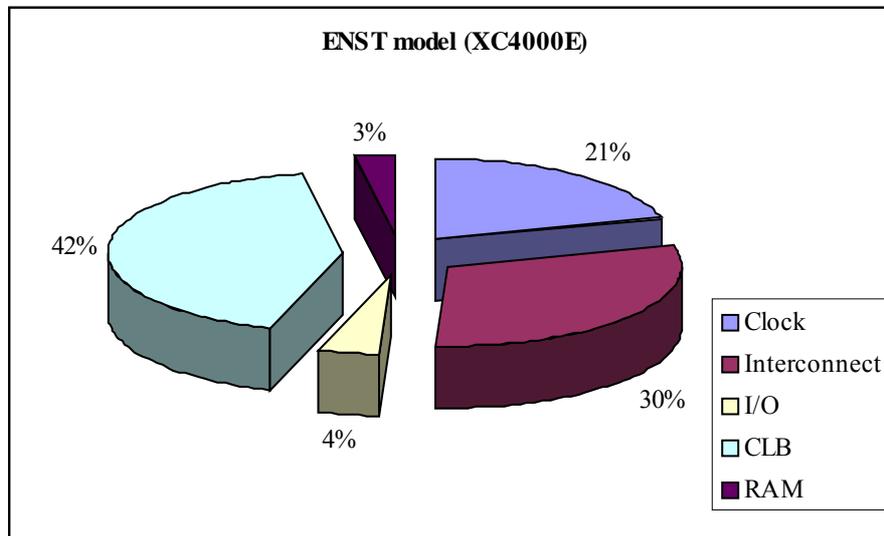


FIGURE 3.5.4. XC4000E POWER DISTRIBUTION WITH TOG = 12.5%

In the last example, 25% of the CLB used were configured as RAM.

Even if the power consumed by a CLB configured as Memory is higher than a EAB from Flex10K, the percentages that corresponds to the CLB used as logic and the interconnect are similar to the percentages for the same features in Flex10K devices.

### 3.5.2 Other Power Consumption Models

#### 3.5.2.1 Berkeley's Power Consumption Model

Low power in FPGAs has inspired several research works, including power estimation and power optimization. A power consumption model of FPGAs has been proposed by the Berkeley Wireless Research Center. In their work untitled "The Design of a Low Energy FPGA" (1999) [83], Varghese, Zhang and Rabaey, from this center, propose a power consumption model for the Xilinx™ XC4000A family using a distribution graphic. In this work, most of the power consumed by the FPGA comes from the interconnect. The following figure illustrates their results under the set of considerations used in section 3.5.1:

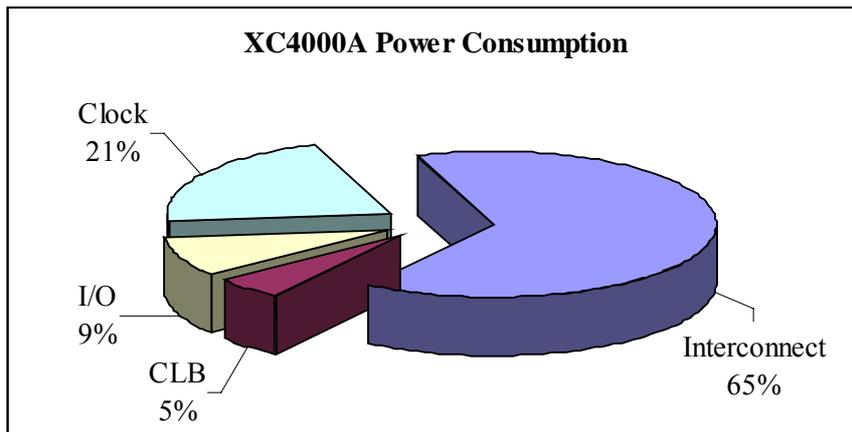


FIGURE 3.5.5. POWER CONSUMPTION OF A XC4000A FPGA

Figure 3.5.5 shows that the power dissipated by interconnect resources is almost 65% of the global power consumption. Logic only represents 5% while the clock tree represents 21% and the I/O cells only represent 9% of the power consumption. Figure 3.5.6 illustrates the interconnect power breakdown of the XC4000A family.

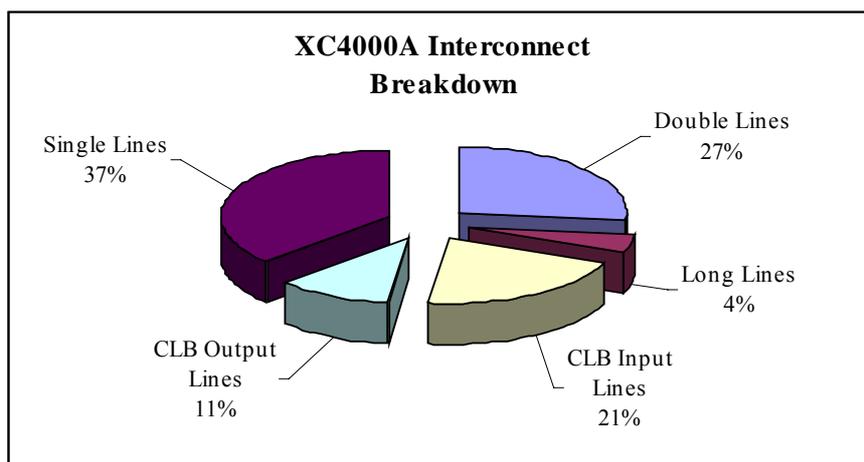


FIGURE 3.5.6. INTERCONNECT POWER BREAKDOWN

The difference between this model and the model proposed in this dissertation will be discussed in section 3.5.3.

### 3.5.2.2 Power Consumption Models from vendors

In order to compare the ENST model with models proposed by vendors, some figures illustrating the power breakdown (or the internal power distribution) will be presented in this section. All of them were constructed following the set of assumptions defined in section 3.5.1:

- A global toggling rate equals 12.5%.
- 80% of internal resources are used (including LUTs, DFFs and interconnect resources).
- 80% of the embedded memory cells are used.
- 40% of the I/Os are configured as Input and another 40% are configured as output.
- For the XC4000 series 40 % of CLBs are configured as logic cells and another 40% are configured as distributed RAM.
- All outputs drive an equivalent load capacitance of 50 pF.

Figure 3.5.7 illustrates the power distribution of a Flex 10K100 using the power consumption model from Altera. Using this model, only four components can be identified: The internal AC power, the internal DC power, the power caused by the AC outputs and the power dissipated by the DC outputs.

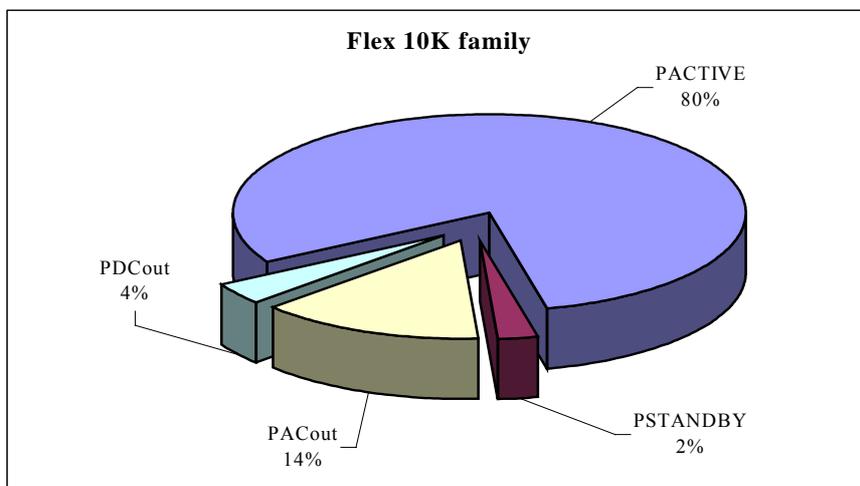


FIGURE 3.5.7. POWER CONSUMPTION MODEL FROM ALTERA.

Figure 3.5.8 is the power breakdown of the AT40K family using the model proposed by Atmel™.

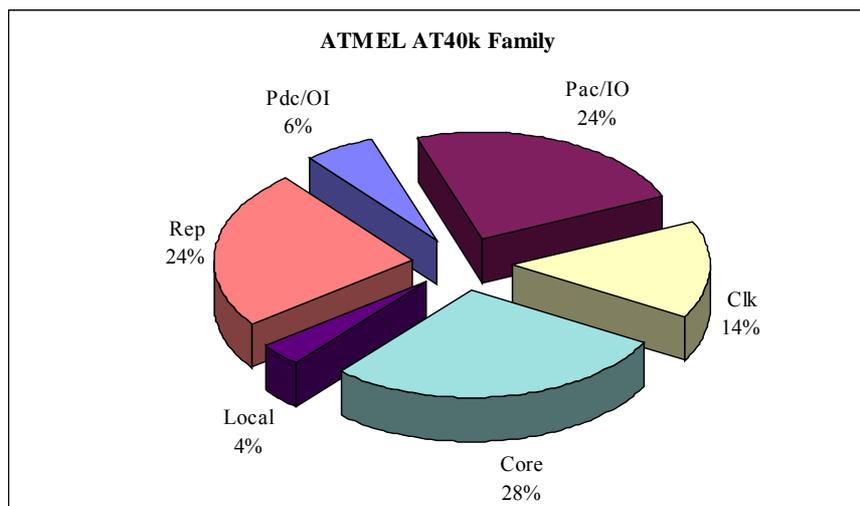


FIGURE 3.5.8. POWER DISTRIBUTION OF AN AT40K DEVICE

The model proposed by Atmel™ to estimate the power consumption of the AT40K family permits an estimation of the power of almost all the internal resources. Unfortunately, this model does not include either the power consumed by long wires or the embedded memory cells.

The following figure shows the power distribution of a CX4000E device using the model proposed by Xilinx™.

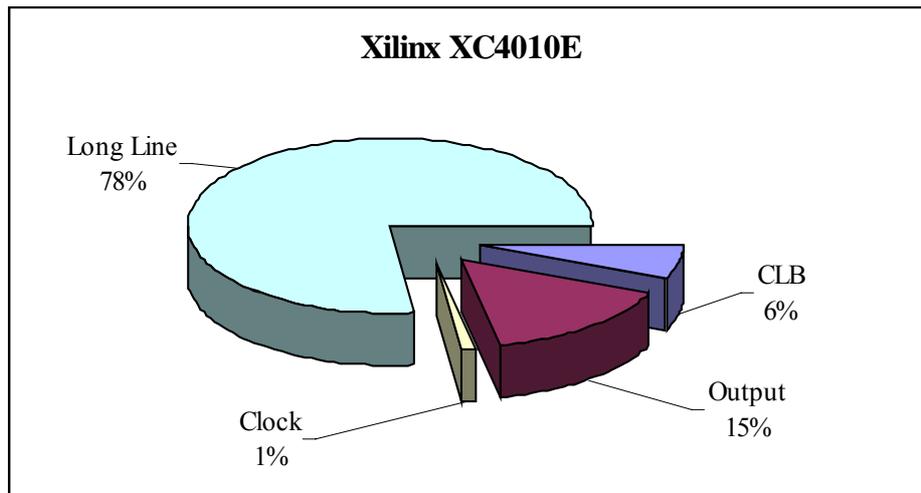


FIGURE 3.5.9. POWER BREAKDOWN OF A XC4010E DEVICE

The model proposed by Xilinx™ does not permit an estimation of the power consumed by distributed RAM cells. It does not include either other interconnect resources such as local wire segments, Single-Length Lines, Double-Length Lines, and Programmable Switch Matrix (PSM).

### 3.5.3 Comparisons between all Power Consumption Models

The differences between the power consumption model proposed in this dissertation, models proposed by vendors and by the Berkeley Wireless Research Center are very significant.

Before comparing all the models exposed in the last sections, some measurements with a fixed toggling rate were taken in order to show that the model proposed in this dissertation can bring us more accurate results than models proposed by vendors.

Figure 3.5.12 shows the result of a 20 DFF chain with a number of outputs from 1 to 18. In this case, the toggling rate is equal to 50%, and the load capacitance was fixed to 12 pF. We can see that the offset values caused by the Clock tree and a single DFF with one output are different. Measurement results are bigger than estimated results using the vendor's equations.

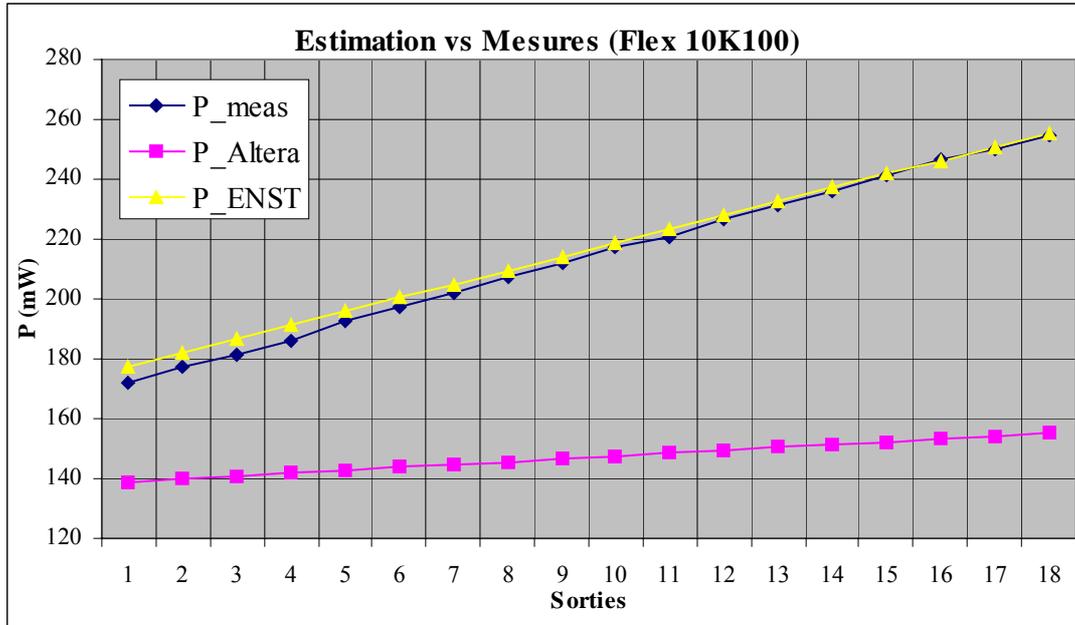


FIGURE 3.5.10. POWER ESTIMATED USING ALTERA™ MODEL AND ENST MODEL VS MEASUREMENT

The power estimation model proposed by this vendor is based on hypothetical cases and it does not permit an estimation of the contribution of the different sub-elements. This model is also based on the CMOS power consumption model, which consider that power is a function of  $V_{DD}^2$ .

On the other hand, using the ENST model to estimate the power consumption of a design with a fixed (or known) toggling rate, we can obtain results closer to the measurement results. It proves that the results obtained from this study could be useful to design techniques of power reduction.

In order to compare all the models, let us assume that power consumption is only decomposed into three components: Power from clock distribution, power dissipated by the outputs and internal power consumption. The following table illustrates the percentage of power consumption from the different models exposed in this chapter:

Feature	Altera™	Atmel™	Berkeley (Xilinx™)	Xilinx™	ENST (Xilinx™)	ENST (Altera™)
Interconnect	N.S.	28%	65%	78%	30%	34%
Outputs	18%	30%	9%	15%	4%	19%
Logic	N.S.	28%	5%	6%	42%	33%
Clock	N.S.	14%	21%	1%	21%	14%

TABLE 3.5.1. POWER CONSUMPTION MODELS

It is clear that power consumed by the I/O cells represents a low percentage of the global power consumption for most models. On the other hand, big differences between all the models come from the distribution of power consumed by the internal resources.

The Berkeley model considers that most internal power consumption comes from the interconnect. This model includes the interconnect resources used to distribute the clock signal into the clock feature. On the other hand, the ENST model considers that most of the power consumed by a FPGA comes from the logic (LUTs and DFFs). Differences between both models can be explained as follows: The ENST model considers some little interconnect resources into the percentage of power consumed by Logic Cells. In other words, the percentage of power consumed by a logic cell, in the ENST model, contains the power consumption of LUTs, DFFs and local (or internal) wires called in the Berkeley model "input CLB" and "output CLB" lines.

Finally, models from vendors don't allow us to estimate the percentage of power consumed by each of the internal resources nor to estimate the global power consumption. From figure 3.5.10 we can notice that the model proposed in this work could deliver more accurate values than models proposed by vendors.

### 3.6 Summary

Power consumption caused by the internal resources (logic elements and programmable wires) represents almost 80% of the global power consumption. I/O cells consume less than 15% and I/Os have an average percentage of power consumption around 10 %. Therefore, power consumption in FPGAs could be optimized by reducing the power consumed by the use of unnecessary internal resources (logic cells and wires). It can be possible by using some architectural techniques that are normally used to optimize speed and surface (i.e. partitioning, reducing critical paths, etc.) [21].

A high percentage of power comes from the interconnect resources. The use of long wires, and/or several wires segments used for one single signal (including programmable switches), increases power consumption.

Techniques to save power consumption have to consider the optimization of the place and route. The main idea is to use long wires for heavily load signals and exploiting the use of local wires for the other signals. Recently, some authors like Vaughn Betz [11, 12] from the University of Toronto have developed some methods and tools to optimize the place and route. This work could be useful to decrease critical data paths, and therefore the power consumption of the device will be reduced. Since the power consumed by logic cells represent a high percentage of the power dissipated by the FPGA, the optimal use of logic cells and the use of embedded memory cells (that consumes less than 2%) to build logic functions must be improved.

Some devices such as XC4000 series use their CLBs to build RAM blocks. The power consumed by those elements is higher than the power consumed by embedded memory cells. Another factor that increases the power consumption of the logic cells is the use of DFFs in synchronous process. The use of embedded memory allows us to build low power memory blocks and high glue logic blocks also.

Finally, partial reconfiguration could be a good solution to reduce power consumption in applications based on FPGAs. The reconfigurability property of these devices (that has inspired the term of virtual hardware) [69] can be used to reduce the power of systems with one or many FPGAs [31,47]. Power consumption of the reconfigurable mechanism must be carefully modeled, measured and optimized for dynamic reconfiguration to be a viable power saving technique [51, 52].

## Chapter IV. Optimizing Power Consumption in FPGAs

### 4.1 Introduction

In chapter 3, a power consumption model of FPGAs based on measurements has been explained. This model shows that an important percentage of the global power consumption is due to the interconnect resources and the logic elements.

An optimal use of the internal resources allows the power consumption to be reduced. Based on all the measurement results obtained, power dissipation in FPGAs can be saved by using some design rules: avoiding the use of Long Lines (Xilinx™) or Fast Tracks (Altera™) and using local interconnect. This can be possible by using a manual partitioning process to optimize critical pads and blocks. If the embedded cells are not used to build memory blocks, they must be used to build glue logic as needed. Finally, some architectural techniques can be useful to save power in FPGAs [14, 16, and 85].

Measurement results have also shown that power consumption in FPGAs behaves like a 3-degree polynomial function of  $V_{DD}$  (as shown equation 2.2.46); this is caused by the complex architecture of commercial FPGA which contains different elements such as SRAM, pure CMOS and pass-transistor structures. The most obvious and effective method to save power is decreasing the supply voltage. Unfortunately, this measure increases the internal delay time of the device and, consequently, decreases the performance of the circuit. This loss of performance can be recovered using some architectural techniques such as pipeline and parallelism.

A technique for low power operation in VLSI using the lowest possible supply voltage coupled with an architectural optimization proposed by Chandrakasan (192) [23] has shown that power can be saved at the expense of an increase of the silicon area. The use of this technique in FPGAs can represent several advantages: In most commercial FPGAs, D-type flip-flops (DFFs) are for free inside each Logic Cell, pipeline architectures match naturally in FPGAs and the increment of internal resources is minimum. The purpose of this chapter is to develop circuit-architectural techniques to save power consumption in prototype applications based on FPGAs.

In the following section, the result from some measurements is presented to probe that some commercial FPGAs can work using a low supply voltage, then, a technique to save power in FPGAs using pipeline architectures coupled with low supply voltages is proposed in section 4.3. Finally, the results using this technique in commercial FPGAs are presented in section 4.4.

## 4.2 Frequency VS Power Consumption

As mentioned in chapter 3, most of the global power consumed by a commercial FPGA is caused by the on-chip logic (logic cells and wires). Those elements have also the largest propagation delay (as described in [1, 60, 83, 92]). Since both factors (Power and time delay) are associated and can change in a different way when the supply voltage changes, it is important to estimate the performance losses caused by a reduction of the supply voltage. In other words, it is important to estimate the increment of the propagation delay when reducing the supply voltage to save power consumption. It is also important to obtain the minimum supply voltage tolerated by the device.

Using a ring oscillator, we have measured the Voltage/Frequency ratio in order to obtain the maximum propagation delay with the lowest possible supply voltage. Figure 4.2.1 shows the ring oscillator implemented in a Flex10K100 and in a XC4010E.

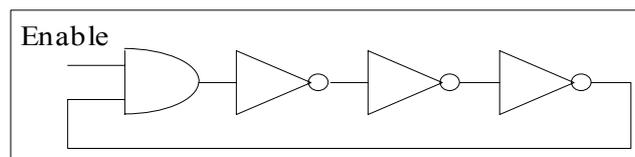


FIGURE 4.2.1. THE RING OSCILLATOR

Figures 4.2.2 and 4.2.3 present the results using the Flex device. The ring oscillator works even when the power supply is lower than 2 Volts. Comparing both figures, some conclusions can be obtained.

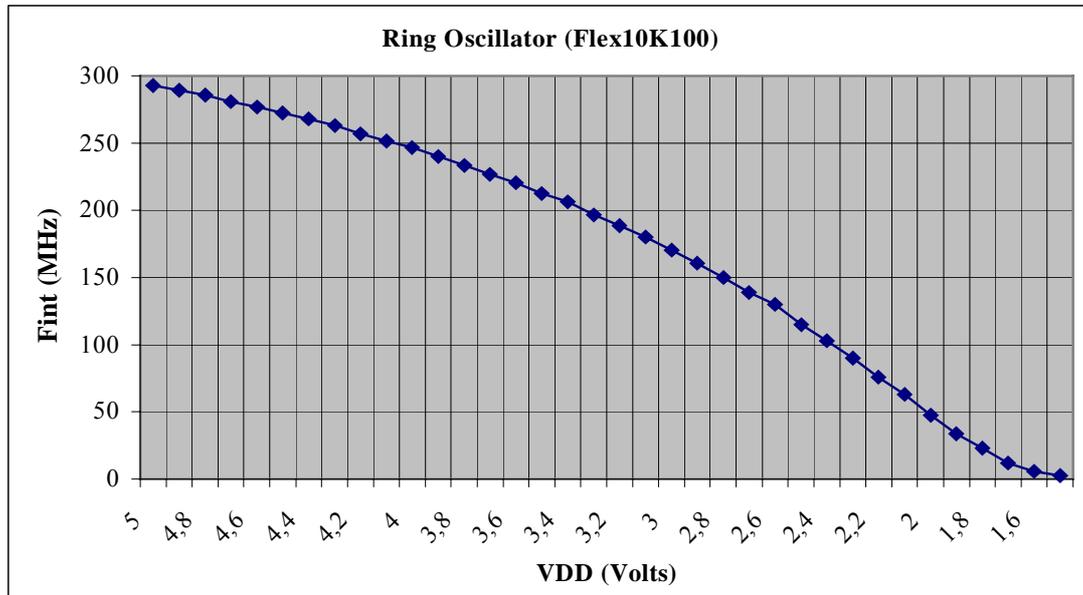


FIGURE 4.2.2. RING OSCILLATOR: INTERNAL FREQUENCY

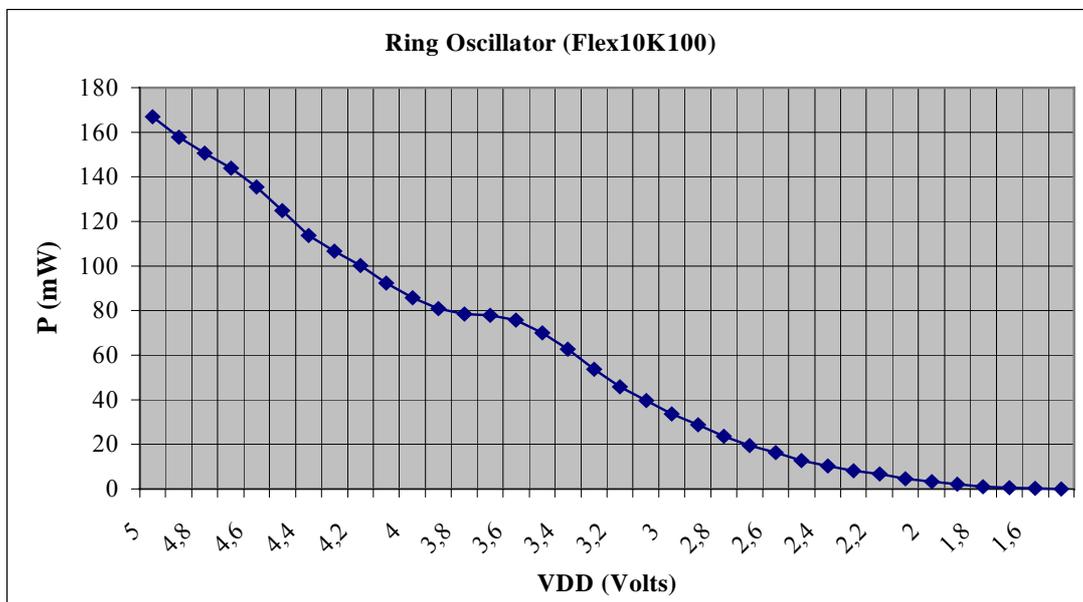


FIGURE 4.2.3. RING OSCILLATOR: POWER CONSUMPTION

This experiment shows that power consumption decreases faster than frequency (i.e. when power is reduced in 60%, the internal frequency is higher than 200 MHz). Power can be largely saved by reducing the power supply. According to A. Bellaouar, M. I.

Elmasry (1995) [10], CMOS circuits can work using a very low supply voltage, it depends on  $V_T$ , whose value can be scaled.

Equations from section 2 show that when  $V_T$  is low, we can drastically reduce  $V_{DD}$ . Figures 4.2.4 and 4.2.5 shows the same experiment using the XC4010 device. In this case the maximum frequency obtained is lower than the maximum frequency obtained using the Flex10K device. It is because of the delay time of wires between CLBs.

On the other hand, the power consumed by both devices for the same applications are similar. Nevertheless, if we consider that those circuits are 5-volt and/or 3.3-volt devices, the results show that they can be used beyond the limits specified by constructors.

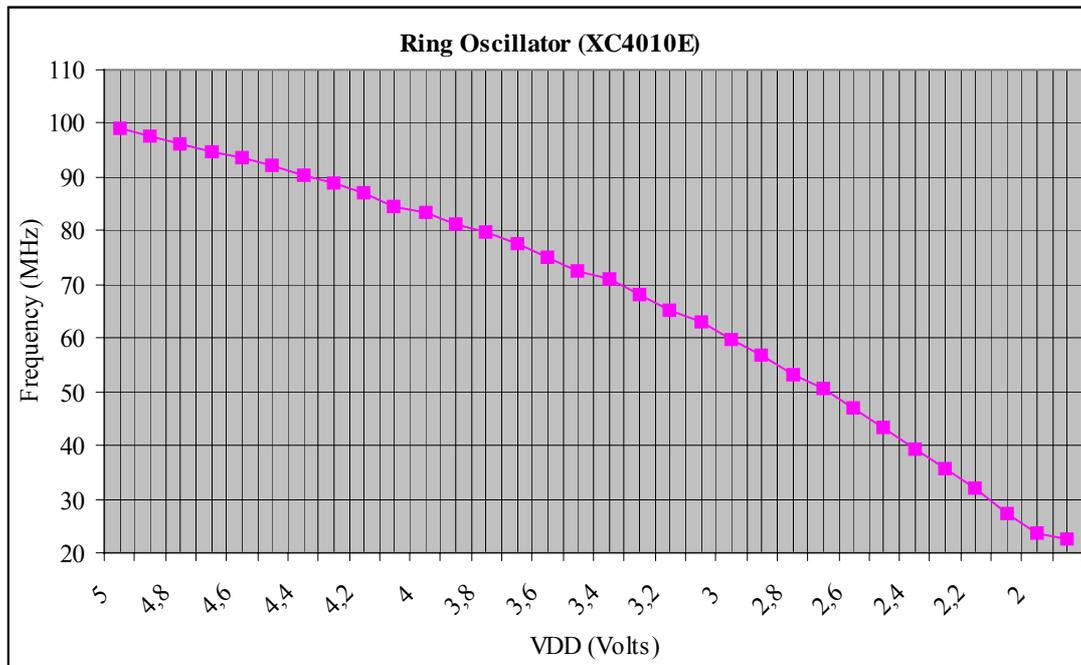


FIGURE 4.2.4. RING OSCILLATOR: INTERNAL FREQUENCY

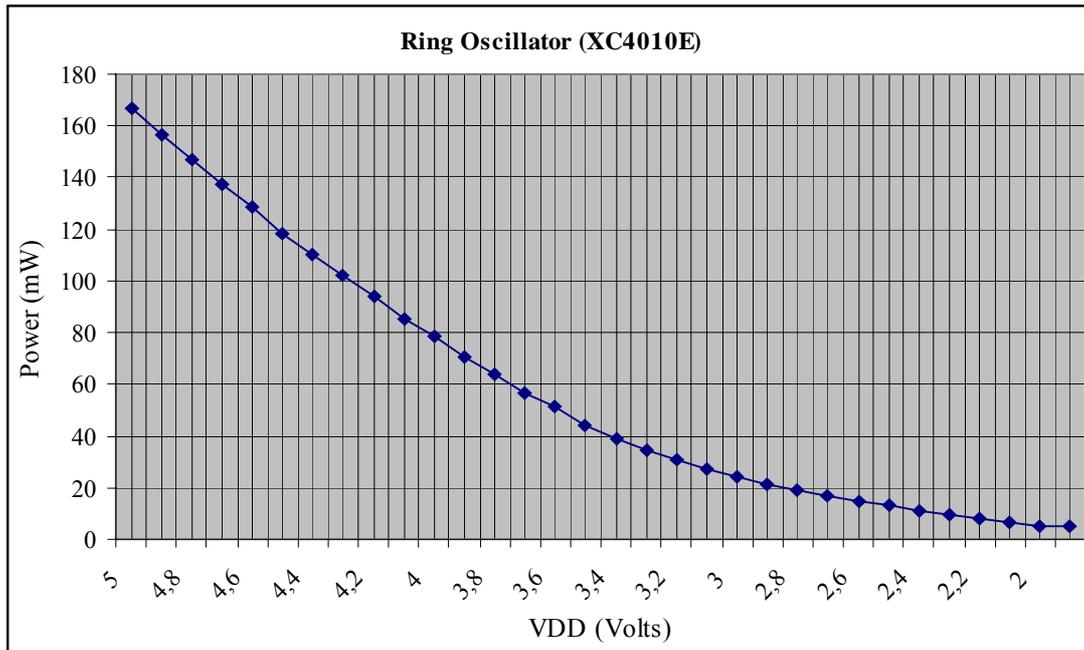


FIGURE 4.2.5. RING OSCILLATOR: POWER CONSUMPTION

Power consumption decreases faster than performance (measured in terms of the propagation delay). Results also show that power consumption in FPGAs present a cubic behavior, in other words, power decreases by more than a quadratic factor when  $V_{DD}$  decreases. Unfortunately, Xilinx™ devices can not reach the same level of performance of Altera™ devices when decreasing supply voltage. Xilinx™ devices probably contain a system that disables the circuit when  $V_{DD}$  is very low. Since the Altera™ device brings better results, the measurements presented in further sections are obtained using a Flex10K100.

### 4.3 Power Optimization

In the last section, measurement results using a ring oscillator show that commercial FPGAs can be used with a low supply voltage, beyond the lowest voltage proposed by vendors [1, 92], without a drastic loss of performance. Results show that reducing  $V_{DD}$  allows us to save power consumption by more than a quadratic factor (see equation 2.2.49). Performance loss can be recovered using architectural techniques such as pipeline. This technique to save power in FPGAs has been proposed by A. Chandrakasan (1992) for ASICs [23]. The results obtained using this methodology in ASICs is described in the following sub-section.

### 4.3.1 Results from ASICs

The use of pipeline and parallel arrays using low supply voltages proposed by Chandrakasan [23] represent a good technique to reduce power in ASIC circuits. In his work, the author has shown that, for a fixed clock frequency, using pipeline or/and Parallel data path architectures, we can save more than 50 % of power consumption.

This work considers only the dynamic power consumption caused by the load capacitance from equation [2.2.26]. According to this, the power consumption in CMOS will be reduced by a quadratic factor when  $V_{DD}$  is reduced. However, power consumption increases because of the increment of the internal capacitance (Silicon Area) due to new elements.

Table 4.3.1 illustrates the results obtained in [23] using pipeline architectures coupled with a low supply voltage. The reference voltage is equal to 5 volts, and the architecture is an adder followed by a comparator.

VOLTAGE	AREA	POWER
5.0 volts	1.0	1.0
2.9 volts	1.3	0.39

TABLE 4.3.1. RESULTS FROM [23]

The minimum supply voltage, with a fixed clock frequency is equal to 2.9 volts. The power consumption reduces by a factor of almost 2.5.

For the purpose of this study, we will consider the power consumption model proposed in section 3.5 and equation 2.2.46.

### 4.3.2 Dynamic power distribution in FPGAs

In section 3.5, a power consumption model for FPGAs is proposed. This model shows the influence of each internal element that consumes and its percentage of power consumption. The results were represented by a distribution graphic. This pie chart contains the five internal elements (Logic Cells, interconnect, I/O cells, clock tree and embedded memory) that consume and their contribution to the total power consumption. Figure 4.3.1 represents the power distribution of a 16-bit multiplier implemented in a Flex10K100.

Let us consider that all the involved parameters (netlist, Temperature,  $C_L$  and  $T_{og}$ ) are constant. If we decrease the supply voltage, the power distribution graphic will be almost the same, only the pie size changes. In addition, if we insert a stage of pipeline, only the power consumption caused by DFFs increases, the number of Logic Cells and wires will be almost the same. Using this technique, power consumption can be reduced by more than a quadratic factor and the performance of the applications can be recovered by inserting stage of pipeline.

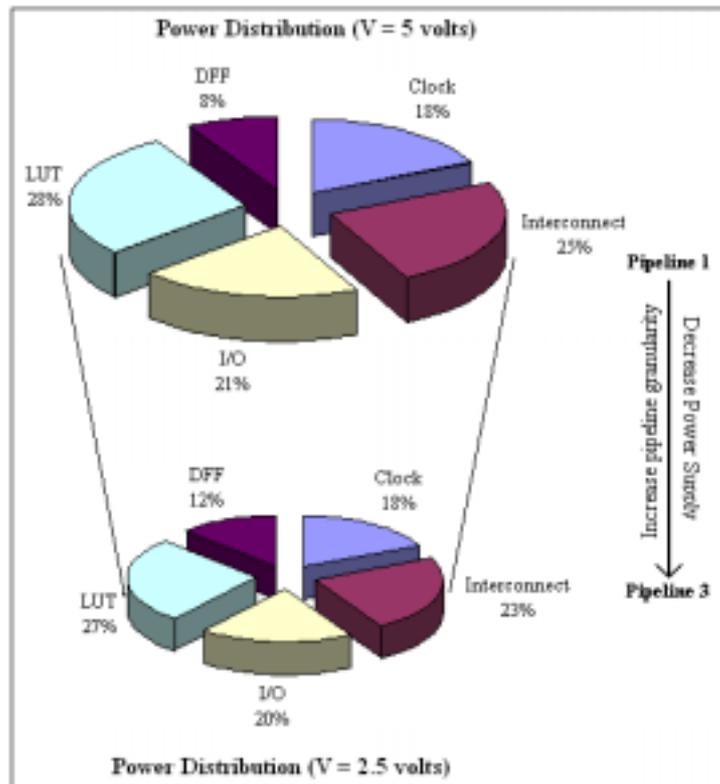


FIGURE 4.3.1. DYNAMIC POWER DISTRIBUTION

We can notice that the use of pipeline architectures has more advantages in FPGAs than in ASICs.

Since DFF are for free inside the LCELL, the percentage of the power consumption that corresponds to the interconnect resources, I/O cells and memory cells (RAM) is almost the same. Only the percentage of power consumption that corresponds to the LCELLs increases since it contains the power consumed by both LUTs and DFFs as we exposed in [32]. The increment of power consumption caused by the increase of internal resources in FPGAs is lower than the power consumed in ASICs when increasing the silicon area.

### 4.3.3 Architectural Optimization in FPGAs

In this work we have measured the impact of pipeline data path architecture using low supply voltages in FPGAs. For the purpose of this article we have used two circuits: a 16-bit adder with an absolute comparator, and a 16-bit multiplier. Pipeline granularity is increased while supply power decreases from 5 volts to the minimum supply voltage possible. These experiments allowed us to find the best pipeline granularity coupled with the minimum supply voltage that improves the maximum power reduction.

#### 4.3.3.1 Circuit 1: Adder with comparator

Figure 4.3.2 shows circuit 1. It consists on a 16-bit adder with an absolute comparator.

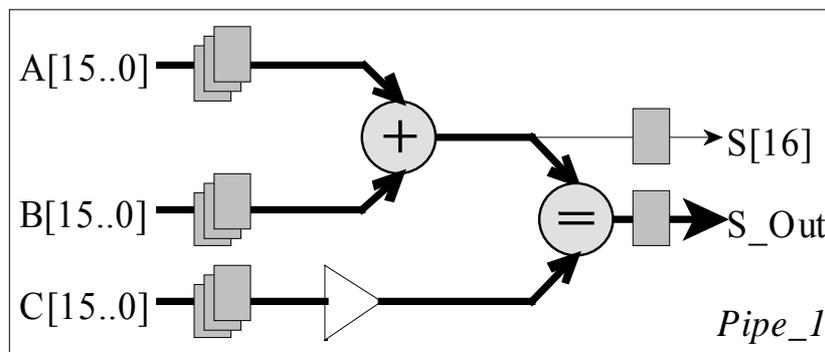


FIGURE 4.3.2. CIRCUIT 1. PIPELINE 1

Between the adder and the absolute comparator, a stage of pipeline can be inserted after the adder, as shown in the following figure:

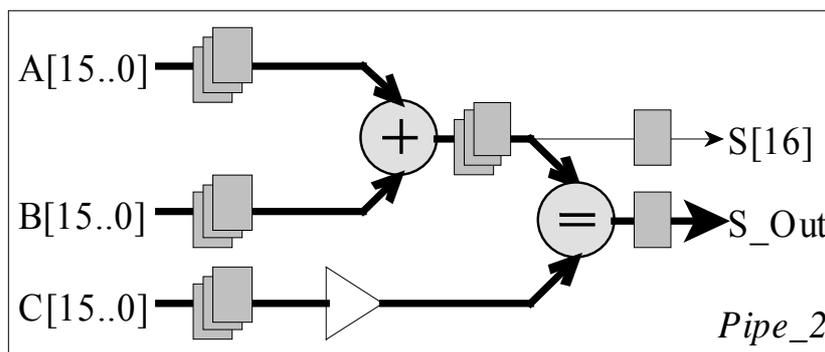


FIGURE 4.3.3. CIRCUIT 1. PIPELINE 2

Finally, another stage of pipeline to balance the propagation delay can be inserted in the second input of the comparator:

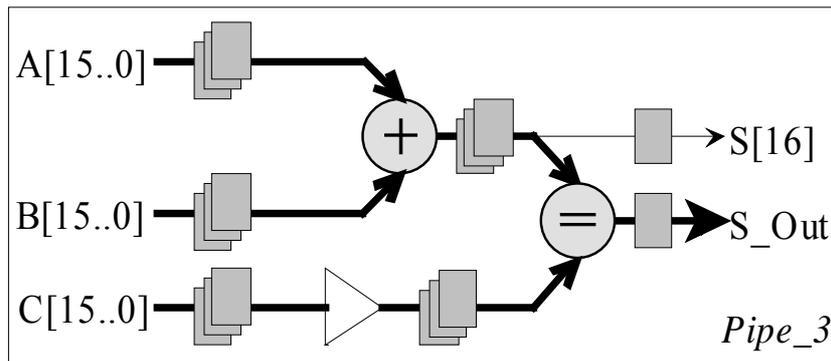


FIGURE 4.3.4. CIRCUIT 1. PIPELINE 3

#### 4.3.3.2 Circuit 2: 12-bit Multiplier

Circuit 2 is a 16-bit multiplier from the LPM ALTERA™ library. In this case, the input vectors have been generated by two counters. Figure 4.3.5 shows the multiplier.

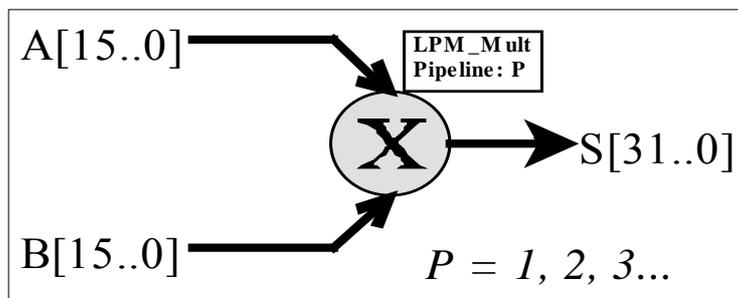


FIGURE 4.3.5. CIRCUIT 2

The LPM multiplier enables us to manually increase the number of pipeline stages. Using the floorplan editor of MAX+PLUS II, some physical constraints have been imposed in order to maintain constant the number of interconnect resources, and to place the architecture always in the same place using the same logic cells.

## 4.4 Results

### 4.4.1 Pipeline Coupled with Single Supply Voltage

#### 4.4.1.1 Results from Circuit 1

##### a) Propagation delay

First of all, the maximum clock frequency improved for each of the pipeline granularities has to be obtained. Supply voltage takes values from 5 volts to the minimum supply voltage possible. The minimum propagation delay has been measured per each supply voltage.

This experiment allows us to obtain the performance of the circuit and to define a fixed clock frequency for further measurements.

The time analyzer from MAX+PLUSII proposes a maximum frequency of 39.37 MHz for the pipeline 1, 57.80 MHz for the pipeline 2, and 60.24 MHz for the pipeline 3 (at  $V_{DD} = 5$  volts).

Measurement results show that this device can work beyond the timing model proposed by the vendor depending on the activity of input vectors. The maximum clock frequency tolerated by the circuit when using three stages of pipeline with a supply voltage of 1.75 volts is equal to 85 MHz. Pipeline 3 improves the best results.

*b) Minimum Power Supply*

Using a clock frequency of 100 MHz, power consumption is measured for each supply voltage value from 5 volts to the minimum supply voltage.

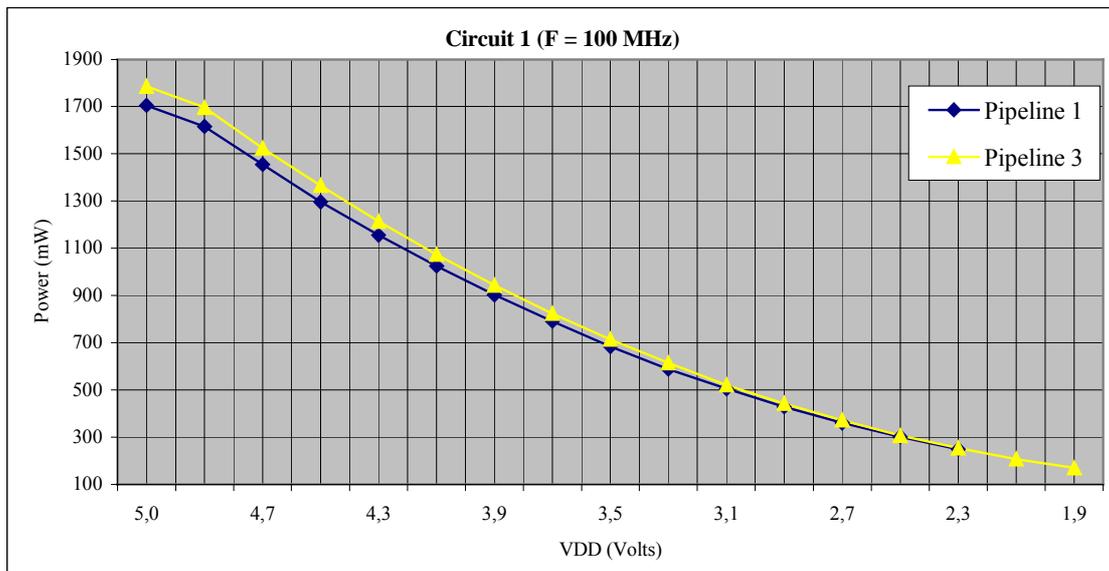


FIGURE 4.4.1. POWER CONSUMPTION OF CIRCUIT 1.

Figure 4.4.1 shows that pipeline 3 improves the best results. The following table shows the minimum supply voltage of each one of the pipeline granularity.

PIPELINE GRANULARITY	Minimum Supply Voltage (F = 100 MHz)
PIPELINE 1	2.3 volts
PIPELINE 2	2.1 volts
PIPELINE 3	1.9 volts

TABLE 4.4.1. MINIMUM SUPPLY VOLTAGE OF CIRCUIT 1

### c) Power Saving

The following table summarizes the results from circuit 1(in mWatts) using a fixed clock frequency equal to 100 MHz, the reference is pipeline 1 coupled with a supply voltage of 5 volts:

STAGE	5 Volts	3.5 Volts	2.2 Volts
PIPELINE 1	1700 mW	689.5 mW	X
PIPELINE 3	1785 mW	728 mW	215.6 mW

TABLE 4.4.2. CIRCUIT 1

It is evident that pipeline at the same supply voltage is not a good technique to reduce power. However, this technique is used to maintain the performance of the circuit when the supply voltage is reduced. The following table shows the increment of Logic Cells and the power saved when implementing pipeline 3.

STAGE	P(3.5) / P(5)	P(2.2) / P(5)	AREA (L.C.)
PIPELINE 1	0.4056	X	1.0
PIPELINE 3	0.4078	0.1208	1.24

TABLE 4.4.3. POWER OPTIMIZATION OF CIRCUIT 1

From Tables 4.4.2 and 4.4.3, we can notice that the power consumed by the circuit when using  $V_{DD} = 2.2$  volts with 3 stage of pipeline is 8.3 times lower than the reference ( $V_{DD} = 5$  volts and 1 stage of pipeline). The results show also that the insertion of pipeline stages in FPGAs does not increase critically the number of internal resources.

#### 4.4.1.2 Results from Circuit 2

##### a) Propagation Delay

Figure 4.4.2 illustrates the behavior of circuit 2. In this case, the best results are obtained by using 3 stages of pipeline the multiplier. It allows us to increase drastically the performance of the circuit. It also guarantees that the circuit will reach the original performance when using a very low supply voltage.

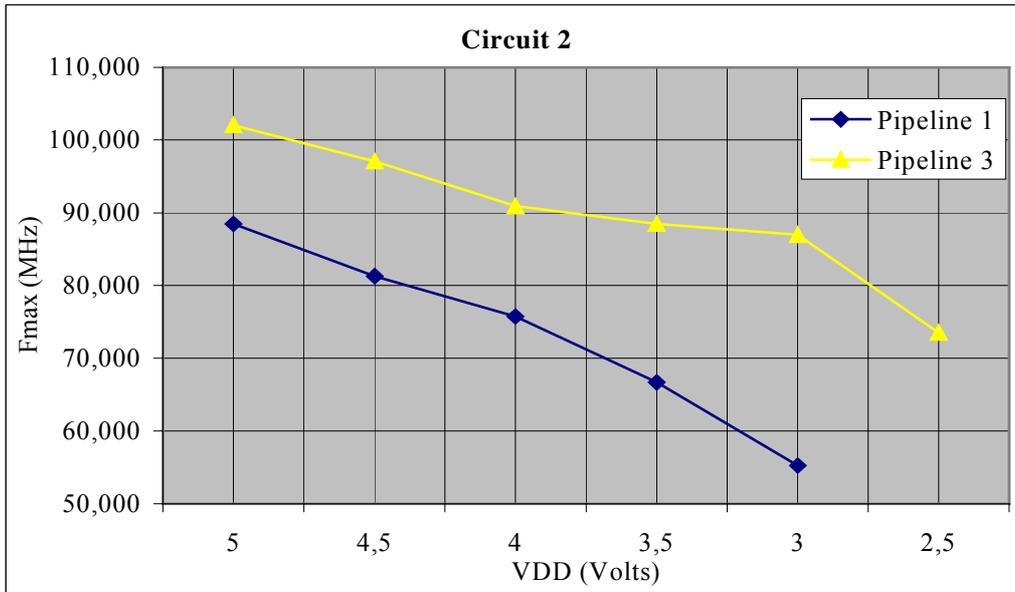


FIGURE 4.4.2. MAXIMUM CLOCK FREQUENCY OF CIRCUIT 2

*b) Minimum Power Supply*

Figure 4.4.3 shows the minimum supply voltage reached by pipeline 1,2 and 3. In this case, the clock frequency is 67 MHz.

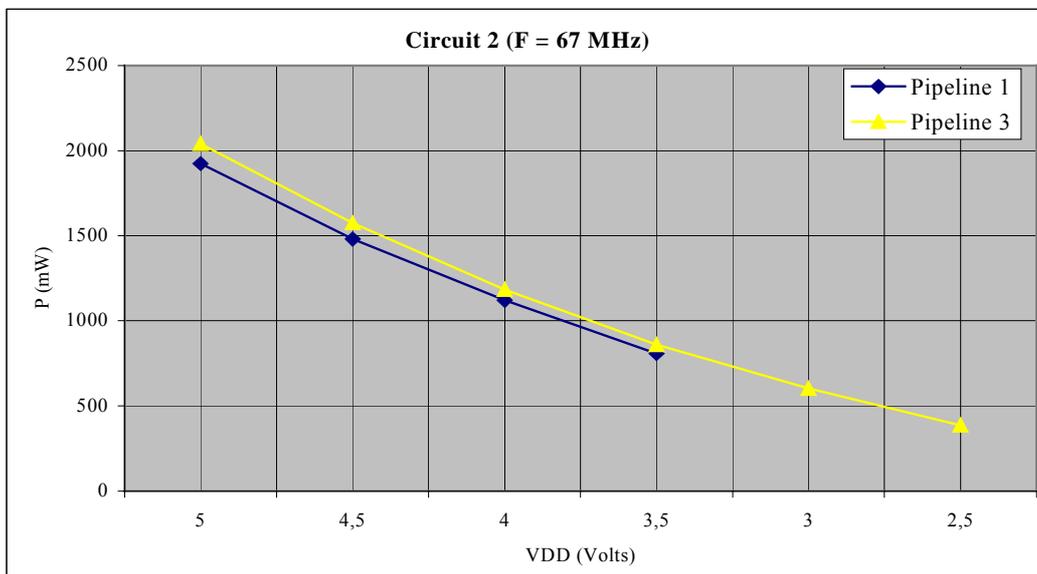


FIGURE 4.4.3. POWER CONSUMPTION OF CIRCUIT 2

Pipeline 2 is closer to the reference (Pipeline 1); the best results are obtained using 3 stages of pipeline. The minimum supply voltage reached using pipeline 3 is equal to 2.5 volts.

*c) Power Saving*

Results from circuit 2 are summarized in table 4.4.4.

PIPELINE GRANULARITY	5 Volts	3.5 Volts	2.5 Volts
PIPELINE 1	1925 mW	808.5 mW	X
PIPELINE 3	2040 mW	861 mW	387.5 mW

TABLE 4.4.4. POWER CONSUMPTION BEHAVIOR OF CIRCUIT 2

The power optimization results obtained from circuit 2 are presented in the following table:

PIPELINE GRANULARITY	P (3.5) / P (5)	P (2.5) / P (5)	Area
PIPELINE 1	0.420	X	1.0
PIPELINE 3	0.447	0.201	1.14

TABLE 4.4.5. POWER OPTIMIZATION IN CIRCUIT 2

The power consumed by the multiplier using 3 stages of pipeline with  $F = 67$  MHz and  $V_{DD} = 2.5$  volts is 4.97 times lower than the reference (5 volts with one stage of pipeline).

#### 4.4.2 Pipeline Coupled with Double Supply voltage

Recent FPGAs have at least two kinds of dedicated input pads for the supply voltage, one group of pad for the core supply-voltage ( $V_{CC\_Core}$ ) and other group of pad for the I/O cells supply-voltage ( $V_{CCIO}$ ).  $V_{CCIO}$  is normally used to define the swing voltage of the outputs allowing the circuit to communicate with different logic-level standards.

Power Consumption of the on-chip logic (or Core) can be reduced by reducing the  $V_{CC\_Core}$ .  $V_{CCIO}$  must be kept constant in order to use the I/O cells as interface between the Core and external devices. The following figure illustrates the test platform to obtain results using double supply voltage:

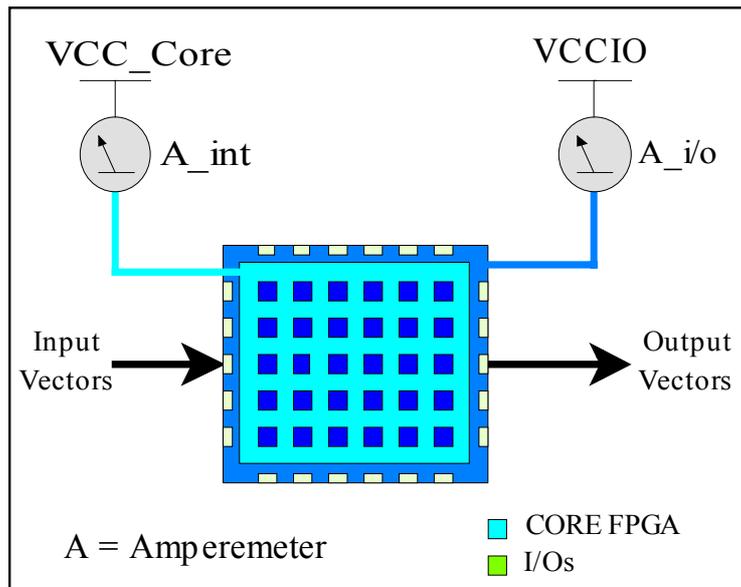


FIGURE 4.4.4. DOUBLE SUPPLY VOLTAGE

In this set of experiments,  $V_{CCIO}$  takes fixed values like 5 volts, 3.3 volts or/and 2.5 volts.  $V_{CC\_Core}$  is reduced from the  $V_{CCIO}$  used to the minimum possible supply voltage.

#### 4.4.2.1 Results from circuit 1

a)  $V_{CCIO} = 5$  Volts

The following measurements have been obtained using a clock frequency of 12 MHz and a supply voltage from 5 volts to 1.9 volts. Figure 4.4.5 shows the internal power consumption of circuit 1.

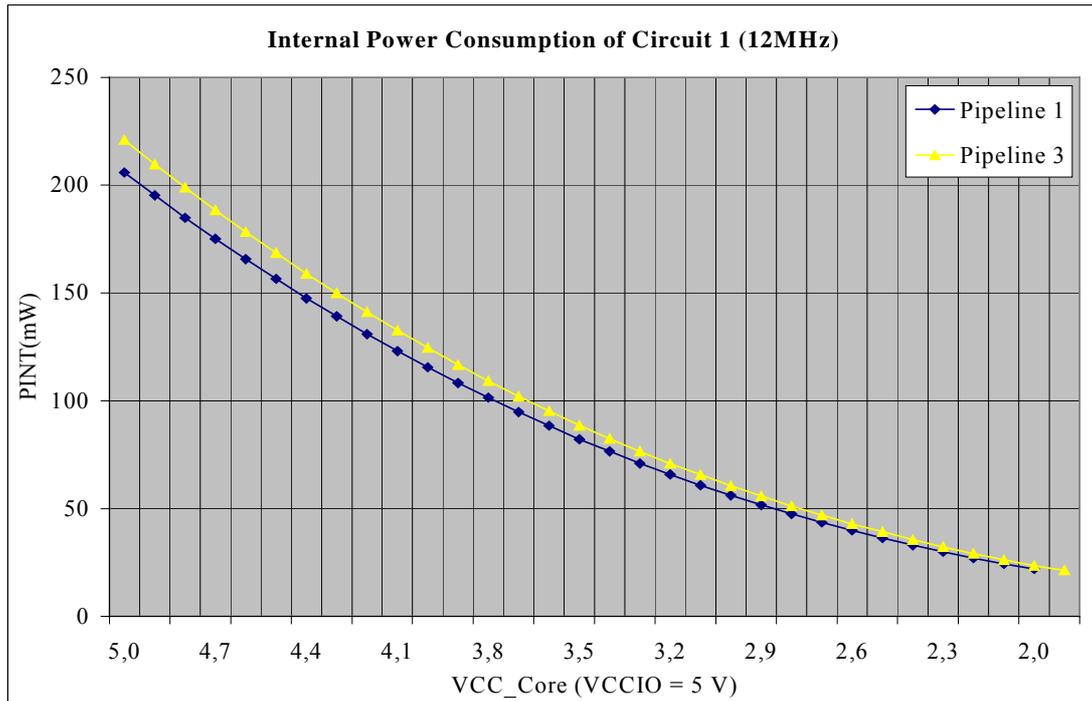


FIGURE 4.4.5 INTERNAL POWER CONSUMPTION OF CIRCUIT 1.

The difference between the 3 measurements (Pipeline 1, 2 and 3) is not significant because of the power consumed by I/O cells. On the other hand, power from I/Os decreases when  $V_{CC\_Core}$  decreases and when increasing the pipeline granularity. It must be due to the increment of DFFs. When the pipeline granularity increases, the internal glitches that are normally propagated through the I/O cells are reduced also. Another advantage using pipeline is that the number of incomplete transitions between logic blocks is reduced. It allows power consumption to be reduced. The following figure shows the power dissipated by I/O cells:

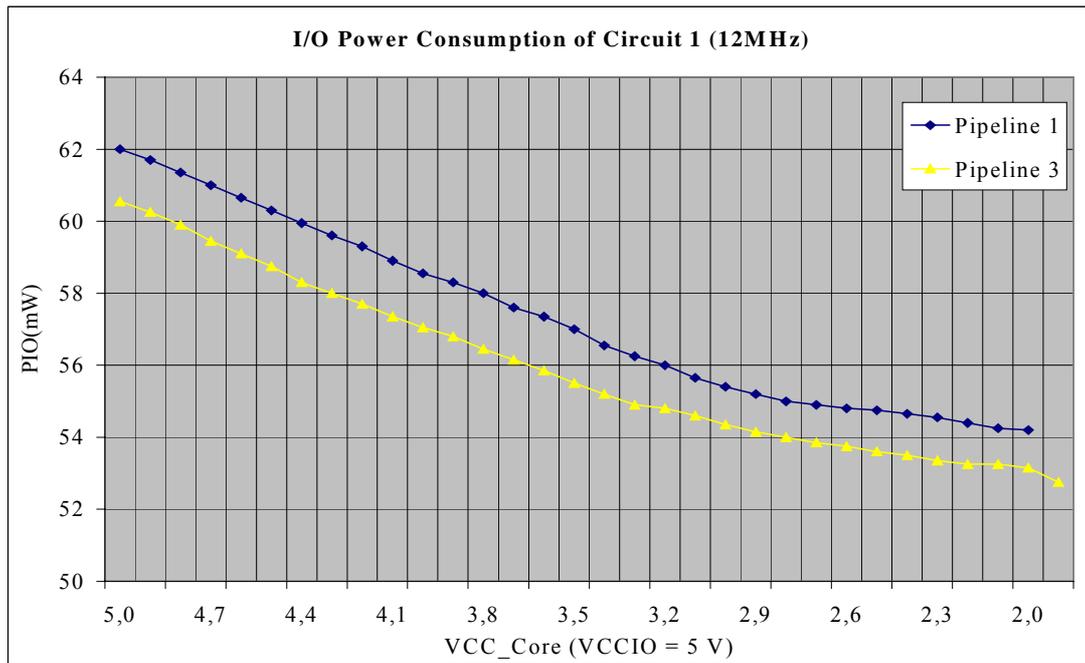


FIGURE 4.4.6. I/O POWER CONSUMPTION OF CIRCUIT 1

Finally, the global power consumption of circuit 1 is illustrated in figure 4.4.8.

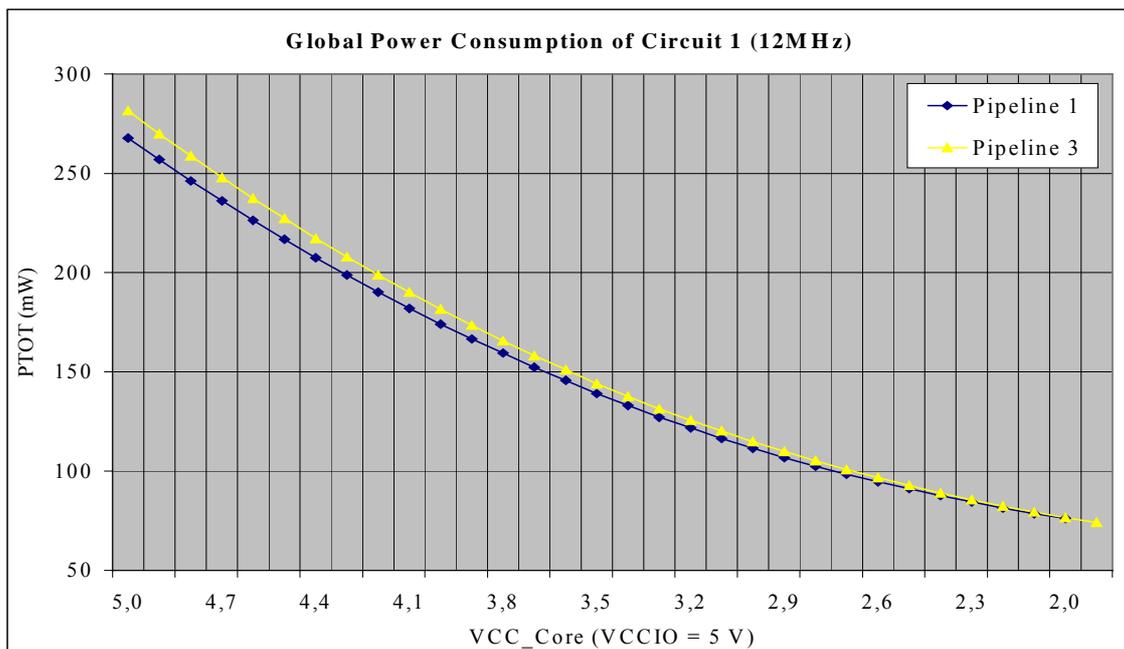


FIGURE 4.4.7. GLOBAL POWER CONSUMPTION OF CIRCUIT 1

Table 4.4.6 summarizes the results obtained from circuit 1.

GRANULARITY	5 volts	4 volts	3 volts	1.9 volts
PIPELINE 1	267.85 mW	174.03 mW	111.53 mW	X
PIPELINE 2	279.75 mW	179.54 mW	112.11 mW	72.59 mW
PIPELINE 3	281.55 mW	181.61 mW	114.89 mW	74.13 mW

TABLE 4.4.6. CIRCUIT 1: V<sub>CCIO</sub> = 5 VOLTS

Finally, the power optimization obtained using circuit 1 with a double-supply voltage is shown in the following table:

GRANULARITY	P (4) /P (ref)	P (3) /P (ref)	P (1.9) /P (ref)
PIPELINE 1	0.650	0.416	X
PIPELINE 3	0.678	0.429	0.277

TABLE 4.4.7. POWER OPTIMIZATION OF CIRCUIT 1 WHEN  $V_{CCIO} = 5$  VOLTS

In this case, power is reduced 3.61 times when  $V_{CC\_Core}$  is equal to 1.9 volts and  $V_{CCIO}=5$  volts.

b)  $V_{CCIO} = 3.3$  volts

In this case, similar results are obtained. The clock frequency is fixed to 12 MHz and the internal supply voltage ( $V_{CC\_Core}$ ) takes values from 3.3 volts to 1.9 volts. The power consumed by the circuit 1 is strongly influenced by the pipeline granularity. As described, power consumed by I/O cells decrease when increasing the number of DFFs. The following table summarizes the results obtained from circuit 1 when  $V_{CCIO}$  is equal to 3.3 volts:

GRANULARITY	3.3 volts	2.6 volts	1.9 volts
PIPELINE 1	98.901 mW	66.200 mW	X
PIPELINE 3	104.244 mW	69.500 mW	47.120 mW

TABLE 4.4.8. CIRCUIT 1:  $V_{CCIO} = 3.3$  VOLTS

Finally, table 4.4.9 shows the percentage of power saved by using double supply voltage.

GRANULARITY	P (2.6) /P (REF)	P (1.9) /P (REF)
PIPELINE 1	0.669	X
PIPELINE 3	0.703	0.476

TABLE 4.4.9. POWER OPTIMIZATION OF CIRCUIT 1 WHEN  $V_{CCIO} = 3.3$  VOLTS

In this case, the power consumption is reduced 2.1 times. Nevertheless, power is already reduced when using 3.3 volts. When comparing tables 4.4.7 and 4.4.9, It must be noticed that power consumption when using  $V_{CCIO} = 3.3$  volts and  $V_{CC\_Core} = 1.9$  volts is 5.68 times lower than the power consumed when using  $V_{CC\_Core} = V_{CCIO} = 5$  volts.

#### 4.4.2.2 Results from Circuit 2

a)  $V_{CCIO} = 5$  volts

Table 4.4.10 summarizes the behavior of circuit 2 when  $V_{CCIO}$  is equal to 5 volts. In this case, the clock frequency was also fixed to 12MHz.

GRANULARITY	5 volts	4 volts	2.9 volts	1.8 volts
PIPELINE 1	365.650 mW	232.310 mW	133.394 mW	X
PIPELINE 3	364.550 mW	230.060 mW	130.919 mW	73.052 mW

TABLE 4.4.10. CIRCUIT 2:  $V_{CCIO} = 5$  VOLTS

The power saved for this circuit when using this technique with double supply voltage is illustrated in the following table:

GRANULARITY	P (4) / Ref	P (2.9) / Ref	P (1.8) / Ref
PIPELINE 1	0.635	0.365	X
PIPELINE 3	0.629	0.358	0.200

TABLE 4.4.11. POWER OPTIMIZATION OF CIRCUIT 2 WHEN  $V_{CCIO} = 5$  VOLTS

In this case, power is reduced by a factor 5. Figures and tables summarizing all the results obtained can be consulted in the appendix B.

b)  $V_{CCIO} = 3.3$  volts

Table 4.4.12 shows the power consumed by circuit 2 when  $V_{CCIO}$  is equal to 3.3 volts.

GRANULARITY	3.3 volts	2.9 volts	1.8 volts
PIPELINE 1	133.962 mW	105.023 mW	X
PIPELINE 3	134.139 mW	104.309 mW	49.698 mW

TABLE 4.4.12. CIRCUIT 2:  $V_{CCIO} = 3.3$  VOLTS

Pipeline 1, 2 and 3 present almost the same behavior. Pipeline 3 allows the circuit to reach a  $V_{CC\_Core}$  of 1.8 volts. The following table shows the percentage of power optimized using this technique with circuit 2:

GRANULARITY	P (2.9) / Ref	P (1.8) / Ref
PIPELINE 1	0.784	X
PIPELINE 3	0.779	0.371

TABLE 4.4.13. POWER OPTIMIZATION OF CIRCUIT 2 WHEN  $V_{CCIO} = 3.3$  VOLTS

In this case, the power is reduced almost 2.7 times. But if we compare this result with the original reference (5 volts and no pipeline optimization) the power is reduced 7.35 times in a 5-volt device.

## **4.5 Summary**

The use of pipeline architectures in FPGA seems natural because of the high number of programmable DFFs inside the device. It is easy to keep a good level of performance when the power supply is reduced. Moreover, internal glitches are reduced when the pipeline depth is increased. It allows the I/O cells to propagate signals with little number of incomplete transitions.

We have obtained better results than ASICs [23] because of the advantages of FPGA architectures. As mentioned, DFFs are for free inside the Logic Cells and the I/O cell also. The internal capacitance does not increase dramatically when we insert registers to create a stage of pipeline. Finally, most part of FPGA architectures contain interconnect resources based on pass-transistors. The use of pass-transistors when  $V_{DD}$  is low can reduce dramatically power consumption [23, 91].

Using this technique we can reduce the global power consumption of the FPGA by more than 75 % when using a single supply voltage, and by almost 50 % when using double supply voltage. This technique could be useful in wireless systems based on FPGA at prototype level.



## Chapter V. Conclusions

### 5.1 Conclusions

FPGA architectures are formed by different structures, such as pure CMOS, Pass-Transistor and SRAM. Hence, the power consumption behavior of these architectures is much more complex than the power behavior of a pure CMOS circuit. In CMOS, static power and dynamic power caused by short-circuit currents can be negligible. In FPGAs, static power due to direct-path currents becomes important because of the use of pass-transistors that allow the internal CMOS logic cells to be interconnected. Measurement results show also that power consumption in FPGAs behaves as a 3<sup>rd</sup>-degree polynomial function of  $V_{DD}$  (represented in equation 2.2.46), different from the pure CMOS power consumption that behaves as a 2<sup>nd</sup>-degree polynomial function of  $V_{DD}$  (see equation 2.2.26). The 3<sup>rd</sup> degree element of the equation corresponds to the power caused by short-circuit currents as explained in sub-section 2.2.2.3.

According to the theoretical overview exposed in section 2.2, the 3<sup>rd</sup>-degree component depends on  $V_{DD}$  as well as  $V_T$  (as shown in equation 2.2.32). If  $V_{DD}$  is much higher than  $2V_T$ , the 3<sup>rd</sup> degree component from equation 2.2.46 has a strong influence on the power consumption behavior. On the other hand, when  $V_{DD}$  is closer to  $2V_T$ , short-circuit currents are drastically reduced and hence the power consumption behavior is dominated by the other components. According to equations 2.2.15 and 2.2.17, static power consumption should increase when  $V_{DD}$  is reduced.

Power behavior in FPGAs should have a non-negligible DC component due to direct-path currents that increase because of the pass-transistor structure, and due to leakage currents that become important when increasing the number of equivalent ASIC gates in FPGAs. The 2<sup>nd</sup>-degree component of the equation 2.2.46 increases when the number of I/O pads and equivalent ASIC gates increase. Even if most recent FPGAs are based on 0.25 $\mu$  technology or less, the number of transistors increases exponentially, and the number of outputs increases also, resulting in an increase of dynamic power consumption.

The empirical model proposed in this dissertation allows us to estimate power consumption more precisely than models proposed by vendors. The ENST model, based on measurements, enables us to identify the sensitive elements of power consumption. This model shows the power distribution inside the device and the percentage of power consumed by each of the basic five elements making up FPGAs: Logic Blocks, Interconnect resources, I/O cells, memory cells (embedded and distributed), and clock tree.

Results show that most of the power consumed by the FPGA is caused by the logic elements (LUTs, DFFs and the local interconnect). They also show that an important percentage of power consumption is caused by the global interconnect. Consequently, we can conclude that an appropriate use of the internal resources will permit us to optimize power consumption in FPGAs. Classical methods to optimize speed and area are also useful to save power of programmable logic devices.

As mentioned above, A study of the supply voltage influence has shown that power consumption in FPGAs behaves as a 3-degree polynomial equation. This power consumption behavior indicates that the most effective way to reduce the power dissipated by a FPGA is by reducing the supply voltage. Based on this, a technique to reduce power has been proposed. This technique consists in a reduction of the supply voltage to the minimum possible voltage supported by the circuit; in this case power consumption can be reduced by more than a quadratic factor. On the other hand, the original performance level of the application is lost when  $V_{DD}$  is reduced.

In order to recover the performance lost when using a very low supply voltage, some extra stage of pipeline must be inserted. Pipeline architectures match naturally in FPGAs because DFFs inside these devices are for free in each logic cell and I/O cell. No new wires or logic cells are needed to increase the pipeline granularity; only the number of DFFs increases. Results obtained from measurements using commercial FPGAs show that, in some cases, pipeline architectures coupled with a very low supply voltage allows power consumption to be reduced by more than 75%.

This technique has been improved to take into account the supply voltage of external components using double supply voltage. Recent FPGAs have two kinds of voltage input pads:  $V_{CC\_Core}$  for the on-chip logic, and  $V_{CCIO}$  for the I/O cells.  $V_{CCIO}$  is fixed to 5 volts and 3.3 volts and  $V_{CC\_Core}$  takes values from the  $V_{CCIO}$  used to the minimum supply voltage possible. Measurements from chapter 4 show that this technique using double supply voltage allows power consumption to be reduced by about 50 percent.

## **5.2 Contributions**

In Section 3.3, a methodology of power measurements has been proposed. As mentioned above, this "incremental" methodology allows us to obtain the power consumed by each internal sub-element of the FPGA by isolating the other sub-elements. It is called "incremental" because the number of internal sub-elements is increased in an incremental way. This methodology was developed because of the poor support from vendors. It forces us to apply a reverse engineering process. The incremental methodology could be useful to make power consumption measurements of PLDs and digital systems based on programmable logic.

Based on all the measurements obtained using the incremental methodology, a power distribution model of commercial FPGAs has been obtained. This model, presented in section 3.5, allows an estimation of the power consumption in a more accurate way for a fixed toggling rate. It enables us to understand the power consumption behavior of FPGAs, to find the sensitive sub-elements of power consumption, and to know where power consumption must be optimized.

On the other hand, since this empirical model was calibrated by current measurements under some conditions (activity rate fixed), it can be used to obtain an accurate estimation of power consumption in digital systems if the activity rate is correctly estimated for each internal element.

Inspired by a well know technique to reduce power consumption in ASICs proposed by Chandrakasan (1992) [23], a technique to optimize power in FPGAs is explained in chapter 4. This technique proposes the use of pipeline architectures coupled with very low supply voltages to save power in FPGAs without performance loss. In order to improve this, the minimum supply voltage and the most optimal pipeline granularity has to be applied. Finally, this technique is applied using double supply voltage. Recent FPGAs can use two power supplies, one for the core ( $V_{CC\_Core}$ ) and the other for the I/O cells ( $V_{CCIO}$ ). In this case,  $V_{CCIO}$  is fixed to 5 volts, 3.3 volts, 2.5 volts or 1.8 volts, and  $V_{CC\_Core}$  is reduced to the minimum value tolerated by the device. The idea is to use the I/O cells like an interface between the on-chip logic using a very low supply voltage, and external devices using a different power supply. This allows the FPGA to be compatible with external components.

### **5.3 Future Work**

Most recent FPGAs use 2.5 and 1.8 volts and are larger than the FPGA families used in this work. Measurements using the new families could be taken in order to obtain the power consumption model of these devices. Results from the power distribution models could be used to build a CAD tool for power estimation that could bring more accurate results. The power estimator tools must take in account the influence of the activity rate of each internal signal.

The technique to reduce power consumption proposed in this dissertation must be explored using recent devices (2.2 volt and 1.8 volt devices). Since the minimum possible supply voltage depends on the  $V_T$  value, this technique could be used in the most recent FPGA families because in these devices (mostly in  $0.3\mu$ ,  $0.25\mu$  or  $0.18\mu$ ),  $V_T$  should have a very low value. Moreover, short circuit currents that are quite high in FPGAs should be drastically reduced when VDD is near to  $2V_T$ .

The conclusions of this dissertation could also be used to propose a new low-power FPGA architecture. It can be possible by optimizing the interconnect resources, scaling  $V_T$  in order to use a very low  $V_{DD}$ , and increasing the size of the embedded memory cells.

Dynamic reconfiguration using the most recent FPGA families can be a good solution to reduce power in digital systems. The reconfiguration of the FPGA can be based on the activity of the internal logic (enable or disable logic elements as needed) or by using a thermal monitoring of the package.

An agreement with FPGA vendors such as Xilinx™ and Altera™ is desirable in order to obtain more accurate results. This research collaboration could permit to analyze the power consumption in FPGAs at circuit level using SPICE models. It could become a new and real low-power FPGA architecture.

Finally, since FPGAs will have more than 4 million ASIC gates at the end of this year and probably 50 million ASIC gates at the end of 2005, power will be always a growing challenge in FPGA-based systems.



## Glossary

**Adder:** Logic circuit which adds two binary numbers together to form a sum.

**Algorithm:** A list of statements that are executed sequentially to implement a process.

**Anti-Fuse:** A programmable switch that is normally open, and which closes when a high voltage is applied to its terminals.

**Architecture:** The internal interconnection of a digital system. It indicates how data are transferred between various functions.

**ASIC:** Application Specific Integrated Circuit. It is an integrated circuit designed at transistor level to implement a specific function that can not be used or modified by anything else.

**Bi-directional:** Circuit or connection, which allows an electrical signal to travel in either directions.

**Boolean:** This term is used to describe the state of some variable which can have only two values, called also true or false represented as 0 or 1.

**Buffer:** A logic circuit that provides additional current/voltage to signals.

**CAD:** Computer Aided Design. It is a software tool used to create logic designs. It helps the user to design a circuit, optimize it, synthesize the logic description, and to generate a netlist that is used to configure a PLD.

**Clock:** A repeated digital waveform, with a regular mark and space that is used to synchronize logic devices.

**CMOS:** Complementary Metal-Oxide Semiconductor. It is a semiconductor production process that creates digital or analog circuits.

**Combinatorial:** This term is used to describe a logic circuit in which the instantaneous output is completely and always determined by the instantaneous input. Those systems have any register and have no method to memory storage.

**Critical path:** The path formed by several logic operators along which a digital signal has to travel. This path has the most important time delay.

**DDF:** a D-type flip-flop where the output is determined only by the value present at the input at the instant of a valid clock pulse. The output value remains valid until the next clock pulse.

**EPROM:** Erasable Programmable Read-Only Memory. This memory can be reprogrammed by first removing the charge load in each ROM cell using ultra-violet rays.

**Field Programmable Gate Array:** A device that contains logic cells, I/O cells and programmable wires. This device can be reprogrammed in the field by the user.

**FLASH:** Flash-erase EPROM technology. This element can be erased even in plastic packages. Some Flash devices can be in-system programmed. Usually, a Flash cell is smaller than an equivalent EEPROM cell and is therefore less expensive to manufacture.

**Florian editor:** Software tool that allows the user to visualize the internal elements (logic cells and interconnect) of a FPGA, and to change them by hand. Using this tool, users can select a specific area to place their design.

**FPGA:** Field Programmable Gate Array. Array of programmable logic block surrounded by programmable I/O cells that can be connected using programmable wires.

**Gate:** The smallest digital circuit possible.

**Glue logic:** Combinatorial and sequential logic for multiplexing, encoding, selecting, registering and designing state machines and other control logic.

**Granularity of a logic cell:** This term can be defined as the number of transistors, or as the number of Boolean operations that can be realized by the LC, or the number of inputs and outputs of the logic block

**Interconnection:** physical low impedance path between two points.

**I/O cell:** A programmable Input-Output cell that permits the on-chip logic to communicate with external devices. This cell can be configured as Input, Output or Bi-directional pads.

**Interconnect resources:** Segments of programmable wires and switches that allow logic cell to communicate with another logic cell or with an I/O cell.

**Logic Cell or Logic Block:** The basic unit of all FPGAs that performs the logic or arithmetic function.

**LUT:** A Look-Up Table is a memory that uses its inputs like address lines to implement any Boolean function by applying the truth table stored into the memory.

**Multiplexer:** Logic function to convert multiple input signals into a single output signal.

**Multiplier:** A logic function that performs the multiply operation.

**Netlist:** A description of the interconnections between the internal components of a PLD. It defines the internal elements that are used by an application. The netlist is used to configure the PLD.

**Pass-Transistor:** A transistor (normally NMOS) that is used as a switch to interconnect the internal wires of the FPGA or to connect a cell to a wire.

**Place and Route tool:** The CAD tool that permits the assignment of the internal resources according to the netlist.

**PLD:** Programmable Logic Device. A limited array of programmable logic blocks that can implement any logic function.

**Power Consumption:** The energy dissipated by the circuit during both operational and idle states. The energy delivered to the circuit from the power supply.

**Programming technology:** The technology used in a FPGA to provide the user-programmability.

**Programmable Switch:** A switch that permits two wire-segments to be interconnected.

**SRAM:** Array based on static memory cells. This technology allows the FPGA to be programmable and re-programmable by the end user.

**Reconfigurable:** A type of logic function that can be altered while the system is executing.

**Synthesis:** The process of converting a program description of a digital system into the hardware to implement it, or in the netlist that is used to configure a PLD.

**Toggling rate:** The percentage of complete transitions per clock cycle (i.e. the clock signal has a toggling rate of 100%).

## References

- [1] Altera. Flex10K Data Sheet. Data Book. Altera Corporation. 1998.
- [2] Altera MAX+PLUS II user guide. Altera Corporation 1998.
- [3] Atmel. AT6K Data Sheet. Data Book. Atmel Corporation. 1996.
- [4] Atmel. AT40K Data Sheet. Data Book. Atmel Corporation. 1997.
- [5] Atmel IDS user guide. Atmel Corporation 1999.
- [6] Atmel Application Notes. DSP Acceleration Using Reconfigurable Co-processor FPGA. Atmel Corporation. 1998.
- [7] Atmel Application Notes. *Saving Power with Atmel PLDs*. Atmel Corporation. 1998.
- [8] M. Aaldering. *XPLA Architecture White Paper*. Internal report. Philips Semiconductors. 1998.
- [9] W. Baker. *Philips Semiconductor's Fast-Zero Power CPLDs Utilize Incredible Power Source*. Internal report. Philips Semiconductors. 1998.
- [10] A. Bellaouar, M. I. Elmasry. *Low-Power Digital VLSI Design. Circuits and Systems*. Kluwer Academic Publishers. 1995.
- [11] V. Betz, J. Rose. *VPR: A New Packing, Placement and Routing Tool for FPGA research*. Proceedings FPL'97.
- [12] V. Betz, J. Rose, A. Marquardt. *Architecture and CAD for Deep-Submicron FPGAs*. Kluwer Academic Publishers. 1999.
- [13] G. M. Blair. *Comments on "New Single-Clock CMOS latches and Flip-Flops with Improved Speed and Power Savings"*. IEEE Journal of Solid-State Circuits. 32-10. October 1997.
- [14] E. Boemo, G. Gonzalez de Rivera, S. Lopez-Buedo, J. M. Meneses. *Some Notes on Power Management on FPGA-based Systems*. Proceedings of FPL'95.
- [15] E. Boemo, S. Lopez-Buedo. *Thermal Monitoring on FPGAs Using Ring-Oscillators*. Proceedings of FPL'97.
- [16] E. Boemo, G. Gonzalez de Rivera, S. Lopez-Buedo, J. M. Meneses. *Some Experiments About Wave Pipelining on FPGAs*. IEEE Transactions on VLSI systems. 6-2. June 1998.
- [17] E. Boemo, S. Lopez-Buedo, C. Santos, J. Jauregui and J. Meneses. *Logic Depth and Power Consumption: A Comparative Study between Standard Cells and FPGAs*. Proceedings of the XIII DCIS Conference (Design of Circuit and Integrated Systems), Madrid, Universidad Carlos III. November 1998.
- [18] E. Boemo. *Computer-Based Tools for Electrical Engineering Education: Some Informal notes*. Invited Paper of CAEE'99.
- [19] S. D. Brown, R. J. Francis, J. Rose, Z. G. Vranesic. *Field Programmable Gate Arrays*. Kluwer Academic Publishers. 1992.
- [20] I. Brynjolfson, Z. Zilic. *FPGA Clock Management for Low-Power Applications*. Poster session of ACM/SIGDA FPGA'2000.
- [21] T. Callahan, P. Chong, A. DeHon, J. Wawrzynek. *Fast Module Mapping and Placement for Datapaths in FPGAs*. Proceedings FPGA'98.

- [22] L. Carloni, P. Chong, E. Kusse. *A 1.5 volt Fine Grain Pass-Transistor FPGA*. EE 241 Project Report. University of California Berkeley. 1997.
- [23] A. Chandrakasan, S. Sheng, R. W. Brodersen. *Low Power CMOS Digital Design*. IEEE Journal of Solid State Circuits. 27-4. April 1992.
- [24] G. Chien, R. Galicia, M. Wan. *A Study on Programmable ASIC for DSP*. Internal Report. University of California Berkeley. 1996.
- [25] P. Cocchini, M. Pedram, G. Piccinini, M. Zamboni. *Fanout Optimization under a Submicron Transistor-Level Delay Model*. Proceedings of Int'l Conference on Computer Aided Design. November 1998.
- [26] K. M. Cuy. *Design Considerations Bring Unity to a Mixed-Voltage World*. EDN Access. February 1995.
- [27] J. De Michelli. *Gestion dynamique de la puissance dans les circuits et systèmes intégrés*. Journées d'étude "FTFC'99". Paris, France.
- [28] B. Dipert. *Programmable Logic: Beat the heat on Power Consumption*. EDN Access. August 1997.
- [29] T. Douseki, & all. *A 0.5-volt MTCMOS/SIMOX Logic Gate*. IEEE Journal of Solid-State Circuits. 32-10. October 1997.
- [30] S. Gailhard, O. Ingreneau, J.-Ph. Diguët, N. Julien, E. Martin. *Une méthode probabiliste pour estimer la consommation à un niveau algorithmique*. Colloque CAO circuits et systèmes, Villars de Lans, 15-17 Janvier 1997.
- [31] A. Garcia, W. Burleson, J. L. Danger. *Etude sur la consommation de puissance d'un décodeur MPEG2 à base des FPGA*. Journées d'étude "FTFC'97". Paris, France.
- [32] A. Garcia, W. Burleson, J. L. Danger. *Power Modelling in FPGAs*. Proceedings FPL'99.
- [33] A. Garcia, W. Burleson, J. L. Danger. *Low Power Digital Design in FPGAs: A Study of Pipeline Architectures Implemented in FPGAs with Low Supply Voltages*. Proceedings WVLSI'2000.
- [34] R. Geiger, P. Allen, N. Strader. *VLSI Design Techniques for Analog and Digital Circuits*. Mc Graw Hill. 1990.
- [35] A. Guyiot. *Power Consumption of Arithmetic Operators*. Journées d'étude "FTFC'97". Paris, France.
- [36] M. Kakumu, M. Kinugawa. *Power-Supply Voltage Impact on Circuit Performance for Half and Lower Submicrometer CMOS LSI*. IEEE Transactions of Electronic Devices. 37-8. August 1990.
- [37] G. Kélemen. *Conception des circuits intégrés pour la basse consommation: Méthodes comparées*. PhD Dissertation. E.N.S.T. Paris, France. 1997.
- [38] T. Kitahara, R. Brayton. *Low Power Synthesis via Transparent Latches and Observability Don't Care*. Proceedings of ISCAS'97.
- [39] P. Kocher, J. Jaffe, B. Jun. *Introduction to Differential Power Analysis and Related Attacks*. DPA Technical Information. 1998.
- [40] D. Lidsky, J. Rabaey. *Low Power Design of Memory Intensive Functions. Case Study: Vector Quantization*. Low Power CMOS Design, edited by A. Chandrakasan and R. Brodersen. IEEE press. 1998.
- [41] J. Lipman. *EDA Tools let you track and control CMOS power dissipation*. EDN Access. November 1995.
- [42] C. Lytle. *Managing Power in High-Speed PLDs*. EDN Access. September 1995.
- [43] S. Lopez-Buedo, J. Garrido, E. Boemo. *Thermal Testing on Reconfigurable Computers*. IEEE Design & Test of Computers, pp.84-90, January-March 2000.
- [44] J. V. McCanny, D. Phil, J. G. McWhirter. *Completely Iterative, Pipelined Multiplier Array Suitable for VLSI*.

- Proceedings of IEE. 129-2, April 1982.
- [45] R. Mehrotra, M. Pedram, X. Wu. *Comparison between nMOS pass-transistor Logic Styles versus CMOS Complementary Cells*. Proceedings of Int'l Conference on Computer Design: VLSI in Computers and Processors, October 1997.
- [46] E. Melcher. *Analyse Temporelle de Circuits Combinatoires*. PhD Dissertation. E.N.S.T. Paris, France. 1993.
- [47] G. Million, M. Doussot, M. Roussel, *De l'algorithme à l'architecture : Définition d'une architecture multi-FPGAs reconfigurable dynamiquement*. 6<sup>e</sup> Colloque GRETSI. Grenoble. 1997.
- [48] R. Min, T. Furrer, A. Chandrakasan. *Dynamic Voltage Scaling Techniques for Distribute Microsensor Networks*. Proceedings of WVLSI'2000.
- [49] W. Nebel, J. Mermet. *Low Power Design in Deep Submicron Electronics*. NATO ASI Series. Vol. 337. Kluwer Academic Publisher. 1997.
- [50] J. Oh, M. Pedram. *Gated Clock Routing Minimizing the Switched Capacitance*. Proceedings of Design Automation and Test in Europe. February 1998.
- [51] S. R. Park, W. Burleson. *Configuration Cloning: Exploiting Regularity in Dynamic DSP Architectures*. FPGA'99.
- [52] S. R. Park, W. Burleson. *Reconfiguration for Power Saving in Real-Time Motion Estimation*. Proceedings of ICASSP'98.
- [53] M. Pedram, X. Wu. *A New Description of MOS Circuits at Switch-Level with Applications*. IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences, Vol. E80-A, No. 10. October 1997.
- [54] M. Pedram, Q. Wu. *Battery-Powered Digital CMOS Design*. Proceedings of Design Automation and Test in Europe. March 1999.
- [55] M. Pedram, X. Wu. *Analysis of Power-Clocked CMOS with Application to the Design of Energy-Recovery Circuits*. Proceedings of Asia and South Pacific Design Automation Conference. January 2000
- [56] Philips Semiconductors. *CoolRunner Data Book*. Philips Corporation. 1998.
- [57] C. Piguet. *Tutorial: Low Power and Low Voltage CMOS Digital Design*. Journées d'étude "FTFC'97". Paris, France.
- [58] QuickLogic notes. *Low Power Clock Enabling Techniques*. QuickNote #68. QuickLogic Corporation. 1999.
- [59] J. Rabaey. *Digital Integrated Circuits: a design perspective*. Prentice Hall. 1996.
- [60] J. Rabaey. *Managing Power-Dissipation in the Generation-after-Next Wireless Systems*. Journées d'étude "FTFC'99". Paris, France.
- [61] V. Rezard. *SRAM Low-power & Ultra Low-Power*. MsC. Dissertation. ISEP and University of Paris VI. 1997.
- [62] W. Röthig. *Modélisation Comportementale de Consommation des Circuits Numériques*. PhD Dissertation. E.N.S.T. Paris, France. 1994.
- [63] T. Sakurai, R. Newton. *Alpha-Power Law MOSFET Model and its Applications to CMOS Inverter Delay and Other Formulas*. IEEE Journal of Solid-State Circuits. 25-2. April 1990.
- [64] G. Saucier. *Special Training on Validation of Complex Systems through Hardware Prototyping. Overview on the FPGA technologies*. Tutorial organized by the INPG-CSI. June 1990.
- [65] H. Schmit, D. Whelihan, P. Kamarchik, F. Gennari. *Scalable Interconnect and Power Distribution for Island-Style FPGAs*. Poster session of AACM/SIGDA FPGA'2000.
- [66] R. C. Seals, G. F. Whapshott. *Programmable Logic. PLDs and FPGAs*. Mc Graw Hill. 1997.

- [67] Sedra, Smith. *Microelectronics Circuits*. Saunders College Publishing. 1989.
- [68] O. Sentieys. *Méthodologie de Conception de Circuits et systèmes enfouis: Application dans le domaine des Télécommunications*. PhD Dissertation. University of Rennes 1 and ENSAT.
- [69] S. Singh, P. Bellec. *Virtual Hardware for graphics Applications Using FPGAs*.
- [70] S. Singh. *Architectural Description for FPGA Circuits*. Proceeding of FCCM'95.
- [71] I. Sirot. *Modélisation et Caractérisation des Signaux Logiques CMOS*. PhD Dissertation. E.N.S.T. Paris, France. 1995.
- [72] K. Skahill. *VHDL for programmable Logic*. Addison Wesley. 1996.
- [73] N. R. Shanbhag. *A Mathematical Basis for Power-Reduction in Digital VLSI Systems*. IEEE Transactions on Circuits and Systems: Analog and Digital Signal Processing. 44-11. November 1997.
- [74] A. K. Sharma. *Programmable Logic Handbook*. Mc Graw Hill. 1988.
- [75] S. Singh. *Architectural Description for FPGA Circuits*. FCCM'95.
- [76] A. Smailagic, D. Reilly, D. P. Siewiorek. *A System-Level Approach to Power/Performance Optimization in Wearable Computers*. Proceedings of WVLSI'2000.
- [77] M. R. Stan. *Low Power Techniques for Global Communication in CMOS VLSI*. PhD Dissertation. UMASS. Amherst, MA, E.U.A.
- [78] A. Tisserand, P. Marchal and C. Piguet. *AN On-line Arithmetic Based FPGA for Power Custom computing*. Proceedings of FPL'99.
- [79] R. Tocci. *Digital Systems. Principles and Applications*. Prentice Hall. 1995.
- [80] S. Turgis, N. Azemard, D. Auvergne. *Explicit Evaluation of Short-Circuit Power Dissipation for CMOS Logic Structures*. Proceedings of ISLPD'95.
- [81] J. P. Uyemura. *Fundamentals of MOS Digital Integrated Circuits*. Addison Wesley. 1988.
- [82] J. Valls, M. Martinez, T. Sansaloni, and E. Boemo. *A Study about FPGA-based Digital Filters*. 1998 IEEE SIPS (IEEE Workshop on VLSI Signal Processing). October 1998.
- [83] Varghese George, Hui Zhang, Jan Rabaey. *Design of a Low Energy FPGA*. Proceedings of ISLPED 1999.
- [84] H. J. M. Veendrick. *Short-Circuit Dissipation of Static CMOS Circuitry and Its Impact on the Design of Buffer Circuits*. IEEE Journal of Solid-State Circuits. 19-4. August 1984.
- [85] C. C. Wang, C. P. Kwan. *Low Power Technology Mapping by Hiding High-Transition Paths in Invisible Edges for LUT-based FPGAs*. Proceedings of ISCAS'97.
- [86] N. Weste, K. Eshraghian. *Principles of CMOS VLSI Design. A System Perspective*. Kluwer Academic. 1990.
- [87] K. Weiß, & all. *Power Estimation Approach for SRAM-Based FPGAs*. Proceedings of ACM/SIGDA FPGA'2000.
- [88] Q. Wu, M. Pedram, X. Wu. *Clock-Gating and Its Application to Low Power Design of Sequential Circuits*. IEEE Transactions on Circuits and Systems. 47-3, March 2000.
- [89] K. Yano & al.. *A 3.8-ns CMOS 16 x 16 multiplier using complementary pass transistor logic*. IEEE Journal Solid State Circuits. 25-2. April 1990.
- [90] H. J. Yoo. *A Study of Pipeline Architectures for High-Speed Synchronous DRAMs*. IEEE Journal of Solid-State Circuits. 32-10. October 1997.

- [91] R. Zimmermann, W. Fichtner. *Low-Power Logic Styles: CMOS Versus Pass-Transistor Logic*. IEEE Journal of Solid-State Circuits. 32-7. July 1997.
- [92] Xilinx XC4000E/XV/A series. Xilinx Data Book. Xilinx Corporation. 1998.
- [93] Xilinx FOUNDATION Series Software. Xilinx Corporation. 1998.

**Internet:**

- [1] <http://www.actel.com>. Actel™ Corporation.
- [2] <http://www.altera.com>. Altera™ Corporation.
- [3] <http://users.ids.net/~randraka/>. Andraka Consulting.
- [4] <http://www.atmel.com>. Atmel™ Corporation.
- [5] <http://brass.cs.berkeley.edu>. Berkeley Reconfigurable Architecture, Systems and Software Group.
- [6] <http://www.enst.fr>. Ecole Nationales Supérieure des Télécommunications.
- [7] <http://www.mrc.uidaho.edu/fpga/fpga.html>. FPGA related WWW Links.
- [8] <http://www-eecs.mit.edu/>. Department of Electrical Engineering and Computer Science. Massachusetts Institute of Technology.
- [9] <http://vsp2.ecs.umass.edu/vspg/vspg.html>. VLSI Signal Processing Group. University of Massachusetts, Amherst.
- [10] <http://optimagic.com/index.shtml>. The Programmable Logic Jump Station.
- [11] <http://www.radiw.net/~futurev/battery.html>. EV Batteries.
- [12] <http://www.eecg.toronto.edu/EECG/RESEARCH/FPGA.html>. FPGA Research Group of the University of Toronto.
- [13] <http://www.usc.edu/dept/ee/>. Department of EE-Systems. University of Southern California.
- [14] <http://www.ii.uam.es/>. School of Computer Science and Engineering Universidad Autónoma de Madrid.
- [15] <http://www.xilinx.com>. Xilinx™ Corporation.



## Appendix A: Power Estimation and Optimization

### Power Estimation :

# LCELLS	6		
# of h F.T.	I_meas (mA)	I per hF.T.	I h F.T.
0	7.25	REF	0
1	7.53	0.28	0.28
2	7.71	0.18	0.230
3	7.78	0.07	0.177
4	8.05	0.27	0.200
5	8.4	0.35	0.230
	Average	0.23	0.223
	P (mW/MHz)	0.115	0.112

TABLE A1. HALF FAST TRACK.

# OF LCELL	6		
# of f F.T.	I_meas (mA)	I per f F.T.	I_est* (mA)
0	7.25	0	6.882
1	7.56	0.31	7.262
2	7.82	0.26	7.642
3	8.28	0.46	8.022
4	8.66	0.38	8.402
5	9.15	0.49	8.782
	Average	0.38	
	P (mW/MHz)	0.19	

TABLE A2. FULL FAST TRACK.

# OF LCELL	6			
# OF COLUMNS	I_MEAS (MA)	I_HF.T.	I PER COLUMN	I_EST (MA)
0	7.25	7.25	0	6.882
1	7.97	7.53	0.440	7.472
2	8.45	7.71	0.370	8.062
3	9.06	7.78	0.427	8.652
4	9.52	8.05	0.368	9.242
5	10.19	8.4	0.358	9.832
		AVERAGE	0.327	
		P (MW/MHz)	0.164	

TABLE A3. COLUMNS

# of LCELLs	I_meas	P_meas (mW)	I per LCELL
1	5.76	28.8	REF
2	5.99	29.95	0.23
3	6.24	31.2	0.25
4	6.46	32.3	0.22
5	6.67	33.35	0.21
6	7.33	36.65	0.66
7	7.45	37.25	0.12
		Average	0.28166667
		P_LE(mW/MHz)	0.14083333

TABLE A4. LOGIC CELLS.

# OF LCELL	7			
CL (pF)	12			
# of Outputs	I_meas (mA)	I half F.T.	I per Output	I_est (mA)
1	7.42	0.23	ref	7.182
2	9.37	0.46	1.72	8.972
3	11.13	0.69	1.53	10.762
4	12.84	0.92	1.48	12.552
5	14.67	1.15	1.6	14.342
6	16.53	1.38	1.63	16.132
		Average	1.592	
		P (mW/MHz*pF)	0.066	

TABLE A5. OUTPUTS.

INPUTS	I_MEAS (mA)	I (NET)	I_OUTPUTS	I_LCELLS	I FFT	I_IN_EST
1	22.1	ref	12.48	2.4	3.42	2.05
2	22.97	0.87	12.48	2.4	3.8	1.27
3	23.37	0.4	12.48	2.4	4.18	0.85
4	24.38	1.01	12.48	2.4	4.56	0.80
5	25.74	1.36	12.48	2.4	4.94	0.83
6	26.33	0.59	12.48	2.4	5.32	0.73
7	26.48	0.15	12.48	2.4	5.7	0.59
8	26.58	0.1	12.48	2.4	6.08	0.48
	Average	0.64				0.95
	P (mW/MHz)	0.32				
	P_Inputs	0.13				

TABLE A6. INPUTS

DFF	I MEAS	I PER DFF	I_EST (mA)
1	29.56	ref	28.183
2	29.66	0.1	28.313
3	29.76	0.1	28.443
4	29.88	0.12	28.573
5	29.98	0.1	28.703
6	30.09	0.11	28.833
7	30.19	0.1	28.963
8	30.3	0.11	29.093
	Average	0.1057	
	P (mW/MHz)	0.1057	

TABLE A7. D-TYPE FLIP FLOP

## Power Optimization :

### a) Circuit 1.

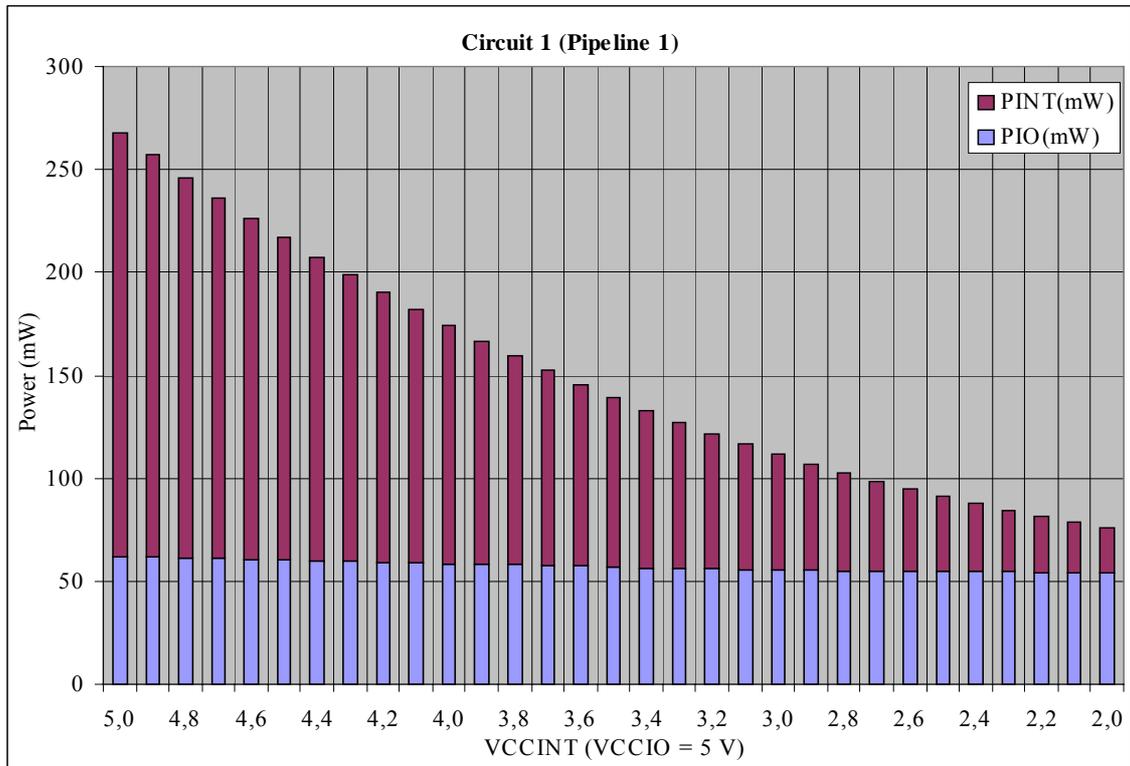


FIGURE A1. PIPELINE 1 ( $V_{CCIO} = 5$  VOLTS)

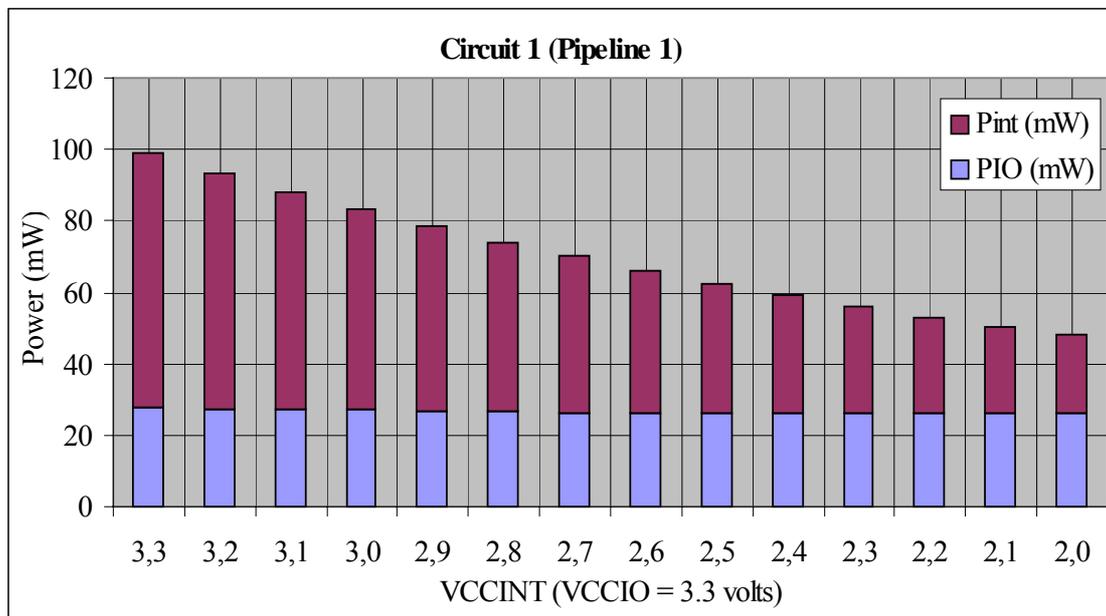


FIGURE A2. PIPELINE 1 ( $V_{CCIO} = 3.3$  VOLTS)

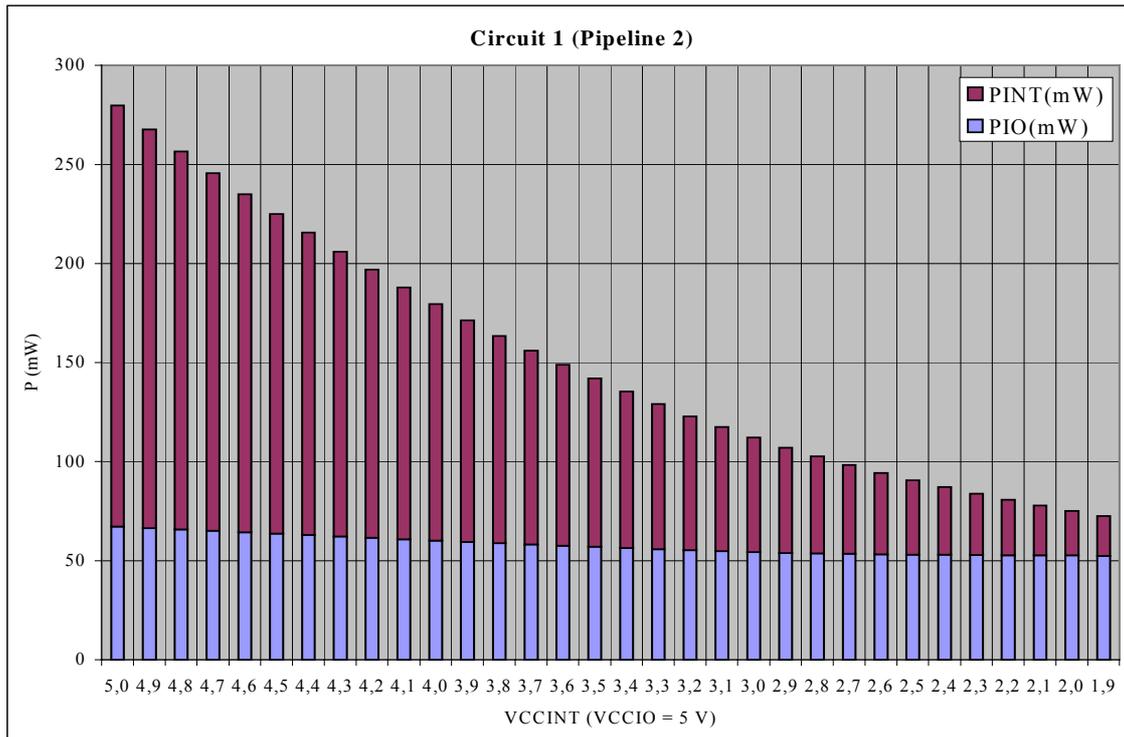


FIGURE A3. PIPELINE 2 ( $V_{CCIO} = 5$  VOLTS)

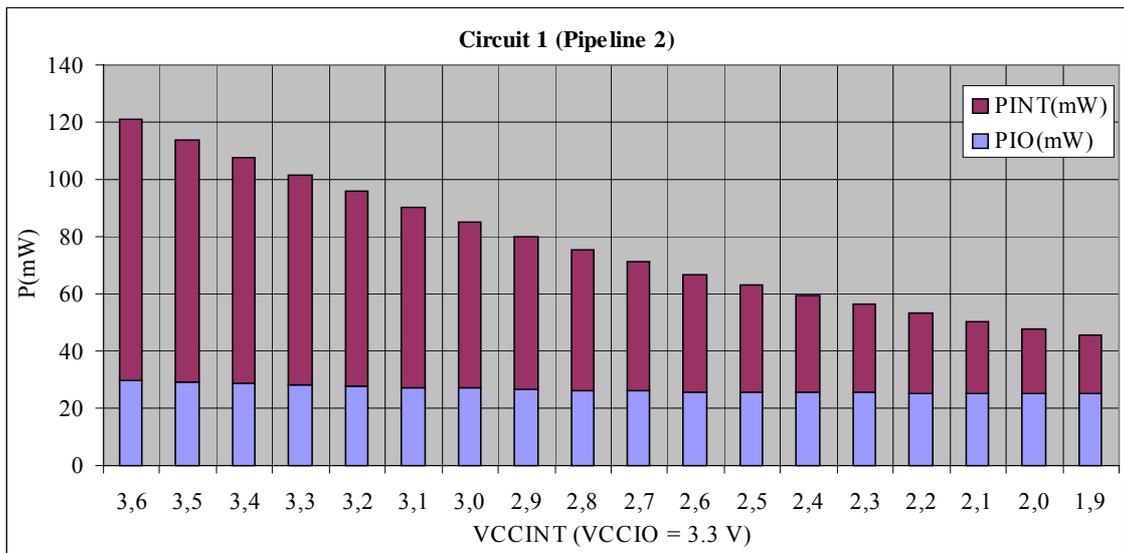


FIGURE A4. PIPELINE 2 ( $V_{CCIO} = 3.3$  VOLTS)

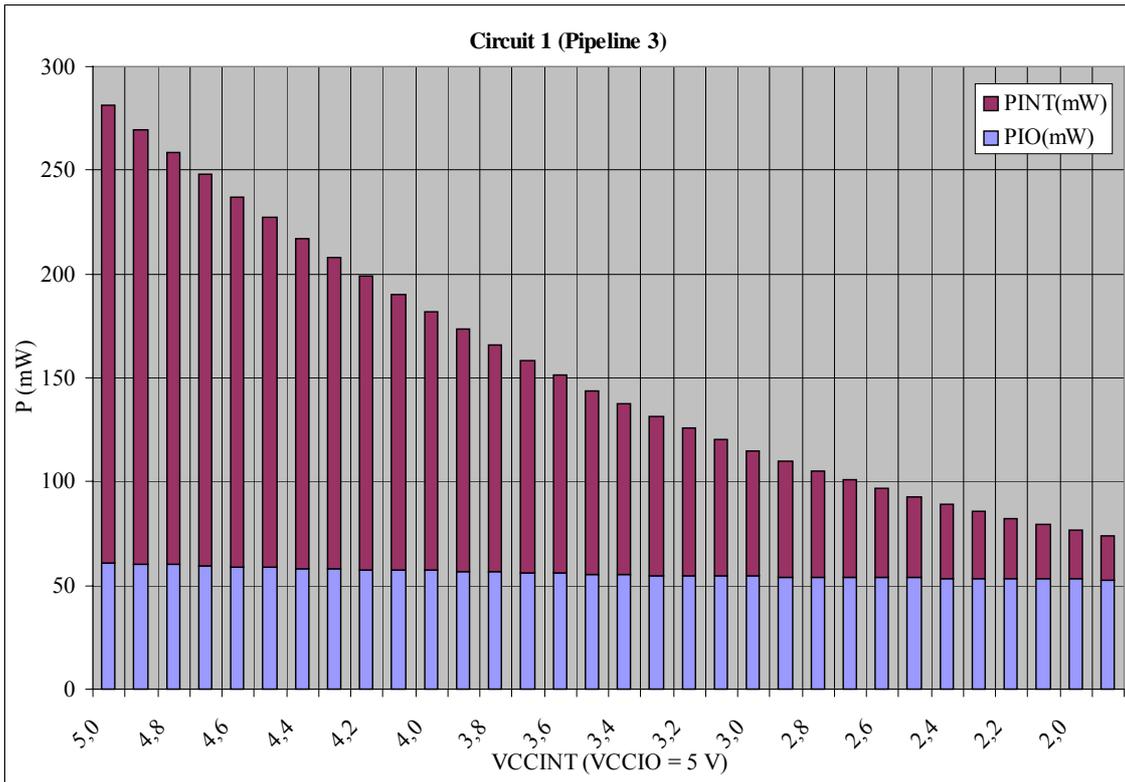


FIGURE A5. PIPELINE 3 (V<sub>CCIO</sub> = 5 VOLTS)

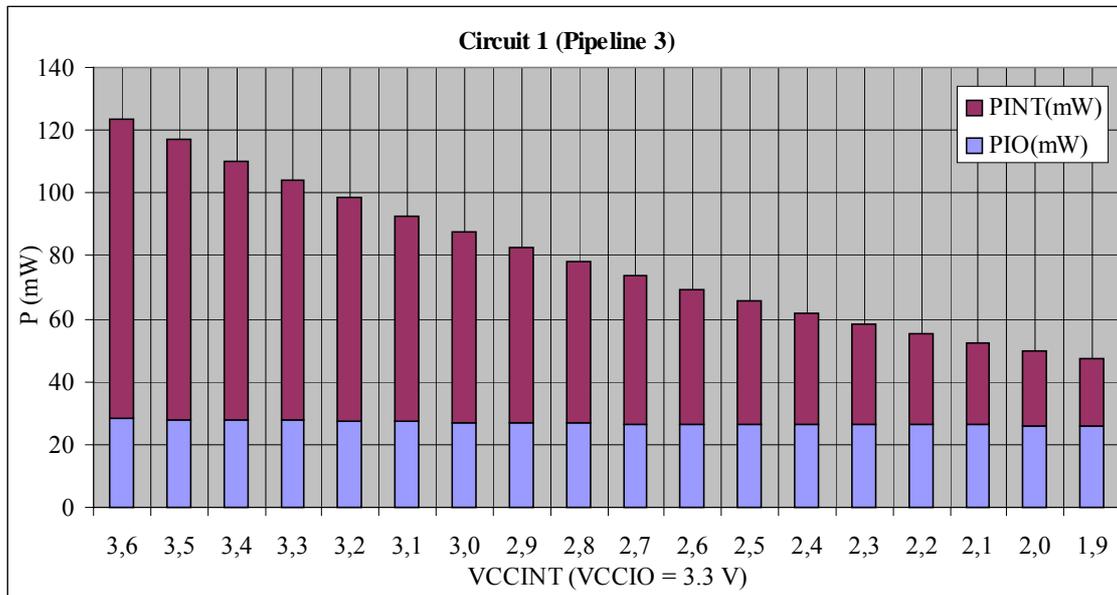


FIGURE A6. PIPELINE 3 (V<sub>CCIO</sub> = 3.3 VOLTS)

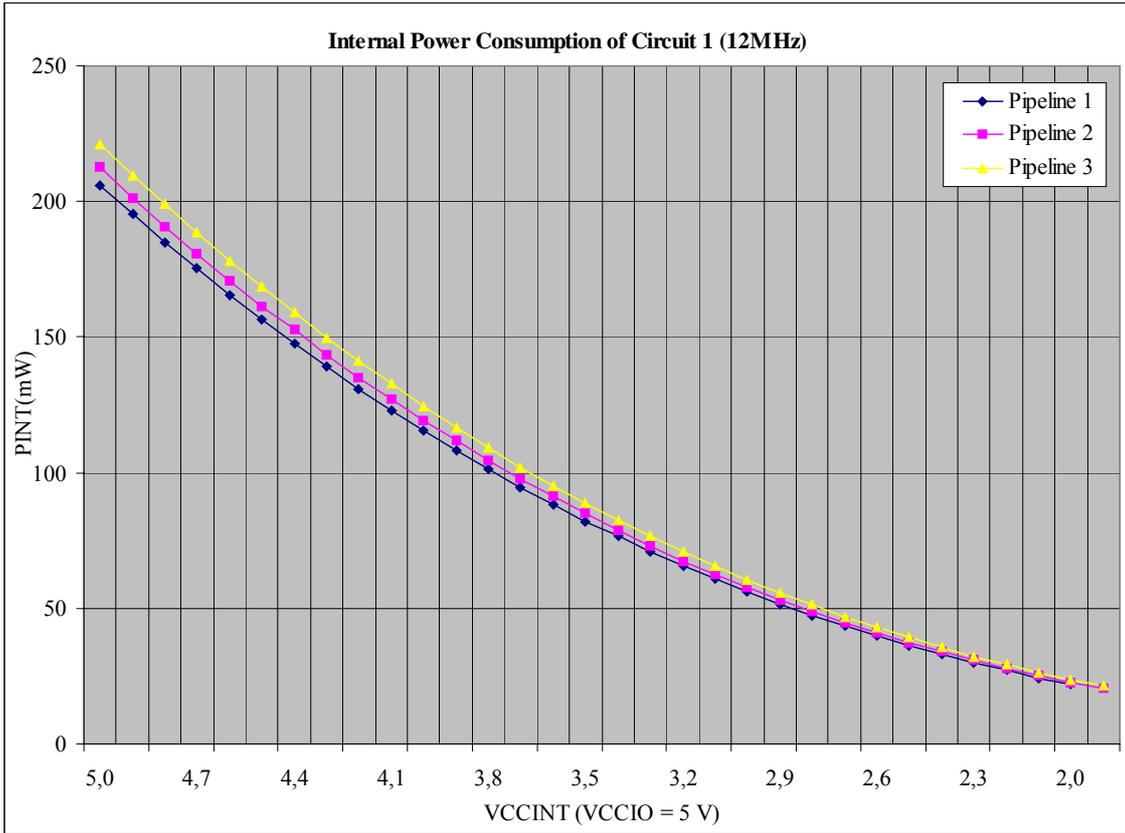


FIGURE A7. INTERNAL POWER CONSUMPTION OF CIRCUIT 1

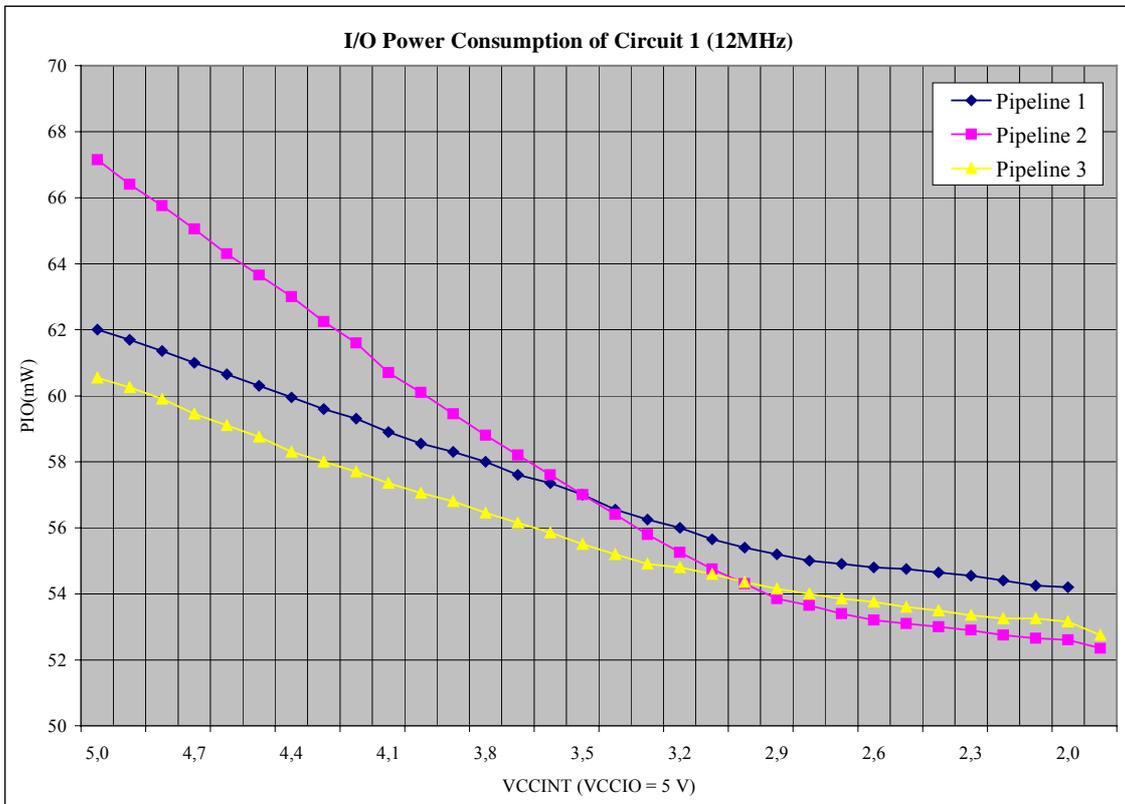


FIGURE A8 I/O CELLS POWER CONSUMPTION OF CIRCUIT 1

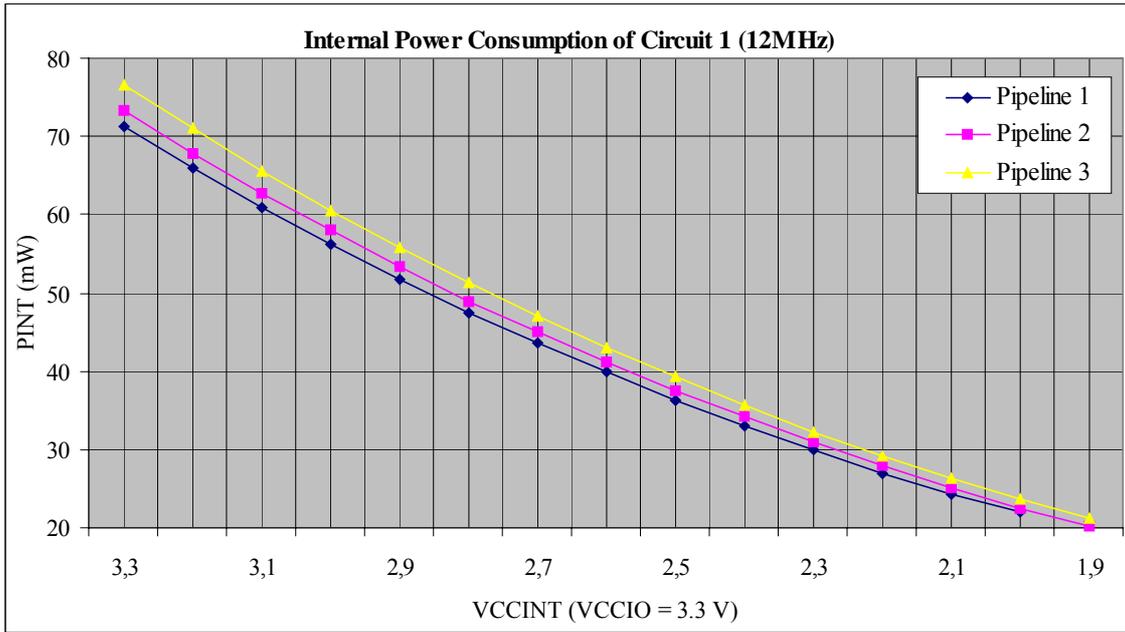


FIGURE A9. INTERNAL POWER CONSUMPTION OF CIRCUIT 1

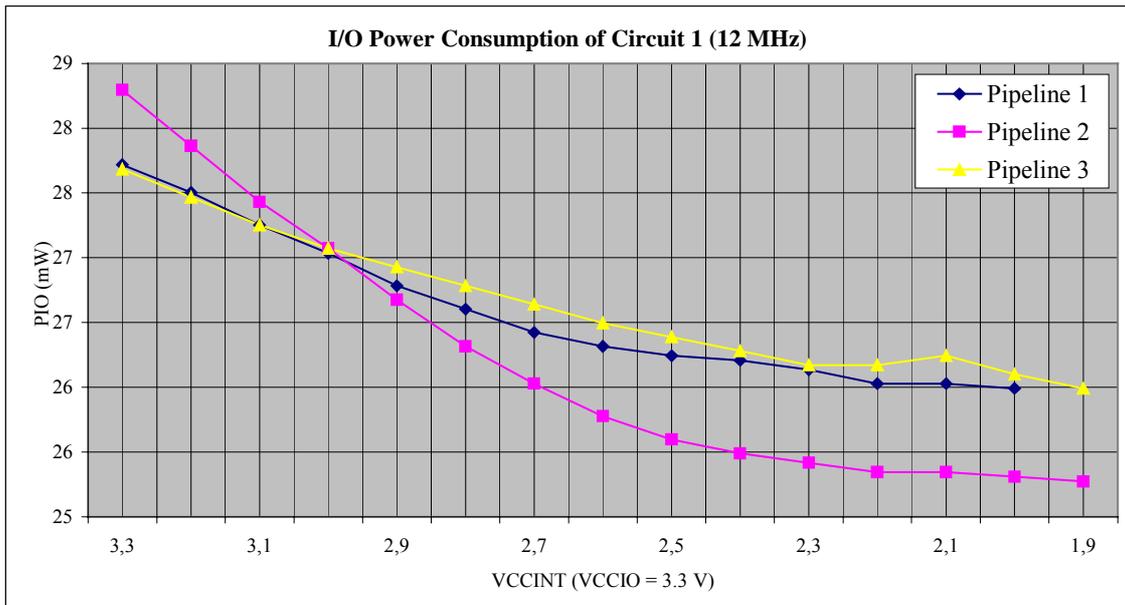


FIGURE A10 I/O POWER CONSUMPTION OF CIRCUIT 1

**b) Circuit 2.**

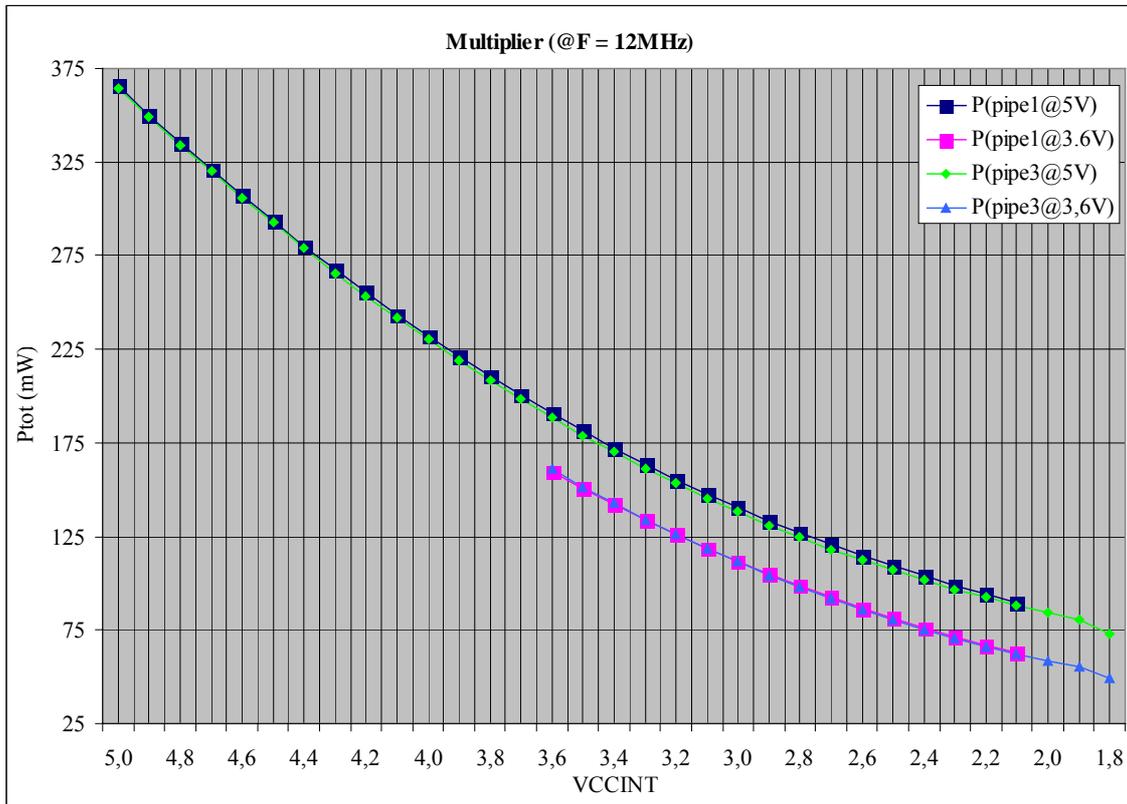


FIGURE A11. POWER OPTIMIZATION IN CIRCUIT 2

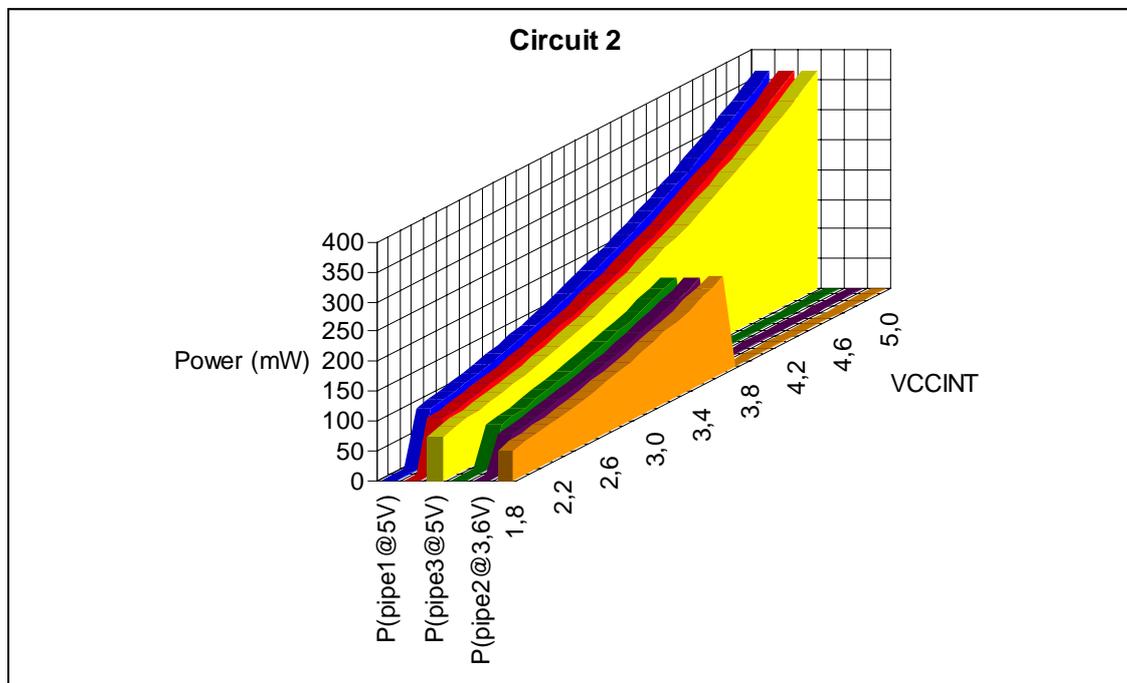


FIGURE A12. POWER CONSUMPTION OF CIRCUIT 2

**Power behavior :**

VDD	P tot (mW)	vdd <sup>2</sup>	vdd <sup>3</sup>	(vdd-2vt) <sup>3</sup>	(vdd-2vt) <sup>3</sup> +vdd <sup>2</sup>	(vdd-2vt) <sup>3</sup> +vdd <sup>2</sup> +vdd
5.0	244.950	244.950	244.950	244.950	244.950	244.950
4.9	232.309	235.24998	230.5449804	227.0342039	229.335	227.5508035
4.8	220.224	225.74592	216.7160832	210.0140063	214.419	213.2610635
4.7	208.492	216.43782	203.4515508	193.866443	200.186	199.6156644
4.6	197.294	207.32568	190.7396256	178.56855	186.621	186.5989908
4.5	186.750	198.4095	178.56855	164.0973633	173.705	174.195427
4.4	175.824	189.68928	166.9265664	150.4299188	161.423	162.3893576
4.3	165.980	181.16502	155.8019172	137.5432523	149.757	151.1651668
4.2	156.114	172.83672	145.1828448	125.4144	138.693	140.5072392
4.1	146.944	164.70438	135.0575916	114.0203977	128.212	130.3999592
4	137.680	156.768	125.4144	103.3382813	118.299	120.8277113
3.9	129.090	149.02758	116.2415124	93.34508672	108.936	111.7748798
3.8	120.954	141.48312	107.5271712	84.01785	100.108	103.2258492
3.7	112.924	134.13462	99.2596188	75.33360703	91.798	95.16500398
3.6	105.228	126.98208	91.4270976	67.26939375	83.989	87.57672855
3.5	97.930	120.0255	84.01785	59.80224609	76.665	80.44540734
3.4	91.052	113.26488	77.0201184	52.9092	69.809	73.7554248
3.3	84.381	106.70022	70.4221452	46.56729141	63.405	67.49116536
3.2	77.728	100.33152	64.2121728	40.75355625	57.435	61.63701345
3.1	71.734	94.15878	58.3784436	35.44503047	51.885	56.17735352
3	66.240	88.182	52.9092	30.61875	46.736	51.09657
2.9	60.958	82.40118	47.7926844	26.25175078	41.974	46.37904733
2.8	56.168	76.81632	43.0171392	22.32106875	37.580	42.00916995
2.7	51.273	71.42742	38.5708068	18.80373984	33.538	37.97132229
2.6	46.696	66.23448	34.4419296	15.6768	29.833	34.2498888
2.5	42.475	61.2375	30.61875	12.91728516	26.447	30.82925391
2.4	38.568	56.43648	27.0895104	10.50223125	23.364	27.69380205
2.3	35.167	51.83142	23.8424532	8.408674219	20.567	24.82791767
2.2	31.790	47.42232	20.8658208	6.61365	18.040	22.2159852
2.1	28.728	43.20918	18.1478556	5.094194531	15.766	19.84238908
2	25.960	39.192	15.6768	3.82734375	13.729	17.69151375
1.9	23.484	35.37078	13.4408964	2.790133594	11.913	15.74774364

TABLE A6. POWER CONSUMPTION OF CIRCUIT 1.

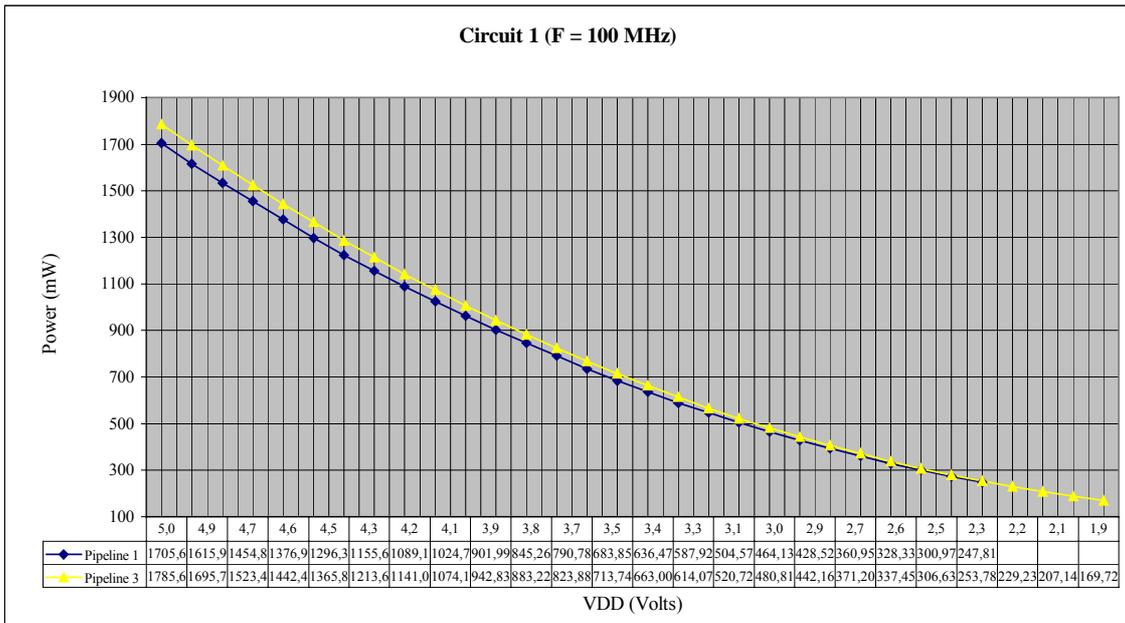


FIGURE A13. POWER CONSUMPTION OF CIRCUIT 1.

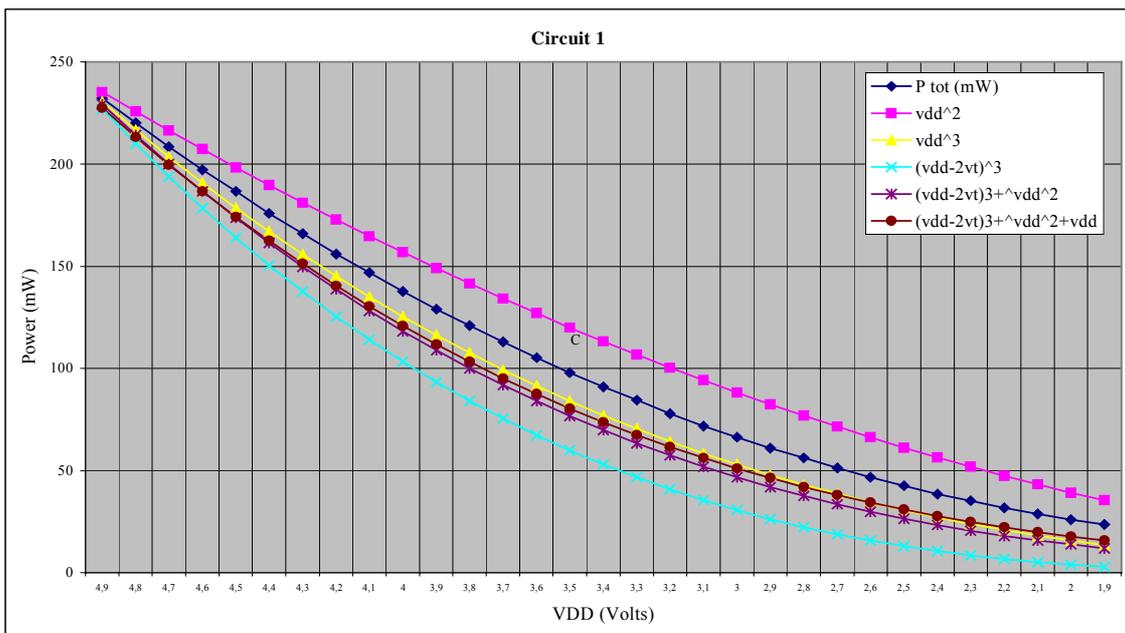


FIGURE A14. POWER BEHAVIOR OF CIRCUIT 1

## Appendix B: SPICE model of a Pass-Transistor Structure

### a) Simple CMOS inverter with pass-transistor

```
MINIMAL_INVERTER
.OPTION post PROBE nopage nomod $brief
*Inverter
.LIB '/elec/produits/technos/es2/ecpd07/hspice/lev6.ecpd07' SLOW
.PAR ldf=.8u wdf=2.8u ljdf=2.2u kdf=2
.GLOBAL ALIM

*MACROS
.MACRO MN D G S wn=wdf ln=ldf lj=ljdf
MN D G S GROUND NMOS W=wn L=ln AD='lj*wn' AS='lj*wn' PD='2*(lj+wn)'
PS='2*(lj+wn)'
.EOM

.MACRO MP D G S wp=wdf lp=ldf lj=ljdf
MP D G S ALIM PMOS W=wp L=lp AD='lj*wp' AS='lj*wp' PD='2*(lj+wp)'
PS='2*(lj+wp)'
.EOM

*BEGIN_INVERTER
.MACRO INVERTER INPUT OUTPUT wnn=wdf psn=kdf
XN OUTPUT INPUT GROUND MN wn=wnn
XP OUTPUT INPUT ALIM MP wp='wnn*psn'

.PROBE I(Vdd)
.PLOT I(Vdd)
.EOM

*BEGIN
Vdd ALIM GROUND DC=5
Vin 1 GROUND PULSE (5 0 50ns 2ns 2ns 200ns)
X1 1 2 INVERTER
C1 2 GROUND 1p

.TRAN 1ns 450ns
.PROBE v(1) v(2) i(C1) i(Vdd)
.PLOT v(1) v(2) i(C1) i(Vdd)
.PRINT v(1) v(2) i(C1) i(Vdd)
.END
```

### **b) Three CMOS inverters**

```
*BEGIN
Vdd ALIM GROUND DC=5
Vin 1 GROUND PULSE (5 0 50ns 2ns 2ns 200ns)
X1 1 ALIM 2 MN wn=wdf
X2 2 3 INVERTER
X3 3 ALIM 4 MN wn=wdf
X4 4 5 INVERTER
X6 5 ALIM 6 MN wn=wdf
C1 6 GROUND 1p

.TRAN 1ns 450ns
.PROBE v(1) v(2) v(3) v(4) v(5) v(6) i(C1) i(Vdd)
.PLOT v(1) v(2) v(3) v(4) v(5) v(6) i(C1) i(Vdd)
.PRINT v(1) v(2) v(3) v(4) v(5) v(6) i(C1) i(Vdd)

.END
```

### **c) Five CMOS inverters**

```
*BEGIN
Vdd ALIM GROUND DC=5
Vin 1 GROUND PULSE (5 0 50ns 2ns 2ns 200ns)
X1 1 ALIM 2 MN wn=wdf
X2 2 3 INVERTER
X3 3 ALIM 4 MN wn=wdf
X4 4 5 INVERTER
X5 5 ALIM 6 MN wn=wdf
X6 6 7 INVERTER
X7 7 ALIM 8 MN wn=wdf
X8 8 9 INVERTER
X9 9 ALIM 10 MN wn=wdf
X10 10 11 INVERTER
X11 11 ALIM 12 MN wn=wdf
C1 12 GROUND 1p

.TRAN 1ns 450ns
.PROBE v(1) v(2) v(3) v(4) v(5) v(6) v(12) i(C1) i(Vdd)
i(Vin)
.PLOT v(1) v(2) v(3) v(4) v(5) v(6) v(12) i(C1) i(Vdd)
i(Vin)
.PRINT v(1) v(2) v(3) v(4) v(5) v(6) v(12) i(C1) i(Vdd)
i(Vin)

.END
```

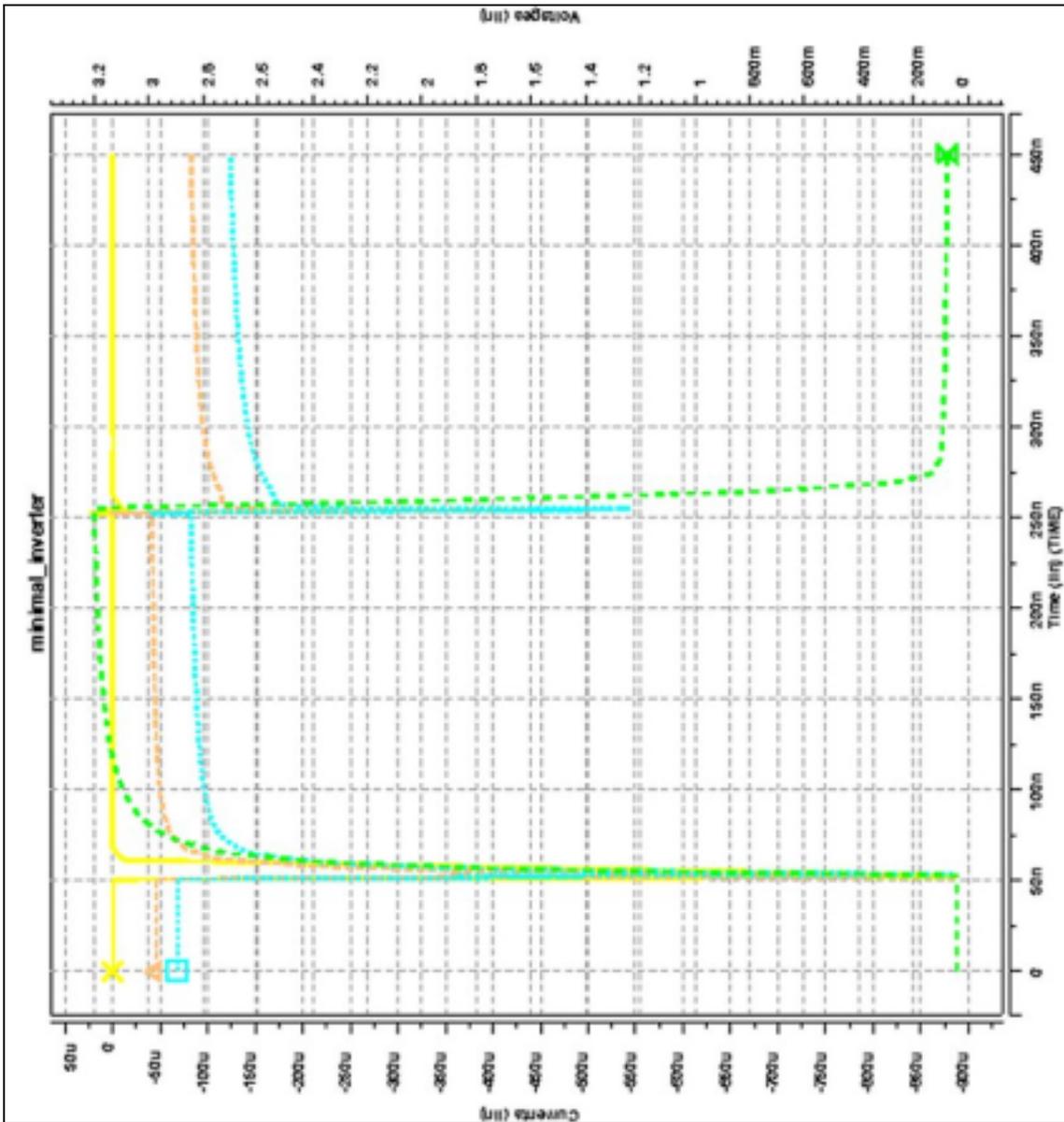


FIGURE B1. PASS-TRANSISTOR STRUCTURE

