UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM MICROELETRÔNICA

CRISTIANO LAZZARI

Transistor Level Automatic Generation of Radiation-Hardened Circuits

Dissertation submitted to UFRGS and
Institut National Polytechnique de Grenoble
in partial fulfillment of the requirements
for the degree of Doctor of Microelectronics

Prof. Dr. Ricardo Augusto da Luz Reis
Advisor

Prof. Dr. Lorena Anghel
Co-advisor

Porto Alegre, December 2007

CIP – CATALOGING-IN-PUBLICATION

Lazzari, Cristiano

Transistor Level Automatic Generation of Radiation-Hardened Circuits / Cristiano Lazzari. – Porto Alegre: PGMICRO da UFRGS, 2007.

125 f.: il.

Thesis (Ph.D.) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Microeletrônica, Porto Alegre, BR–RS, 2007. Advisor: Ricardo Augusto da Luz Reis; Coadvisor: Lorena Anghel.

1. Transistor Level Automatic Layout Generation, Transistor Sizing, Low leakage, Radiation-Hardened Circuits. I. Reis, Ricardo Augusto da Luz. II. Anghel, Lorena. III. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitor: Prof. José Carlos Ferraz Hennemann
Pró-Reitor de Coordenação Acadêmica: Prof. Pedro Cezar Dutra Fonseca
Pró-Reitora de Pós-Graduação: Prof^{a}. Valquíria Linck Bassani
Diretor do Instituto de Informática: Prof. Flavio Wagner
Coordenador do PGMICRO: Prof. Henri Boudinov
Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Gostaria de agradecer primeiramente a meus pais Eloi e Lurdes pelo apoio dado durante minha vida e, principalmente, a educação que sempre me ofereceram, servindo de exemplo para minhas atitudes.

Agradeço a minha amada esposa Patrícia pelo apoio que me tem dado nos momentos difíceis. Ela que sempre me motivou a batalhar por meus objetivos.

Sou grato aos meus orientadores Ricardo Reis e Lorena Anghel. Sem seus conselhos, suporte e encorajamento, este trabalho não teria sido possível. Agradeço pela confiança que depositaram em mim e pela amizade que surgiu a partir deste trabalho.

Agradeço aos professores Altamiro Amadeu Suzim, Luigi Carro, Marcelo Johann, Marcelo Lubaszewski, Fernanda Lima, Gilson Wirth e Sergio Bampi por me mostrarem o valor da pesquisa e me auxiliarem no meio acadêmico.

Gostaria de agradecer as pessoas que trabalharam diretamente comigo, Adriel Ziesemer, Thiago Assis, Rodrigo Bastos, Cristiano dos Santos, Daniel Ferrão, Gustavo Wilke, Renato Hentschke, e aos colegas do grupo de microeletronica, especialmente a Alessandro Girardi, Fernando Paixão Cortes, Alexandre Morais Amory, José Carlos Sant'Anna, Lucas Brusamarello, Lisane Brisolara, Marcio Oyamada e Gustavo Neuberger.

Finalmente, gostaria de agradecer a todos meus colegas do CEITEC, pelo suporte que possibilitou a finalização desta tese.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS AND ACRONYMS 9

LIST OF FIGURES 11

LIST OF TABLES 15

ABSTRACT 17

RESUMO 19

1 INTRODUCTION 21
1.1 Organization of this Thesis 25

2 A TIMING CLOSURE DESIGN FLOW USING A TRANSISTOR LEVEL AUTOMATIC LAYOUT GENERATOR 27
2.1 Introduction 27
2.2 The *Super Library* Generation 29
2.2.1 The Development of a *superlib* 31
2.3 The Transistor Level Design Flow 33
2.3.1 The Transistor Level Optimization 35
2.3.2 Transistor Level Optimization for Leakage Reduction 37
2.3.3 The Transistor Level Layout Generation 42
2.4 Transistor-Level vs Traditional Design Flow Summary 45
2.4.1 A Comparison with the Traditional Standard Cell Design Flow 45
2.5 Conclusion 45

3 ALGORITHMS FOR TRANSISTOR LEVEL AUTOMATIC LAYOUT GENERATION 49
3.1 Introduction 49
3.2 An Overview of the Algorithm Classes 49
3.2.1 Deterministic Algorithms 50
3.2.2 Stochastic Algorithms 52
3.3 Goals of Placement Algorithms 53
3.4 Goals of Routing Algorithms 54
3.5 State-of-the-art Algorithms for transistor placement and routing 54
3.5.1 Transistor Placement Using an O-tree Algorithm 54

3.5.2 Transistor Placement with Symmetry Constraints 57
3.5.3 Non-complementary Transistor Placement 58
3.5.4 Transistor Placement & Routing By Using Linear Integer Programming . 59
3.5.5 A Maze Routing Steiner Tree with Effective Critical Sink Optimization . . 60
3.5.6 Routing with a Negotiation-based Algorithm 61
3.6 Transistor Placement Technique Using Genetic Algorithm And Analytical Programming 62
3.6.1 Initial Population Generation 63
3.6.2 The Placement Parameters 64
3.6.3 The Mathematical Modeling 65
3.6.4 Obtained Results 69
3.7 Compaction and Layout Optimization using Linear Programming . . . 70
3.8 Overview of the Algorithms Developed in the Punch++ 73
3.9 Conclusion 74

4 A RADIATION-INDUCED EFFECTS OVERVIEW 75
4.1 Introduction 75
4.2 Single Event Effects (SEEs) 76
4.2.1 Soft SEE 77
4.2.2 Single Hard Errors (SHE) 82
4.3 Other Radiation-Induced Effects 82
4.4 Conclusion 82

5 STATE-OF-THE-ART TECHNIQUES FOR SOFT ERROR PROTECTION 83
5.1 Introduction 83
5.2 The Classic Techniques 84
5.3 Gate Duplication Methods 85
5.4 Gate Sizing Techniques 86
5.5 Protecting Sequential Elements Through Feedback Control 86
5.6 Using Time Redundancy To Protect Sequential Elements 87
5.7 Conclusion 91

6 AN EFFICIENT TRANSISTOR SIZING METHODOLOGY FOR SOFT ERROR PROTECTION IN COMBINATIONAL LOGIC CIRCUITS . . . 93
6.1 Introduction 93
6.2 Combinational Circuits Sensitivity 93
6.2.1 The Logical Masking 94
6.2.2 The Electrical Masking 95
6.3 An Analytical Model for Single Event Transients 96
6.3.1 Modeling Resistances and Capacitances 96
6.3.2 The Single Event Transients Model 98
6.3.3 Single Event Transient Propagation 99
6.4 The Transistor Sizing Strategy 99
6.5 The Transistor Sizing Model 101
6.6 Results 102
6.7 Conclusions 104

7 CONCLUSION 107
REFERENCES 109
APPENDIX A RESUMO EM PORTUGUÊS 119

LIST OF ABBREVIATIONS AND ACRONYMS

ASIC	Application Specific Integrated Circuit
CAD	Computer Aided Design
CTS	Clock Tree Synthesis
CMOS	Complementary Metal-Oxide-Semiconductor
CLIP	Cell Layout via Integer Programming
CVSL	Cascode Voltage Switch Logic
CWSP	Code Word State Preserving
DD	Displacement Damage
DICE	Dual Interlocked storage Cell
DSM	Deep Submicron
EDA	Electronic Design Automation
EDD	Energy Times Delay Squared
GDS	Graphic Data System format
GDSII	Graphic Data System format, second version
GND	Ground
ILP	Integer Linear Programming
IP	Integer Programming
LET	Linear Energy Transfer
LP	Linear Programming
PI	Primary Inputs
PO	Primary Outputs
PTL	Pass Transistor Logic
QP	Quadratic programming
RTL	Register Transfer Language

SAA	South Atlantic Anomaly
SCCG	Static CMOS Complex Gates
SEB	Single Event Burnout
SEE	Single Event Effects
SEGR	Single Event Gate Rupture
SEL	Single Event Latchup
SET	Single Event Transient
SEU	Single Event Upset
SHE	Single Hard Errors
SRAM	Static Random Access Memory
TID	Total Ionizing Dose
TMR	Triple Modular Redundancy
V_{DD}	Supply voltage
VLSI	Very Large Scale Integration

LIST OF FIGURES

Figure 1.1: Gate and interconnect delay versus technology generation. Delays for feature size below 100nm is estimated. 21
Figure 1.2: Active and static power in microprocessors. 22
Figure 1.3: Clock frequency of high performance ASIC and custom processors in different process technologies. 23

Figure 2.1: The proposed design flow overview. 28
Figure 2.2: Runtime and occupied memory as a function of the number of gates in a library. 30
Figure 2.3: Number of different complex logic functions in circuits in a commercial 0.35μm technology. Results were obtained with Cadence RTL Compiler. 31
Figure 2.4: Generation of the liberty file. This file is used later in the logic synthesis. 34
Figure 2.5: The proposed transistor level design flow. A) Logic synthesis with the *superlib*. B) Transistor level optimization tool. C) Layout generation tool, D) Developed tools to cope with the proposed transistor level design flow. 35
Figure 2.6: Delay and power consumption results for some circuits from the ISCAS'85 benchmarks. 38
Figure 2.7: Delay paths of the circuit c1355. 39
Figure 2.8: An NMOS two-stack inverter reduces the subthreshold leakage when input stays at logic “0”. 39
Figure 2.9: An example of fine-grained sleep transistors. 40
Figure 2.10: Leakage current in deep submicron technology process. 41
Figure 2.11: Delay paths distribution before and after transistors length biasing to the benchmarks ISCAS'85. 43
Figure 2.12: Layout style of two transistor level automatic layout generators. The *Punch* and *Punch++* tools. 44
Figure 2.13: Some layouts of circuits generated by the proposed transistor level design flow. 46

Figure 3.1: A force-directed example. 50
Figure 3.2: An Euler path example. 51
Figure 3.3: Example of a 8-node tree. 55
Figure 3.4: The O-tree representation placement. 55

Figure 3.5: Admissible o-tree. 56
Figure 3.6: Possible insertion position of a external node. 56
Figure 3.7: Datapath tile placement. 57
Figure 3.8: Placement with symmetry group and different encodings. 58
Figure 3.9: Stages of the layout synthesis. 59
Figure 3.10: The CLIP layout style. 60
Figure 3.11: Comparison of the best trees generated by (a) AMAZE, (b) AHHK and (c) P-Trees algorithms. 61
Figure 3.12: Data structure for the negotiation-based algorithm. 62
Figure 3.13: Euler path example. 63
Figure 3.14: Transistor orientation constraints. 64
Figure 3.15: Transistor behavior constraints. 64
Figure 3.16: Width and height transistors parameters. 65
Figure 3.17: A row-based boundary representation. 66
Figure 3.18: Horizontal neighborhood. 67
Figure 3.19: The Δ_{n_i} representation. 68
Figure 3.20: Preliminary placement examples. 69
Figure 3.21: Example of layout. 71
Figure 3.22: An example of layout compaction. The first figure is a layout compacted without cost assignment. Second layout is compacted with costs. 73

Figure 4.1: Location where Single Event Upsets (SEU) occured in a spacecraft into a polar orbit of altitude 700km. 75
Figure 4.2: Classification of radiation-induced effects. 76
Figure 4.3: Heavy ions and protons striking the silicon device. (a) heavy ion increasing the depletion region (b) *Spallation* caused by a proton or neutron. 77
Figure 4.4: The current curve as result of a α-particle striking a device according. 78
Figure 4.5: The propagation of a transient fault as results of a particle striking a node of a combinational block. 80
Figure 4.6: Example of a bit flip in a classic latch. 81

Figure 5.1: The classic TMR. 84
Figure 5.2: Two TMR versions with delayed clock and delayed inputs to avoid transient faults coming from the combinational blocks. 84
Figure 5.3: Latch using the DICE technique. 86
Figure 5.4: INV, NOR2 and NAND2 gates using the CWSP technique. 87
Figure 5.5: Perturbation tolerant circuit based on time redundancy. 88
Figure 5.6: Using CWSP logic inside the latch. 90

Figure 6.1: The logical masking. 94
Figure 6.2: The electrical masking. 95
Figure 6.3: Equivalent circuit for calculating circuit response to an energetic particle hit. 96
Figure 6.4: A transistor modeled as a resistance. 97

Figure 6.5: A transient pulse propagation example. 101
Figure 6.6: Timing penalty versus area overhead for TMR, CWSP and the proposed sizing technique. 105

LIST OF TABLES

Table 2.1: Maximum number of logic functions obtained by stacked transistors. 29

Table 2.2: Synthesis runtime of the ISCAS85 benchmarks with a commercial $0.35 \mu m$ standard cell library and the *superlib* with Cadence RTL Compiler. 32

Table 2.3: The gate length biasing technique for some ISCAS'85 benchmarks with a $65 nm$ technology process. 42

Table 2.4: Summary of differences of between a traditional standard cell and the proposed transistor level design flow. 47

Table 2.5: Comparison between layouts generated by the standard cell approach and the proposed transistor level design flow. 48

Table 3.1: Parameters to the transistors relationship. 65

Table 3.2: Horizontal neighborhood constraints. 67

Table 3.3: Placement results. 69

Table 3.4: Review of the challenges targeted by physical synthesis algorithms. . 74

Table 4.1: Summary of the worst case depositing charge for some technology processes for particles of with an LET of 15 $MeV cm^2/mg$ 79

Table 5.1: Area and delay of the library cells INV, NAND and NOR in comparison with CWSP cells automatically generated. 88

Table 5.2: Total area and delay of CWSP cells. 89

Table 5.3: Area overhead comparison of TMR and CWSP d-flipflops. 89

Table 5.4: CWSP and TMR techniques on microprocessors. 90

Table 5.5: Review of the techniques presented in this chapter. 92

Table 6.1: Probability of a node as a function of the gate equation. 95

Table 6.2: Approximation of intrinsic MOS gate capacitance. 97

Table 6.3: The proposed transistor sizing to single event transient attenuation. Results show the area, timing and average power overhead for symmetric and asymmetric sizing techniques for particles with charge $Q = 0.3pC$ with final sensitivity of 50%. 102

Table 6.4: The proposed transistor sizing to single event transient attenuation. Results show the area, timing and average power overhead for symmetric and asymmetric sizing techniques for particles with charge $Q = 0.3pC$ with final sensitivity of 0%. 103
Table 6.5: A comparison among TMR, CWSP and the proposed sizing techniques. Results show area overhead and timing penalties to protect some circuits against particles with charge $Q = 0.3pC$ with final sensitivity of 0%. 103

ABSTRACT

Deep submicron (DSM) technologies have increased the challenges in circuit designs due to geometry shrinking, power supply reduction, frequency increasing and high logic density. The reliability of integrated circuits is significantly reduced as a consequence of the susceptibility to crosstalk and substrate coupling. In addition, radiation effects are also more significant because particles with low energy, without importance in older technologies, start to be a problem in DSM technologies. All these characteristics emphasize the need for new Electronic Design Automation (EDA) tools. One of the goals of this thesis is to develop EDA tools able to cope with these DSM challenges. This thesis is divided in two major contributions. The first contribution is related to the development of a new methodology able to generate optimized circuits in respect to timing and power consumption. A new design flow is proposed in which the circuit is optimized at transistor level. This methodology allows the optimization of every single transistor according to the capacitances associated to it. Different from the traditional standard cell approach, the layout is generated on demand after a transistor level optimization process. Results show an average 11% delay improvement and more than 30% power saving in comparison with the traditional design flow. The second contribution of this thesis is related with the development of techniques for radiation-hardened circuits. The Code Word State Preserving (CWSP) technique is used to apply timing redundancy into latches and flipflops. This technique presents low area overhead, but timing penalties are totally related with the glitch duration is being attenuated. Further, a new transistor sizing methodology for Single Event Transient (SET) attenuation is proposed. The sizing method is based on an analytic model. The model considers independently pull-up and pull-down blocks. Thus, only transistors directly related to the SET attenuation are sized. Results show smaller area, timing and power consumption overhead in comparison with TMR and CWSP techniques allowing the development of high frequency circuits, with lower area and power overhead.

Keywords: Transistor Level Automatic Layout Generation, Transistor Sizing, Low leakage, Radiation-Hardened Circuits.

Geração Automática de Circuitos Tolerantes a Radiação no Nível de transistores

RESUMO

Tecnologias submicrônicas (DSM) têm inserido novos desafios ao projeto de circuitos devido a redução de geometrias, redução na tensão de alimentação, aumento da freqüência e aumento da densidade de lógica. Estas características reduzem significativamente a confiabilidade dos circuitos integrados devido a suscetibilidade a efeitos como crosstalk e acoplamento de substrato. Ainda, os efeitos da radiação são mais significantes devido as partículas com baixa energia começam a ser um problema em tecnologias DSM. Todas essas características enfatizam a necessidade de novas ferramentas de automação. Um dos objetivos desta tese é desenvolver novas ferramentas aptas a lidar com estes desafios. Esta tese é dividida em duas grandes contribuições. A primeira está relacionada com o desenvolvimento de uma nova metodologia com o objetivo de gerar circuitos otimizados em respeito ao atraso e ao consumo de potência. Um novo fluxo de projeto é apresentado na qual o circuito é otimizado no nível de transistor. Esta metodologia permite otimizar cada transistor de acordo com as capacitâncias associadas. Diferente da metodologia tradicional, o leiaute é gerado sob demanda depois do processo de otimização de transistores. Resultados mostram melhora de 11% em relação ao atraso dos circuitos e 30% de redução no consumo de potência em comparação à metodologia tradicional. A segunda contribuição está relacionada com o desenvolvimento de técnicas de geração de circuitos tolerantes a radiação. Uma técnica CWSP é usada para aplicar redundância temporal em elementos seqüenciais. Esta técnica apresenta baixa utilização de área, mas as penalidades no atraso estão totalmente relacionadas com a duração do pulso que se planeja atenuar. Além disso, uma nova metodologia de dimensionamento de transistores para falhas transientes é apresentada. A metodologia de dimensionamento é baseada em um modelo analítico. O modelo considera independente blocos de transistores PMOS e NMOS. Então, somente transistores diretamente relacionados à atenuação são dimensionados. Resultados mostram área, atraso e consumo de potência reduzido em comparação com as técnicas CWSP e TMR, permitindo o desenvolvimento de circuitos com alta freqüência.

Palavras-chave: geração automática de leiaute, dimensionamento de transistores, baixo consumo de potência, circuitos tolerantes à radiação.

1 INTRODUCTION

Deep submicron (DSM) technologies have created new challenges to the design of circuits due to geometry shrinking, power supply reduction, frequency increasing and high logic density (CONG; SARRAFZADEH, 2000). These characteristics reduce significantly the reliability of the integrated circuits due to the susceptibility to crosstalk and substrate coupling (IEEE Design & Test Staff, 2002).

Interconnections have presented an increased importance in DSM technologies. New technologies have shifted the design paradigm from a conventional logic-dominated to an interconnect-dominated design process (CHENG, 2007).

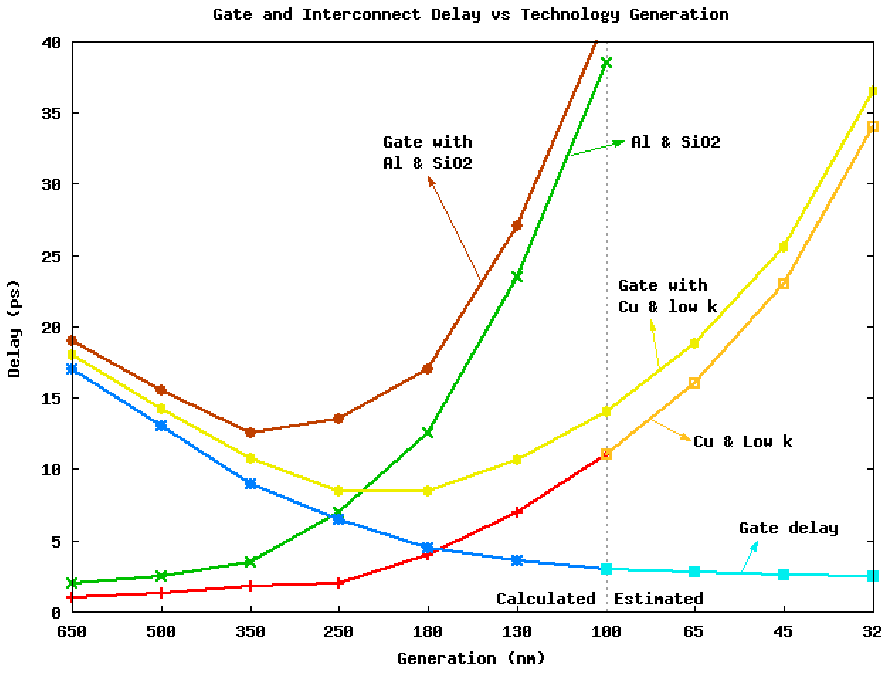


Figure 1.1: Gate and interconnect delay versus technology generation (SIA, 1997). Delays for feature size below $100nm$ is estimated.

Figure A.1 shows the amount of timing as a function of the technology generations (SIA, 1997). Technologies below $100nm$ are estimated. It is possible to note that the interconnection delay exceeds the gate delay in the $250nm$ technology process when *Al* is used for interconnections and SiO_2 is used as dielectric.

Interconnection delay presents more importance in the $0.18\mu m$ technology process when *Cu* is used in the interconnections and a *low k* dielectric is used as insulator.

The amount of delay due to interconnections is not the only important factor in DSM technologies. The high logic density and the increased number of metal layer make interconnection charactaristics prediction a very hard task. In a $180nm$ technology, for example, the capacitance per unit length may variate up to 35 times (VUJKOVIC, M. et al., 2004).

Usually, the logic of a circuit is optimized with respect to timing, area and power with assumed capacitances, but their actual values are not known until after the layout phase. Timing closure cannot be achieved with this inaccurate prediction of interconnections without the integration of the layout with the whole process.

The advent of deep submicron technologies also makes mandatory the control of the static power consumption. Designers have been concerned with the dynamic power consumption over all the technology evolution. On the other hand, the power leakage was not taken into account due to its low amount in the total power consumption. However, static power is drastically increasing in DSM technologies.

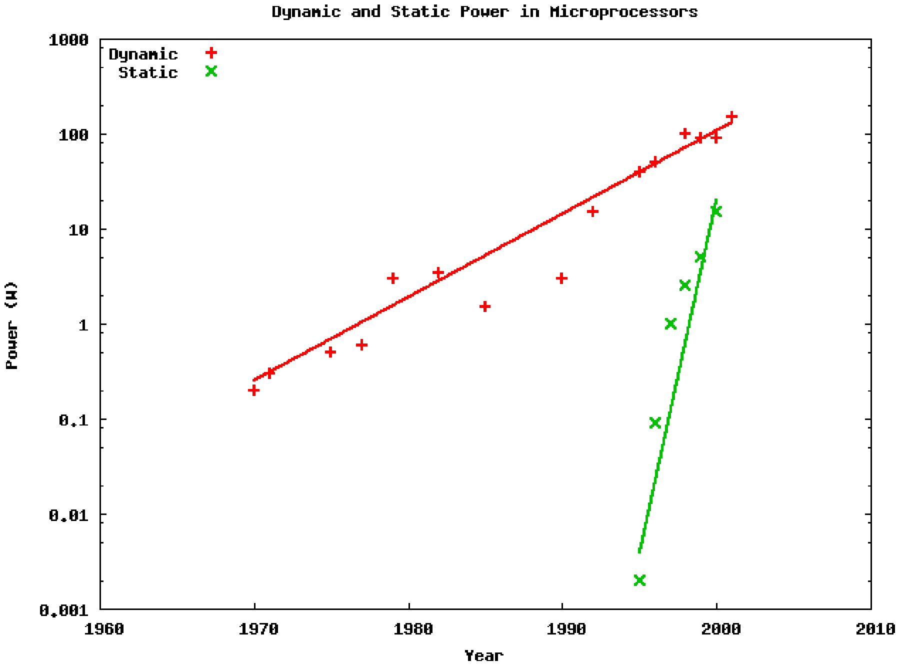


Figure 1.2: Active and static power in microprocessors (MOORE, 2003).

Figure A.2 shows the dynamic and static power in microprocessors as a function of the year (MOORE, 2003). The power leakage is projected to exceed the dynamic power consumption in $65nm$ technologies (KIM, N. S. et al., 2003).

Dealing with these deep submicron challenges require careful planning (OTTEN; CAMPOSANO; GROENEVELD, 2002) and emphasize the need for electronic design automation (EDA) tools able to automatically generate and validate integrated circuits.

Standard cell based designs have dominated the layout generation of digital VLSI circuits due some virtues (GOPALAKRISHNAN; RUTENBAR, 2001). Standard cells hide the increasingly unpleasant details of shape-level design rules, IO pins are arranged on individual gates in geometry accessible locations, cells are quite easily assembled with relative ease into row-based blocks and cells can be pre-characterized for timing and power.

However, the gap between the standard cell and custom applications are well know in the literature (ANDREW, 1998; ERIKSSON et al., 2003).

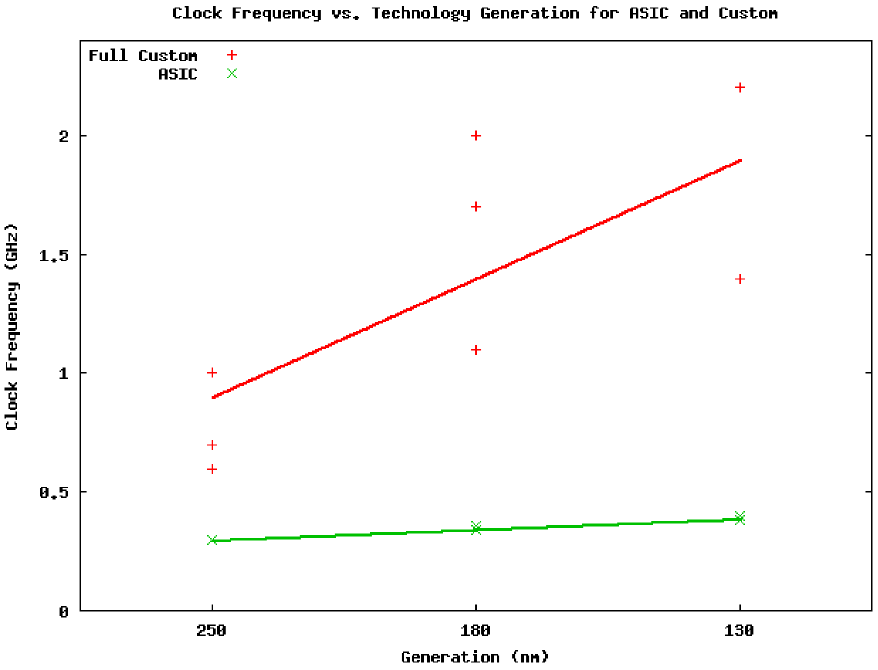


Figure 1.3: Clock frequency of high performance ASIC and custom processors in different process technologies (CHINNERY; KEUTZER, 2002).

Figure A.3 quantifies the differences between ASIC and custom chips speeds. Chinnery and Keutzer classify some ASIC and custom processors and evaluates the frequency as a function of the generation of technology (CHINNERY; KEUTZER, 2002). Some of the custom circuits are Intel, AMD and IBM Power PC processors. Tensilica, Lexra and ARM are classified as ASIC processors.

Figure shows that the frequency gap between custom and ASICs is increasing. Authors report some of the factors that contributes to the superior performance in custom applications. Factors are mainly related to microarchitecture modifications, efficient clock tree design, cells and wire design and transistor sizing. Besides, custom design offers the advantage of full control over the size and the location of each transistor for performance tuning.

Although standard cells are effective to address the layout generation problem, some works have proposed some changes in the traditional design flow in order to cope with the gap between ASIC and custom circuits.

A post-layout optimization is presented in (HASHIMOTO; ONODERA, 2001) in which they propose a downsizing methodology with interconnects preserving. They obtained a 65% power reduction without increasing the circuit delay.

Vujkovic *et al* (VUJKOVIC, M. et al., 2004) show that a standard cell library usually cannot handle the huge connection capacitance variation with a small number of versions of a cell. Furthermore, if a cell is able to deal with the required load capacitance, the cell wastes a lot of power due to the oversized transistors.

The concept of *flex cells* is presented in (ROY; BHATTACHARYA; BOPPANA, 2005). They propose the post-layout identification and optimization of a minimum number of critical cells aiming at enhancing the target design performance.

The first contribution of this thesis is to propose a new paradigm in the design of circuits. In the traditional standard cell design flow, the layout of cells are generated and characterized for timing, area and power consumption. These cells are grouped in a library and used during the whole design process.

We propose a transistor level design flow in which the layout generation is integrated within the whole design process. Transistor level design flow consists on tailoring which individual gate of a circuit. Thus, transistors in critical paths of the circuit are optimized but gates in non-critical paths are maintained with minimum transistors sizes.

The reduction on the dynamic power consumption is a direct consequence of the optimized sizing. Transistor level optimization includes a gate length biasing methodology in order to cope with the power leakage. The length of the transistors outside the critical path are increased to reduce the static power without causing timing penalties in the circuit.

Radiation effects have been also increased in DSM technology process. Reduced transistor gate dimension and low voltage levels cause an increase in the soft error failure rate in integrated circuits.

In the 20^{th} century, single event effects (SEE) were assumed to be concerned mainly in the space environment. Thus, hardening techniques were employed to these applications to avoid loss of information or functional failure.

However, the technology scaling has reduced the reliability of integrated circuits concerning radiation effects. The energy of neutrons, for example, varies between $1MeV$ to $10MeV$ at the atmosphere (NORMAND, 2001). Worthless in older technologies, the energy in the atmospheric neutron flux is enough to drastically affect the functionality of current integrated circuits.

Kastensmidt reports that memory elements in a $0.25\mu m$ technology and combinational logic composed of transistors with length smaller than $0.13\mu m$ may be subject to SEE while operating in the atmosphere (BAUMANN, 2005). For this reason, the development of electronic design automation tools able to cope with the SEE is strongly encouraged.

The second contribution of this thesis is related to the generation of radiation-hardened circuits. A transistor level design flow allows the development of any structure or methodology at transistor level. Different from the conventional standard cell, new techniques could be directly applied in the design flow.

A new technique for protecting sequential elements is proposed in which temporal redundancy is inserted inside the latches and flipflops structures. Besides, a new analytical transistor sizing methodology is proposed in order to cope with single event effects in combinational blocks.

The main characteristic of this new sizing model is the possibility to optimize independently pull-down and pull-up structures of each gate. This allows radiation-hardened combinational circuit to be performed with smaller penalties to the circuit functionality.

1.1 Organization of this Thesis

This thesis is organized in two major parts. The first part is related to the generation of optimized circuits at transistor level.

The proposal of a transistor level design flow is presented in Chapter 2. Main aspects about the proposed methodology and its impact on the generation of integrated circuits are discussed. A comparison with the traditional design flow is also presented in which some results concerning timing and power consumption are reported.

Algorithms for physical synthesis are presented in Chapter 3. This chapter presents algorithms implemented in the layout generator. Basically, algorithms are used to place, route and compact the layout. The chapter includes a discussion about the importance of the EDA tools on the design of integrated circuits.

The second part of the thesis is related to the protection of integrated circuits against SET. A discussion about radiation effects are given in Chapter 4. The discussion is focused mainly in the Soft Single Event Effects (SEE) due to their relevance with the development of this thesis. The main principles about Single Event Transients (SET) and Single Event Upsets (SEU) are highlighted.

Based on these definitions, Chapter 5 presents some state-of-the-art techniques for soft error protection. These techniques involve methodologies for radiation hardening in sequential and combinational circuits. A special attention is given to the Code Word State Preserving (CWSP) technique (ANGHEL, 2000) because it was the basis for the development of a new technique aiming at protecting sequential elements.

The last chapter proposes a new transistor sizing methodology for the protection of combinational circuits against SEE. The main contribution of this methodology is the reduced overhead. The main characteristic of the proposed methodology is to find the smallest transistor widths to attenuate SETs in the nodes of a combinational circuit. Another important point is that pull-up and pull-down transistors are independently sized, minimizing the area overhead and power consumption.

2 A TIMING CLOSURE DESIGN FLOW USING A TRANSISTOR LEVEL AUTOMATIC LAYOUT GENERATOR

2.1 Introduction

Traditionally, standard cell libraries are used in digital circuits due some virtues such as the possibility to hide unpleasant details of layout rules and the pre-characterization for timing and power (GOPALAKRISHNAN; RUTENBAR, 2001). Cells are implemented in different versions allowing to attack diferent drive strenghts. Usually, a standard cell library presents around ten versions of inverters/bufers and only four or five versions of the other cells (VUJKOVIC, M. et al., 2004).

The advent of deep-submicron technologies shifted the design paradigm from conventional logic-dominated to an interconnect-dominated design process (CHENG, 2007). Thus, the huge variance in the wire capacitances cannot be efficiently handled wiht the limited number of versions of a cell in a standard cell library. In order to drive the output load respecting the required timing, cells may waste a lot of power due to oversized transistors (VUJKOVIC, M. et al., 2004).

In the last years, some works have been presented in order to deal with this layout optimization problems. Vujkovic *et al* presented in (VUJKOVIC, M. et al., 2004) a design flow, where the number of versions of each standard cell is increased. Obtained results show an improvement of up to 20% related to the energy times delay squared (EDD) mesurements. The authors consider that the best performance metric is the effectively EDD.

The concept of *flex cells* is presented in (ROY; BHATTACHARYA; BOPPANA, 2005). They proposed the identification and optimization of critical cells. In this context, the synthesis of a minimum number of cells is performed aiming at enhancing the target designs performance. For this, the process must take into account both functionality and timing contexts in which each unique cell is used.

A post-layout optimization is presented in (HASHIMOTO; ONODERA, 2001). They proposed a method for reducing the power consumption by downsizing transistors preserving interconnections. Results presented by authors show that power can be reduced by 65% on average without delay increase when applying post-layout transistor sizing.

The main idea of the proposed work is to explore switch level characteristics in order to reduce the gap between standard cell and full custom design. For this, a complete RTL-to-Layout design flow generation is proposed. The design flow includes the generation of a database used as basis to the logic synthesis and the layout generation. The design flow is based on academic and commercial tools aiming at achieving timing closure and reduced power consumption.

Timing closure may be obtained by sizing transistors of a circuit in a wide range of possibilities. In other words, a transistor level design flow allows to tailor transistors in critical paths of the circuit maintaining other paths with minimum transistors sizes. The reduction on the power consumption is a direct consequence of this optimized sizing.

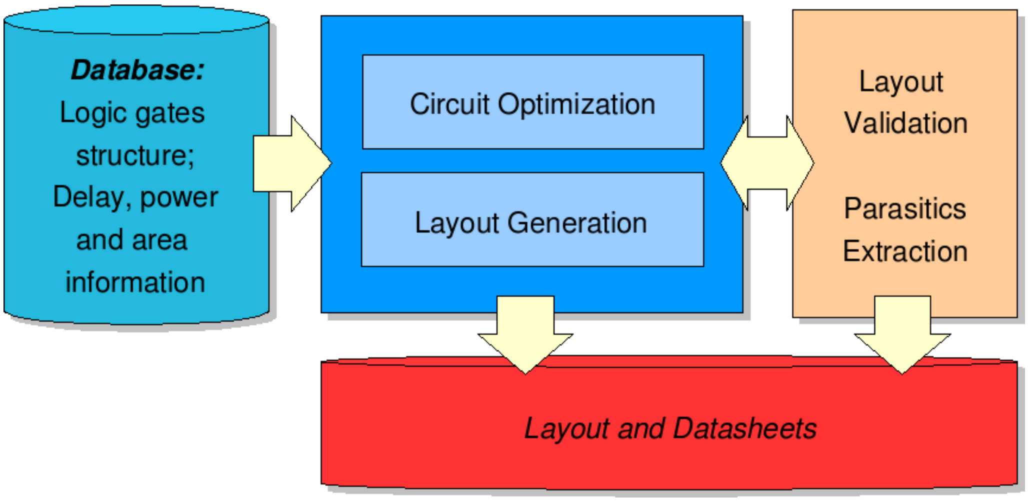


Figure 2.1: The proposed design flow overview.

Figure 2.1 shows an overview of the proposed design flow. First of all, the design flow is based on a database where the structure of each cell, its estimated area, timing and power consumption are stored.

Circuit optimization is the first step in our design flow. Thus, the circuit descrition is taken with minimum transistor sizes. For each iteration, the longest path is extracted, transistors are resized and the power consumption is analyzed. A power-delay tradeoff can be plot in the end of the optimization process. These preliminarily results allow the designer to choose the best delay and power characteristics according to the design specifications.

The next step is the layout generation. Once the power-delay trade off was chosen by the designer, the layout can be generated. In this step, transistors are placed, folded and routed according to state-of-the-art algorithms. Details about these algorithms are presented in Chapter 3.

Layout extraction allows the evaluation of the generated layout and give to designers important information about the optimization process. If the extracted delay and power met the designer specifications, the flow is finished. Otherwise, extracted data are used to a new optimization phase and the layout is generated again with the optimized layout.

Further details about the design flow are given in the next sections.

2.2 The *Super Library* Generation

Timing closure can be a hard task due to the limited number of logic functions and drive strengths available in standard cell libraries of traditional design flows. Libraries have up to 200 different logic functions and around four drive strengths. Inverters and buffers usually have up to ten drive strengths. The total number of cells is approximately 3,000 cells. These characteristics are hard limitations to the efficiency of the synthesis and sometimes may degrade power and timing results.

In a standard cell design flow, cells are generated and characterized for timing and power. The whole process takes long time and needs an enormous effort from designers. As consequence, a limited number of cells and drive strengths are available for logic synthesis and technology mapping.

Table 2.1: Maximum number of logic functions obtained by stacked transistors (DETJENS, E. et al., 1987).

		Number of PMOS stacked transistors				
		1	2	3	4	5
Number of NMOS stacked transistors	1	1	2	3	4	5
	2	2	7	18	42	90
	3	3	18	87	396	1677
	4	4	42	396	3503	28435
	5	5	90	1677	28435	425803

A research published in (DETJENS, E. et al., 1987) shows the possibilities on generating libraries according to the number of stacked transistors. Table 2.1 shows the result of this research where the number of possible combinations are very larger than the number of logic functions available in a library.

Another aspect concerning the number of library cells is related to the quality of the algorithms for logic and physical synthesis. Algorithms presented in the literature in the last years usually show a linear behavior in respect to the complexity of circuits. This linear behavior allows to increase significantly the number of cells in the libraries.

Figure 2.2 shows the performance and occupied memory of synthesis tools as a function of the number of cells in a library. Studies of the globally based synthesis algorithms show that they can effectively take advantage of sets of 10K or more cells due to the linear runtime and memory usage (CADENCE, 2007a).

The concept of *library-free mapping* was introduced in (REIS et al., 1997). The library-free is a method for mapping a set of boolean equations into a set of static CMOS complex gates (SCCG) under a constraint in the number of stacked transistors. Reis highlights that the number of transistors in synthesized circuits is inversely proportional with the static power consumption.

The most important characteristic of the library-free mapping is related to the reduced number of transistors in comparison with standard cell libraries with a restricted number of logic functions. The library free mapping may result in 20-30% reduction in the number of transistors.

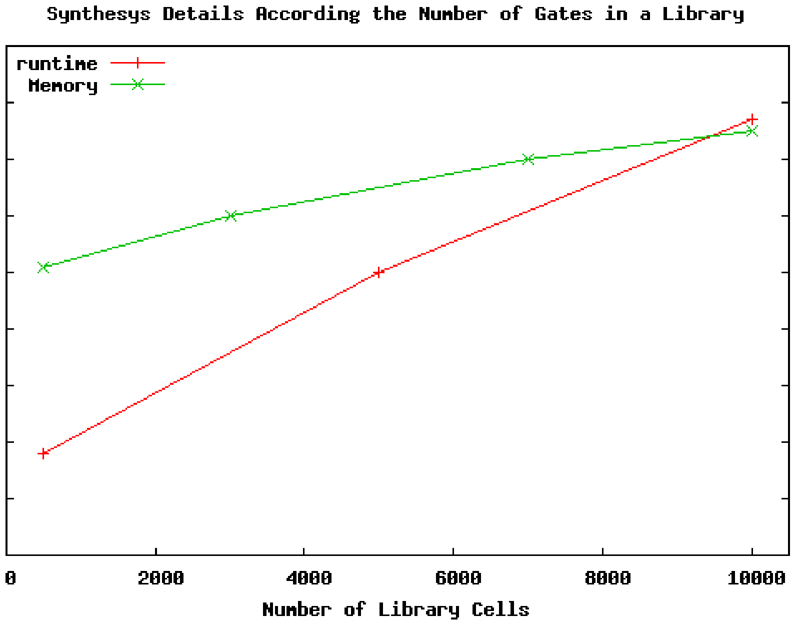


Figure 2.2: Runtime and occupied memory as a function of the number of gates in a library (CADENCE, 2007a).

On the other hand, a library-free methodology may not lead to good results due to the absence of timing and/or power information. It is clear that the synthesis cannot be efficient with this lack of information. Although the library-free mapping presents good characteristics concerning the reduced number of transistors, timing closure cannot be reached without the delay information of each cell.

For this reason, the development of a *super library* is proposed in this thesis, which consists on developing a library with a very large number of cells enriched by the timing information. Thus, the synthesis is able to generate a circuit with small number of transistors as well as targets timing closure.

The cells layout does not exist in the *superlib*. Only the logic function, cells structure (transistors and connections) and the estimated timing and occupied area are known. These informations are enough to allow timing closure logic synthesis.

The ability of synthesis algorithms on exploring a wide number of logic functions is not so clear in the literature. One important question about the efficacy of our *superlib* was whether commercial tools really take advantage of a wide number of logic functions at the logic synthesis.

Figure 2.3 shows that commercial tools are able to explore the synthesis when a wide number o logic functions are available. These results were obtained with Cadence RTL Compiler (CADENCE, 2007b). Analyzing these ten circuits, we note that circuits mapped with our *superlib* have around 52% more complex logic functions. Simple gates such as NANDs, NORs, inverters and buffers are not included in these results.

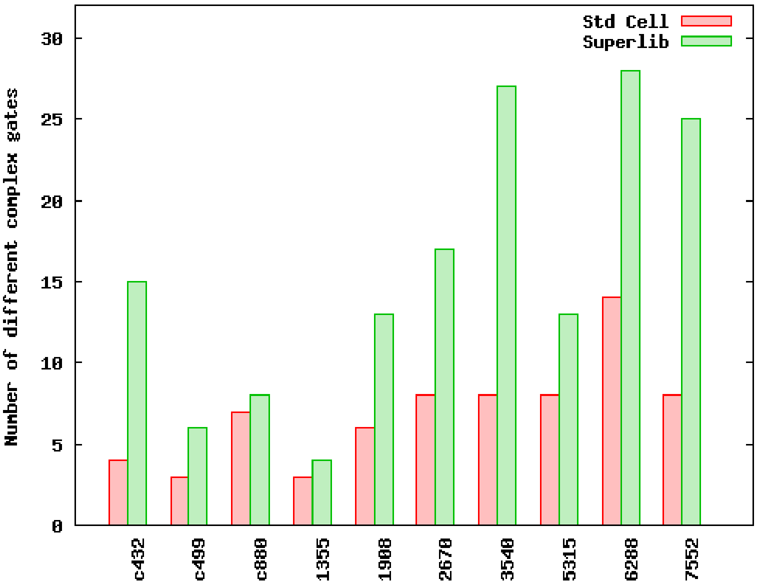


Figure 2.3: Number of different complex logic functions in circuits in a commercial $0.35\mu m$ technology. Results were obtained with Cadence RTL Compiler (CADENCE, 2007b).

2.2.1 The Development of a *superlib*

The *superlib* is generated in Synopsys Liberty library format (SYNOPSYS, 2007) and contains delay information about different versions of each logic function. This library is composed of 3,503 different logic functions, which is composed by every logic function with up to four stacked transistors, as shown in Table 2.1.

Each logic function was implemented in four different drive strengths by sizing its transistors from 1 to 4 times the minimum possible transistor width. Inverters and buffers were implemented with 8 and 32 different versions, respectively. The resulting library is composed by almost 15,000 cells.

The number of drive strengths of the *superlib* was limited due to the runtime and memory usage. Although the linear behavior of new synthesis tools, very large libraries may take long time to converge to a solution. In other words, the number of drive strengths was limited in order to deal with the tradeoff between quality of results and runtime during the synthesis.

Table 2.2 shows a comparison between the runtime of the logic synthesis with a standard cell library and the *superlib* for the ISCAS85 benchmarks. We consider a standard cell library with around 500 cells while the *superlib* is composed by 14,048 cells in this comparison. The drawback of our methodology is visible when runtime is considered and these results justify the limitations we impose to the *superlib* concerning the number of available cells.

This limitation is attenuated by sizing the transistors just after the synthesis. Further details are given in Section 2.3.

One important remark is that we do not take into account the power consumption at

Table 2.2: Synthesis runtime of the ISCAS85 benchmarks with a commercial $0.35\mu m$ standard cell library and the *superlib* with Cadence RTL Compiler (CADENCE, 2007b).

Benchmark	**Standard Cell**	**superlib**
c432	3s	2min21s
c499	9s	2min31s
c880	8s	1min49s
c1355	9s	2min28s
c1908	8s	1min34s
c2670	14s	2min12
c3540	22s	4min38s
c5315	32s	7min52s
c6288	47s	26min30s
c7552	48s	14min36s

this stage. This is done in order to reduce the library generation execution time. The power consumption is proportional to the transistors area. Thus, this information may be omitted in the library characterization.

Algorithm 1 presents the method for generating the *superlib*. Considering a library composed by a set of cells C associated with a set of input slopes S and a set of output capacitances O, the algorithm simulates the spice netlist of each cell for rise and fall time. The timing characterization of the cells is done by electric simulations with Synopsys HSPICE (SYNOPSYS, 2007).

In order to reduce the number of simulation runs in the characterization process, we simulate only input vectors that stimulate the biggest path between supply source and the cell output for each cell input. We describe a logic function as a graph G (line 4), where each variable of the logic function is represented by an input signal (vertex in the graph). Drain/Source nodes are represented by edges.

The algorithm searches the biggest path crossing the variable i between the VDD/GND and the output. After the biggest path is assumed to be known (line 8 for rise time, and line 18 for fall time), transistors in the path are configured to switch from the “OFF” state to the “ON” state (line 10 and 20) and transistors outside the biggest path are set to the “OFF” state. Thus, for each input signal of a cell c, only one input vector is necessary for measuring the rise time and another input vector for measuring the fall time.

Once the path is known and input signals are defined, a spice simulation is done for each input slope $s \in S$ and each output capacitance $o \in O$ (lines 12-15 and 22-25). At the end, the sets of delay times D_{rise} and D_{fall} contain the whole information about the cells timing.

Figure 2.4 shows the process of generating the propagation time of each cell. Only the pull-up is represented in the figure. Considering the transistor schematic in Figure 2.4(a), the graph is created. The graph is used to find the longest path between the V_{DD} and the cell output (Figure 2.4(b)). An example of the representation of the rise and fall time in liberty format is shown in Figure 2.4(c). The field `values` is a 2D table and contains the

Algorithm 1 : The proposed *superlib* generation process.

```
Require: Set of cells $C$, Set of input slopes $S$, Set of output Capacitances $O$
Ensure: Liberty file
for all $c \in C$ do
    $T \Leftarrow$ getTransistors( $c$ );
    $I \Leftarrow$ getInputs( $c$ );
    $G \Leftarrow$ createGraph( $T$ );
    for all $i \in I$ do
        {Finding the rise time from the input $i$ to the cell output}
        $D_{rise} \Leftarrow \emptyset$ {$D_{rise}$ are rise times to each input slope and output capacitance}
        $P \Leftarrow$ findTransistorsInPullUpPath( $T$, $i$ );
        $N \Leftarrow T \setminus P$;
        connectInputsToVDD( $N$ );
        setInputSwitchingFromOneToZero( $P$ );
        for all $s \in S, o \in O$ do
            $d \Leftarrow$ getDelay( $G$, $P$, $N$, $s$, $o$ ); {Run hspice}
            $D_{rise} \Leftarrow D_{rise} \bigcap \{d\}$;
        end for
        {Finding the fall time from the input $i$ to the cell output}
        $D_{fall} \Leftarrow \emptyset$ {$D_{fall}$ are fall times to each input slope and output capacitance}
        $P \Leftarrow$ findTransistorsInPullDownPath( $T$, $i$ );
        $N \Leftarrow T \setminus P$; {$N$ is the set of transistors outside the biggest path}
        connectInputsToGND( $N$ );
        setInputSwitchingFromZeroToOne( $P$ );
        for all $s \in S, o \in O$ do
            $d \Leftarrow$ getDelay( $G$, $P$, $N$, $s$, $o$ ); {Run hspice}
            $D_{fall} \Leftarrow D_{fall} \bigcap \{d\}$;
        end for
    end for
end for
```

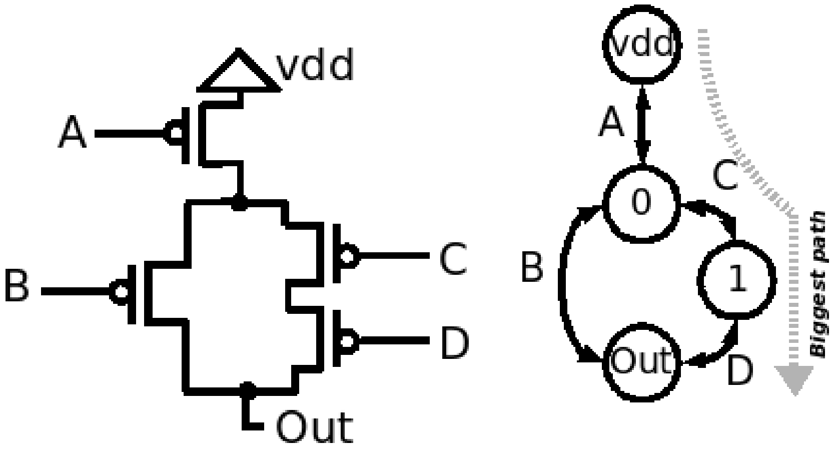
delay information as a function of input slopes (rows) and output capacitances (columns).

The whole process (more than 750,000 simulations) was automated by scripts. The generation of the *superlib* took around 6 days in a SunfireV890 with 8Gb of RAM.

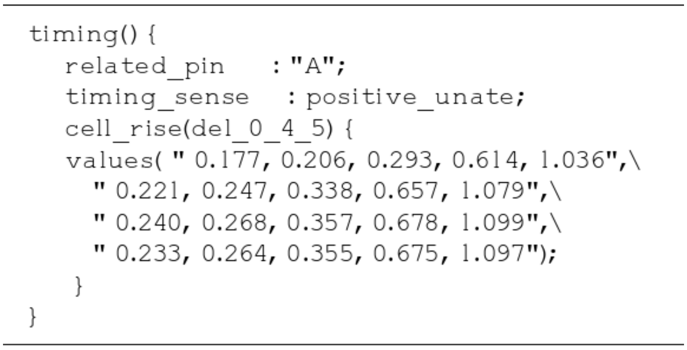
2.3 The Transistor Level Design Flow

The transistor level optimization has been considered in several works when timing closure is achieved (HASHIMOTO; ONODERA, 2001; VUJKOVIC, M. et al., 2004; ROY; BHATTACHARYA; BOPPANA, 2005). Vujkovic *et al.* report in (VUJKOVIC, M. et al., 2004) that the capacitance per unit length varies around 35 times in a $0.18\mu m$ technology. This shows how the optimization of each cell as a function of the output load capacitance is important, specially in the critical path.

In this section we present a complete design flow based on transistors optimization. The main goal of this methodology is to explore the capabilities of a transistor level design



(a) Schematic of a cell (b) The graph of the same cell



```
timing() {
   related_pin   : "A";
   timing_sense   : positive_unate;
   cell_rise(del_0_4_5) {
   values( " 0.177, 0.206, 0.293, 0.614, 1.036",\
     " 0.221, 0.247, 0.338, 0.657, 1.079",\
     " 0.240, 0.268, 0.357, 0.678, 1.099",\
     " 0.233, 0.264, 0.355, 0.675, 1.097");
    }
}
```

(c) Example of a cell representation in liberty format

Figure 2.4: Generation of the liberty file. This file is used later in the logic synthesis.

flow to deal with the challenges present in deep submicrom technologies.

Once the *superlib* was created, the layout of the circuit can be generated. Figure 2.5 shows in details the design flow proposed in this work. The proposed transistor level design methodology consists of a set of commercial and academic tools that deal with the timing closure challenges at the same time as it offers to designers a complete RTL-to-GDS design flow. Developed tools are labeled with letter D.

Main differences of the traditional standard cell flow are the following:

- The possibility to synthesize a circuit with a wide range of cells by using the *superlib* (Figure 2.5 Label A);
- A transistor level optimization tool called **T-Factor** able to size each transistor of the circuit in order to deal with timing closure (Figure 2.5 Label B);
- A new transistor level layout generation tool called **Punch++** able to generate any kind of Static CMOS gate (Figure 2.5 Label C).

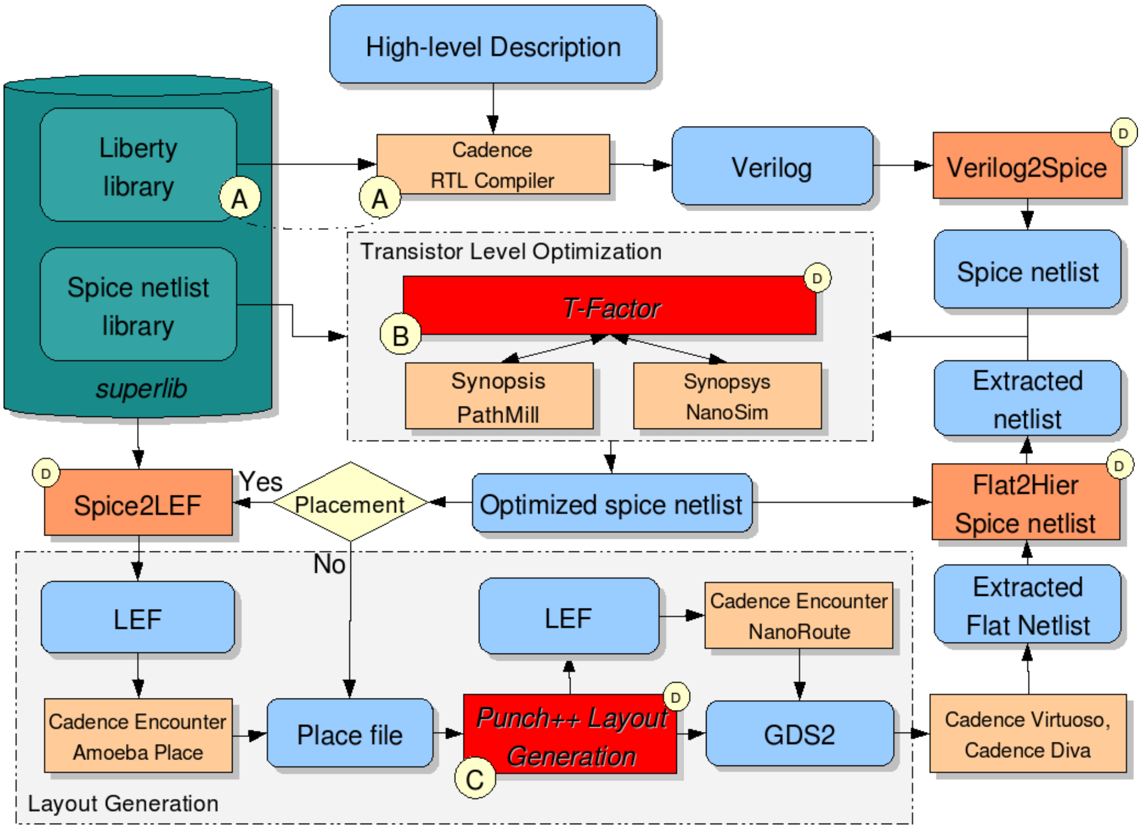

High-level Description
Liberty library
A
Cadence RTL Compiler
A
Verilog
Verilog2Spice
D
Spice netlist library
superlib
Transistor Level Optimization
T-Factor
D
B
Synopsis PathMill
Synopsys NanoSim
Spice netlist
Extracted netlist
D
Spice2LEF
Yes
Placement
No
Optimized spice netlist
Flat2Hier Spice netlist
D
LEF
LEF
Cadence Encounter NanoRoute
Extracted Flat Netlist
Cadence Encounter Amoeba Place
Place file
Punch++ Layout Generation
D
C
GDS2
Cadence Virtuoso, Cadence Diva
Layout Generation

Figure 2.5: The proposed transistor level design flow. A) Logic synthesis with the *superlib*. B) Transistor level optimization tool. C) Layout generation tool, D) Developed tools to cope with the proposed transistor level design flow.

The first step in the design flow is to generate a spice-like netlist of the circuit. In the proposed design flow, circuits described in high level languages such as VHDL or Verilog are converted to netlist descriptions by using Cadence RTL Compiler (CADENCE, 2007b). After synthesis, the verilog netlist is converted to a spice netlist description.

The transistor level optimization starts with the spice netlist description. The tool **T-Factor** finds transistors in the critical paths and these transistors are sized in order to meet timing specifications while transistors in less important paths are maintained with minimum sizes.

The cells placement is based on area estimates with the Cadence Amoeba placer (CADENCE, 2007b). After the placement, the layout of the entire circuit is generated and routed by the Cadence Nanoroute. The last step of the design flow is the DRC, LVS and parasitics extraction, which makes possible to evaluate the correctness of the circuit.

Details about the most important steps in the transistor level design flow are discussed in the following.

2.3.1 The Transistor Level Optimization

As previously discussed, the number of cells was limited to 3,503 targeting runtime reduction in the logic synthesis. The number of drive strengths was also limited by the same reason.

The limitation concerning the number of drive strengths is compensated by optimizing the circuit with a transistor sizing tool. *T-Factor* works in association with the Nanosim e Pathmill from Synopsys (SYNOPSYS, 2007) in order to optimize the circuit at transistor level. Dynamic power simulation is performed by Nanosim and static timing analysis is done by Pathmill.

Algorithm 2 : The transistor level optimization process used by *T-Factor*.

```
Require: The netlist spice $N$, Number of iterations $k$, Maximum transistor width $m$,
    Transistor size step $p$, Input vector size $s$
Ensure: Set of optimized netlists $N_{new}$ with timing and power information
 1: $N_{new} \Leftarrow N$;
 2: $i \Leftarrow 0$;
 3: $V \Leftarrow$ generateInputVectors( $s$ );
 4: while $i < k$ do
 5:    $pm \Leftarrow$ getCriticalPath( $N$ );            {run Pathmill}
 6:    $ns \Leftarrow$ getPowerInformation( $N$, $V$ ); {run Nanosim}
 7:    $c \Leftarrow$ findCellWithWorstLoadFactor( $pm$ );
 8:    $T \Leftarrow$ findTransistorToSize( $c$ );
 9:    for all $t \in T$ do
10:       $w \Leftarrow t.w + p$;
11:       if $w > m$ then
12:          stop; {Stop and go to line 21}
13:       else
14:          $t.w \Leftarrow w$;
15:       end if
16:    end for
17:    $c_{new} \Leftarrow$ updateTransistors( $c$, $T$ );
18:    $N_{new} \Leftarrow N_{new} \bigcup \{c_{new}\} \setminus \{c\}$;
19:    $i++$;
20: end while
21: plotDelayPowerTradeoff( );
```

The transistor level optimization is shown in Algorithm 2. The optimization process starts with the generation of a set of input vectors V (line 3). These vectors are used to analyze the dynamic power consumption. The number of input vectors are defined by s because the power analysis runtime is directly proportional to s. A smaller s may be defined to big circuits in order to reduce the runtime when analyzing the power consumption.

Function `getCriticalPath(` N `)` (line 5) consists on running Pathmill and extracting the critical path. Pathmill reports the worst path with the delay in each stage and the capacitances of each node. The power consumption is detected by the function `getPowerInformation(` N`,` V `)` (line 6) as a function of the input vector V.

The load factor is used to find a cell among all the candidates to sizing. Load factor is used in many works as criterion to size cells in circuits (CHEN; HSIEH; PEDRAM, 2000; SANTOS et al., 2003). The load factor is defined by

$$F_{load} = \frac{C_{load}}{C_{in}} \tag{2.1}$$

where C_{load} is the load capacitance on the cell outputs and C_{in} is the input gate capacitance. The candidate cell is given by the function `findCellWithWorstLoadFactor(` pm `)` at line 7 where the cell with the worst load factor is chosen to be sized.

Function `findTransistorToSize(` c `)` finds the transistors path between the supply line to the cell output by a graph structure similar to Figure 2.4(b). Lines 9 to 16 increase the width of every transistor $t \in T$ by s.

After the transistors sizing, the netlist is updated and the whole process is repeated until the number of iterations meet the maximum number of iterations k defined by the designer or the gates have the maximum width m.

Figure 2.6 shows the delay and power consumption as result of the first optimization step for the C1908 circuit. Figures (a) presents the delay and power consumption for each iteration. One important aspect is the linear grown of the power consumption as a function of the sizing algorithm.

Huge amount of power is inevitable when searching fastest circuits. In most designs, the smallest delay is not the best option due to the increased power consumption and circuit surface. Furthermore, aiming at low power devices, the consequence is a bigger delay of the circuit.

Sometimes the criterion for designing a circuit is neither an extremely low power nor a high performance, but a good tradeoff between delay and power consumption. Thus, Figure 2.6 (b) shows the power-delay tradeoff for the circuits where the designer is able to evaluate the optimization process. Each point in the power-delay tradeoff plot corresponds to a possible circuit design. In other words, according to design specifications, the designer is able to choose the best power-delay tradeoff in order to generate the layout.

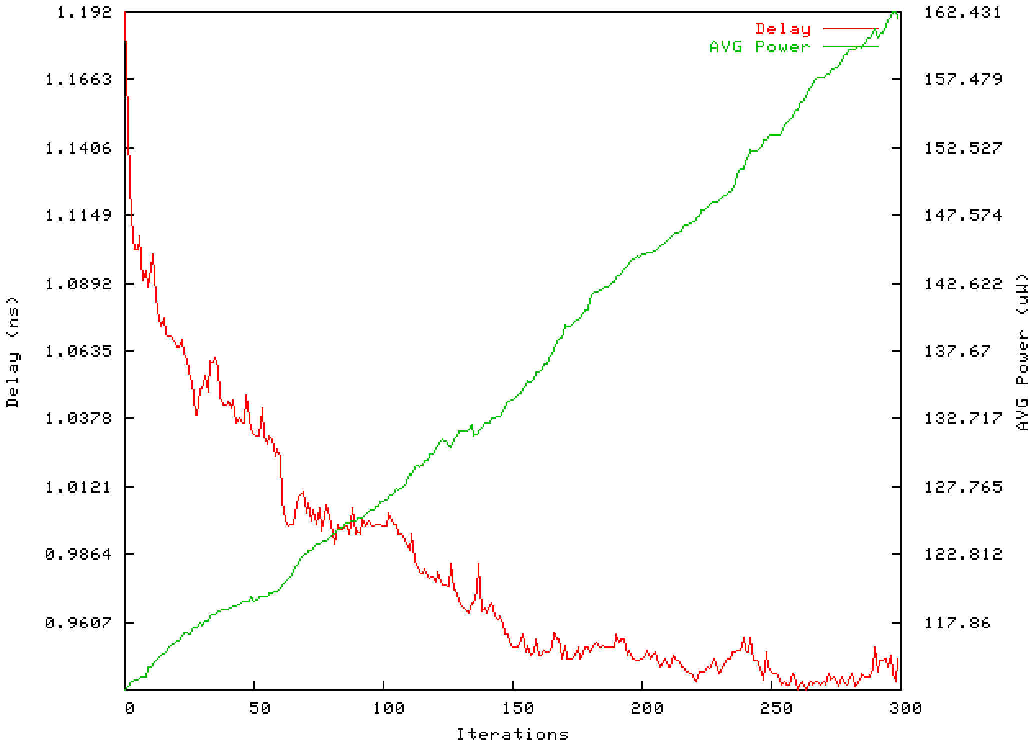
The efficiency of the transistor optimization process cannot be measured while analyzing only the worst delay of a circuit. The efficient is proven by analyzing all paths of the circuit.

The operation frequency of a combinational circuit is given by the worst path delay. A big number of paths with very small delay may signify oversized transistors due to an inefficient sizing algorithm. As a consequence, an inefficient sizing strategy may lead to an excessive and undesirable power consumption.

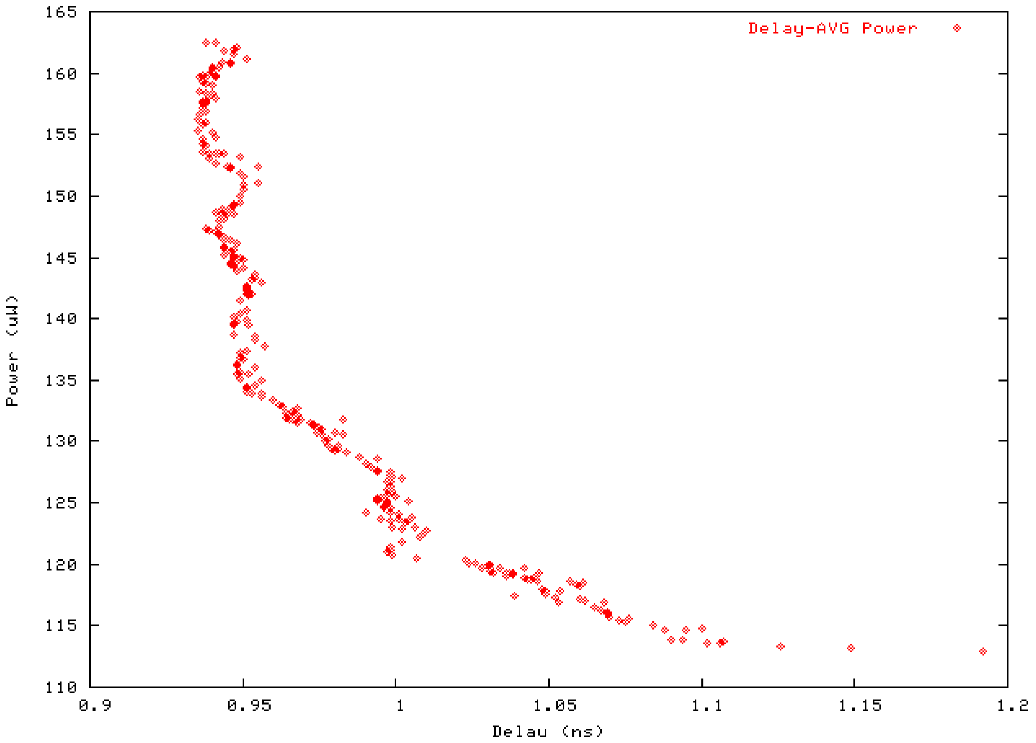
Figure 2.7 illustrates delay paths before the transistor sizing and after the transistor sizing. The X axis shows all the paths in the circuit c1355 while the Y axis represents the delay of each path. The almost horizontal line given by the delay paths after sizing shows the efficiency of the sizing algorithm. This uniformity in delay paths means that gates in the fastest paths present adequate size and are not over consuming.

2.3.2 Transistor Level Optimization for Leakage Reduction

The static power has become an important amount of the total power consumption. The power dissipation due to leakage current is approaching the dynamic power, and off-state subthreshold leakage is projected to exceed the dynamic power consumption as technology drops below the 65nm feature size (KIM, N. S. et al., 2003).



(a) C1908 - Delay and Power



(b) C1908 - Power-delay tradeoff

Figure 2.6: Delay and power consumption results for some circuits from the ISCAS'85 benchmarks.

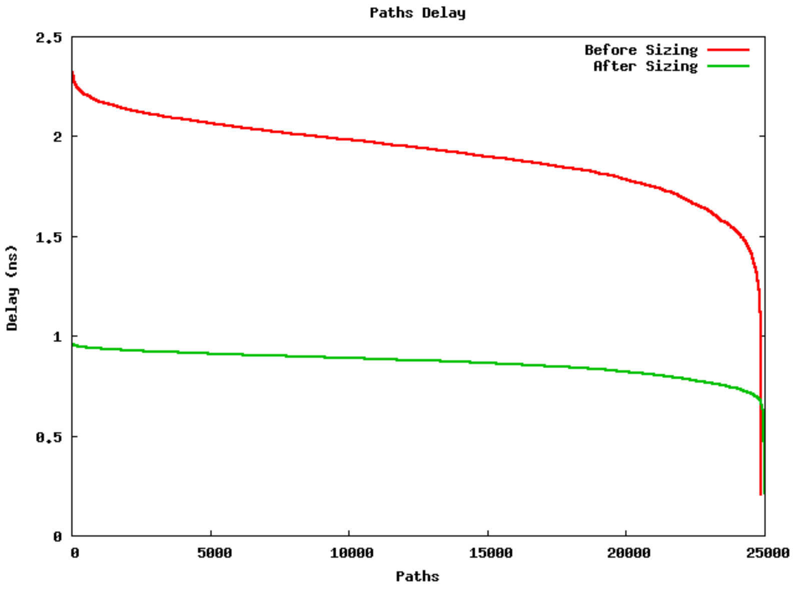


Figure 2.7: Delay paths of the circuit c1355.

For these reasons, the static power consumption has to be incorporated in the design of a systems.

Many techniques have been presented in the last years aiming at reducing the static power consumption. The leakage problem has been addressed at design stage by various techniques such as transistor stacking, sleep transistor insertion, V_{DD} assignment and transistor length biasing.

Transistor stacking is the technique to duplicate transistors aiming at reducing the subthreshold leakage when transistors are in standby mode (NARENDRA, S. et al., 2002). Figure 2.8 shows an example of stacked transistors structure.

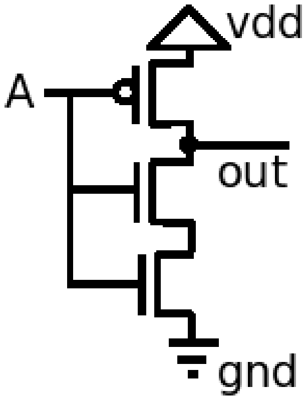


Figure 2.8: An NMOS two-stack inverter reduces the subthreshold leakage when input stays at logic “0”.

Sometimes the stacked transistor widths are divided by two in order to ensure the same input load. Thus, the previous gate delay and the switching power remains unchanged. However, the gate delay is increased and timing closure may not be achieved.

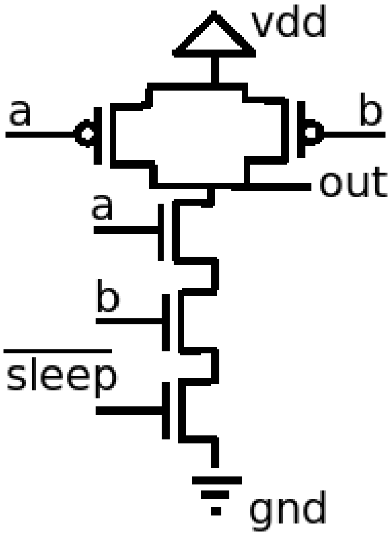


Figure 2.9: An example of fine-grained sleep transistors.

The main idea in the insertion of *sleep transistors* is to cut the subthreshold current when CMOS gates are in standby mode (LONG; HE, 2003; BABIGHIAN, P. et al., 2004). The methodology seems to be very interesting because a whole block can be turned off, but the main drawback of this technique is related to delay penalties.

The activation time needed by the sleep transistors when they change from the state “OFF” to the state “ON”, may be longer than the response time of many cells in the design (specially those placed close to the primary inputs (PI)). This includes an extra delay to the circuit.

Figure 2.9 illustrates an example of fine-grained sleep transistor. Usually sleep transistors do not only control the shutoff of a gate but also a set of gates in order to reduce the area overhead.

V_{DD} *assignment* consists of assigning different supply power voltages to different cells in the circuit (LIN; HE, 2005). Thus, cells with lower V_{DD} voltage present a smaller power consumption.

The V_{DD} assignment technique demands careful analysis in relation to the cells placement and supply lines routing. Cells must be grouped in order to reduce the supply routing, increasing the wire length and routing congestion.

Considering that the power planning is already a hard task in submicron technologies due to local hot spots, insufficient power supply and signal integrity problems, V_{DD} assignment insert more challenges in the traditional design flow.

Gate length biasing consists on adjusting the length of the transistors to reduce the power leakage (GUPTA et al., 2004; KAHNG; MUDDU; SHARMA, 2005; BHARDWAJ; CAO; VRUDHULA, 2006). The leakage is inversely proportional with the gate length. However, the delay of a transistor increases with the gate length. Thus, fastest paths can be sized while the transistors in the critical path may maintain its length in order to cope with timing closure.

From the point of view of the transistor level layout generation, all the techniques above can be applied. However, the gate length is a technique witch the whole process can be performed at switch level. This technique is very hard to be implemented in the traditional standard cell approach because new versions of each cell must be generated and characterized.

The second possibility to gate length biasing in standard cells is the post layout optimization. The length of the transistor in the fastest paths may be manually modified. Identify and modify the layout is a very complex task and add many hours to design.

In a transistor level layout generation, the gate length biasing is simple to be included in the design flow because the layout is the last step of the whole process. For this reason, we include the gate length biasing in our design flow. Transistors are sized for power leakage reduction after the circuit is sized for timing closure.

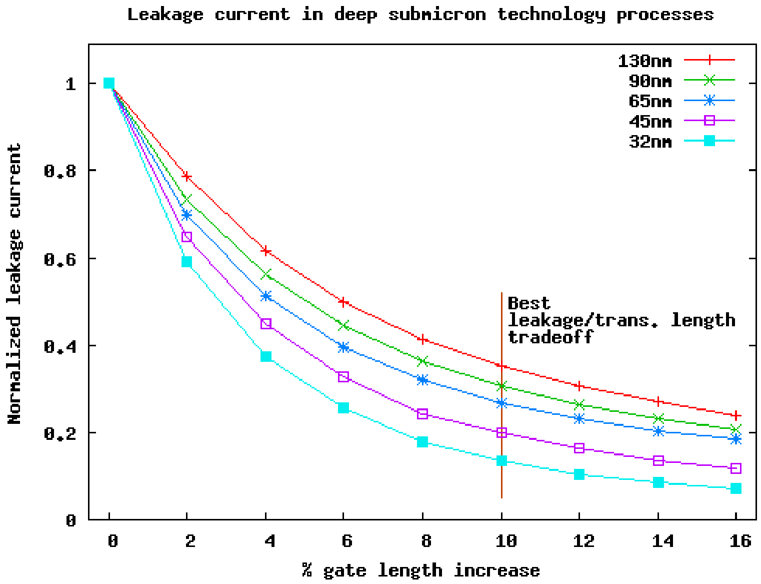


Figure 2.10: Leakage current in deep submicron technology process.

Figure 2.10 shows the normalized leakage current in deep submicron technologies as a function of the transistor length biasing. These data were obtained by spice simulations with the predictive technology models presented in (ZHAO; CAO, 2007).

Results show that the transistor length biasing is more efficient with the technology shrinking. However, an upper bound to the transistor length sizing may be defined due to the exponential reduction in the leakage current.

The leakage reduction gain is very small when the transistor length is bigger than 10% for all technology processes. For this reason, we define a 10% upper bound to transistor length sizing for our experiments.

Results about the gate length biasing are shown in Table 2.3. Results show that the gate length biasing is very efficient on the reduction of the power leakage. An average

Table 2.3: The gate length biasing technique for some ISCAS'85 benchmarks with a $65nm$ technology process.

Benchmark	**Sized cells**		**Power leakage**		
	#Cell	**(%)**	**Before Biasing**	**After Biasing**	**After Biasing (%)**
C432	139/209	66%	3.5 μW	2.1 μW	60%
C499	208/296	70%	12.5 μW	8.1 μW	64%
C880	290/359	80%	10.6 μW	5.9 μW	55%
C1355	247/446	55%	10.3 μW	6.6 μW	64%
C1908	245/372	65%	14.4 μW	11.0 μW	76%
C3540	526/704	74%	20.1 μW	12.3 μW	61%
Average leakage after gate length					63%

of 70% of the cells were sized and the resulting circuit spend 60% of the initial power leakage in average. It is important to remark that the delay and power dynamic is not increased.

The distribution of the delay paths before and after the transistors length biasing is shown in Figure 2.11. Curves show the bigger number of paths close to the target delay after the gate length biasing.

One important remark is related to the process variability and the big number of paths close to the target delay. The variability of the process technology may increase the delay of some paths causing a timing violation. For this reason, worst case corners must be used at simulation time to reduce the impact of the variability in the circuit delay.

2.3.3 The Transistor Level Layout Generation

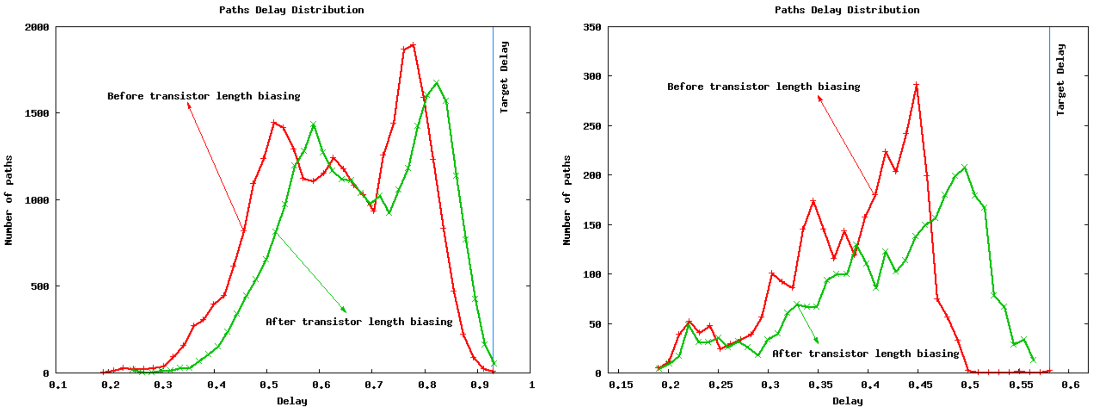
As previously discussed in this chapter, the layout of the whole circuit is fully generated on demand, without any previously defined layout. Only the netlist with the sized transistors and the technology rules are needed.

The transistor level layout generation involves several tasks as illustrated in Figure 2.5. First, the layout cannot be generated without the placement of cells. Thus, estimated information (delay, power and area) of each cell is used to place the cells in the circuit. In order to estimate this information, the spice netlist of each cell is used. The transistors structure allows the power and timing estimation. The area can be estimated with the cells structure and some technology rules. Cadence Amoeba (CADENCE, 2007b) is used to place the cells.

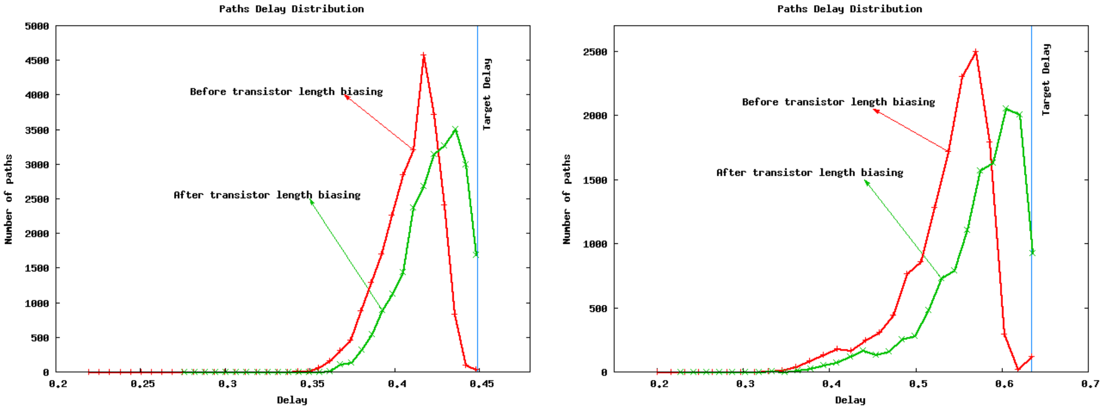
After the cells placement, the layout generation can be performed. Algorithm 3 shows the process to generate the circuit layout.

The algorithm starts with the transistors folding. The folding technique consists on breaking big transistors in equivalent parallel transistors (line 1). Usually, big transistors are not easy to place and may increase the area occupied by the cells. For these reasons, the folding technique is applied to the netlist.

The placement is applied to the netlist in Function `applyPlacement(` N, P `)` (line 2) where cells are organized in the set of rows R and placed side by side.



(a) C499 (b) C880



(c) C1355 (d) C1908

Figure 2.11: Delay paths distribution before and after transistors length biasing to the benchmarks ISCAS'85.

Algorithm 3 : The Punch++ layout generation.

Require: Set of cells N in spice format, Technology Rules T, Placement P

Ensure: Layout in GDSII format L

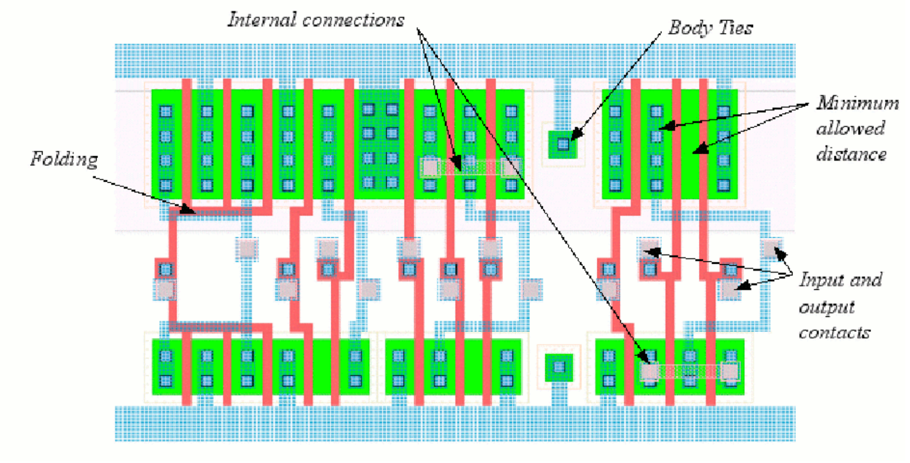
```
foldTransistors( $N$ );
$R \Leftarrow$ applyPlacement( $N$, $P$ );          {$R$ are the rows}
for all $r \in R$ do
    $L_r \Leftarrow$ placeTransistors( $N$, $r$ );
    $L_r \Leftarrow$ routeTransistors( $L_r$ );
    $L_r \Leftarrow$ compactLayout( $L_r$, $T$ );
end for
$L \Leftarrow$ routeCircuit( $L$ );          {Call Cadence nanoroute}
writeLayout( $L$ );
```

For each row $r \in R$, transistors are placed and routed in Functions `placeTransistors(` N, r `)` and `routeTransistors(` L_r `)`, respectively. Transistors are placed and routed without taking into account the technology rules. The

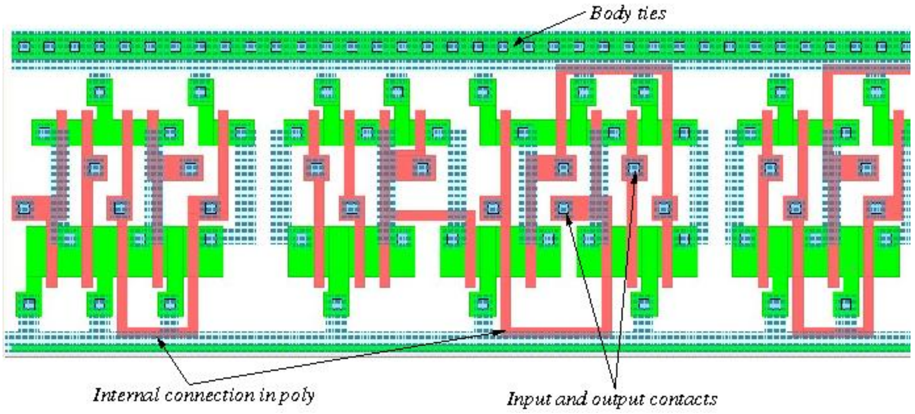
Function `compactLayout(` L_r`,` T `)` is responsible for compacting the layout and applying the technology rules to the layout. More details about the algorithms used in these three steps are given in Chapter 3.

Once the layout of each row is performed, the circuit is routed by the Cadence nanoroute (CADENCE, 2007b) (Function `routeCircuit(` L `)` in line 8) and the circuit is stored in GDSII format (Function `writeLayout(` L `)` in line 9). The GDSII is the standard layout description format used by the industry.

When the layout is fully generated, the designer is able to extract the parasitics using Cadence DIVA (CADENCE, 2007b). Layout validation is also done with LVS and DRC tools. The extracted netlist is used to evaluate electric characteristics such as power and timing. In addition, if the circuit does not meet the specifications, the designer can repeat the optimization process (Section 2.3.1) in order to improve the electrical characteristics of the circuit.


Internal connections
Body Ties
Minimum allowed distance
Folding
Input and output contacts

(a) Layout from the work proposed in (LAZZARI, 2003)


Body ties
Internal connection in poly
Input and output contacts

(b) The new layout style

Figure 2.12: Layout style of two transistor level automatic layout generators. The *Punch* and *Punch++* tools.

Figure 2.12 shows a comparison between the layout style of the tool presented in (LAZZARI, 2003) and the layout style of the tool developed in this work. Basically, differences can be seen in the placement of body ties and interconnections.

The layout style from (LAZZARI, 2003) presents internal connections implemented with the first and second metal layers. The new tool only connects transistor with the first metal layer, but small connection can be performed with polysilicon lines. Details about the algorithms implemented in the new tool are given in Chapter 3.

2.4 Transistor-Level vs Traditional Design Flow Summary

Table 2.4 shows a summary about some characteristics of the traditional standard cell and the proposed transistor level design flow. The main steps in the whole process are discussed. The design flow includes logic and physical synthesis.

2.4.1 A Comparison with the Traditional Standard Cell Design Flow

Table 2.5 presents results of the proposed method in comparison with the standard cell approach for a commercial 0.35 μm technology. Results are very interesting because they show the efficiency of our transistor level design flow.

The design process was done based on high effort to meet the minimum possible delay. We noted that this high effort resulted in the insertion of many buffers in the critical path. This explains the low gain concerning the timing (around 11%) of our methodology in comparison with the standard cell approach. The power consumption gain in these circuits is between 15% and 42% due to the number of complex gates in the circuit and the optimized transistor width.

These results were obtained by attempt for improving the layout quality concerning polysilicon and metal connections and by the possibility to optimize the circuit in relation to the wide number of logic functions and drive strengths present in a library.

Figure 2.13 shows two layouts of circuits generated by the proposed transistor level design flow.

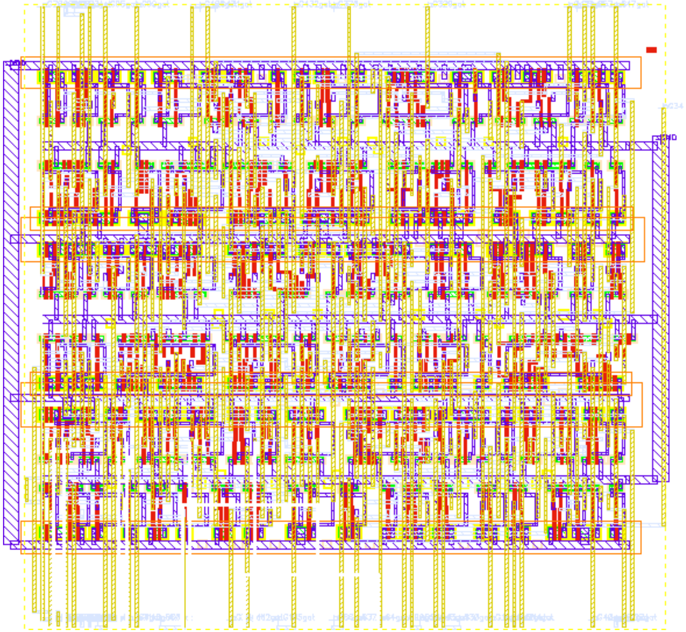
2.5 Conclusion

This chapter explores the possibilities to develop a transistor level design flow as alternative to the traditional standard cell. In a traditional design flow, the layout of cells is generated, characterized and grouped in cell libraries. These cell libraries contain information about occupied area, timing and power consumption.

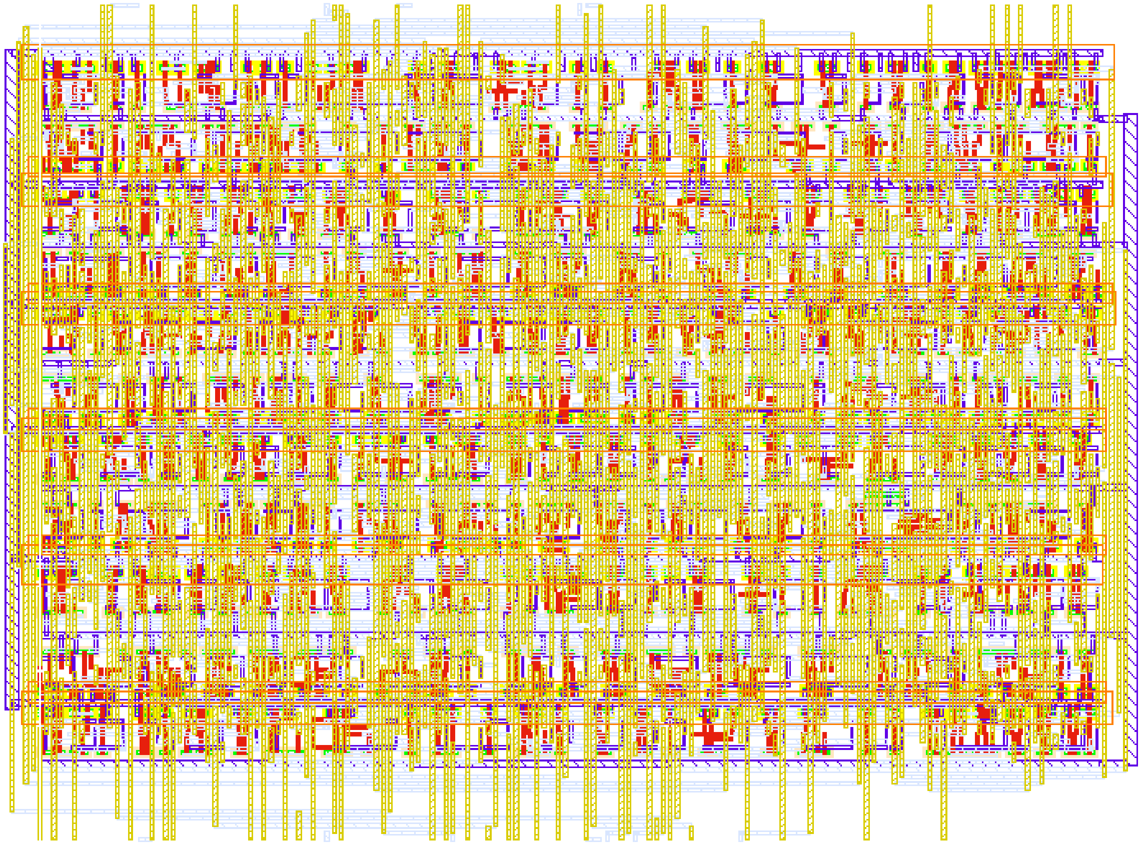
The transistor level design flow is a different paradigm. Libraries do not contain the layout of the cells, but only timing and power information. This information is obtained by spice simulations. The advantage of this methodology is that a library may contain thousands of logic functions and not hundreds as in standard cells.

Thus, the logic synthesis tool is able to optimize the circuit with a very large range of logic functions available in the library. After the logic synthesis, an additional step optimize independently transistors according to the loading applied to each gate.

This transistor optimization allows to optimize critical paths while gates in non-critical paths have minimum sizes to reduce the power consumption. Besides, techniques such as the gate length biasing are easily employed in the design process to reduce also the power leakage.



(a) C432



(b) C1908

Figure 2.13: Some layouts of circuits generated using the proposed transistor level design flow.

Table 2.4: Summary of differences of between a traditional standard cell and the proposed transistor level design flow.

Library Generation	**Traditional design Flow:** The cell library contains the layout of cells with area, timing and power information. Cell library usually contains a few versions of each logic function. **Proposed design Flow:** Layout of the logic functions does not exist. Information concerning timing and power is done based on the spice simulations. A cell library contains thousands of logic functions.
Logic Synthesis	**Traditional design Flow:** Synthesis based on the cell library. **Proposed design Flow:** Synthesis based on the cell library.
Transistor Level Optimization	**Traditional design Flow:** N/A **Proposed design Flow:** Consists on optimizing transistors for timing and power. Static timing analysis is done to find and optimize critical paths with a bigger range of possibilities. Transistors outside the critical paths are maintained with minimum sizes to reduce power consumption. Power leakage is attenuated by gate length biasing.
Placement	**Traditional design Flow:** Based on the cells information. **Proposed design Flow:** Based on the estimated area of cells and timing from electrical simulation.
Layout Generation	**Traditional design Flow:** N/A. Layout is done using a cell library. **Proposed design Flow:** Layout is generated by algorithms applied to an optimized transistor level netlist.
Routing	**Traditional design Flow:** Traditional Routing. **Proposed design Flow:** Traditional Routing.

The layout is only generated after the transistor level optimization with computationally efficient algorithms.

The proposed design flow allows designers to mitigate the timing closure problem. Results show that this methodology is very promising. Comparisons between the transistor level methodology and the standard cell approach show interesting results where our methodology presented around 11% of delay improvement and more than 30% power

Table 2.5: Comparison between layouts generated using the standard cell approach and the proposed transistor level design flow.

Circuit	**Timing (ns)**			**Total Power Consumption (uW)**		
	Std Cells	**Proposed**	**Gain**	**Std Cells**	**Proposed**	**Gain**
C432	3.97	3.68	7.3%	4416	3726	15.6%
C499	2.36	1.89	19.9%	11881	7122	40.0%
C880	1.88	1.85	1.5%	5592	3984	28.7%
C1355	2.50	2.45	2%	12071	6965	42.2%
C1908	2.39	2.06	13.8%	9493	6007	36.7%
C3540	5.15	4.05	21.4%	21141	15235	27.9%
C6288	9.46	7.98	15.6%	211593	145660	31.1%
Average Gain			11.6%			31.7%

savings.

3 ALGORITHMS FOR TRANSISTOR LEVEL AUTOMATIC LAYOUT GENERATION

3.1 Introduction

The automation of the layout design reduces the design time due to a rapid synthesis and enables the designer to deal with a great range of challenges present in the new process technologies. New technologies challenges require additional functionality as performance-driven placement, area-efficient placement of substrate and well ties, performance-driven detailed routing and layout compaction with preference to critical nets (GURUSWAMY, M. et al., 1997).

Old physical synthesis algorithms usually present an exponential complexity and cannot lead with these challenges. Algorithms proposed in the last years are focused in attempting linear complexity or hierarchical methodologies. Hierarchical models can be a solution because they reduce the runtime of exponential algorithms to acceptable levels.

A transistor level design flow is presented in Chapter 2. Algorithms work with a huge number of variables and careful attention must be taken concerning the complexity and memory usage efficiency.

Algorithms used in the transistor level design flow are presented in this chapter. These algorithms include transistor placement & routing, and compaction. A brief discussion on the algorithm classes and state-of-the-art physical synthesis algorithms are also introduced.

3.2 An Overview of the Algorithm Classes

The synthesis of the layout has been widely explored in academic and commercial researches. These algorithms present different characteristics and the resulting efficiency is measured by the runtime and memory usage, but also by the quality of the generated layout.

One fundamental point when developing a new algorithm is the tradeoff between the algorithm complexity and the quality of results. According to the problem complexity, algorithms with low complexity may lead to an acceptable quality of results while complex algorithms present unacceptable runtime efficiency.

On the other hand, complex algorithms may present excellent quality of results for a problem with a small number of variables, while runtime efficient algorithms may lead to simplistic results.

The physical synthesis algorithms can be broadly divided into two classes:

- *Deterministic* : A deterministic algorithm is based on a model in which no randomness is involved. Deterministic models use mathematical representations of the underlying regularities that are produced by the entities being modeled and generate theoretically perfect data.
- *Stochastic* : Stochastic models use computational elements that represent the entities and the processes by which they interact and create a procedural algorithm to generate realistic data. Stochastic algorithms present certain randomness. This randomness is usually based on fluctuations observed in historical data or nature's phenomena.
- *Mixed Deterministic-stochastic* : Some algorithms include a mixed solution to solve physical synthesis problems. They usually converge to a solution by a set of deterministic iterations. For each iteration, variables are fed by stochastic models.

3.2.1 Deterministic Algorithms

The class of deterministic algorithms include *constructive* and *Analytical* methods.

Constructive methods model the convergence to a solution by placing one element at a time. Many algorithms have been presented to solve physical design problems. The sequence of elements inserted in the model influences the convergence to a solution.

A typical example of the constructive method may be the *force-directed.* A force-directed based algorithm models a system as a graph where the position of the vertices indicates the solution.

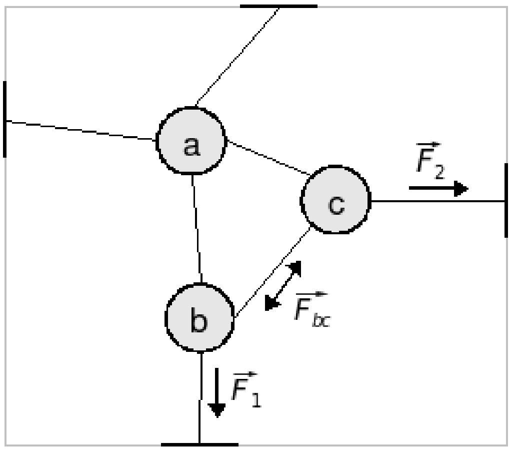


Figure 3.1: A force-directed example.

The force-directed model is represented as a mechanical system of objects connected by springs. The spring behavior is defined by the Hooke's law, which states that the force pushing back the spring is linearly proportional to the distance from its equilibrium length. The Hooke's law is given by

$$F = -kx \tag{3.1}$$

where F is the restoring force exerted by the spring, x is the distance the spring is elongated by, and k is the spring constant or force constant of the spring.

Figure 3.1 exemplifies the force-directed model where only the three forces $\vec{F_1}$, $\vec{F_2}$ and $\vec{F_{bc}}$ are shown. Other forces are omitted. The position of the nodes b and c is a consequence of these forces. Forces $\vec{F_1}$ and $\vec{F_2}$ pull the nodes to the boundaries while force $\vec{F_{ab}}$ controls the attraction between them.

Force-directed models are usually used in algorithms to solve placement problems (WILSON; SMITH, 1974; MO; TABBARA; BRAYTON, 2000; HUR et al., 2003; CHAN; CONG; SZE, 2005).

Another constructive algorithm is the placement of elements based on the Euler's path (REKHI; TROTTER; LINDER, 1995; RIEPE; SAKALLAH, 1999, 2003). The Euler path algorithm also models the placement as a graph. The idea of the Euler path is to walk on the graph edges where each graph edge is visited once (WEISSTEIN, 2007a).

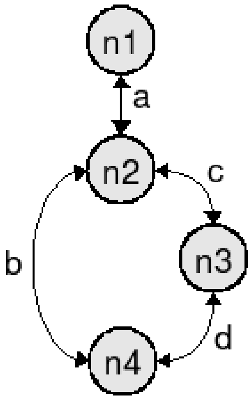


Figure 3.2: An Euler path example.

The Euler path graph is shown in Figure 3.2 where two possible Euler paths could be the sequence of edges `a,b,d,c` or `a,c,d,b`.

Analytical methods solve a given problem at once as a system of equations for all components. Analytical formulations are solved by mathematical programming such as *linear programming* (LP) and *quadratic programing* (QP). Linear programming in which variables assume on integer values is known as *integer linear programming* (ILP) or only *integer programming* (IP).

Linear programing is the optimization of an outcome based on some set of constraints using a linear mathematical model (NOYES; WEISSTEIN, 2007). Thus, a LP can be used to solve analytical problems if the problem may be formulated as

$$\begin{array}{rc} \textit{minimize} & c^T x \\ \textit{subject to} & Ax \leq b \\ \textit{where} & x \geq 0 \\ & L \leq x \leq U \end{array}$$

where $x \in \Re^n$ is a vector of variables that are continuous real numbers. $c^T x$ is the objective function and is represented as $f(x_1, x_1, \ldots, x_n) = a_1 x_1 + a_2 x_2 + \ldots + a_n x_n + b$, and $Ax \leq b$ represents the set of constraints. L and U are vectors of lower and upper bounds on the variables.

The most known linear programming solver is the `lp_solve` (BERKELAAR, 2006). lp_solve is able to minimize an objective function taking into account linear equalities and inequalities.

Linear programming has been used in placement & routing algorithms (GUPTA; HAYES, 2000; BEHJAT; CHIANG, 2005; BEHJAT; VANNELLI; ROSEHART, 2006; REDA; CHOWDHARY, 2006). The main characteristic of these works is related to the fast convergence to a solution. Besides, solutions tend to be optimal according to the linear formulations.

Although the excellent characteristics of the linear programming, most part of the problem cannot be represented by a linear objective function.

The quadratic programming is a problem with quadratic objective and linear constraints.

$$\begin{array}{rc} \textit{minimize} & q(x) = g^T x + \frac{1}{2} x^T H x \\ \textit{subject to} & Ax \geq b \\ \textit{where} & L \leq x \leq U \end{array}$$

where $x \in \Re^n$ is a vector of variables that are continuous real numbers. $Ax \geq b$ represents the set of constraints. L and U are vectors of lower and upper bounds on the variables.

Some placement algorithms such as the proposed in (ASKAR; CIESIELSKI, 1999; CIESIELSKI; ASKAR; LEVITIN, 2002; KLEINHANS, J. et al., 1991; VISWANATHAN; CHU, 2004) are based on quadratic programming. Thus, the placement problem is described in a mathematical language. Once the formulation converges to a solution, the result is the position of the transistors in the layout.

3.2.2 Stochastic Algorithms

Some design problems have a large range of possible solutions. These problems are computational hard or even impossible to be solved. Some stochastic methods may be used to reduce the search space.

Main examples of stochastic methods are *simulated annealing* and *genetic algorithms*. Simulated annealing is analogous to physical annealing process. It basically involves perturbing independent variables by random values while the temperature controls the standard deviation used by the random number generator. Many placement algorithms based on the simulated annealing have been proposed (MALLELA; GROVER, 1988; SECHEN, 1988; HENTSCHKE, 2002; TAYU, 2003; HENTSCHKE et al., 2006).

Genetic algorithms use basic principles of biology and emulates the natural process of evolution to find solutions to a problem.

In the genetic algorithm, each solution is represented by a *chromosome*. A chromosome is usually composed by a binary vector where variables formed by one or more bits are described. A population of chromosomes (possible solutions) is then created and genetic operators as mutation and crossover are applied in order to evolve the solutions to better results.

Some applications of the genetic algorithms on the physical design are presented in (BAHUMAN; BISHOP; RASHEED, 2002; LAZZARI; ANGHEL; REIS, 2005a). A new transistor placement technique is presented in Section 3.6 that consists on a genetic algorithm integrated with analytical programming.

3.3 Goals of Placement Algorithms

Hentschke (HENTSCHKE, 2002, 2007) presents an interesting discussion concerning placement algorithms. In this work, it is presented some characteristics about classic placement algorithms and the importance of these algorithms in modern physical designs. Furthermore, a comparison study on the most important placement algorithms is reported in order to present the consequences of these methods in the wire length and occupied area. Some of these placement algorithms are briefly reported in the following, based on the Hentschke's considerations.

Routability is defined by the ability of routing algorithms to route all wires under electrical and topological restrictions. The responsibility by a non-routable circuit is not only due to the router but also to the placer. Otherwise, a router will never be able to route a circuit whether cells are not adequately placed. Thus, the placer must be able to estimate the wire length of a circuit. The main estimation techniques are:

- *Semiperimeter:* Find the smaller *bounding box*, including all cells connected by a net and calculates the Semiperimeter (width + height).
- *Complete Graph:* Calculates the distance between each two points of a net.
- *Source to Target:* Every net has at least one source and one target. This method estimates the wire length from an output pin (source) to the input pins (targets) from a net. This estimation method is important when *timing* is considered.
- *Spanning Tree:* Spanning Tree is a tree where one point is connected only with the nearest point of the net.
- *Steiner Tree:* This wire length method is very realistic and close to the results given by the *Maze routing* algorithm. In Steiner Trees, connections can be done not only by two points but between a point and connection.

The wire length of a circuit is important but the balanced distribution of the connections is mandatory to avoid the congestion. The congestioning is indirectly minimized with the wire length, but it is not a guarantee that there is no congestion points in the circuit.

Power dissipation is another important factor in VLSI designs. Higher Clock frequencies associated with the device scaling in deep submicron technologies are increasing the power dissipation in modern circuits. Thus, total power reduction techniques must be taken into account when developing new algorithms for physical design.

However, placement oriented to a total power dissipation reduction may provoke areas with high power concentration. Furthermore, placement algorithms targeting power consumption must be able not only to reduce the power dissipation, but they must deal with the distribution of power in the whole circuit.

Deep submicron technology process present more challenges to physical designers concerning *timing* of connections. The wire length is not the only factor considered by a placer, also electric problems such as timing and signal integrity must be taken into account when placement is done.

In nowadays technologies, several metal layers are available and connections are each time smaller. The increase of the resistance due to the reduction of the connections widths and the larger capacitance between lines may cause signal degradation and bigger delay. In the worst case, crosstalk may change the logic signal of a net.

The *area* occupied by a circuit must be considered by the placement algorithm. In order to deal with this constraint, placement algorithms must be able to generate balanced rows and to manage very well empty spaces. A balanced distribution of cells in the rows is the key to reduce the area occupied by the circuit.

3.4 Goals of Routing Algorithms

One of the most important issues imposed by recent technologies is related to circuit wires (HENTSCHKE, 2007). Designs are getting bigger while component sizes are becoming dramatically smaller. This *scaling* scenario imposes larger, denser and more complex wiring nets.

Considering *timing* issues, the amount of delay of the logic is being reduced in comparison with the interconnection delay. Interconnection delay is responsible for more than 50% of a circuit delay in submicron technologies.

Considering the *power consumption*, which is strongly affected by the capacitance of circuit nodes. Large wires represents considerably large capacitance to be charged or discharged.

In other words, routability, timing, power and manufacturability are strongly affected by interconnect complexity of a design. In order to cope with wire related problems, the effort of reducing wire length is a very relevant issue on physical design research. Shorter wires are faster, dissipate less power, lead to less complex wiring networks affecting routability and manufacturability.

3.5 State-of-the-art Algorithms for transistor placement and routing

Many algorithms have been presented in the literature during the last decades. These algorithms are related with *Placement & Routing* and *Compaction* techniques and aims to face new technology process challenges. Some of them are discussed in this Section.

3.5.1 Transistor Placement Using an O-tree Algorithm

A non-slicing floorplaning based on ordered trees (O-tree) is presented in (GUO; CHENG; YOSHIMURA, 1999). The O-tree placement algorithm presented in this paper is characterized by representing the placement of blocks using an ordered tree structure.

A n-node O-tree is a tree with $n+1$ nodes and encoded by a 2-bit string **T** to identify the branching structure of the tree, and a permutation π as the labels of the n nodes. The bit string **T** is a realization of the tree structure. We write a '0' for a traversal which descends an edge and a '1' when it subsequently ascends that edge in the tree. The permutation π is the label sequence when we traverse the tree in depth-first search order. The first element in permutation π is the root of the tree.

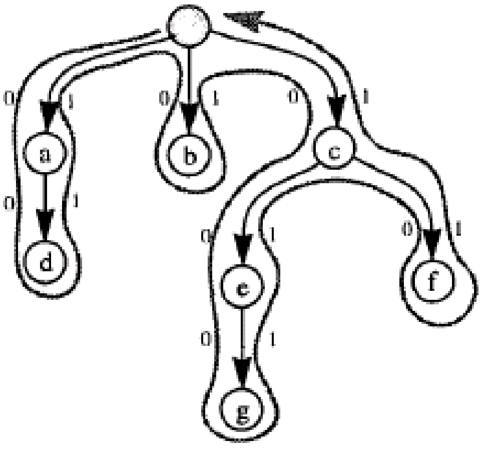


Figure 3.3: Example of a 8-node tree.

Given the tree shown in Figure 3.3, it can be represented by (00110100011011, adbcegf). Thus, starting from the root, we visit node *a* first and record a bit '0' to **T** and a label 'a' to π. Then we visit node *d* and record a bit '0' to **T** and a label 'd' to π. On the way back to the root from nodes *d* and *a*, we record two bits "11" to **T**. Then we visit subtrees *b* and *c* in sequence, and record the remaining of **T** and π respectively.

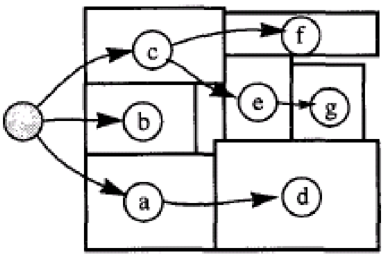


Figure 3.4: The O-tree representation placement.

The root of the O-tree represents the left boundary. Thus, the x-coordinate is given by

$$x_j = x_i + w_i \tag{3.2}$$

where the element *i* is the parent of the element *j*, w_i is the width of the element *i*, x_i and x_j are the left most position of the elements *i* and *j*, respectively. In the O-tree placement, x_{root} and w_{root} must be considered as ZERO. Figure 3.4 shows the placement which is represented by the horizontal O-tree in Figure 3.3.

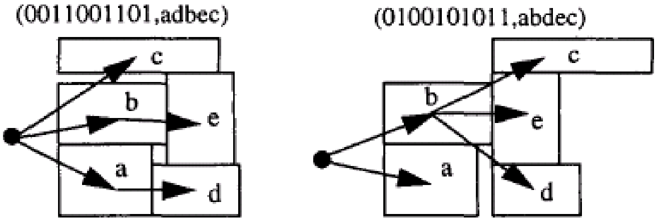
The permutation *x* determines the vertical position of the component when two blocks have proper overlap in their x-coordinate projections. For each element B_i, let $\Psi(i)$ be the set of B_k with its order lower than B_i in permutation π and interval (x_k, $x_k + W_k$) overlaps interval (x_i, $x_i + w_i$) by a non-zero length. If $\Psi(i)$ is non-empty, we have

$$x_i = max_{k \in \Psi(i)} y_k + k_k \tag{3.3}$$

otherwise

$$y_i = 0 \tag{3.4}$$

The definition of **LB-compact** placement is used to guarantee the best placement according to the O-tree representation. Thus, for a given tree encoding, the algorithm returns the most possible compacted placement. When all elements are placed in the left side, the solution is considered as **L-compact**. A solution is **B-compact** whether all elements are placed in the bottom.



(a) Admissible (b) Not admissible

Figure 3.5: Admissible o-tree.

A solution is considered as admissible whenever the resulting placement is a **LB-compact** placement, being both L-compact and B-compact. Figure 3.5(a) illustrates an admissible solution whose elements are placed on the left and the bottom boundaries. Figures 3.5(a) shows an example of not admissible placement solution due to the x-coordinate of the transistors *c* and *d*.

Giving an initial O-tree, as shown in Figure 3.3, a new placement configuration can be generated by deleting a component from the O-tree and placing it in another insertion position. For *n* elements, there is *2n-1* possible perturbed positions. If the element *d* from the O-tree is chosen to be deleted and permuted, then the possible insertion positions for this element are shown in Figure 3.6. In order to simplify the algorithm, the insertion positions are considered only at the external nodes of the tree.

An automatic datapath placer is presented in (SERDAR; SECHEN, 2001) whose transistors placement are encoded following the O-tree representation presented in (GUO;

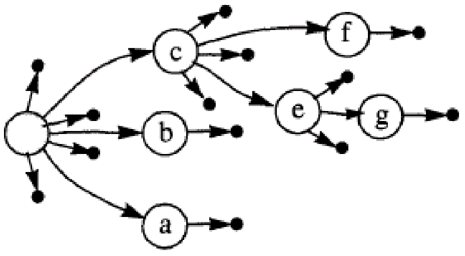
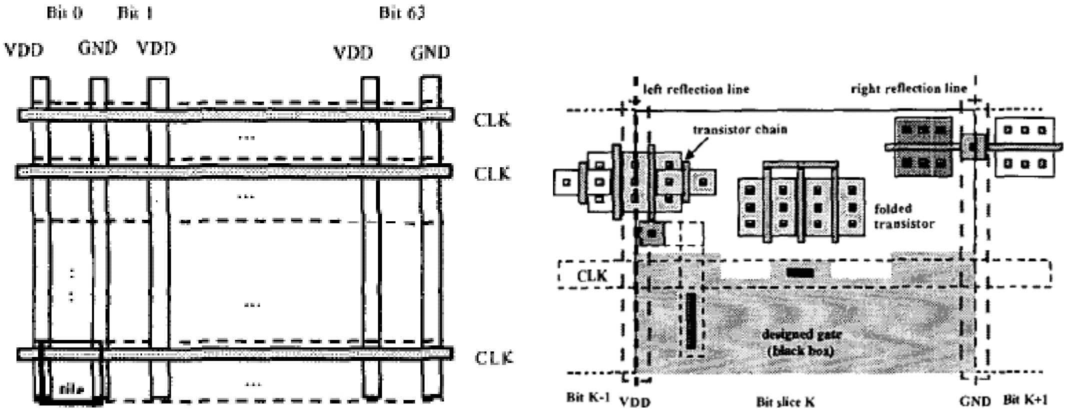


Figure 3.6: Possible insertion position of a external node.

CHENG; YOSHIMURA, 1999). Some modifications were done in order to deal with datapath tile placement characteristics. The main characteristics are the ability to handle fixed tile width, placement on the reflection line (essential to connect adjacent tiles) and non-rectangular shapes.


Bit 0
Bit 1
Bit 63
VDD
GND
VDD
VDD
GND
CLK
CLK
CLK
tile
left reflection line
right reflection line
transistor chain
folded transistor
CLK
designed gate (black box)
Bit K-1
VDD
Bit slice K
GND
Bit K+1

(a) View of a regular datapath structure.

(b) Example of possible placeable devices in a datapath tile.

Figure 3.7: Datapath tile placement.

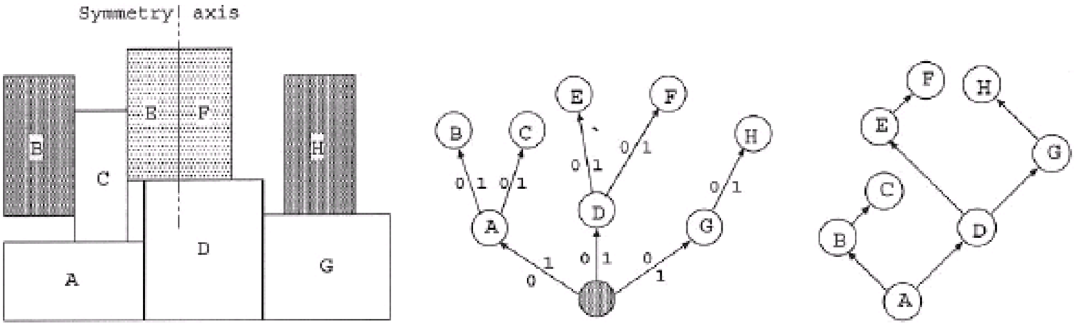
Figure 3.7 shows the basic idea of the datapath structure. In Figure 3.7(a), the regular structure of a datapath is shown where tiles are placed side-by-side in order to generate the layout. Clock and Supply lines are distributed by the circuit in such a way they connect every tile. Figure 3.7(b) illustrates important characteristics of the datapath generation, where supply lines, the clock line and transistor connections are shared between adjacent tiles.

3.5.2 Transistor Placement with Symmetry Constraints

In high-performance analog circuits, it is often required that groups of devices are placed symmetrically with respect to one or several axis to match the layout-induced parasitics in the two halves. Failure to match these parasitics in differential analog circuits can lead to higher offset voltages and degraded power-supply rejection ratio (BALASA; MARUVADA; KRISHNAMOORTHY, 2004).

Binary trees were used in (BALASA, 2000, 2001; BALASA; MARUVADA; KRISHNAMOORTHY, 2002, 2004), instead of O-trees, for representing the transistor placement in analog design at device level. Results presented in these works shown that binary trees can efficiently represent the placement of transistors with smaller complexity and saving CPU time.

An important characteristic of these works is related to the possibility to deal with symmetric transistor placement. Symmetry constraints were inserted into the tree encoding, where a subset of tree representations called *symmetric-feasible* is taken into account during the search of the solution space. In these works, simulated annealing is used to explore the solution space.



(a) Placement with symmetry. (b) O-tree. (c) Binary tree.

Figure 3.8: Placement with symmetry group and different encodings.

Figure 3.8 shows the placement of elements with symmetric constraints. In Figure 3.8(a), an example of symmetric placement is illustrated where the pair of transistors *(B,H)* and *(E,F)* are symmetrically placed. The O-tree and binary tree encodings are presented in Figure 3.8(b) and 3.8(c), respectively.

Considering the placement of two transistors *i* and *j*, symmetry constraints are given by

$$|x_{symAxys} - x_i + w_i| = |x_{symAxys} - x_j| \tag{3.5}$$

and

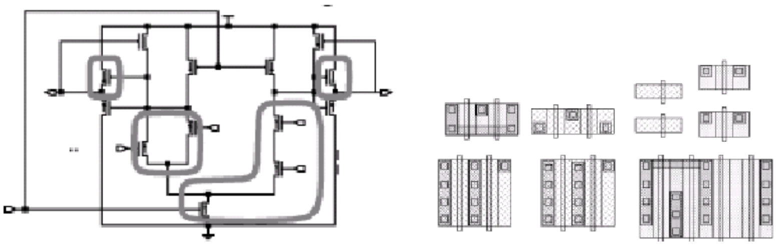
$$y_i = y_j \tag{3.6}$$

where the element *i* is always in the left side of the element *j* and $x_{symAxys}$ is the x-coordinate of the axis of symmetry.

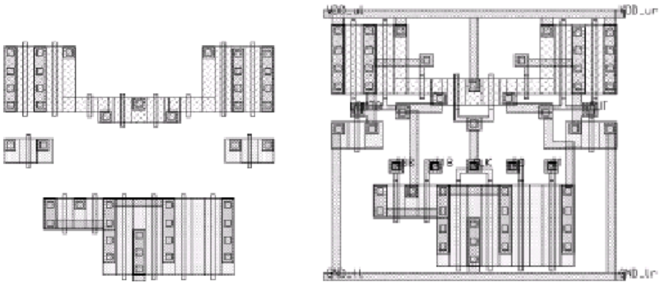
3.5.3 Non-complementary Transistor Placement

There is an increasing need in modern VLSI designs for circuits implemented in high-performance logic families such as Cascode Voltage Switch Logic (CVSL), Pass Transistor Logic (PTL), and domino CMOS. Circuits designed in these non-complementary logic families can be highly irregular, with complex diffusion sharing and nontrivial routing. Traditional digital cell layout synthesis tools derived from the highly stylized “functional cell” style break down when confronted to such circuit topologies. These cells require a full-custom, two-dimensional layout style, which currently requires skilled manual design.

A methodology for the synthesis of such non-complementary digital cell layouts is presented in (RIEPE; SAKALLAH, 2003). The methodology permits the concurrent optimization of transistor chain placement and the ordering of the transistors within these diffusion-sharing chains. The mechanism for supporting this concurrent optimization is the placement of transistor subchains, diffusion-break-free components of the full transistor chains. When a chain is reordered, transistors may move from one subchain (and therefore one placement component) to another. This permits the chain ordering to be optimized for both intra-chain and inter-chain routing. The placement algorithm is combined with third-party routing and compaction tools in order to finish the synthesis process.



(a) Transistor clustering (b) Chain formation



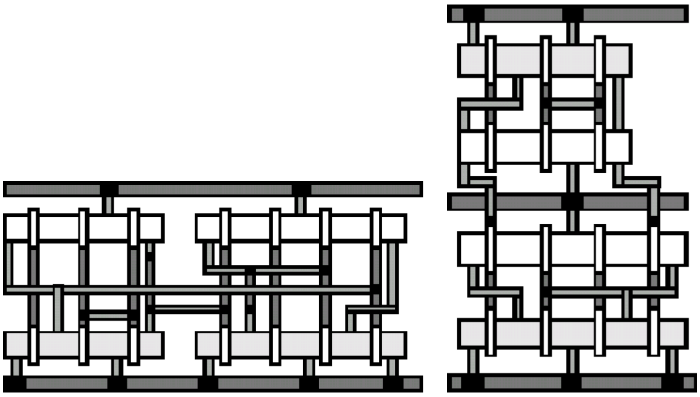
(c) Placement (d) Routing and compaction

Figure 3.9: Stages of the layout synthesis proposed in (RIEPE; SAKALLAH, 2003).

The methodology presented by Riepe and Sakallah (Figure 3.9) can be summarized by the following points when defining the cell-level transistor placement and routing problem:

1. The input to the system is a sized transistor netlist, the process design rules and technology parameters, and a description of the cell layout style.
2. Transistor source/drain geometry sharing is encouraged, but is not the primary optimization objective. Obtaining a routed cell of minimum area is the primary objective.
3. Individual transistors, or chains of source/drain connected transistors, may be placed in any position or orientation to optimize the objective as long as the design rules and template constraints are satisfied.
4. Routing inside the cell may be performed in two layers: Polysilicon and first-level metal. There is no preferred direction for routing in either layer.

3.5.4 Transistor Placement & Routing By Using Linear Integer Programming



(a) One-dimension layout (b) Two-dimension layout

Figure 3.10: The CLIP layout style.

In (GUPTA; HAYES, 2000), it is presented a technique for the automatic generation of layouts of CMOS cells in the two-dimensional (2D) style. The technique, CLIP (Cell Layout via Integer Programming) is based on integer-linear programming and solves both width and height minimization problems for 2D cell.

Width minimization is formulated in form that combines factors influencing the 2D cell width in a common problem space: transistor placement, diffusion sharing and vertical inter-row connections. This space is searched in a systematic manner by the branch-and-bound algorithms used in ILP solvers. For height minimization, cell height is modeled based on the horizontal wire density.

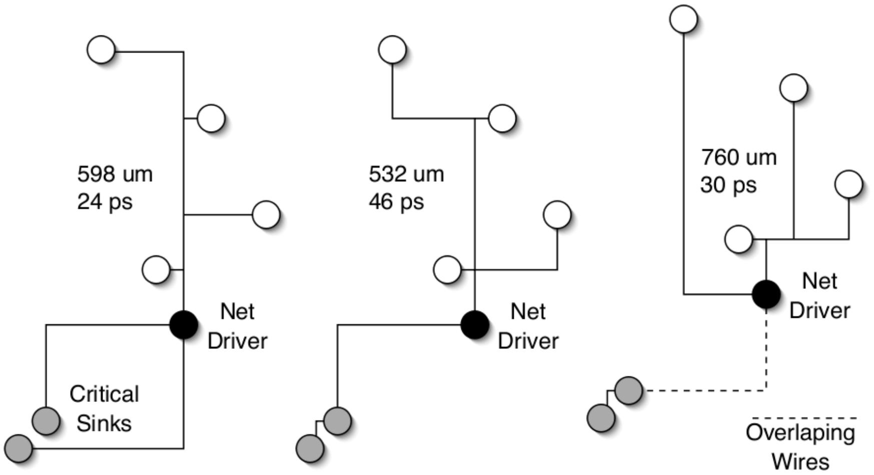
The CLIP run time for width minimization is in seconds for circuits with 30 or more transistors. For both height and width optimization, the CLIP is practical for circuits with up to 20 transistors. To extend the algorithms to larger circuits, hierarchical methods are necessary.

Figure 3.10 shows a one-dimension layout in Figure 3.10(a) and, the same two-dimension layout in figure 3.10(b). It is important to note that the three routing horizontal tracks in the two-dimension layout are distributed in the two-dimensional layout.

3.5.5 A Maze Routing Steiner Tree with Effective Critical Sink Optimization

An algorithm for optimized steiner tree generation is presented in (HENTSCHKE et al., 2007). The algorithm called AMAZE consists on the application of a biasing technique as the key to achieve wire length reduction by maximizing wire sharing. On the other hand, repulsive biasing, path length and sharing factors were introduced to isolate critical paths so that delay to the identified critical sinks is minimized.

Figure 3.11 compares nets produced by 3 algorithms. We observe that, with the AMAZE algorithm, the path for the critical sinks has minimum length and minimum sharing, while the rest of the tree is optimized for wire length. The best effort of AHHK (ALPERT et al., 1995) algorithm found the minimum path length to all sinks (arbores-



(a) AMAZE (b) AHHK (c) P-Tree

Figure 3.11: Comparison of the best trees generated by (a) AMAZE (HENTSCHKE et al., 2007), (b) AHHK (ALPERT et al., 1995) and (c) P-Trees (LILLIS, J. et al., 1996) algorithms.

cence), however it didn't help much for the delay of the critical sinks, since the path is fully shared by both sinks.

With the P-Trees algorithm (LILLIS, J. et al., 1996), among 35 different topologies evaluated, the one that was best for minimizing delay to critical sinks has separated wires to the critical sinks. The P-Trees's drawback that impacted the delay is the fact that the overall wire length is too large (affecting the product of driver resistance per total capacitance).

Among the various topologies generated, the one with better isolation of the critical path failed to provide good wirelength for the rest of the tree. Another drawback is the fact that overlapping wires could be used for Steiner trees and global routing but not for detailed routing.

The AMAZE algorithm outperformed algorithms used in the industry and in the state-of-the-art academic research, such as AHHK by 25%–40% and P-Trees by 1%–30%.

3.5.6 Routing with a Negotiation-based Algorithm

A negotiation-based algorithm for cells routing is presented by Ziesemer in (ZIESEMER, 2007). The negotiation methodology was previously proposed to routing FPGA circuits (MCMURCHIE; EBELING, 1995). Ziesemer has used the methodology to generate cells very efficiently.

The algorithm is based on the competition of resources. Thus, nets compete to get congested nodes and nets with more difficulties to find an alternative path have more probability to get the node. Authors guarantee that the negotiation-based algorithm is better than the traditional *rip-up and re-rout* algorithms.

Figure 3.12 illustrates the data structure for the negotiation-based algorithm. The layout is represented by a graph where each node has an associated cost.

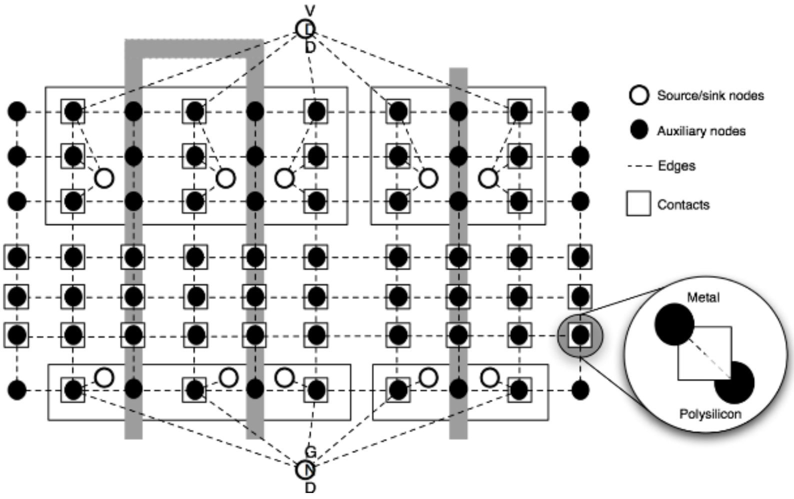


Figure 3.12: Data structure for the negotiation-based algorithm.

The cost to use a given node in an iteration is given by

$$C_N = (B_n + H_n) \times P_n \tag{3.7}$$

where B_n is the basis cost for the edge to achieve the node n, H_n is the congestion history of the previously iterations and P_n is the number of paths crossing node n.

To achieve better electrical characteristics of the resulting cell layout, different weights are applied to the graph edges according to its layer and position. Connections in polysilicon mean a bigger cost than in metal and contacts have an even bigger cost. Connections in metal over the transistor gates have an increased cost since they frequently insert additional space in the cell width and therefore must be avoided.

Input/output port connections require additional space to be placed and also there are fixed rules that must be followed. A chain of two or more serial transistors in the same diffusion row can be placed without diffusion contacts between the gates. This allows an area reduction if no ports are placed in the closest track to the transistors. For this reason, better area results are achieved when increasing the routing cost of the graph edges that lead to these ports.

3.6 Transistor Placement Technique Using Genetic Algorithm And Analytical Programming

We propose a new transistor placement technique using genetic algorithm associated to analytical programing in (LAZZARI; ANGHEL; REIS, 2005a). The approach presented in this work is basically divided in three phases. First, a classical genetic algorithm is used to generate some parameters concerning transistor orientation and the relationship between them. These parameters are used as placement constraints described in an algebraic modeling language. The second phase consists on solving the placement constraints by a nonlinear solver in order to find the optimal solution according to given constraints.

After that, the best solutions are propagated and genetic operators are applied to the solutions.

Algorithm 4 : The genetic algorithm.

```
Require: Set of Transistors $T$, Population Size $N$, Number of Iterations $I$
Ensure: Transistor placement $p$
$P \Leftarrow$ generatePopulation( $T$, $N$ );
while $i < I$ do
   for all $k \in P$ do
      solveConstraints( $k$ );
      calculateFitness( $k$ );
   end for
   $P \Leftarrow$ doEvolution( $P$ );
   $i++$;
end while
$p \Leftarrow$ getBestSolution( $P$ );
```

The pseudo code of the proposed approach is presented in Algorithm 4. An initial set of solutions is generated in the function `generatePopulation(` N `)` where each chromosome in the population *P* has a set of constraints about the transistor placement problem. The generation of this initial population is explained in Section 3.6.1.

In function `solveConstraints(` k `)`, the parameters of the chromosome *k* are converted to an algebraic modeling language and the placement problem is solved.

The fitness of a chromosome is generated in the function `calculateFitness(` k `)`. The fitness of a chromosome is calculated based on the objective function as described in Section 3.6.3.4.

The function `doEvolution(` P `)` is basically the reproduction of the chromosomes in the population *P* to generate a new population with better results. In the generation of this new population, operations of elitism, mutation and crossover are applied to the chromosomes in order to propagate the best solutions and to evolve the other chromosomes.

3.6.1 Initial Population Generation

The range of possible solutions in the process of layout generation is related to the number of elements in a cell or macro-block. Moreover, the relation between these elements makes a solution better than others. Thus, some techniques can be used for reducing the number of elements and, consequently, decreasing the complexity of the layout generation problem.

Transistor chaining is a technique that consists of grouping transistors when their drain/source diffusions can be shared. Figure 3.13 illustrates the transistor chaining generation where the *Euler* path is searched to PMOS and NMOS transistors. Dashed lines illustrate *Euler* paths in which a chain of transistors is performed based on the sharing of the source/drain diffusion areas. In this example, the two transistors chains are (*Z,B,2,A,1,A,VCC,C,1,B,2,C,Z*) to PMOS transistors and (*GND,B,3,A,Z,A,4,C,GND,B,5,C,Z*) to NMOS transistors.

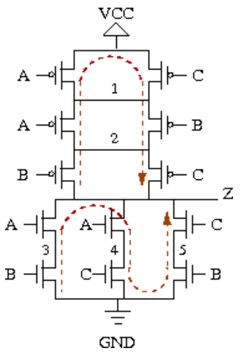


Figure 3.13: Euler path example.

Its is clear that many solutions can be found to these set of transistors. In the approach proposed in this work, an *Eulerian* graph is used in order to generate the *N* solutions related to the initial population. Transistor chainings are randomly chosen to be used in the genetic algorithm.

3.6.2 The Placement Parameters

Each chromosome in the genetic algorithm is a set of parameters used in the placement constraints. Parameters used in transistor placement are basically the description of transistors orientation and the relationship between these transistors. Transistor orientation means whether a transistor must be placed horizontally or vertically and where the drain/source contacts are located, while the relationship between transistors is the relative placement of a transistor in relation to each other transistor.

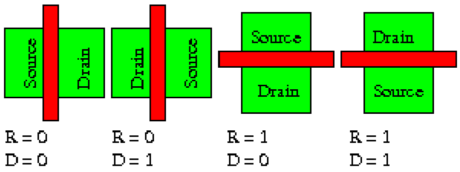


Figure 3.14: Transistor orientation constraints.

Figure 3.14 illustrates the orientation constraints `R` and `D`. The parameter `R` represents the orientation of the transistors. `R = 0` indicates that the transistor must be placed horizontally and `R = 1` means that the transistor must be placed vertically.

The parameter `D` indicates where drain/source diffusion areas are located. `D = 0` means that the transistor source area is located in the left/top and `D = 1` means that the drain area is located in the left/top of the transistor.

The relationship between transistors is shown in Figure 3.15. The parameters `C` and `Pc` are used to describe these relationship. `C` indicates whether the placement constraints are related to horizontal or vertical coordinates and `Pc` represents the relative position of these transistors.

Taking as example the transistors *M1*, *M2* and *M3* illustrated in Figure 3.15, `C[M1,M2] = 0` means that the transistors *M1* and *M2* are placed side by side hori-

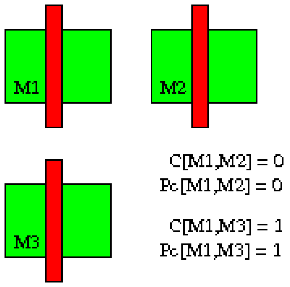


Figure 3.15: Transistor behavior constraints.

zontally and `Pc[M1,M2] = 0` indicates that the transistor *M1* is placed in the left side of *M2*. In other words, $X_{M1} < X_{M2}$ and there is no requirements to coordinate Y.

The same idea is used to `C[M1,M3] = 1` and `Pc[M1,M3] = 1`. In this case $Y_{M1} > Y_{M3}$ and any horizontal constraint is applied. Table 3.1 shows the possible constraints resulting of the parameters `C` and `Pc`.

Table 3.1: Parameters to the transistors relationship.

Parameters		**Constraints**	
`C`	**`Pc`**	**Horizontal**	**Vertical**
0	0	$X_1 < X_2$	–
0	1	$X_1 > X_2$	–
1	0	–	$Y_1 < Y_2$
1	1	–	$Y_1 > Y_2$

Based on these parameters, each chromosome is a binary vector containing information about orientation and relationship between transistors. The size of a chromosome is given by Equation 3.8:

$$L_{chrom} = T * 2 + \sum_{i=1}^{T-1} i * 2 \tag{3.8}$$

where *T* is the number of transistors. The first part of the equation 3.8 is related to parameters `R` and `D`, and the second part is related to parameters `C` and `Pc`.

3.6.3 The Mathematical Modeling

Once the parameters are defined in the chromosomes, they can be applied in an algebraic modeling in order to obtain the optimal placement solution to given parameters and constraints. The main idea of this approach is to use a nonlinear solver to find the solution to the placement of transistors.

Figure 3.16 shows width and height parameters used in the placement constraints. For each transistor $i \in T$, the parameters wd_i and hd_i are the width and height of the diffusion area while wp_i and hp_i are the parameters for the polysilicon area. Besides, three integer parameters $drain_i$, $source_i$ and $gate_i$ represent the connections of the transistors and the parameter $type_i$ is also used to indicate PMOS and NMOS transistors.

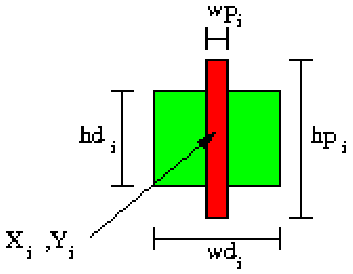

wp_j
hd_j
hp_j
X_j ,Y_j
wd_j

Figure 3.16: Width and height transistors parameters.

The variables X_i and Y_i are the central coordinates of the transistor i. Their values are given by the minimization of the objective function. The goal of the used objective function is to find the optimal X_i and Y_i by the minimization of the wire lengths. The specification of the objective function is given in more details in Section 3.6.3.4.

The constraints are divided in three groups: 1) Boundary Constraints, 2) Neighborhood Constraints and 3) Connections Constraints.

3.6.3.1 Boundary Constraints

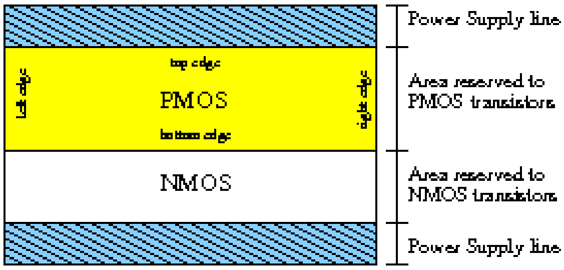

Power Supply line
top edge
left edge
PMOS
right edge
bottom edge
Area reserved to PMOS transistors
NMOS
Area reserved to NMOS transistors
Power Supply line

Figure 3.17: A row-based boundary representation.

The layout of standard-cells and macro-blocks is usually structured in rows. In these structures, layout boundaries must be regular in order to allow the connection between adjacent cells at the moment of circuit generation. Figure 3.17 illustrates the boundaries in a row-based layout.

Regions for PMOS and NMOS transistors can be determined by the implant areas and boundary constraints can be formulated according to the edges of these areas. Thus, the boundary constraints are given by

$$B_{left} + \Delta_x + \frac{1}{2}W_i \leq X_i \leq B_{right} - \Delta_x - \frac{1}{2}W_i \tag{3.9}$$

$$B_{bottom} + \Delta_y + \frac{1}{2}H_i \leq Y_i \leq B_{top} - \Delta_y - \frac{1}{2}H_i \tag{3.10}$$

where B_{left}, B_{right}, B_{bottom} and B_{top} are the edges of the placement region, Δ_x and Δ_y are the minimal distances from the transistor i to the boundaries, and W_i and H_i are the the width and height of the transistor.

3.6.3.2 Neighborhood Constraints

Neighborhood constraints are related to the possibility to connect transistors. These constraints are separated in categories and they are responsible to give the correct distance between two adjacent transistors.

In order to verify the possibility of connection between transistors, the variables $left$, $right$, top and $bottom$ are used. They are given by

$$left_i = (D_i + R_i * D_i) * drain_i + (1 - D_i + R_i * D_i) * source_i + R_i * gate_i \quad (3.11)$$

$$right_i = (1 - D_i + R_i * D_i) * drain_i + (D_i + R_i * D_i) * source_i + R_i * gate_i \quad (3.12)$$

$$top_i = (R_i - R_i * D_i) * source_i + (R_i * (1 - D_i)) * drain_i + (1 + R_i) * gate_i \quad (3.13)$$

$$bottom_i = (R_i - R_i * D_i) * drain_i + (R_i * (1 - D_i)) * source_i + (1 + R_i) * gate_i \quad (3.14)$$

where $i \in T$, R_i and D_i are the parameters given by the current chromosome. $drain_i$, $source_i$ and $gate_i$ are integer parameters related to the list of connections C.

Considering κ_c the number of points of the connection c and assuming that $c \in C$, it is possible to know when two transistors are connected in series or parallel. Thus, two transistors are in series whenever $\kappa_c = 2$. In all other cases the transistors are in parallel or they are not connected.

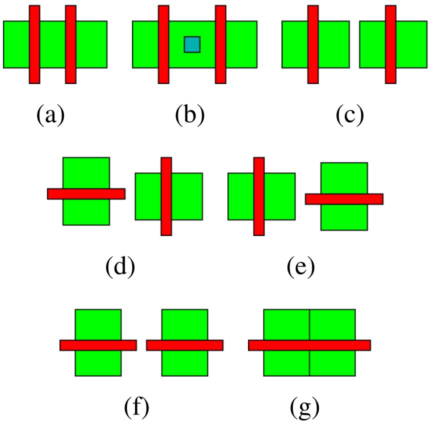
Table 3.2: Horizontal neighborhood constraints.

#	**Figure**	**Orientation Parameters**		κ_c	**Situation**	**Constraint**
		R_i	R_j			
1	3.18(a)	0	0	$= 2$	$right_i = left_j$	$X_j - X_i \geq \frac{1}{2}wp_i + sp + \frac{1}{2}wp_j$
2	3.18(b)	0	0	$\neq 2$	$right_i = left_j$	$X_j - X_i \geq \frac{1}{2}wp_i + 2 \times spc + wc + \frac{1}{2}wp_j$
3	3.18(c)	0	0	×	$right_i \neq left_j$	$X_j - X_i \geq \frac{1}{2}wp_i + sd + \frac{1}{2}wp_j$
4	3.18(d)	1	0	×	×	$X_j - X_i \geq \frac{1}{2}hp_i + sdp + \frac{1}{2}wd_j$
5	3.18(e)	0	1	×	×	$X_j - X_i \geq \frac{1}{2}wd_i + sdp + \frac{1}{2}hp_j$
6	3.18(f)	1	1	×	$right_i = left_j$	$X_j - X_i \geq \frac{1}{2}hd_i + \frac{1}{2}hd_j$
7	3.18(g)	1	1	×		$X_j - X_i \geq \frac{1}{2}hp_i + sp + \frac{1}{2}hp_j$

From the definition of these variables, it is possible to understand how the neighborhood constraints are formulated. Figure 3.18 illustrates every neighborhood possibility to the horizontal placement and Table 3.2 presents the neighborhood constraints where sp is the spacing between polysilicon lines, sdc is the distance between a polysilicon line and a contact, wc is the width of a contact, sd is the spacing of two diffusion areas and sdp is the distance between a polysilicon line and a diffusion area.

Neighborhood constraints are separated in categories with the effort to deal with every possible relationship between two transistors. Only horizontal constraints are discussed here but similar equations are used vertically.

Seven different constraints are shown in Table 3.2. In the case of $R_i = 0$ and $R_j = 0$, equation 1 treats situations where transistors are in series, equation 2 deals with parallel transistors and equation 3 takes situations where transistors are not connected.


(a) (b) (c)

(d) (e)

(f) (g)

Figure 3.18: Horizontal neighborhood.

The equation 3 and 4 treat situations where there are different transistor orientation parameters ($R_i \neq R_j$). In these cases, the sharing of diffusion areas is impossible.

When transistors are placed vertically ($R_i = 1$ and $R_j = 1$), the connection between two transistors is possible only if $top_i = top_j$, $right_i = left_i$ and $bottom_i = bottom_j$ (Equation 6). Equation 7 takes all other cases to $R_i = 1$ and $R_j = 1$, in which the connection between transistors cannot be done.

3.6.3.3 Connection Constraints

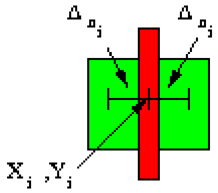


Figure 3.19: The Δ_{n_i} representation.

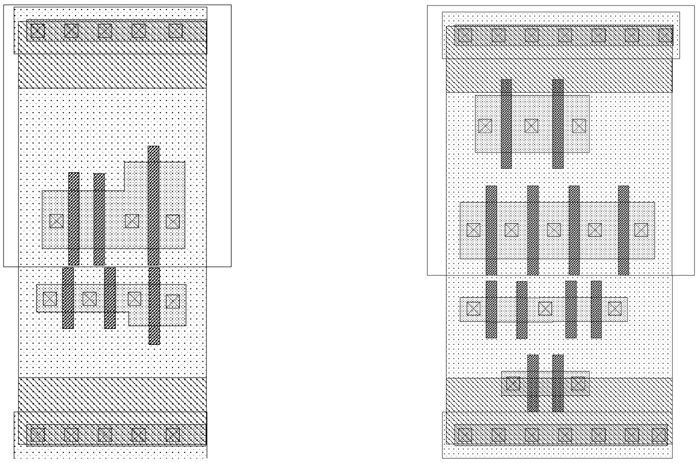
Let n be the number of connections and m the number of transistors, the position to gate, drain and source can be inserted in matrix notation to the horizontal and vertical coordinates, Q_x and Q_y. Thus, Q_x and Q_y are $n \times m$ matrices where the coordinates X and Y of the nets are given by

$$Q_x(drain_i, i) = (1-R_i)*D_i*(X_i-\Delta_{n_i})+(1-R_i)*(1-D_i)*(X_i+\Delta_{n_i})+R_i*X_i \quad (3.15)$$

$$Q_x(source_i, i) = (1-R_i)*(1-D_i)*(X_i-\Delta_{n_i})+(1-R_i)*D_i*(X_i+\Delta_{n_i})+R_i*X_i \quad (3.16)$$

$$Q_x(gate_i, i) = X_i \quad (3.17)$$

where $i \in T$ and Δ_{n_i} is the distance from the center of the transistor to the point where the connection is located as shown in Figure 3.19. The matrix to vertical coordinates Q_y is composed based on the same idea.



(a) An OR2 layout (b) An AOI222 Layout

Figure 3.20: Preliminary placement examples.

3.6.3.4 The Objective Function

The goal of the proposed technique is to reduce the wire length connecting the transistors. Thus, the objective function is based on the connection constraints and it is obtained by

$$OBJ : min\left\{\sum_{c \in C} W_c * S(c)\right\} \tag{3.18}$$

where $S(c)$ is the half perimeter wire length and W_c is the weight of the connection c. The wire length of a connection c is calculated by the coordinates of the points of a net in the matrices Q_x and Q_y. Then, $S(c)$ is given by

$$S(c) = \sum_{i \in T, j \in T, i \neq j} HP(c, i, j) * I(c, i) * I(c, j) \tag{3.19}$$

and

$$HP(c, i, j) = |Q_x(c, i) - Q_x(c, j)| + |Q_y(c, i) - Q_y(c, j)| \tag{3.20}$$

where $I(c, i)$ are binary values indicating whether the wire c is connected to the transistor i. The same principle is used to $I(c, j)$ with the connection c and the transistor j.

3.6.4 Obtained Results

Figure 3.20 shows the placement of two cells using the proposed algorithm. The transistor placement of an OR2 gate is shown in Figure 3.20(a) and the placement of an AOI222 is shown in Figure 3.20(b).

Table 3.3 shows some results of the comparison between the proposed technique and a pure Eulerian placement algorithm used in (LAZZARI et al., 2003). Results show that the proposed technique deals with the transistor placement problem. The area gain is around 4.5 % .

The drawback of this technique is the execution run time. While a pure Eulerian algorithm executes the placement task very quickly, the proposed technique take hours in

Table 3.3: Placement results.

Cell Name	Area (μm)		Gain (%)	Execution Time
	(LAZZARI et al., 2003)	Proposed		
NOR2	7.9	8.1	-3	2s
OR2	10.8	10.9	-1	1m 15s
AOI22	13.6	13.0	4	4m 10s
AOI222	19.4	17.9	8	18m 30s
Full Adder	52.3	45.1	13	3h 15m

some cases to solve the placement problem. As the genetic algorithm works with random information, the execution time presented in Table 3.3 is the average time of at least 5 executions of each cell.

3.7 Compaction and Layout Optimization using Linear Programming

Linear programming has been used to solve many design problems such as placement (WANG; LILLIS; SANYAL, 2005), routing (BEHJAT; CHIANG, 2005) or even complete cell generation (GUPTA; HAYES, 2000). However, describing a problem using only linear constraints may impose undesired restrictions to the design and non optimal resulting layout.

Layout compaction and migration between technologies are applications where the linear programing is very well accepted. In this section, a layout compaction algorithm using linear programing is presented. In order to place each polygon in the layout, its coordinates and technology rules are represented in form of linear equalities and inequalities. The generic linear *LPSolve* (BERKELAAR, 2006) is called to solve the compaction constraints.

In order to understand how the linear programing compaction works, it is necessary to have in mind how the constraints are generated. When placement and routing are performed, the relative information of each polygon position is stored in the data structure. Thus, it is possible to know whether a polygon is placed on the left side or on the right side in relation to another polygon.

Before generating linear constraints, the whole layout generation process is done without considering the technology design rules. Transistor placement and routing algorithms do not need any information about the technology rules. The only needed information is which polygon is on the left/right (top/bottom in the case of vertical compaction) side of another polygon. In other words, the relative position of each polygon is known.

Once the relative position of the polygons is known, it is possible to describe the relation between these polygons as the linear programming. Figure 3.21 illustrates an example of layout, in which two transistors are placed side by side. Constraints related to these polygons can be described as shown in Algorithm 5.

WP is the polysilicon width and SP is the spacing between two adjacent polysilicon lines. $x1$ and $x2$ are left and right horizontal coordinates of the left polysilicon line. $x3$

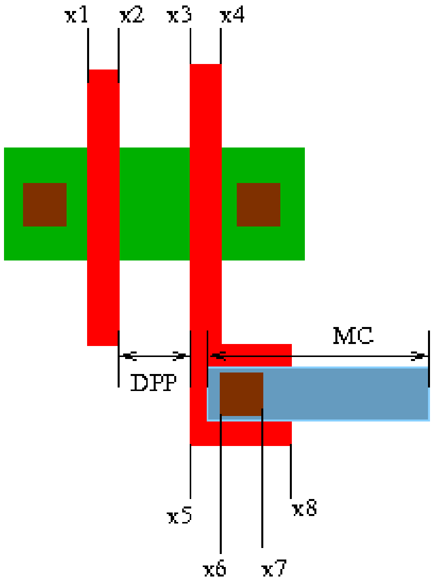


Figure 3.21: Example of layout.

Algorithm 5 : An example of layout representation in linear equalities and inequalities.

```
1: $x2 - x1 = WP$
2: $x4 - x3 = WP$
3: $x3 - x2 \geq SP$
4: $x7 - x6 = WC$
5: $x6 - x5 \geq EPC$
6: $x8 - x7 \geq EPC$
7: $x5 \leq x3$
8: $x8 \geq x4$
```

and $x4$ are the left and right coordinates of the right polysilicon line. The first and second lines of this example shows how the minimum transistor width is guaranteed.

The third line illustrates how the minimum spacing between two polygons can be guaranteed, where the spacing between the polysilicon lines are placed with a spacing equal or greater than the minimum spacing technology rule.

Enclosure constraints are shown in lines 04 to 06. WC is the contact width and EPC is the minimum enclosure of polysilicon over contact. These three constraints impose to the solver a way to find the coordinates for the polygons without violating the technology rules.

The connection between the contact and its enclosure with the polysilicon transistor is depicted by lines 07 and 08. Thus, there is a certain freedom to place the contacts, but the connection is respected.

The main idea of using linear programming in the layout generation is to reduce the area occupied by the circuit. Besides, some important aspects are taken into account in order to generate optimized layout. The main aspects are:

- *Stacked transistors* : Stacked transistor are known to form a resistive path. The distance between two transistors must be the minimum possible in order to reduce the area and perimeter of the transistors drain/source.

- *Polysilicon and diffusion lines* : The resistance of polysilicon and diffusion lines is very high. For this reason, small connections in these layers is mandatory.
- *Transistor active region* : As stacked transistors, active regions must be reduced in order to reduce the area and perimeter of the transistor drain/source regions.
- *Metal lines* : The resistance in metal connections is smaller than polysilicon/diffusion ones, but the reduction of these lines is also important.

The example shown in the Algorithm 5 can be described as follows in order to exemplify how this characteristics are taken into account in the linear programming. In this example, the spacing between the transistors DPP is inserted in the objective function aiming at reducing the distance between them.

Algorithm 6 : An example of layout representation in linear equalities and inequalities with an objective function.

```
$min : DPP$
$x2 - x1 = WP$
$x4 - x3 = WP$
$x3 - x2 = DPP$
$DPP >= SP$
$x7 - x6 = WC$
$x6 - x5 \geq EPC$
$x8 - x7 \geq EPC$
$x5 \leq x3$
$x8 \geq x4$
```

Each one of the aspects shown in the Algorithm 6 can be considered in linear programming by applying costs to the constraints and inserting them in the objective function. Constraint costs are directly related to the priority of a constraint. For example, the spacing between stacked transistors is very important due to the resistance of the active region of a transistor. A connection in metal has smaller priority. Thus, to improve the quality of the layout, the objective function can be written as

$$min : 2DPP + MC$$

where DPP is the distance between two polysilicon lines and MC is the length of a metal line. An objective function described in this form results in a better layout. Otherwise, if no costs are used, the spacing between the transistors can be harmed as a result of a small metal connection.

The complete compaction algorithm is presented in Algorithm 7. Function `describePolygonsAsLP(` B `)` converts the relative position of the polygons to linear equalities and inequalities. Costs are applied to the linear programming in Function `applyCosts(` D `).` After, the formulations are solved in Function

Algorithm 7 : A compaction algorithm using linear programming.

```
Require: Set of polygons $B$
Ensure: The polygons position $P$
1: $D \Leftarrow$ describePolygonsAsLP( $B$ );
2: applyCosts( $D$ );
3: $L \Leftarrow$ callLPSolver( $D$ );
4: $P \Leftarrow$ placePolygons( $L$ );
```

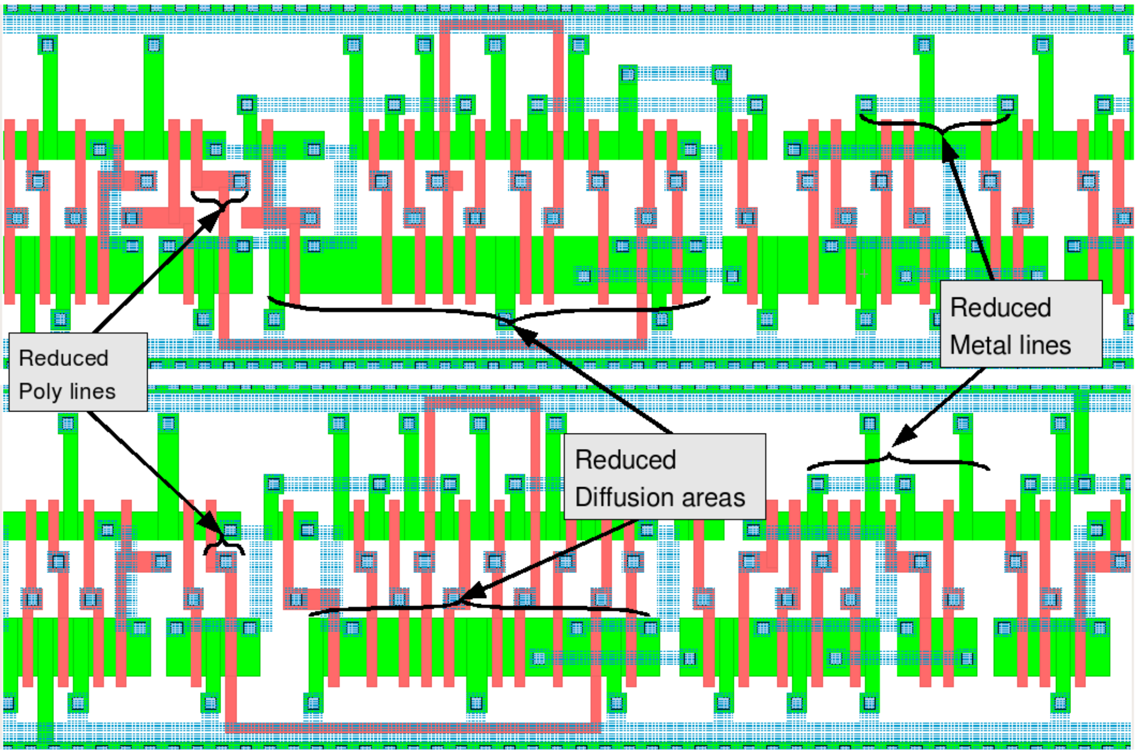


Figure 3.22: An example of layout compaction. The first figure is a layout compacted without cost assignment. Second layout is compacted with costs.

`callLPSolver( D )` and Function `placePolygons( L )` places each polygon in the layout.

Figure 3.22 illustrates the layout compaction without costs assignment and with costs attributed to the polysilicon lines, diffusion areas and metal connection. For this example, diffusion areas have higher cost (4), polysilicon lines have a smaller cost (3) and metal lines have cost equal to 1.

3.8 Overview of the Algorithms Developed in the Punch++

As discussed in Chapter 2, the runtime is one of the most important issues when generating a whole circuit at a time. There is a tradeoff between the quality of circuits and the performance of algorithms that must be carefully analyzed.

Thus, for the development of the automatic layout generator **Punch++**, we choose to sacrifice some layout aspects such as the occupied area to have computationally efficient algorithms.

Basically, the algorithms in an automatic layout generator are separated in three types: *Placement*, *Routing* and *Compaction*.

The **Euler path algorithm** (Section 3.2.1) is used to place the cells. The algorithm is computationally efficient and gives reasonable results. Transistors are separated concerning the logic functions. Thus, the complexity is reduced because the algorithm deal with only tenths of transistors.

The routing algorithm is based on the **negotiation algorithm** presented in Section 3.5.6. After the placement & routing, the **linear programming-based compaction algorithm** discussed in Section 3.7 is used to compact the layout. The routing and compaction is done in the whole row.

These three algorithms are basically the kernel of the layout generator. They are computationally very efficient and they result in a good rapport between layout quality and algorithms performance.

3.9 Conclusion

Physical synthesis algorithms are presented in this chapter. These algorithms are used in placement and routing of transistors aiming at generating whole blocks with up to thousands of transistors.

Algorithms presented in this chapter are the state-of-the-art of the methodologies existing in literature. The main characteristics of the placement, routing and compaction algorithms is to achieve with the recent and growth technology challenges.

A review of the main submicron technology process challenges are shown in Table 3.4.

Table 3.4: Review of the challenges targeted by physical synthesis algorithms.

Runtime	The computational complexity is one of the most important factors when developing new algorithms. Algorithms with linear complexity deal very well with a big number of variables. However, sometimes physical design problems are not easily represented in a linear form.
Routability	Routability is defined by the ability of routing algorithms to route all wires under electrical and topological restrictions. Placement and routing algorithms must be able to route the whole circuit and respect design and electrical rules.
Timing	The timing is not only result of the circuit topology, but also how physical synthesis algorithms deal with the placement, routing and compaction. Algorithm must be able to balance charges in the nets in respect to the driving cell and control the placement to avoid long wires.
Power Consumption	Power consumption is totally related with the transistor placement and routing. There are an enormous set of possibilities of placement and routing possibilities for a given set of transistors. The power consumption may considerably vary among this range of solutions according to the transistors sharing areas and the capacitances associated to the connections.

4 A RADIATION-INDUCED EFFECTS OVERVIEW

4.1 Introduction

Radiation effects such as Total Ionizing Dose (TID), Displacement Damage (DD) and Single Event Effects (SEE) provide aerospace designers' a myriad of challenges for the design of a system (LABEL, K. A. et al., 2000).

This chapter introduces some effects induced by the radiation. Principles and effects about the total ionizing dose and the displacement damage are briefly discussed. For the purpose of this thesis, single event effects are the focus of this chapter. More specifically, Soft SEE are discussed in details.

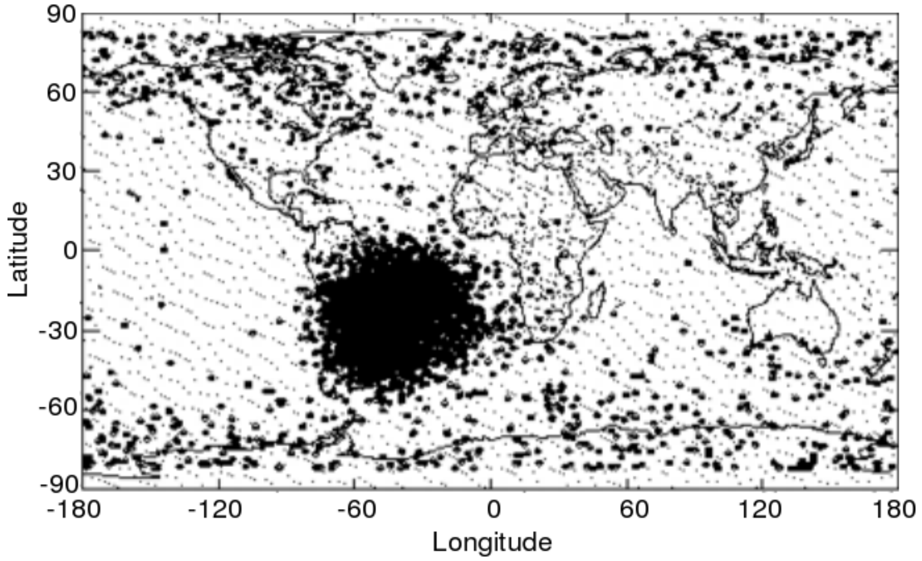


Figure 4.1: Location where Single Event Upsets (SEU) occured in a spacecraft into a polar orbit of altitude 700km (HARBOE-SORENSEN, R. et al., 1990).

Radiation-induced spacecraft anomalies have been known since the Explorer I launch on January 31, 1958, when a Geiger counter put aboard by Van Allen suddenly stopped counting. It turned out that the counter was in fact saturated by an extremely high count rate. This event led to the discovery of the Van Allen belts (MATIS, H. et al, 2003). The inner belt, beginning at about 1,000 km above the surface of the Earth, contains primarily protons with energies between 10-100 $MeVcm^2/mg$.

The offset between Earth's geographical and magnetic axes causes an asymmetry in the radiation belt above the Atlantic Ocean in the Brazilian coast, allowing the inner belt to reach a minimum altitude of 250 km.

This *South Atlantic Anomaly* (SAA) is important because it occupies a region in which low-orbiting satellites spend as much as 30% of their time. During a solar flare, which can happen anytime, the number of protons suddenly increases by more than a million. Figure 4.1 shows the location where Single Event Upsets (SEU) occured in a spacecraft into a polar orbit of altitude 700km as presented in (HARBOE-SORENSEN, R. et al., 1990).

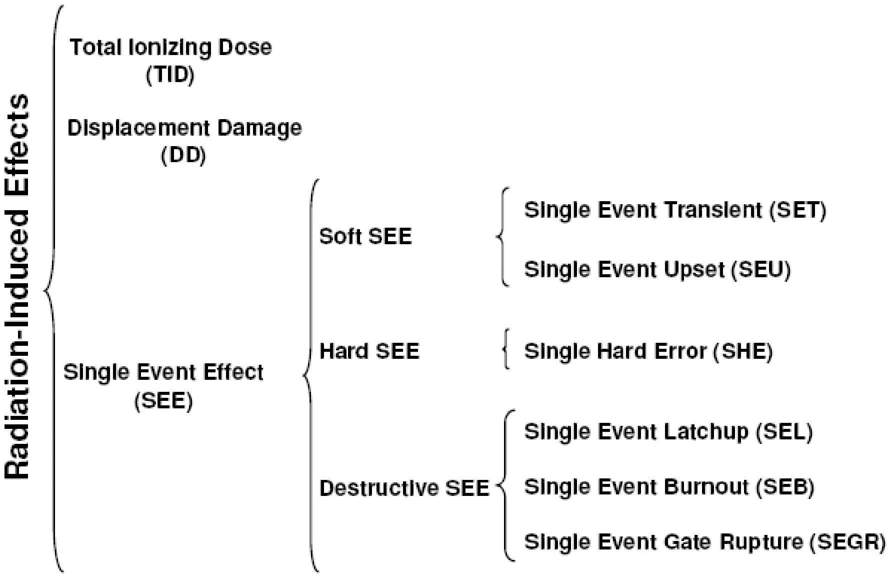

Radiation-Induced Effects
Total Ionizing Dose (TID)
Displacement Damage (DD)
Single Event Effect (SEE)
Soft SEE
Hard SEE
Destructive SEE
Single Event Transient (SET)
Single Event Upset (SEU)
Single Hard Error (SHE)
Single Event Latchup (SEL)
Single Event Burnout (SEB)
Single Event Gate Rupture (SEGR)

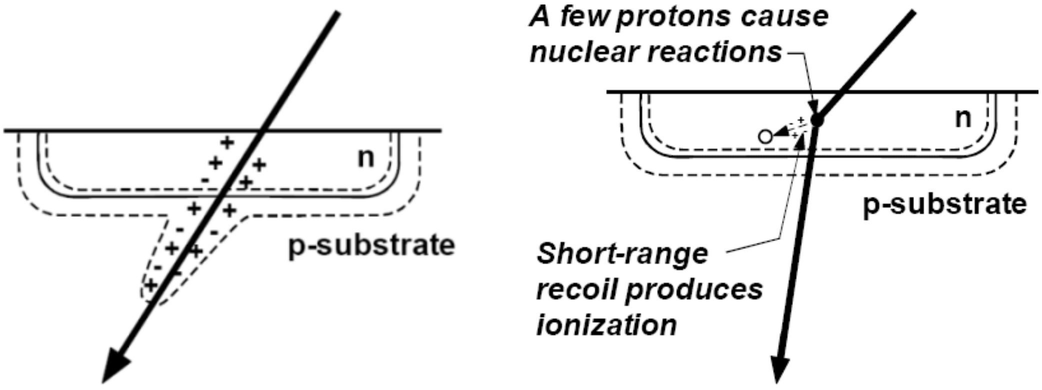
Figure 4.2: Classification of radiation-induced effects (BASTOS, 2006).

Bastos (BASTOS, 2006) classifies radiation-induced effects as shown in Figure 4.2. Basically, radiation-induced effects are divided in three categories: Total Ionizing Dose (TID), Displacement Damage (DD) and Single Event Effects (SEE).

Total Ionizing Dose (TID) is due to the degradation as a result of the cumulative energy deposited in the material. This degradation occurs in long term, but the effects are permanent in devices and usually lead to functional failures. Displacement Damage (DD) is also a long term degradation of devices, but is a different physical mechanism. Different from TID and DD, Single Event Effects (SEEs) occur when a single ion strikes the material with sufficient energy in the device to cause a system failure.

4.2 Single Event Effects (SEEs)

Single Event Effects (SEEs) occur when a single ion strikes the material, depositing sufficient energy in the device to cause an error. SEE are divided in soft and hard errors (LABEL, K. A. et al., 2000). Basically, a soft error occurs when a transient pulse or bit-flip causes an error detectable at the device output. Hard errors are physically destructive



(a) Heavy ions (Direct Ionization)

(b) Protons (Indirect Ionization)

Figure 4.3: Heavy ions and protons striking the silicon device. (a) heavy ion increasing the depletion region (b) *Spallation* caused by a proton or neutron.

to the device and cause permanent functional effects.

4.2.1 Soft SEE

Several types of particles are generated by the sun activity. These particles are classified in two groups: 1) *Charged particles* such as electrons, protons and heavy ions, and 2) *Electromagnetic radiation (Photons)* as x-ray, gamma ray and ultraviolet light (KASTENSMIDT, 2003).

A heavy ion is defined as any ion with atomic number greater than two. A single heavy ion striking a silicon device looses its energy as a result of the production of free electron-hole pairs. The creation of these electron-hole pairs generates a dense ionized track increasing the depletion region as shown in Figure 4.3(a). In other words, heavy ions cause direct ionization within the device.

The direct ionization is the primary charge deposition mechanism for single events. The particle looses its energy during the directing ionization, when crossing the semiconductor material.

The term *linear energy transfer* (LET) is used to describe the energy loss by a particle. The LET of a particle can be easily related with its charge deposition (DODD; MESSENGILL, 2003). An LET of 97 $MeV - cm^2/mg$ in silicon devices corresponds to a charge deposition of 1 $pC/\mu m$.

Lighter particles such as proton, electron and neutrons do not usually produce enough charge by direct ionization to cause a single event. Typically, protons and neutrons cause a transient pulse or a bit-flip through complex nuclear reactions such as emission of alpha and gamma particles or spallation in the vicinity of the sensitive node (Figure 4.3(b)). *Spallation* is a nuclear reaction in which two or more fragments or particles are ejected from the target nucleus.

Any one of these reactions may deposit energy along their paths. The particles created by these reactions are much more heavier than the original neutron or proton. This resulting heavy particles deposit higher charge capable to provoke a single event.

When the energy of a particle is enough to increase the current in a transistor node, a temporary current disturbance is generated. When this disturbance occurs in a combinational logic circuit, the effect is known as single event transient (SET). SETs may lead a system to an unexpected response whether it propagates to a memory element or a primary output (PO) of a circuit.

On the other hand, when the increase of current occurs in storage elements such as latches, flip-flops and RAMs, it may provoke the modification of the logic value stored in these elements. This modification in the value stored in the memory element as a function of a particle is called Single Event Upset (SEU).

4.2.1.1 A Single Event Model

Messenger presents in (MESSENGER, 1982) a fault model aiming at estimating the effects of single events. The model represents the effects of an α-particle striking a device as a double exponential current curve. This curve is obtained by

$$I(t) = \frac{Q}{\tau_\alpha - \tau_\beta}\left(e^{-t/\tau_\alpha} - e^{-t/\tau_\beta}\right) \tag{4.1}$$

where Q is the injected charge and may be positive or negative, τ_α is the collection time constant of the junction and τ_β is the time constant for initially establishing the ion track. τ_α and τ_β are constants and depend on several process-dependent factors.

In the double exponential, the τ_β is responsible to shape the rising of the current pulse and the τ_α shapes the fall time of the curve. The curve presents a fast rise time with smaller τ_β, as well as the fall time is faster with smaller τ_α.

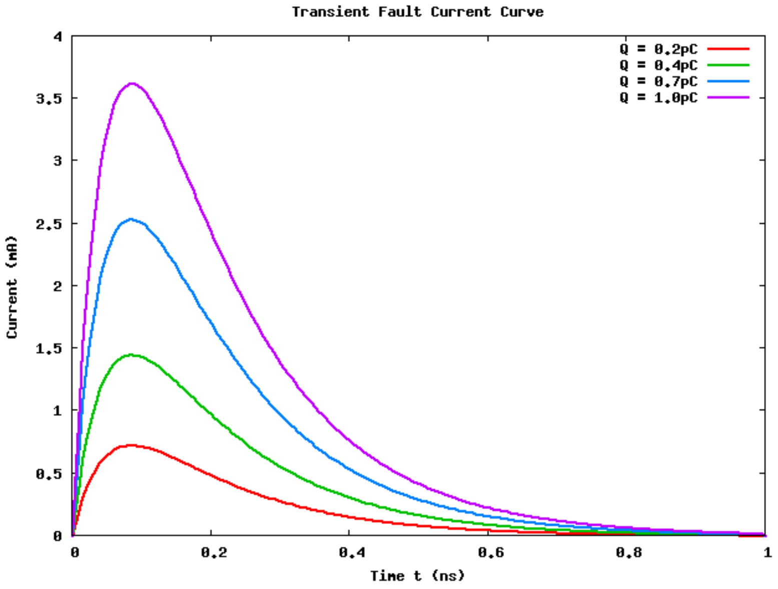


Figure 4.4: The current curve as result of a α-particle striking a device according (MESSENGER, 1982).

Figure 4.4 illustrates the current curve according the double exponential equation 4.1 for Q =0.2pC, Q =0.4pC, Q =0.7pC and Q =1.0pC. In these curves, τ_α and τ_β were defined as $1.06ns$ and $0.05ns$, respectively (DHARCHOUDHURY, A. et al, 1994).

4.2.1.2 The Worst Case Depositing Charge

The amount of charge deposited by a particle in a device is widely discussed in the literature (MAVIS; EATON, 2002; GADLAGE, M. et al., 2004; BAUMANN, 2005). These works consider the charged deposited by a particle as a function of the energy of the ionizing particle and the device charge collection depth.

A particle with 1 $MeVcm^2/mg$ deposits approximately $10fC/\mu m$ of electron-hole pairs along each micron of its tracks. In addition, the LET of very few ionizing particles is higher than 15 $MeVcm^2/mg$ (ZHOU; MOHANRAM, 2006).

A typical charge collection depth for heavy ions in a bulk silicon is approximately 2 μm. Thus an ionizing particle with an LET of 15 $MeVcm^2/mg$ deposits approximately 0.3 pC of charge in any sensitive region it passes through.

Zhou and Mohanram present in (ZHOU; MOHANRAM, 2006) a discussion about upper bounds for the deposited charges in some submicron process technologies. In other words, they suggest worst cases when choosing the worst case for a particle charge Q at ground level. For this, the linear energy transfer (LET) and the charging collection depth are taken into account. LET is a measure of the energy transferred to material as a function of an ionizing particle traversing through it.

Table 4.1: Summary of the worst case depositing charge for some technology processes for particles with an LET of 15 $MeVcm^2/mg$

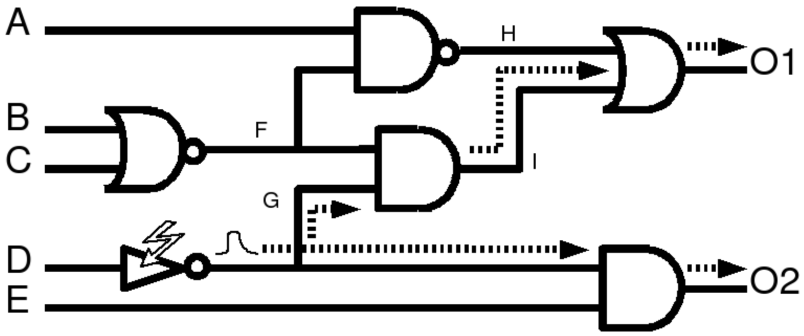
Technology process	Worst case depositing charge
180 nm	0.30 pC
130 nm	0.21 pC
100 nm	0.15 pC
70 nm	0.11 pC

Table 4.1 summarizes the worst case depositing charge for some technology processes for particles with an LET of 15 $MeVcm^2/mg$ according (ZHOU; MOHANRAM, 2006). It is important to remark that the amount of deposited charge is not linearly reduced with the technology scaling.

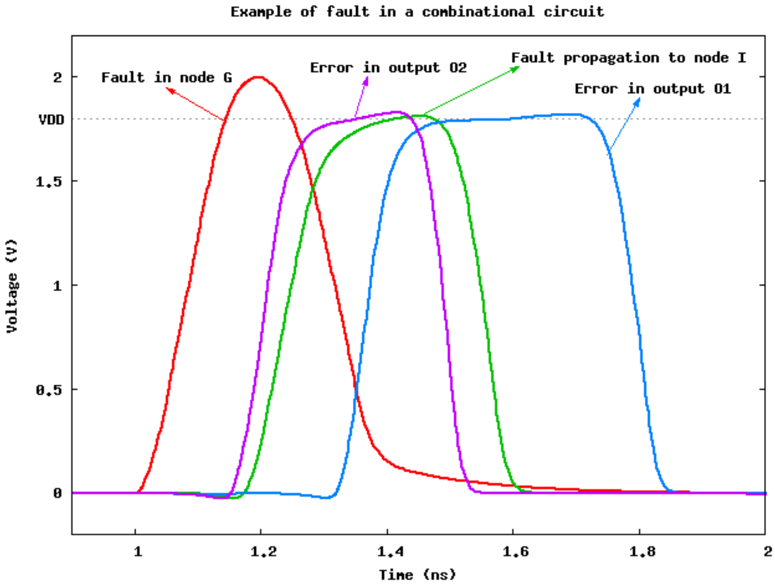
They emphasize that in smaller technologies, the charge collection efficiency decreases due to the higher channel doping density and a reduction of the active layer thickness, which reduces the depletion width and channel funneling. Considering this characteristics of smaller feature sizes, their experiments give upper bounds of 0.21 pC, 0.15 pC and 0.11 pC for the 130nm, 100nm and 70nm process technologies.

4.2.1.3 Single Event Transient (SET)

When a radiation-induced particle with enough energy hits a node of a circuit, a Single Event Transient (SET) pulse may be generated. Indeed, SETs are characterized as transient voltage fluctuations on circuit nodes. They can be caused by radiation-induced particles as previously discussed as well as by electrical noise in a noisy power supply, crosstalk noise, electromagnetic interference and radiation from lightning.



(a) Combinational block



(b) The fault propagation

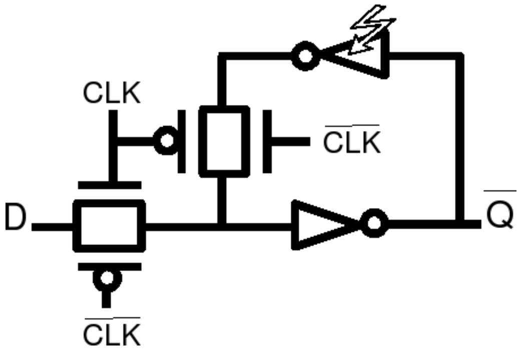
Figure 4.5: The propagation of a transient fault as results of a particle striking a node of a combinational block.

Figure 4.5 illustrates the effects of a particle striking a combinational block. Assuming the combinational block in Figure 4.5(a) with five inputs and two outputs and considering the logic values in the inputs of *A*='1', *B*='0', *C*='0',*D*='1' and *E*='1'. The Figure exemplifies an inverter being hit by a particle and the transient fault is propagated by the path *G* and *I*. Thus, the outputs *O1* and *O2* have their states changed as consequence.

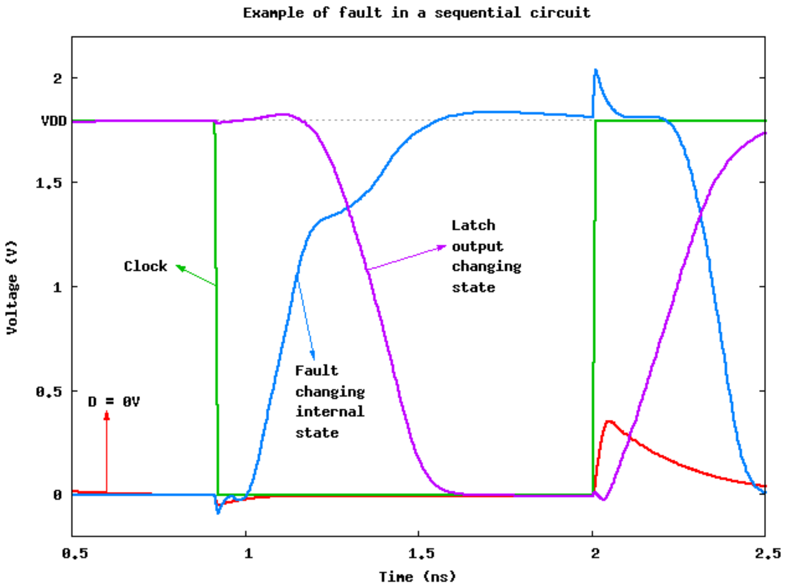
Waveforms are shown in Figure 4.5(b) where the transient pulse is propagated through the combinational circuit. The first path (node G to output $O1$ is longer than the second path (node G to output $O2$). Despite, the transient pulse is propagated through both paths provoking an error in the circuit outputs.

4.2.1.4 Single Event Upset (SEU)

A single event upset (SEU) is a change of state, or voltage pulse caused when a high-energy particle strikes a sensitive node in a micro-electronic device, such as in a micro-



(a) Classic latch



(b) The bit flip in the latch

Figure 4.6: Example of a bit flip in a classic latch.

processor, semiconductor memory, or power transistors.

SEUs represent the radiation-induced hazard which is most difficult to avoid in spaceborne applications, particularly in high density submicron CMOS ICs. Experimental results have shown that the critical charge collected at a sensitive node which is able to produce an upset decreases as the inverse square of the feature size (NICOLAIDIS; CALIN, 1997).

Figure 4.6 exemplifies a bit flip in a classic latch. The schematic of the latch is shown in Figure 4.6(a), which is composed by two inverters and two transmission gates controlled by the clock signal.

The waveform is shown in Figure 4.6(b). We consider the input D always set to “0” for this example. The single event disturbance starts at 1 ns at the internal node of the latch. At the same time, the output $\bar{Q}$ starts to change from V_{DD} to “0”. When the voltage in the internal node achieves $\frac{V_{DD}}{2}$, the inverters force the latch to change its state and the bit-flip occurs.

4.2.2 Single Hard Errors (SHE)

Single Hard Error (SHE) is as SEE, which causes a permanent change to the operation of a device. An example is a stuck bit in a memory device. In other words, we consider that a SHE occurs when the total dose of a single ion is sufficient to create a stuck bit. Further details of the characterization of this kind of error in memories is presented in (DUFOUR, C. et al., 1992; POIVEY, C. et al., 1994).

4.3 Other Radiation-Induced Effects

Single Event Latchup (SEL)

Single event Latchup (SEL) is a condition that causes loss of device functionality due to a single-event induced current state. During a SEL, the device exceeds the maximum current specified for the device. In this condition, it is mandatory that the power is removed. On the contrary, the device will be destroyed.

Single Event Burnout (SEB)

Single event burnout (SEB) is a condition that can cause permanent device destruction due to a high current state in a power transistor.

Single Event Gate Rupture (SEGR)

Single Event Gate Rupture (SEGR) is a single ion induced condition in power MOSFETs which may result in the formation of a conducting path in the gate oxide.

4.4 Conclusion

This chapter presents basic aspects about radiation-induced effects in silicon devices. Some radiation effects are introduced in order to give some details about the behavior of a device with respect to a particle hitting it. The main radiation-induced effects are the total ionizing dose (TID), the displacement damage (DD) and the Single Event Effects (SEE).

The discussion is focused mainly in the Soft Single Event Effects due to their relevance with the development of this thesis. The main principles about Single Event Transients (SET) and Single Event Upsets (SEU) are highlighted.

The worst case depositing charge is also discussed, whose upper bounds for the critical charge as a function of some submicron technology processes are given. These values for the worst case depositing charge are used as basis to the development of the experiments in this work.

5 STATE-OF-THE-ART TECHNIQUES FOR SOFT ERROR PROTECTION

5.1 Introduction

Several techniques for soft error protection have been proposed in the literature in the last years. Most techniques are based on redundant structures. This redundancy may be temporal or spatial.

Temporal redundancy consists on inserting additional logic in the design, which guarantee the evaluation of a signal in different instants of operation. If an energetic particle hits a device creating a pulse voltage, this additional logic filters the signals and guarantee the attenuation of the pulse.

Spatial redundancy is usually based on the replication. The main idea in the spatial replication is that a particle hitting one of the elements does not affects the others. Thus, the outputs of the replicated elements are compared and the filtered signal is propagated.

All these techniques are characterized by the high area overhead and important delay and power penalties. The techniques are usually very effective against soft error effects, but the consequences are usually related to the inefficiency of the circuit concerning high frequency or power consumption.

Historically, sequential elements have been concerned for single event upsets. Efficient solutions to memory elements protection are presented in (BESSOT; VELAZCO, 1994; CALIN; NICOLAIDIS; VELAZCO, 1996; ROCKETT, 1988; WHITAKER; CANARIS; LIU, 1991). However, since the transition time of the logic gates is getting shorter and clock frequencies are increasing significantly in nanometric technologies, errors in combinational logic parts are increasing and error rates will reach the same levels as in memories in the near future.

The constant decreasing size of microelectronics devices, associated with the reduction of voltage supply and the higher operating frequencies, leads to an increased vulnerability of logic circuits to soft errors (JASAREVIC; JERIN, 2005). In (MOHANRAM; TOUBA, 2003), a study projects the soft error rate in combinational logic circuits comparable to unprotected memory elements by 2011.

This chapter presents some state-of-the-art techniques to mitigate the effects of single events in sequential and combinational circuits. The advantages and drawbacks of each technique are also discussed. Besides, a temporal redundancy method is proposed to protect sequential elements against SEEs.

5.2 The Classic Techniques

Many techniques have been presented in the literature aiming at protecting combinational blocks and storage elements against Single Event Error (SEE). The techniques are usually based on redundancy of elements or sizing transistors.

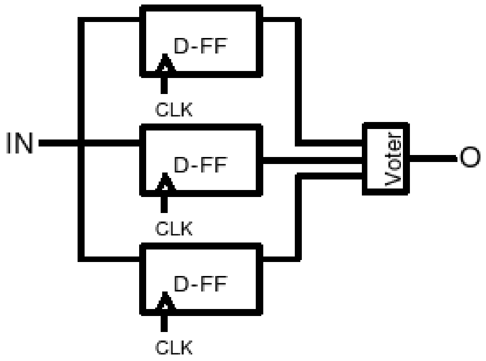
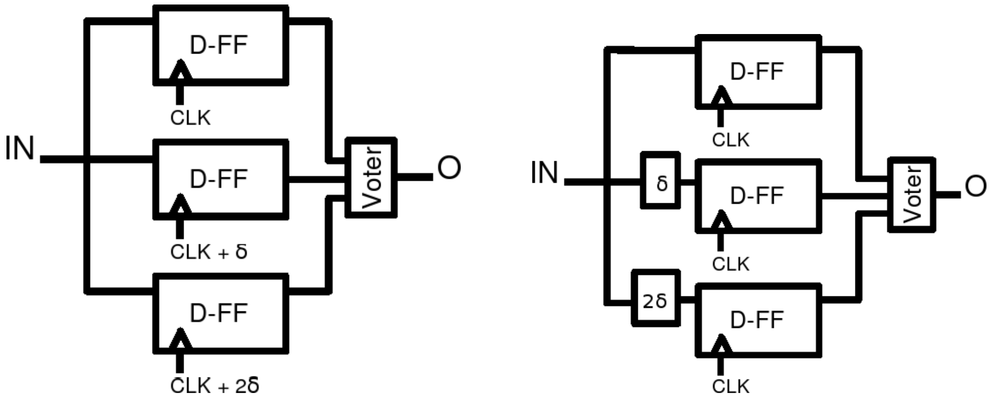


Figure 5.1: The classic TMR.

Triple Modular Redundancy (TMR) is the most common used technique to protect sequential circuits. The TMR technique consists on triplicating elements (spatial redundancy) in such way the logic value resulting from at least two elements are propagated. Figure 5.1 shows the classic TMR technique applied to D-FlipFlops. This technique can be used to prevent an error as a result of a single fault occurring inside the elements. On the other hand, when a SET occurs before the inputs (e.g. a combinational block), the transient pulse may be captured by the flipflops.



(a) TMR with delayed clock
(b) TMR with delayed inputs

Figure 5.2: Two TMR versions with delayed clock and delayed inputs to avoid transient faults coming from the combinational blocks.

A way to provide perturbation tolerance on both combinational and sequential blocks is using the TMR technique in such way we can guarantee that the fault is only propagated to one Flip-Flop. In other words, providing perturbation tolerance by inserting temporal redundancy.

The Figure 5.2 shows two implementations of the TMR technique targeting perturbation tolerance to combinational and sequential blocks. In the first implementation (Figure 5.2(a)), the clock signal of each Flipflop (CLK,$CLK + \delta$ and $CLK + 2 \times \delta$) is delayed in such a way the input signal is captured at three different moments. Thus, a transient fault at the input is captured only by one of the flipflops.

The same idea is used in Figure 5.2(b), where the same clock signal is used in the tree Flip-Flops, but the input signal is delayed by *Delay Blocks*. The time penalty of these TMR techniques over the time of a D-Flipflop is $2 \times \delta + T_{DelayVoter}$.

The TMR is usually applied to sequential elements but it can be used to combinational blocks and a whole circuit but they present a very high overhead in area (more than 200%). Furthermore, techniques that modify clock signal as shown in Figure 5.2(a) may present additional and usually unnecessary design challenges due to clock skew and clock tree distribution.

5.3 Gate Duplication Methods

Many techniques based on gate duplication have been proposed to reduce the soft error failure rate in combinational logic. The gate duplication consists of connecting inputs and outputs of a gate with an identical gate.

A partial gate duplication method for soft error failure rate reduction in combinational circuits is presented in (MOHANRAM; TOUBA, 2003). The partial gate duplication consists of selectively duplicating the most sensitive gates.

Results show that a 90% soft error failure rate reduction for energetic particles of 10 MeV is obtained with 50% area overhead. The drawback of this method is related to the area overhead to achieve a 0% sensitivity circuit.

The area overhead in this methodology presents an exponential behavior, which small area overhead is obtained with up to 80% soft error reduction rate. Soft error reduction rates higher than 80% result in area overheads close to 100%.

A SER analysis in combinational logic circuits and a partial gate duplication methodology for soft error protection is proposed in (HEIJMEN; NIEUWLAND, 2006; NIEUWLAND; JASAREVIC; JERIN, 2006). These works present a very important contribution in relation to SET analysis. The SET propagation is analyzed with respect to logical and electrical masking.

They consider that the logical and electrical masking analysis contribute in the SER of the circuit. Thus, the logical masking contributes by given the probability of a single event to be masked by the circuit logic and the electrical masking contributes to the circuit SER by analyzing the electrical attenuation of the single event. More details about the logical and electrical masking are given in the Chapter 6.

The drawback of this technique is mainly related to the resulting overhead. The duplication of a gate increases twice the occupied area, but also increases the input capacitances. These capacitances increase the delay of the connected gates. This factor causes disturbs in the delay paths of the circuit and makes it difficult the timing closure.

5.4 Gate Sizing Techniques

The gate sizing method consists on changing the cells of the circuit by cells with bigger transistors with the same logic function. A library of cells is composed by some versions of each gate. Thus, a sensitive cell may be changed by a more robust one in order to reduce the sensitivity.

Gate sizing techniques for soft error failure rate are presented in (DHILLON et al., 2004; CAZEAUX et al., 2005; ZHOU; MOHANRAM, 2006). The contribution of these works are related to the techniques to analyze the sensitivity of the circuits and the selection of the gates. The sizing of the gates is done with analytical model, which are computationally efficient.

In (ZHOU; MOHANRAM, 2006), candidate gates are selected according to the sensitivity level. The sensitive level is determined by the logical masking of each node in the circuit.

Design penalties of the gate sizing technique are positives in comparison with the duplication method due to the reduction on the area overhead. However, the reduction of penalties could be more efficient if the sizing algorithm could size pull-up and pull-down blocks of the gates.

A new sizing algorithm is proposed in the Chapter 6. This algorithm takes into account different characteristics of PMOS and NMOS transistors in order to reduce the penalties provoked by a radiation-hardened design.

5.5 Protecting Sequential Elements Through Feedback Control

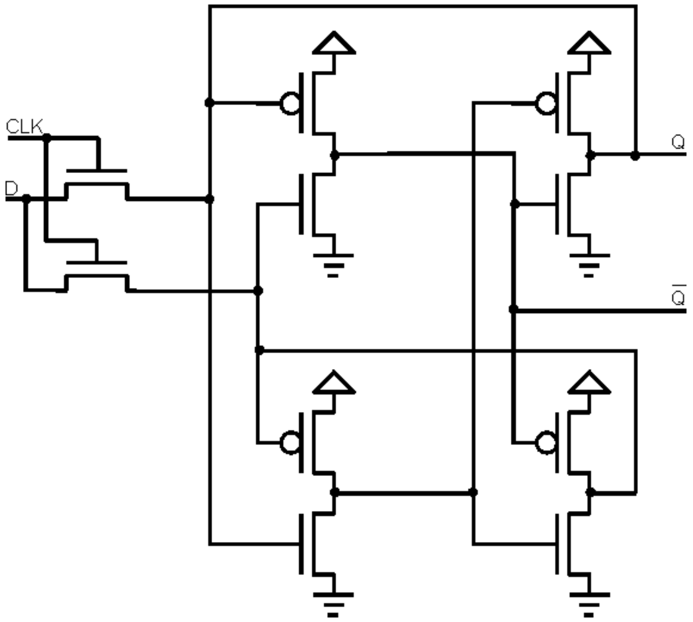


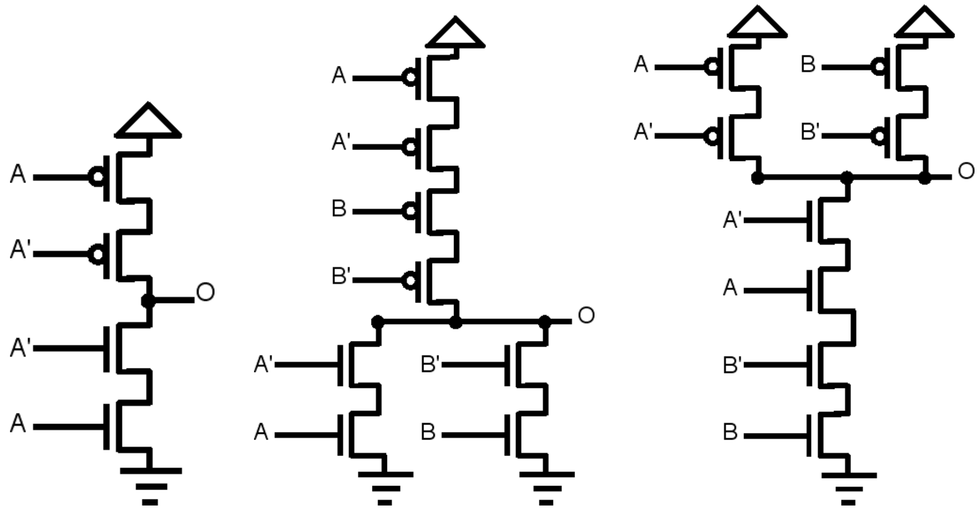
Figure 5.3: Latch using the DICE technique as proposed in (CALIN; NICOLAIDIS; VELAZCO, 1996).

An interesting technique to protect storage elements is the Dual Interlocked storage Cell (DICE) (CALIN; NICOLAIDIS; VELAZCO, 1996). The main idea of this technique relies on the principle of *dual node feedback control* in order to achieve immunity to upsets. In other words, this means that the logic of each node of the cell is controlled by two nodes. Thus, a perturbation in a node is removed after the upset due to the state-reinforcing feedback ensured by the other two nodes.

Figure 5.3 shows a latch using the DICE technique, where each inverter will change its state only if two other inverters change their state. This technique presents an important characteristic for the design of SRAM memories with upset immunity. Further details can be seen in (CALIN; NICOLAIDIS; VELAZCO, 1996).

5.6 Using Time Redundancy To Protect Sequential Elements

In (ANGHEL, 2000), it is presented a technique aiming at taking advantage from the temporal nature of transient faults, and achieve transient fault tolerance by using time redundancy. This technique should lead to a significant reduction of hardware cost compared to the TMR classic solution because the main idea is to combine self-checking design with time redundancy.



(a) Inverter (b) NOR2 (c) NAND2

Figure 5.4: INV, NOR2 and NAND2 gates using the CWSP technique proposed in (ANGHEL, 2000).

The fact that soft-errors affect the outputs of the circuit only for a short time duration can be exploited by using asynchronous sequential elements. These elements produce on the outputs a determined state for each correct input. This state corresponds to the circuit fault-free operation. In addition, the element preserves its previous output state for each erroneous input. In addition, if a transient pulse changes a fault free input into an erroneous one, the output state produced by the correct input is preserved.

A way to generate a *Code Word State Preserving* (CWSP) element is to replace each transistor of the gate by a pair of transistors connected in series and driven by duplicated inputs. In this gate, when the inputs of a pair of transistors are equal, the two transistors behave as a single transistor driven by one of the duplicated inputs.

When the inputs of one or more transistor pairs have not equal values due to a transient error, the two transistors of the pairs behave as a single transistor in off state. This situation preserves the same state at the output of the CWSP element, as the same state obtained before the transient errors drive some input into non-equal values.

Figure 5.4 shows examples of the inverter, NOR and NAND gates using this principle. In the time redundancy approach, instead of circuit duplication we can duplicate the output signal of the circuit in the time domain, by observing this signal at two different instants. One of the inputs of the CWSP element is coming directly from the combinational circuit output while the other input is delayed.

Table 5.1: Area and delay of the library cells INV, NAND and NOR in comparison with CWSP cells automatically generated with the tool presented in (LAZZARI et al., 2003).

	Area (μm^2)			**Delay** (ps)		
	Typ.	CWSP	Over.	Typ.	CWSP	Over.
INV	8.19	11.64	42%	83	160	92%
NAND	12.28	19.07	55%	98	172	75%
NOR	12.28	19.07	55%	102	260	154%

Table 5.1 presents the occupied area and propagation time of CWSP cells according to Figure 5.4 in comparison to typical cells presented in a $0.18\mu m$ standard cell library. It is shown that the area overhead is between 42% and 55% and the propagation time is between 92% and 154% applying the same output capacitance to both gates. The tool presented in (LAZZARI et al., 2003; SANTOS et al., 2003; BASTIAN et al., 2004) was used to automatically generate these cswp cells.

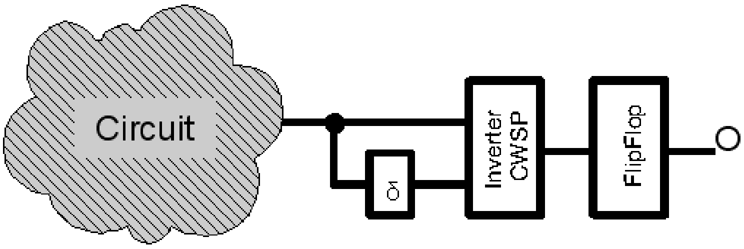


Figure 5.5: Perturbation tolerant circuit based on time redundancy.

The implementation of perturbation tolerant combinational circuits based on time redundancy is presented in Figure 5.5. The delay block must be able to degrade the signal at the input of the CWSP cell according to the time of the transient fault that we achieve to tolerate. The time penalty in this case is **Dcw + 2 × Dtr**, where Dcw is the logic transition time and Dtr is the duration of the transient pulse.

Table 5.2 presents the total area and propagation time of the new CWSP cells developed as shown in Figure 5.5. The development of CWSP cells supporting transient

Table 5.2: Total area and delay of CWSP cells.

	Trans. (ps)	**Area** (μm^2)	**Delay** (ps)
INV	250	28.8	323
	500	46.0	538
NAND	250	59.2	370
	500	91.2	559
NOR	250	59.2	352
	500	91.2	572

faults ranging from $250ps$ of $500ps$ are assumed. Thus, delay blocks were developed and inserted in the CWSP gates in order to obtain hardened cells.

We proposed in (LAZZARI; ANGHEL; REIS, 2005b) a technique targeting perturbation tolerance to both combinational and sequential logics. It uses the timing redundancy technique presented in (ANGHEL, 2000) to provide fault tolerance in circuits. To tolerate transient faults in combinational circuits as well as in the sequential elements (e.g. latches) our approach uses a modified latch where the last inverter stage of the combinational circuit is replaced by a CWSP inverter, while a delay block has been introduced in the feedback path of the latch.

Figure 5.6 shows the implementation of a latch using the CWSP logic. The technique uses a CWSP inverter and *Delay Blocks* intend to achieve fault tolerance through timing redundancy. It was proved by transient fault simulation that these CWSP d-latches have 100% transient fault coverage, assuming that the *delay blocks* have a delay greater than the time of the transient pulse.

Table 5.3: Area overhead comparison of TMR and CWSP d-flipflops.

	Area	
	μm^2	Overhead
Standard Cell	57.6	–
CWSP ($250ps$)	181.7	215%
CWSP ($500ps$)	249.6	333%
TMR ($250ps$)	206.1	258%
TMR ($500ps$)	206.1	258%

Table 5.3 presents the comparison between a classic D-FlipFlop found in a $0.18\mu m$ standard cell library and the transient robust latch proposed in (LAZZARI; ANGHEL; REIS, 2005b) (Figure 5.6) and the TMR FlipFlop. We assume that delay blocks in the TMR D-FlipFlops can be shared between all FlipFlops in the same clock domain, reducing the occupied area. Despite of that, results show that the robust CWSP FlipFlops present smaller area overhead against faults of $250ps$ in comparison to the TMR technique.

A case study was done in order to verify the penalties of the insertion of CWSP cells in a MIPS-like processor and a 8051 controller. The logic synthesis and technology mapping

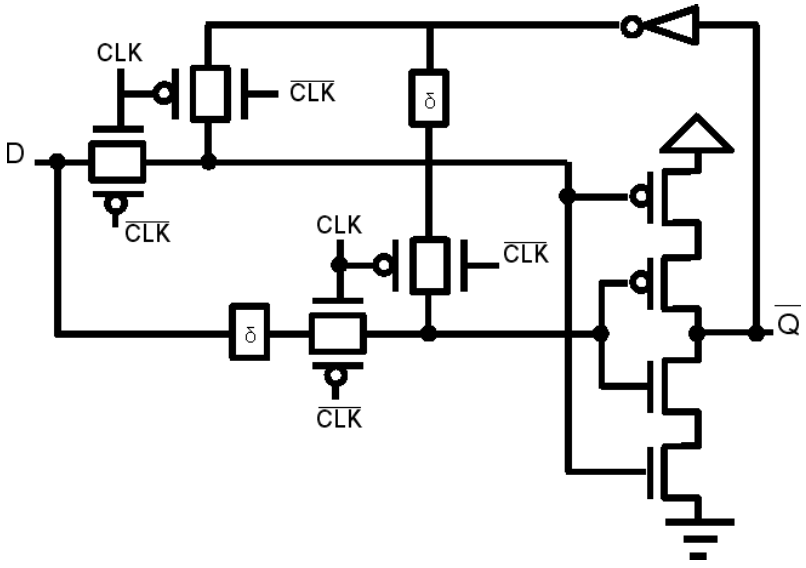


Figure 5.6: Using CWSP logic inside the latch as proposed in (LAZZARI; ANGHEL; REIS, 2005b).

were done with Synopsys Design Compiler and the whole circuit layout was generated with Cadence Silicon Ensemble. The CWSP robust latch layout was generated by using Parrot Punch tool and inserted in the system layout.

Table 5.4: CWSP and TMR techniques on microprocessors.

	MIPS				**8051**			
No. Comb. El.	11,968				5,408			
No. Flipflops	1,793				1,359			
	Area		**Frequency**		**Area**		**Frequency**	
	μm^2	Over	MHz	Penalty	μm^2	Over	MHz	Penalty
Classic	480,317	–	77.7	–	234,720	–	58.2	–
CWSP ($250ps$)	746,480	55%	75.8	2.4%	436,240	85%	57.2	1.8%
CWSP ($500ps$)	890,172	85%	72.7	6.7%	550,560	134%	55.4	5.0%
TMR ($250ps$)	808,200	68%	73.9	5.1%	491,400	109%	56.0	3.8%
TMR ($500ps$)	808,200	68%	71.2	9.0%	491,400	109%	54.5	6.8%

Table 5.4 shows the occupied area and frequency in the MIPS and 8051 architectures mapped to a $0.18\mu m$ technology. The area overhead of the TMR is constant in the implementation to cope with faults of $250ps$ and $500ps$ because we assume that the delay blocks are shared with every D-FlipFlop. We also assume that three clock lines are not a problem in the design of these microprocessors.

The results in Table 5.4 show that we can deal with the problem of transient faults in combinational and sequential parts by using CWSP robust latch with smaller area and time penalties than using TMR at latch level. To deal with fault duration of $250ps$, the

CWSP technique is always the best solution concerning area overhead and delay penalty. Hardening designs aiming at faults of $500ps$ present the best area overhead with the TMR technique but the CWSP technique always provide better time results.

In addition, the TMR technique with three clock signals can be a problem in bigger designs. Thus, additional steps as buffer insertion or transistor sizing may be necessary in the clock tree in order to guarantee the functioning of the technique, due toinherent CTS (Clock Tree Synthesis) and routing problems.

This case study shows the layout generation of time redundant cells to be used in the synthesis of integrated circuits. The importance of an automated process to generate these kind of cells is related to the need for generating hardened circuits for several applications. Besides, typical cell libraries do not present any kind of fault tolerant cells.

5.7 Conclusion

Some techniques for designing hardened circuits are reported in this Chapter. These techniques are usually based on the redundancy of elements or transistor sizing techniques.

A special attention is given to the Code Word State Preserving (CWSP) technique (ANGHEL, 2000) because of its importance to the development of a technique proposed in (LAZZARI; ANGHEL; REIS, 2005b). This technique consists on protecting sequential elements as latches and flipflops by inserting CWSP elements in their structures. A case study was done in order to validate this technique and it is presented in Section 5.6.

Table 5.5 presents an overview of the techniques presented in this chapter, where advantages and drawbacks are highlighted.

Table 5.5: Review of the techniques presented in this chapter.

Technique	**Description, advantages and drawbacks**
Classic TMR	Consists on triplicating elements. A voter propagates the state coming from two or more replicated elements. TMR works whenever the single event occurs in only one element, independently of the particle energy. Faults may be captured by the inputs. Timing penalties are insignificant but it presents high area and power overhead, mainly if the whole circuit is triplicated. (Figure 5.1)
TMR w/ delayed clock	Consists on triplicating elements with the delayed clocks CLK,$CLK + \delta$ and $CLK + 2 \times \delta$. Transient pulses with duration smaller than δ are captured by only one element. Additional complexity is inserted in the design due to the clock management. (Figure 5.2(a))
TMR w/ delayed inputs	Consists on triplicating elements with delay blocks in the inputs. Transient pulses with duration smaller than δ are captured by only one element. The technique does not insert additional complexity in the clock synthesis but increases the clock period by $2 \times \delta$. (Figure 5.2(b))
Gate Duplication	Gate duplication places two gates, whose inputs and outputs are connected. The gates are selectively duplicated according to sensitivity levels. The main drawback is related to area overhead and delay penalties. (Section 5.3)
Gate Sizing	The gate sizing technique consists on changing the cells by bigger ones. The area overhead is less significant than gate duplication but delay penalties are also important. (Section 5.4)
Feedback Control	Feedback control relies on the principle that dual nodes together are able to attenuate the single event. The area overhead is considerably smaller than the TMR and time penalties are insignificant. Faults may be captured by the inputs. (Section 5.5)
CWSP	Used in combinational blocks. The CWSP techniques replaces each transistor of the gate by a pair of transistors connected in series and driven by duplicated inputs. A block with delay δ must guarantee that a transient pulse arrives in only one of the inputs. Area overhead is insignificant, but clock period increases by δ (Section 5.6)
CWSP Sequential	This technique inserts CWSP elements inside sequential structures as flipflops and latches. The sequential element is able to filter single events with duration smaller than δ at internal blocks and coming from the combinational block. The area is increased by the insertion of two delay blocks, and the clock period increases by δ. (Section 5.6)

6 AN EFFICIENT TRANSISTOR SIZING METHODOLOGY FOR SOFT ERROR PROTECTION IN COMBINATIONAL LOGIC CIRCUITS

6.1 Introduction

A radiation-induced effects overview is given in Chapter 4. The chapter introduces the causes and effects of the radiation-induced effects and shows a way to model the behavior of current injected by a particle in the devices.

State-of-the-art methods for soft error failure rate reduction are presented in Chapter 5. These techniques are very effective on protecting circuits against single event upsets. However, radiation-hardened designs are usually inefficient concerning area, delay or power consumption. Most techniques fail in these three aspects.

A new transistor sizing method is presented in this chapter. The main characteristic of the proposed methodology is to find the smallest transistors width to attenuate SETs in the nodes of a combinational circuit.

Another important point is that pull-up and pull-down transistors are independently sized, minimizing the area overhead and the power consumption. In other words, we apply asymmetric transistor sizing to attenuate SETs with minimized area overhead. Works presented in the literature are based in symmetric models to size pull-up and pull-down blocks.

6.2 Combinational Circuits Sensitivity

The sensitivity analysis of circuits has been presented in several works (NIEUWLAND; JASAREVIC; JERIN, 2006). Most of them include the structure of the gates and layout details. The analysis of the structure of a gate consists on evaluating the propagation of a fault as a functions of transistors connections. For example, a fault in the drain node of a transistor has more probability to be propagated than a node connected to the V_{DD} or GND.

The sensitivity analysis considering the layout takes into account the probability of a particle to hit a region of the layout. For example, a big drain area has as higher probability to be hit than a smaller one.

In this work, the structure of the gate and its layout is not considered. Differently, we consider only the output node of each cell due to its higher sensitivity in comparison with the internal nodes of the gate. Layout characteristics are not taken into account in the sensitivity analysis because we consider the sensitivity of a gate after sizing becomes zero as function of a given critical charge Q_c.

It is important to remark that layout aspects are not considered only for sensitivity analysis purposes. Layout details are essential for the proposed transistor sizing methodology.

We consider the logical and electrical masking as the sensitivity of a circuit. The logical masking represents the probability of a transient pulse to be masked by the logic function of the circuit, and the electrical masking describes if a transient pulse in a node is not propagated to the primary outputs (PO) or flipflops. Thus, the sensitivity of a circuit is given by

$$S_{circuit} = \sum_{n=1}^{N} (1 - L_n) \cdot (1 - E_n) \tag{6.1}$$

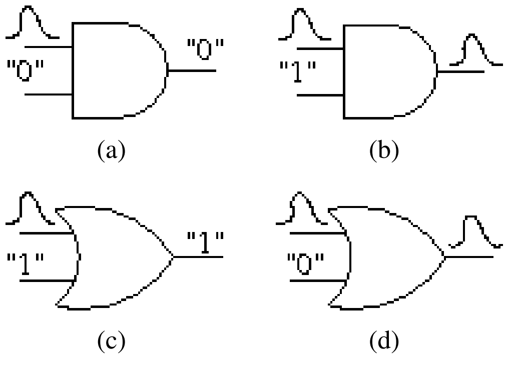
where L_n is the logical masking and E_n is the electrical masking. The logical masking L_n is a probability value. Bigger logical masking means smaller probability of a transient pulse to be detected in the circuit outputs. The electrical masking E_n is a boolean value where “0” means that the transient pulse is totally attenuated and “1” indicates that the transient can be seen in the outputs.

6.2.1 The Logical Masking

The *logical masking* occurs when a SET provoked by a particle is not propagated to a primary output (PO) due to the logic of the circuit. In other words, the pulse is masked as function of the vector applied in the primary inputs (PI) of the circuit. Controllability and observability techniques are used to define the logical masking of a node.

Controllability in combinational logic circuits denotes the ability to a state be set in a node. Observability is a measure of how well a state in a internal node can be known at the primary outputs (PO).

The controllability of the gate output node is obtained by the logic function of the gates as shown in Table 6.1 (JASAREVIC; JERIN, 2005). Thus, the propagation of the controllability probability is done for the entire circuit, from the PIs through each gate until the POs are reached.



(a) (b)

(c) (d)

Figure 6.1: The logical masking.

Table 6.1: Probability of a node as a function of the gate equation (JASAREVIC; JERIN, 2005).

Logic Function	**Resulting Probability**
AND	$P_Z(1) = P_a(1) * P_b(1)$
NAND	$P_Z(1) = 1 - P_a(1) * P_b(1)$
OR	$P_Z(1) = 1 - (1 - P_a(1)) * (1 - P_b(1))$
NOR	$P_Z(1) = (1 - P_a(1)) * (1 - P_b(1))$
XOR	$P_Z(1) = P_a(1) + P_b(1) - 2 * P_a(1) * P_b(1)$
XNOR	$P_Z(1) = 1 - P_a(1) - P_b(1) + 2 * P_a(1) * P_b(1)$
BUF	$P_Z(1) = P_a(1)$
INV	$P_Z(1) = 1 - P_a(1)$

Figure 6.1 illustrates the logical masking in a gate. A pulse in one of the gate inputs is propagated through the gate only if a non-controlling value is applied at the other input. Figure 6.1(a) shows the logical masking in the AND gate as a function of the controlling logic value “0” at the input. Otherwise, the logical masking does not happen if a non-controlling value is applied (Figure 6.1(b)).

In an OR gate, the same situation is considered, where the pulse propagates through the gate only if the non-controlling value is applied to the other input. Figure 6.1(c) shows the logical masking as function of a controlling value and Figure 6.1(d) shows the case where there is no logical masking.

6.2.2 The Electrical Masking

Electrical masking can be defined as the electrical attenuation of a pulse in a node by the gates in a path to the point that the SET does not affect the result of a circuit.

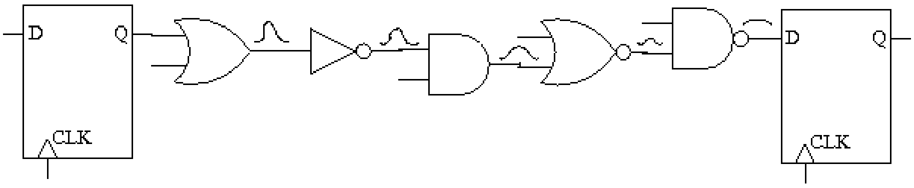


Figure 6.2: The electrical masking.

Figure 6.2 shows an example of SET degradation. This degradation is the base of the electrical masking, where the pulse is degraded as a function of the electrical characteristics of the gates in the path. The pulse can be captured by the memory element if it is not enough degraded. More details about electrical masking and SET propagation are given in Section 6.3.3.

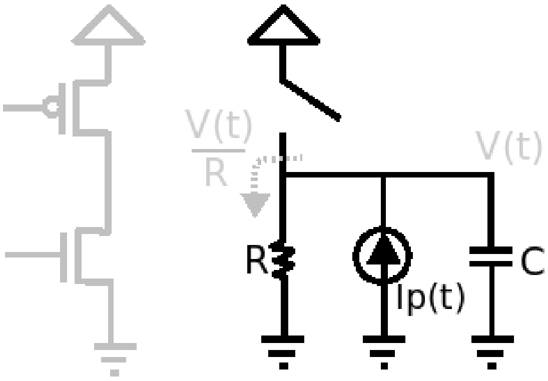


Figure 6.3: Equivalent circuit for calculating circuit response to an energetic particle hit.

6.3 An Analytical Model for Single Event Transients

The sensitivity model used in our transistor sizing strategy was proposed in (WIRTH; VIEIRA; KASTENSMIDT, 2007). The model is based on two electrical device parameters. The effective loading capacitance C lumped onto the output node of a gate g and the effective resistance R of the "ON" transistors of this gate.

For modelling purposes, circuit response for the energy particle is modeled as the network depicted in Figure 6.3, and may be represented by

$$C\frac{dV(t)}{dt} + \frac{V(t)}{R} = I_p(t) \tag{6.2}$$

where the term $\frac{V(t)}{R}$ is related to the discharging current in the transistor, represented by the resistor R. $I_p(t)$ represents the current caused by the particle hitting the device and the last term $C\frac{dV(t)}{dt}$ represents the current in the capacitor C.

The model derivation has a strong relation with the electrical devices behavior and allows the evaluation of the critical charge Q_c needed to induce a SET in a node, and the transient pulse duration, as well.

6.3.1 Modeling Resistances and Capacitances

The use of linear resistors to model transistor paths is a widely known method (WESTE; ESRAGHIAN, 1993). Thus, the effective resistance R can be analytically determined by

$$R = \frac{1}{\mu_0 C_{ox}(\frac{W}{L})(V_{gs} - V_{th})} \tag{6.3}$$

where μ_0 is the mobility of the transistor channel. C_{ox} is the oxide capacitance, which is given by $\frac{\varepsilon_0 \varepsilon_{SiO_2}}{t_{ox}}$. ε_0 is the dielectric constant, ε_{SiO_2} is the oxide relative dielectric constant and t_{ox} is the gate oxide thickness. V_{gs} is the gate-source voltage and V_{th} is the threshold voltage.

All these parameters are constants related with the technology process, except the $(\frac{W}{L})$ ratio that represents the transistor dimensions. Based on this aspect ratio, we are able

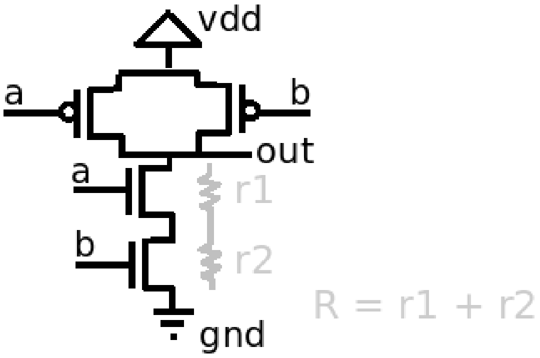


Figure 6.4: A transistor modeled as a resistance.

Table 6.2: Approximation of intrinsic MOS gate capacitance.

Parameter	Off	Non-saturated	Saturated
C_{gb}	$C_{ox}A$	0	0
C_{gs}	0	$\frac{1}{2}C_{ox}A$	$\frac{2}{3}C_{ox}A$
C_{gd}	$C_{ox}A$	$\frac{1}{2}C_{ox}A$	0
$C_g = C_{gb} + C_{gs} + C_{gd}$	$C_{ox}A$	$C_{ox}A$	$\frac{2}{3}C_{ox}A$

to explain the relation between the transistor width and the resistance. The smaller the transistor width, the higher the resistance.

Figure 6.4 illustrates two stacked transistors modeled as resistors. Assuming NMOS transistors are “ON” in the NAND gate of the example due to input signals $a =$ “1” and $b =$ “1”. The effective resistance R is given by the sum of the resistances r_1 and r_2.

The effective capacitances C is defined by the sum of three capacitances connected to the output node.

$$C = C_{diffusion_{g1}} + C_{connection} + C_{gate_{g2}} \tag{6.4}$$

where $C_{diffusion_{g1}}$ is the sum of all PN junction capacitances of the driving gate. $C_{connection}$ is the wiring and parasitic capacitances, and $C_{gate_{g2}}$ is the gate capacitance of all transistors connected to the output node.

The $C_{diffusion_{g1}}$ is given by

$$C_{diffusion_{g1}} = \sum_{d}^{D} \times C_{ja}A_d + C_{jp} \times P_d \tag{6.5}$$

where C_{ja} is the junction capacitance per μ^2, A_d is the diffusion area, C_{jp} is the periphery capacitance per μ and P_d is the diffusion perimeter.

The third term of the capacitance C is the gate capacitance. Thus, $C_{gate_{g2}}$ is defined according to the region the gate $g2$ is operating.

Table 6.2 presents the gate capacitance according to the region of operation. Based on this information, the gate capacitance is defined by

$$C_{gate_{g2}} = \sum_{g_{off}} C_{ox} A_g + \sum_{g_{on}} \frac{2}{3} C_{ox} A_g \tag{6.6}$$

These analytical equations allow to model the behavior of the transient pulse as a function of the electrical characteristics of the devices.

6.3.2 The Single Event Transients Model

The single event model uses the double exponential equation discussed in Section 4.2.1.1. Important characteristics about the transient pulse can be obtained by (4.1). Models presented in (WIRTH; VIEIRA; KASTENSMIDT, 2007) are derivations of the double exponential to obtain the peak time t_{peak} and the voltage peak V_{peak}.

It is important to remark that the τ_β is considered to be much smaller than τ_α ($\tau_\alpha \gg \tau_\beta$) in the formulations. In other words, the model assumes a very fast rise time to the double exponential.

This differential equation (6.2) can be solved in order to obtain the voltage $V(t)$ at the struck node. Thus, $V(t)$ is given by

$$V(t) = \frac{I_0 \tau_\alpha R}{\tau_\alpha - RC} \left(e^{\frac{-t}{\tau_\alpha}} - e^{\frac{-t}{RC}}\right) \tag{6.7}$$

The time t_{peak} at which the node voltage reaches its maximum value can be evaluated by

$$t_{peak} = \frac{ln\left(\frac{\tau_\alpha}{RC}\right) \tau_\alpha RC}{\tau_\alpha - RC} \tag{6.8}$$

and, inserting (6.7) into (6.8) leads to the peak transient voltage V_{peak} reached at the struck node.

$$V_{peak} = \frac{I_0 \tau_\alpha R}{\tau_\alpha - RC} \left(\left(\frac{\tau_\alpha}{RC}\right)^{\frac{RC}{RC - \tau_\alpha}} - \left(\frac{\tau_\alpha}{RC}\right)^{\frac{\tau_\alpha}{RC - \tau_\alpha}} \right) \tag{6.9}$$

where R is the effective resistance of the pull-up path (if PMOS transistors are "ON") or the effective resistance of the pull-down path (if NMOS transistors are "ON") and C is the effective capacitance loading lumped onto the output node.

The critical charge Q_c can be derived from (6.9) once the V_{peak} of a transient pulse is known. Thus, the critical charge Q_c is given by

$$Q_c = \frac{V_{peak}(\tau_\alpha - RC)}{R\left(\left(\frac{\tau_\alpha}{RC}\right)^{\frac{RC}{RC - \tau_\alpha}} - \left(\frac{\tau_\alpha}{RC}\right)^{\frac{\tau_\alpha}{RC - \tau_\alpha}} \right)} \tag{6.10}$$

The voltage at the struck node shows a double exponential behavior in which the transient voltage V_{peak} is reached at time t_{peak}. The voltage starts to decrease exponentially after t_{peak}.

$$\tau_n = t_{peak} - RCln\left(\frac{\frac{1}{2}VDD}{V_{peak}}\right) - \tau_\alpha ln\left(\frac{\frac{1}{2}VDD}{V_{peak}}\right) \tag{6.11}$$

Equation (6.11) shows the transient pulse duration τ_n, where the second term corresponds to the analytical solution if RC time is much greater than τ_α and the last term corresponds to the analytical solution if τ_α time is much greater tan RC.

6.3.3 Single Event Transient Propagation

The analysis of the transient pulse propagation shows that the pulse degradation is directly influenced by the propagation delay τ_g. In other words, larger τ_g leads to greater degradation of the transient pulse.

Wirth *et al* proposed a pulse degradation model based on curve fitting (WIRTH et al., 2007). The model considers a k parameter equal to the minimum ratio τ_n/τ_g needed to propagate a SET to the next stage in a circuit path.

The transient pulse degradation is given by

$$\tau_{n+1} = \begin{cases} 0 & \text{if (} \tau_n \leq k\tau_g \text{),} \\ (k+1)\tau_g(1 - e^{(k-(\tau_n/\tau_g))}) & \text{if (} (k+1)\tau_g < \tau_n \leq (k+3)\tau_g)\text{),} \\ \frac{\tau_n^2 - \tau_g^2}{\tau_n} & \text{if (} (k+1)\tau_g < \tau_n \leq (k+3)\tau_g \text{),} \\ \tau_n & \text{if (} \tau_n > (k+3)\tau_g \text{).} \end{cases} \tag{6.12}$$

For an input transient with small duration, the output voltage peak does not reach $\frac{1}{2}VDD$ and complete attenuation must be considered. Thus, the first case models situations where the transient pulse is totally suppressed.

Second and third cases are related to a partial degradation in the transient pulse according to the relation with the SET duration τ_n and the gate delay τ_g. The fourth degradation case consists on situations where the pulse is not degraded from a stage to another or it can be neglected.

These four degradations cases are the basis to the sizing algorithm because of its propagation properties. These properties can be useful also to obtain the maximum acceptable transient pulse duration in a node.

We consider maximum acceptable transient pulse in a node the maximum SET duration that is attenuated before the primary outputs. In other words, the maximum acceptable SET duration is a pulse in which is not propagated to any PO.

One important remark is that a transient pulse does not need to be attenuated in the net n (except in cases where gates are connected to outputs). The SET may be attenuated through the gates in the whole path between the node n and the primary outputs. The complete attenuation of a SET in a net results in unnecessary oversized transistors.

6.4 The Transistor Sizing Strategy

The transistor sizing strategy proposed in this thesis consists of finding the smallest transistor width of each circuit gate for SET attenuation. The formulations previously discussed are the basis to the sizing algorithm.

The proposed transistor sizing strategy is presented in Algorithm 8. First lines (2-6) define the cicuit sensitivity as shown in (6.1). The transistor sizing strategy starts at line 8, where every node n of the circuit is visited in order to find the minimum transistor width

Algorithm 8 : The transistor sizing for SET attenuation.

```
Require: Set of gates $G$, Set of Nets $N$, Set of outputs $O$, Maximum sensitivity $M$, Max
    critical charge $Q_c$, Desired circuit sensitivity $S_{desired}$
Ensure: Set of gates with sized transistors $G_{new}$
 1: $G_{new} \Leftarrow \emptyset$
 2: for all $n \in N$ do
 3:     $L_n \Leftarrow$ calculateLogicalMasking( $n$ );
 4:     $E_n \Leftarrow$ calculateElectricalMasking( $n$, $Q_c$ );
 5:     $S_n \Leftarrow (1 - L_n) \cdot (1 - E_n)$
 6: end for
 7: $V \Leftarrow O$        {Nets to visit, starting from the outputs.}
 8: while $V \neq \emptyset$ do
 9:     for all $n \in V$ do
10:         $g \Leftarrow$ getFaninGateConnectedToNet( $n$ );
11:         if $S_n > M$ then
12:             $\tau_n \Leftarrow$ getMaximumSET( $n$, $g$ );
13:             $g_{new} \Leftarrow$ sizeTransistors( $s$, $g$, $\tau_n$ );
14:             $G_{new} \Leftarrow G \bigcup \{g_{new}\} \setminus \{g\}$
15:         end if
16:         $I \Leftarrow$ getGateInputs( $g$ );
17:         $V \Leftarrow V \bigcup I \setminus \{n\}$
18:     end for
19: end while
```

to each gate g connected to this node. It is important to note that only nodes with the sensitivity bigger than the maximum defined sensitivity M are sized (line 11).

Function `getMaximumSET(` n, g `)` (line 12) finds the maximum pulse duration τ_n in the node n that is suppressed before the primary outputs. The transistor sizing algorithm to a gate g is function of this SET duration τ_n.

Function `sizeTransistors(` s, g, τ_n `)` (line 13) continuously increases the transistors width until the SET in the node n is smaller than τ_n. When this situation is reached, we consider the transistors of the gate g are sized as expected to the charge Q_c.

The implementation of these functions are discussed in details in Section 6.5.

Other lines of the strategy shown in Algorithm 8 give some idea about the navigation in the nets. The algorithm evaluates every node of the combinational logic, from the primary outputs (PO) to the primary inputs (PI). This is done because the delay of the gates is changed after sizing. When transistors of a gate are sized, the delay usually becomes smaller and a transient pulse propagates with a smaller degradation.

Erroneous interpretation concerning the SET propagation must happen if the transient pulse is evaluated before the sizing of the gates in the path to the POs. Thus, when the SET is evaluated in a node n, we guarantee that every gate in the path between this node n and the POs were already sized.

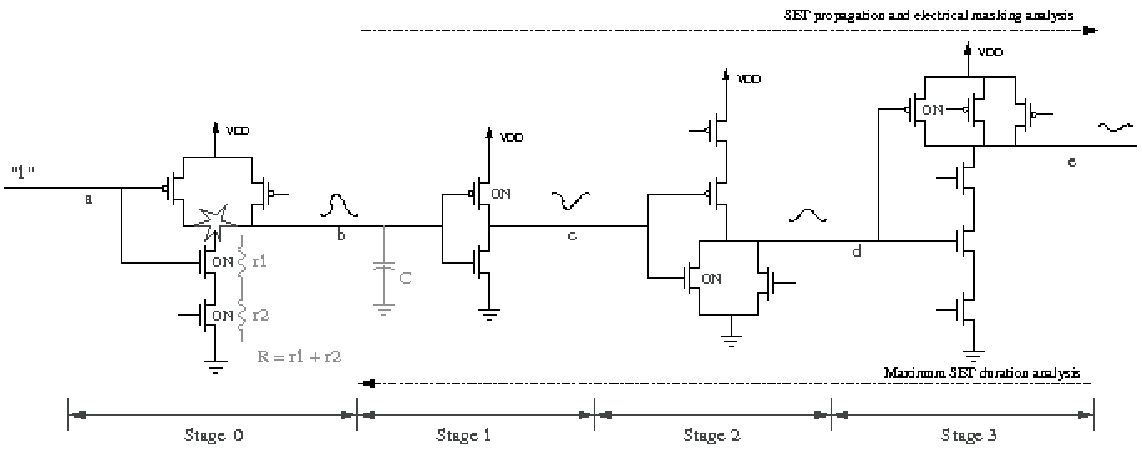

SET propagation and electrical masking analysis
VDD
"1"
a
ON
r1
ON
r2
R = r1 + r2
b
C
VDD
ON
c
VDD
ON
d
VDD
ON
e
Maximum SET duration analysis
Stage 0
Stage 1
Stage 2
Stage 3

Figure 6.5: A transient pulse propagation example.

6.5 The Transistor Sizing Model

The transistor sizing technique proposed in this thesis is basically separated in three steps. These steps are related to the sensitivity of a circuit node as a function of a particle hitting the circuit and the minimum transistor width needed to attenuate a SET (electrical masking).

The first step is the sensitivity analysis, which is represented in lines 2 to 6 (Algorithm 8). The sensitivity of a node n is given by the logical and electrical masking.

Assume the example in Figure 6.5 where a particle with charge Q hits the output of the NAND gate at the *stage 0*. The transient pulse is propagated from the net b through each stage to the circuit output. The transient pulse duration at the struck node is defined by equations described in Section 6.3.2 and is a function of the resistance R, the capacitance C, the charge Q and some technology process constants.

The degradation of a SET depends on the delay of each gate in the path as explained in Section 6.3.3. The electrical masking is basically the analysis of the SET degradation through the path. A node is assumed to be electrically masked if the transient pulse is suppressed before the outputs.

For this reason, function `getMaximumSET(` n, g `)` (line 18) finds the maximum SET duration τ_n for a net n that is attenuated just before the primary outputs. Assuming SET duration at the primary outputs $\tau_{out} = 0$, equations from Section 6.3.3 were modified to find an acceptable SET duration in the net n. In other words, we derived those equations to obtain the SET duration τ_n at the inputs of each gate as a function of the SET duration τ_{n+1} at its output.

The transient pulse propagation equations in Section 6.3.3 consider four situations where the SET duration may be totally degraded, partially degraded or propagated.

Cases where the pulse is totally attenuated or propagated without any degradation are shown in first and last cases of (6.12). Partial degradation in the transient pulse is presented in the other equations. These equations were derived as follows.

$$\tau_n = \tau_g \left(k - ln \left(1 - \frac{\tau_{n+1}}{\tau_g(k+1)} \right) \right) \tag{6.13}$$

Situations where τ_n is bigger than $k\tau_g$ and smaller than $(k+1)\tau_g$ are treated by (6.13).

$$\tau_n = \frac{\tau_{n+1} + \sqrt{\tau_{n+1}^2 + 4\tau_g^2}}{2} \tag{6.14}$$

Propagation cases where τ_n is bigger than $(k+1)\tau_g$ and smaller than $(k+3)\tau_g$ are treated by (6.14).

Function `sizeTransistors(` s, g, τ_n `)` finds the smallest transistors width for a gate g according to the transient pulse duration τ_n. In Figure 6.5, NMOS transistors of the first gate are "ON" and these transistors are the main responsible for the SET duration. Thus, only the NMOS transistors are sized in order to reduce the resistances $r1$ and $r2$ (the capacitance is indirectly increased as result of the diffusion areas).

The transistor sizing is modeled based on equations presented in Section 6.3. The algorithm consists on applying the *bisection method* (WEISSTEIN, 2007b) to find the width of each pull-up and pull-down transistor of the gate.

6.6 Results

Table 6.3: The proposed transistor sizing to single event transient attenuation. Results show the area, timing and average power overhead for symmetric and asymmetric sizing techniques for particles with charge $Q = 0.3pC$ with final sensitivity of 50%.

Combinational Circuit	**Sizing Methodology**	**Overhead**		
		Area (%)	**Power (%)**	**Timing (%)**
C432	Symmetric	47.4	63.8	0.0
	Asymmetric	35.5	50.7	2.0
C880	Symmetric	88.0	72.4	0.0
	Asymmetric	69.2	51.6	0.0
C1355	Symmetric	62.4	38.6	16.0
	Asymmetric	50.6	29.5	15.8
C1908	Symmetric	47.0	35.5	12.0
	Asymmetric	37.0	29.0	8.8
Average overhead	Symmetric	61.2	52.7	7.0
	Asymmetric	48.0	40.2	6.65

Table 6.3 and 6.4 show some results obtained by the proposed transistor sizing strategy. Results include a comparison between symmetric and asymmetric sizing methodologies for a $180nm$ technology process (ZHAO; CAO, 2007). The transient pulse propagation parameter k was defined by hspice simulations as 0.8 for this technology. The transistor sizing was done aiming at reducing the sensitivity to a 50% sensitivity (Table 6.3) and 0% (Table 6.4).

Table 6.4: The proposed transistor sizing to single event transient attenuation. Results show the area, timing and average power overhead for symmetric and asymmetric sizing techniques for particles with charge $Q = 0.3pC$ with final sensitivity of 0%.

Combinational Circuit	**Sizing Methodology**	**Overhead**		
		Area (%)	**Power (%)**	**Timing (%)**
C432	Symmetric	69.8	105	1.2
	Asymnetric	50.0	59.7	0.0
C880	Symmetric	115.3	88.7	12.3
	Asymmetric	86.9	59.1	13.2
C1355	Symmetric	80.0	61.6	24.8
	Asymmetric	58.6	37.2	17.1
C1908	Symmetric	69.2	20.89	13.0
	Asymmetric	49.2	17.4	10.16
Average overhead	Symmetric	83.5	69.0	12.82
	Asymmetric	61.1	43.3	10.11

As discussed in Section 4.2.1.2, a study presented in (ZHOU; MOHANRAM, 2006) shows that the deposited charge of very few particles is higher than $0.3pC$ for $180nm$ technologies at the atmosphere. We use this value in our experiments by considering as the worst case deposited charge.

The first important point shown by these results is the small overhead presented by the proposed methodology. The worst case was a 87% area overhead for complete protection (0% sensitivity) against particles with charge $Q = 0.3pC$. Results show an average 83% area overhead for the symmetric sizing and 61% for the asymmetric sizing. Power consumption presents 70% average overhead for the symmetric sizing against 43% for the asymmetric. Results show small timing penalties of 10% for the circuit with 0% sensitivity.

The asymmetric transistor sizing resulted in smaller area, power consumption and timing in comparison with the symmetric sizing. Despite of the penalties when designing radiation hardened circuits, results show the asymmetric sizing efficacy.

Table 6.5: A comparison among TMR, CWSP and the proposed sizing techniques. Results show area overhead and timing penalties to protect some circuits against particles with charge $Q = 0.3pC$ with final sensitivity of 0%.

Circuit	**TMR**		**CWSP**		**Sizing**	
	Area (%)	**Timing (%)**	**Area (%)**	**Timing (%)**	**Area (%)**	**Timing (%)**
C432	209.8	0.0	23.7	105.9	50.0	0.0
C880	213.8	0.0	44.6	83.4	86.9	13.2
C1355	222.3	0.0	36.6	106.1	58.6	17.1
C1908	219.2	0.0	36.5	64.9	49.2	10.1
Average	216.2	0.0	35.3	90.0	61.1	10.1

Table 6.5 shows a comparison among TMR, CWSP and the proposed sizing technique. It is possible to verify the high area overhead of the TMR technique. On the other hand, timing penalty is insignificant for the TMR due to the inclusion of only a voter in the critical path.

The CWSP technique presents small area overhead but timing penalties are very high due to the insertion of delay blocks. For particles with critical charge $Q_c = 0.3pC$, the transient duration is around $500ps$. A CWSP element with a delay bigger than $500ps$ is necessary to filter the transient.

For the presented combinational circuits, delay varies between $400ps$ and $700ps$. For this reason, the additional delay needed to attenuate the transient significantly increases the timing of the circuits. The area overhead of these combinational circuits is a function of the number of POs.

The proposed sizing technique presents better results in comparison with the TMR and CWSP techniques. The asymmetric sizing technique guarantees smaller area overhead due to selective transistor sizing as well as smaller timing penalties because new elements are not inserted in the critical paths.

6.7 Conclusions

A new transistor sizing algorithm aiming at protecting combinational logic circuit against single event transients is presented in this Chapter. The sensitivity of the circuit is analyzed by taking into account the logical and electrical masking.

The proposed technique consists on sizing only transistors directly related to the SET attenuation. It is known that transistors PMOS and NMOS have different characteristics in relation to mobility, impurities concentration and, as consequence, the delay. For a given particle with charge Q, PMOS and NMOS transistors present different attenuation characteristics. Thus, the model considers independently pull-up and pull-down blocks.

Besides, the model takes into account propagation characteristics in which the degradation of the transient pulse is considered in order to reduced sizing penalties.

The importance of this method is presented in the results. Results show smaller area, timing and power consumption overhead in comparison with a symmetrical methodology. The reduced timing penalties presented by the sizing methodology allows the development of high frequency circuits, with low overhead concerning area and power.

Figure 6.6 summarizes TMR, CWSP and the proposed sizing techniques. Penalties of the proposed transistor sizing technique are grouped in the bottom-left corner. This characteristic highlights the efficiency of the proposed technique. The TMR technique fails because of the huge area overhead, while the CWSP technique presents big timing penalties.

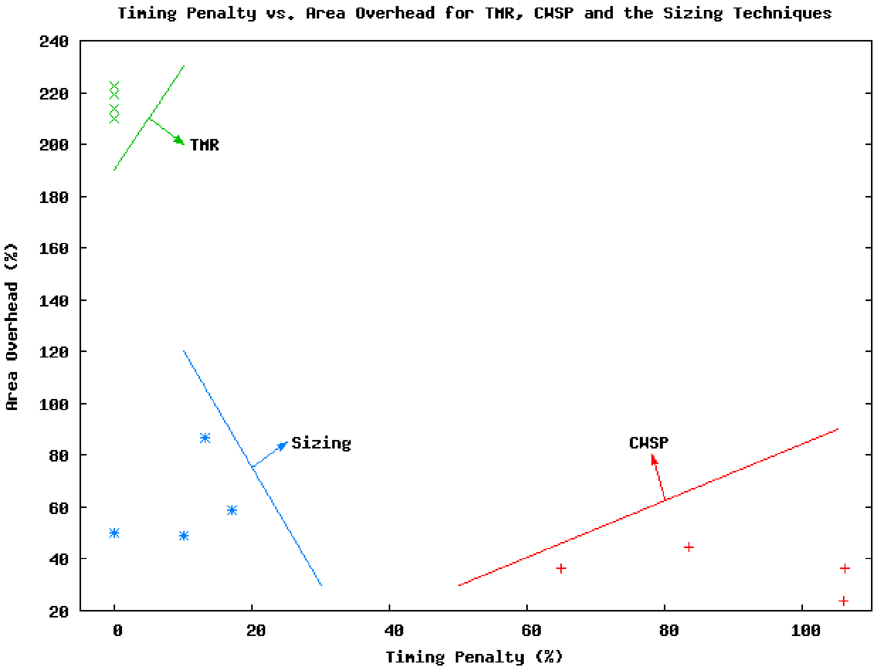


Figure 6.6: Timing penalty versus area overhead for TMR, CWSP and the proposed sizing technique.

7 CONCLUSION

The contributions of this thesis are basically divided in two major parts. The first is related to the development of a new methodology able to generate optimized circuits concerning timing and power consumption. The transistor level design flow, as it is called, optimizes every single gate of a circuit according to the capacitances in which it is involved.

The advent of the deep submicron technologies has included a myriad of new challenges in the design of circuits. The geometries are shrinking, power supplies are getting lower and the logic density is reaching a very high rate. In addition, the increased number of metal layers, associated with these deep submicron characteristics, are shifting the design paradigm from circuits with the delay dominated by the logic to circuits with the delay dominated by interconnections.

Process variability and the huge variance in the capacitance per unit length emphasize the need of a transistor level optimization because it is practically impossible to predict interconnects before the layout phase.

The proposed transistor level design flow is based on academic and commercial tools. Commercial tools include logic synthesis, placement & routing, and switch level static timing & power consumption analysis. Academic tools were developed to cope with the gaps between the conventional standard cell and transistor level methodologies.

Basically, the transistor level design flow presents four differences in comparison with the standard cell methodology:

1. *Library generation:* The layout of cells is not generated at library generation time. This allows significantly to increase the number of logic functions.
2. *Transistor level optimization:* Transistor level optimization allows to find optimized transistors width according to the capacitances involved with the gate.
3. *Layout generation:* The layout generation is performed after the transistor optimization to cope with a very large range of possibilities concerning transistors width.
4. *A feedback flow:* A flow with feedback allows to take into account the extracted capacitances during the transistor optimization process.

A power leakage reduction method is also proposed in which a gate length biasing technique is applied in order to reduce the static current. The gate length biasing technique consists on adjusting the length of transistors aiming at reducing the current flowing through it.

All these features allow to mitigate the timing closure problem. Results show that this methodology is very promising. Comparisons between the transistor level methodology and the standard cell approach show interesting results where the proposed methodology presents around 11% of delay improvement and more than 30% power saving.

The second contribution of this thesis concerns the application of the transistor level design flow in the protection of integrated circuits against single event effects (SEE). The main aspect concerning the transistor level design flow is the possibility to develop a new transistor sizing methodology targeting radiation-hardened combinational circuits.

Technology scaling also has effects in integrated circuits concerning functional failure due to SEEs. Gate length reduction and low supply voltages make circuits sensitive to energetic particles that were worthless in older technologies.

Two contributions are presented in respect to the protection of circuit against SEE. The first is related with the insertion of timing redundancy in sequential elements in order to cope with single event. The main idea consists on applying the concepts proposed by Anghel in (ANGHEL, 2000) about the Code Word State Preserving (CWSP) technique in latches and flipflops. The technique was applied to some microprocessors and results show a worst case of 85% area overhead and 7% timing penalty.

The second contribution related with single event effects protection involves a new transistor sizing methodology. The sensitivity of combinational circuits are defined by logical and electrical masking analysis. The sensitivity of each node allows to size each individual gate as a function of the particle charge.

The logical masking gives the probability of a transient pulse to be masked by the circuit logic, which is computed by controllability and observability techniques. The electrical masking describes whether a transient pulse in a node is not propagated to the primary outputs due to electrical attenuation.

The sizing method is based on an analytical model of transient pulse detection presented in (WIRTH; VIEIRA; KASTENSMIDT, 2007; WIRTH et al., 2007). The model is based on electrical device parameters. A propagation method is used, where the gate delay and the transient pulse duration are evaluated. The propagation model is very important because it assumes that the transient pulse must be attenuated during the whole path and not in the struck node. Thus, oversized transistors are avoided. The model also considers independently pull-up and pull-down blocks. Only transistors directly related to the SET attenuation are sized.

The proposed sizing technique presented small area, timing and power consumption overhead in comparison with the TMR and CWSP techniques, allowing the development of high frequency circuits, with low area and power overhead. Results show an average 61% area overhead, 43% power consumption increase and 10% delay penalty.

REFERENCES

ALPERT, C. J.; HU, T. C.; HUANG, J. H.; KAHNG, A. B.; KARGER, D. Prim-Dijkstra Tradeoffs for Improved Performance-Driven Routing Tree Design. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, New York, v.14, p.890–896, 1995.

ANDREW, C. **VLSI Datapath Choices**: cell-based versus fullcustom. 1998. Master Thesis — Massachusetts Institute of Technology, Massachusetts.

ANGHEL, L. **Les Limites Technologiques du Silicium et Tolerance aux Fautes**. 2000. Ph. D. Thesis — Institut Polytechnique de Grenoble, France.

ASKAR, S.; CIESIELSKI, M. Analytical approach to custom datapath design. In: IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, 1999. **Digest of Technical Papers...** [S.l.: s.n.], 1999. p.98–101.

BABIGHIAN, P. et al. Post-layout leakage power minimization based on distributed sleep transistor insertion. In: LOW POWER ELECTRONICS AND DESIGN, ISLPED, 2004. **Proceedings...** New York: ACM Press, 2004. p.138–143.

BAHUMAN, A.; BISHOP, B.; RASHEED, K. Automated synthesis of standard cells using genetic algorithms. In: IEEE COMPUTER SOCIETY ANNUAL SYMPOSIUM ON VLSI, 2002. **Proceedings...** [S.l.: s.n.], 2002. p.126–133.

BALASA, F. Modeling Non-Slicing Floorplans with Binary Trees. In: ICCAD, 2000. **Proceedings...** [S.l.: s.n.], 2000. p.13–16.

BALASA, F. Device-level placement for analog layout: an opportunity for non-slicing topological representations. In: ASIA SOUTH PACIFIC DESIGN AUTOMATION, ASP-DAC, 2001. **Proceedings...** New York: ACM Press, 2001. p.281–286.

BALASA, F.; MARUVADA, S. C.; KRISHNAMOORTHY, K. Efficient solution space exploration based on segment trees in analog placement with symmetry constraints. In: IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, 2002. **Proceedings...** New York: ACM Press, 2002. p.497–502.

BALASA, F.; MARUVADA, S. C.; KRISHNAMOORTHY, K. On the exploration of the solution space in analog placement with symmetry constraints. **IEEE Transactions on**

Computer-Aided Design of Integrated Circuits and Systems, [S.l.], v.23, n.2, p.177–191, February 2004.

BASTIAN, F.; LAZZARI, C.; GUNTZEL, J.; REIS., R. A New Transistor Folding Algorithm Applied to an Automatic Full-Custom Layout Generation Tool. In: INTERNATIONAL WORKSHOP ON POWER AND TIMING MODELING, 2004. **Proceedings...** [S.l.: s.n.], 2004. p.732–741.

BASTOS, R. P. **Design a Robust Microprocessor to Soft Errors**. 2006. 113p. Dissertacão (Mestrado em Ciencia da Computacão) — Instituto de Informatica, UFRGS, Porto Alegre.

BAUMANN, R. Soft Errors in Advanced Computer Systems. **IEEE Des. Test**, Los Alamitos, CA, USA, v.22, n.3, p.258–266, 2005.

BEHJAT, L.; CHIANG, A. Fast integer linear programming based models for VLSI global routing. In: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, ISCAS, 2005. **Proceedings...** [S.l.: s.n.], 2005. p.6238–6243.

BEHJAT, L.; VANNELLI, A.; ROSEHART, W. Integer Linear Programming Models for Global Routing. **INFORMS J. on Computing**, Linthicum, Maryland, USA, v.18, n.2, p.137–150, 2006.

BERKELAAR, M. **lp solve 5.5.0.9**. [S.l.: s.n.], 2006.

BESSOT, D.; VELAZCO, R. Design of SEU-Hardened CMOS Memory Cells: the hit cell. In: RADECS CONFERENCE, 1994. **Proceedings...** [S.l.: s.n.], 1994. p.563–570.

BHARDWAJ, S.; CAO, Y.; VRUDHULA, S. Statistical leakage minimization through joint selection of gate sizes, gate lengths and threshold voltage. In: ASIA SOUTH PACIFIC DESIGN AUTOMATION, ASP-DAC, 2006. **Proceedings...** New York: ACM Press, 2006. p.953–958.

CADENCE. **Global Synthesis for timing closure - The impact on design closure**. [S.l.: s.n.], 2007.

CADENCE. Available at: <http://www.cadence.com/>. Visited on: August, 2007.

CALIN, T.; NICOLAIDIS, M.; VELAZCO, R. Upset hardened memory design for submicron CMOS technology. **IEEE Transactions on Nuclear Science**, [S.l.], v.43, p.2874–2878, December 1996.

CAZEAUX, J. M.; ROSSI, D.; OMANA, M.; METRA, C.; CHATTERJEE, A. On Transistor Level Gate Sizing for Increased Robustness to Transient Faults. In: IEEE INTERNATIONAL ON-LINE TESTING SYMPOSIUM, IOLTS, 2005. **Proceedings...** Washington : IEEE Computer Society, 2005. p.23–28.

CHAN, T.; CONG, J.; SZE, K. Multilevel generalized force-directed method for circuit placement. In: PHYSICAL DESIGN, ISPD, 2005. **Proceedings...** New York: ACM Press, 2005. p.185–192.

CHEN, W.; HSIEH, C.-T.; PEDRAM, M. Simultaneous Gate Sizing and Fanout Optimization. In: IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, 2000. **Proceedings...** [S.l.: s.n.], 2000. p.374–378.

CHENG, C. **Timing Closure Using Layout Based Design Process**. Available at: <http://www.techonline.com/community/related content/14016>. Visited on: July, 2007.

CHINNERY, D.; KEUTZER, K. **Closing the Gap Between ASIC & Full Custom**: tools and techniques for high-performance asic design. [S.l.]: Kluwer Academic Publisher Group, 2002. 434p.

CIESIELSKI, M.; ASKAR, S.; LEVITIN, S. Analytical approach to layout generation of datapath cells. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, [S.l.], v.02, n.12, p.1480–1488, Dec. 2002.

CONG, J.; SARRAFZADEH, M. Incremental physical design. In: INTERNATIONAL SYMPOSIUM ON PHYSICAL DESIGN, 2000. **Proceedings...** New York: ACM Press, 2000. p.84–92.

DETJENS, E. et al. Technology mapping in MIS. In: ICCAD, 1987. **Proceedings...** [S.l.: s.n.], 1987. p.116–119.

DHARCHOUDHURY, A. et al. Fast timing simulation of transient faults in digital circuits. In: IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, ICCAD, 1994. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1994. p.719–722.

DHILLON, Y. S.; DIRIL, A. U.; CHATTERJEE, A.; SINGH, A. D. Sizing CMOS Circuits for Increased Transient Error Tolerance. In: INTERNATIONAL ON-LINE TESTING SYMPOSIUM, IOLTS, 2004. **Proceedings...** Washington: IEEE Computer Society, 2004. p.11.

DODD, P.; MESSENGILL, L. Basic Mechanisms and Modeling of Single-Event Upset in Digital Microelectronics. **IEEE Transactions on Nuclear Science**, [S.l.], v.50, n.3, p.583–602, June 2003.

DUFOUR, C. et al. Heavy ion induced single hard errors on submicronic memories. **IEEE Transactions on Nuclear Science**, [S.l.], p.1693 – 1697, 1992.

ERIKSSON, H.; LARSSON-EDEFORS, P.; HENRIKSSON, T.; SVENSSON, C. Full-custom vs. standard-cell design flow: an adder case study. In: ASIA SOUTH PACIFIC DESIGN AUTOMATION, ASP-DAC, 2003. **Proceedings...** New York: ACM Press, 2003. p.507–510.

GADLAGE, M. et al. Single event transient pulse widths in digital microcircuits. **IEEE Transactions on Nuclear Science**, [S.l.], v.51, p.3285 – 3290, December 2004.

GOPALAKRISHNAN, P.; RUTENBAR, R. A. Direct transistor-level layout for digital blocks. In: IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, ISCAD, 2001. **Proceedings...** Piscataway: IEEE Press, 2001. p.577–584.

GUO, P.-N.; CHENG, C.-K.; YOSHIMURA, T. An O-Tree Representation of Non-Slicing Floorplan and Its Applications. In: DAC, 1999. **Proceedings...** [S.l.: s.n.], 1999. p.268–273.

GUPTA, A.; HAYES, J. P. CLIP: integer-programming-based optimal layout synthesis of 2d cmos cells. **IEEE Transactions on Design Automation of Electronic Systems**, New York, NY, USA, v.5, p.510–547, 2000.

GUPTA, P.; KAHNG, A. B.; SHARMA, P.; SYLVESTER, D. Selective gate-length biasing for cost-effective runtime leakage control. In: DESIGN AUTOMATION, DAC, 2004. **Proceedings...** New York: ACM Press, 2004. p.327–330.

GURUSWAMY, M. et al. CELLERITY: a fully automatic layout synthesis system for standard cell libraries. In: DESIGN AUTOMATION CONFERENCE, 1997. **Proceedings...** [S.l.: s.n.], 1997. p.327–332.

HARBOE-SORENSEN, R. et al. The behaviour of measured SEU at low altitude during periods of high solar activity. **IEEE Transactions on Nuclear Science**, [S.l.], p.1938 – 1946, 1990.

HASHIMOTO, M.; ONODERA, H. Post-layout transistor sizing for power reduction in cell-based design. In: ASIA SOUTH PACIFIC DESIGN AUTOMATION, ASP-DAC, 2001. **Proceedings...** New York: ACM Press, 2001. p.359–365.

HEIJMEN, T.; NIEUWLAND, A. Soft-Error Rate Testing of Deep-Submicron Integrated Circuits. In: IEEE EUROPEAN TEST SYMPOSIUM, ETS, 2006. **Proceedings...** Washington: IEEE Computer Society, 2006. p.247–252.

HENTSCHKE, R. **Algoritmos para o Posicionamento de Celulas em Circuitos VLSI**. 2002. Dissertacão (Mestrado em Ciencia da Computacão) — Instituto de Informatica, UFRGS, Porto Alegre.

HENTSCHKE, R. **Algorithms for Wire Length Improvement of VLSI Circuits With Concern to Critical Paths**. 2007. Thesis (Ph.D. in Computer Science) — Instituto de Informatica, UFRGS, Porto Alegre. To be published.

HENTSCHKE, R. F.; NARASIMHAM, J.; JOHANN, M. O.; REIS, R. L. Maze routing steiner trees with effective critical sink optimization. In: PHYSICAL DESIGN, ISPD, 2007. **Proceedings...** New York: ACM Press, 2007. p.135–142.

HENTSCHKE, R.; FLACH, G.; PINTO, F.; REIS, R. Quadratic placement for 3d circuits using z-cell shifting, 3d iterative refinement and simulated annealing. In: INTEGRATED CIRCUITS AND SYSTEMS DESIGN, SBCCI, 2006. **Proceedings...** New York: ACM Press, 2006. p.220–225.

HUR, S.-W.; CAO, T.; RAJAGOPAL, K.; PARASURAM, Y.; CHOWDHARY, A.; TIOURIN, V.; HALPIN, B. Force directed mongrel with physical net constraints. In: DESIGN AUTOMATION, DAC, 2003. **Proceedings...** New York: ACM Press, 2003. p.214–219.

IEEE Design & Test Staff. Deep-Submicron Challenges. **IEEE Des. Test**, Los Alamitos, CA, USA, v.19, n.2, p.3, 2002.

JASAREVIC, S.; JERIN, G. **Automated Soft Error Analysis and Protection in Combinational Logic Circuits**. 2005. 55p. Master thesis — Lund University, Sweden.

KAHNG, A. B.; MUDDU, S.; SHARMA, P. Defocus-aware leakage estimation and control. In: LOW POWER ELECTRONICS AND DESIGN, ISLPED, 2005. **Proceedings...** New York: ACM Press, 2005. p.263–268.

KASTENSMIDT, F. G. **Designing Single Event Upset Mitigation Techniques for Large SRAM-Based FPGA Components**. 2003. Tese (Doutorado em Ciencia da Computacão) — Instituto de Informatica, UFRGS, Porto Alegre.

KIM, N. S. et al. Leakage Current: moore's law meets static power. **Computer**, Los Alamitos, CA, USA, v.36, n.12, p.68–75, 2003.

KLEINHANS, J. et al. GORDIAN: vlsi placement by quadratic programming and slicing optimization. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, [S.l.], v.10, p.356–365, March 1991.

LABEL, K. A. et al. A roadmap for NASA's radiation effects research in emerging microelectronics and photonics. In: IEEE AEROSPACE CONFERENCE, 2000. **Proceedings...** [S.l.: s.n.], 2000. v.5, p.535–545.

LAZZARI, C. **Automatic Layout Generation of Static CMOS Circuits Targeting Delay and Power Reduction**. 2003. 113p. Master Dissertation — Instituto de Informatica, UFRGS, Porto Alegre.

LAZZARI, C.; ANGHEL, L.; REIS, R. A Transistor Placement Technique Using Genetic Algorithm and Analytical Programming. In: IFIP WG.5 CONFERENCE ON VERY LARGE SCALE INTEGRATION SYSTEM-ON-CHIP, 2005. **Proceedings...** [S.l.: s.n.], 2005. p.559–564.

LAZZARI, C.; ANGHEL, L.; REIS, R. On Implementing a Soft Error Hardening Technique By Using an Automatic Layout Generator: case study. In: IEEE INTERNATIONAL ON-LINE TESTING SYMPOSIUM, IOLTS, 2005. **Proceedings...** [S.l.: s.n.], 2005. p.29–34.

LAZZARI, C.; DOMINGUES, C. V.; GUNTZEL, J. L.; REIS, R. A. L. A New Macrocell Generation Strategy for Three Metal Layer CMOS Technologies. In: VLSI-SOC, 2003. **Proceedings...** [S.l.: s.n.], 2003. p.143–147.

LILLIS, J. et al. New performance driven routing techniques with explicit area/delay tradeoff and simultaneous wire sizing. In: DESIGN AUTOMATION, DAC, 1996. **Proceedings...** New York: ACM Press, 1996. p.395–400.

LIN, Y.; HE, L. Leakage efficient chip-level dual-Vdd assignment with time slack allocation for FPGA power reduction. In: DESIGN AUTOMATION, DAC, 2005. **Proceedings...** New York: ACM Press, 2005. p.720–725.

LONG, C.; HE, L. Distributed sleep transistor network for power reduction. In: DESIGN AUTOMATION, DAC, 2003. **Proceedings...** New York: ACM Press, 2003. p.181–186.

MALLELA, S.; GROVER, L. K. Clustering based simulated annealing for standard cell placement. In: ACM/IEEE CONFERENCE ON DESIGN AUTOMATION, DAC, 1988. **Proceedings...** [S.l.]: IEEE Computer Society Press, 1988. p.312–317.

MATIS, H. et al. **Space Applications**: radiation-induced effects. [S.l.: s.n.], 2003.

MAVIS, D.; EATON, P. Soft Error Rate Mitigation Techniques for Modern Microcircuits. In: INTERNATIONAL RELIABILITY PHYSICS SYMPOSYUM, 2002. **Proceedings...** [S.l.: s.n.], 2002. p.216–255.

MCMURCHIE, L.; EBELING, C. PathFinder: a negotiation-based performance-driven router for fpgas. In: FIELD-PROGRAMMABLE GATE ARRAYS, FPGA, 1995. **Proceedings...** New York: ACM Press, 1995. p.111–117.

MESSENGER, G. Collection of charge on junction nodes from ion tracks. **IEEE Transactions on Nuclear Science**, [S.l.], p.2024–2031, 1982.

MO, F.; TABBARA, A.; BRAYTON, R. A force-directed macro-cell placer. In: INTERNATIONAL CONFERENCE ON COMPUTER AIDED DESIGN, ICCAD, 2000. **Proceedings...** [S.l.: s.n.], 2000. p.177–180.

MOHANRAM, K.; TOUBA, N. A. Cost-Effective Approach for Reducing Soft Error Failure Rate in Logic Circuits. In: ITC, 2003. **Proceedings...** Los Alamitos: IEEE Computer Society, 2003.

MOORE, G. E. No exponential is forever: but ¨ forever¨ can be delayed! In: IEEE INTERNATIONAL SOLID-STATE CIRCUITS CONFERENCE, ISSCC, 2003. **Digest of Technical Papers...** [S.l.: s.n.], 2003. v.1, p.20–23.

NARENDRA, S. et al. Full-chip sub-threshold leakage power prediction model for sub-0.18 CMOS. In: LOW POWER ELECTRONICS AND DESIGN, ISLPED, 2002. **Proceedings...** New York: ACM Press, 2002. p.19–23.

NICOLAIDIS, M.; CALIN, T. A Theory of Perturbation Tolerant Asynchronous FSM and its Application on the Design of Perturbation Tolerant Memories. In: EUROPEAN TEST WORKSHOP, ETW, 1997. **Proceedings...** [S.l.: s.n.], 1997. p.28–30.

NIEUWLAND, A. K.; JASAREVIC, S.; JERIN, G. Combinational Logic Soft Error Analysis and Protection. In: IEEE INTERNATIONAL SYMPOSIUM ON ON-LINE TESTING, IOLTS, 2006. **Proceedings...** Washington: IEEE Computer Society, 2006. p.99–104.

NORMAND, E. Single Event Upset at Ground Level. **IEEE Transactions on Nuclear Science**, [S.l.], v.43, p.2742 – 2750, December 2001.

NOYES, J.; WEISSTEIN, E. **Linear Programming.** Available at: <http://mathworld.wolfram.com/LinearProgramming.html>, Visited on: August, 2007.

OTTEN, R.; CAMPOSANO, R.; GROENEVELD, P. Design Automation for Deepsubmicron - present and future. In: DESIGN, AUTOMATION AND TESTE IN EUROPE CONFERENCE AND EXHIBITION, 2002. **Proceedings...** [S.l.]: IEE Computer Society, 2002. p.650–657.

POIVEY, C. et al. Characterization of single hard errors (SHE) in 1 M-bit SRAMs from single ion. **IEEE Transactions on Nuclear Science**, [S.l.], p.2235 – 2239, 1994.

REDA, S.; CHOWDHARY, A. Effective linear programming based placement methods. In: PHYSICAL DESIGN, ISPD, 2006. **Proceedings...** New York: ACM Press, 2006. p.186–191.

REIS, A. I.; REIS, R. A. L.; AUVERNE, D.; ROBERT, M. Library Free Technology Mapping. In: INTERNATIONAL CONFERENCE IN VERY LARGE SCALE INTEGRATION, VLSI, 1997. **Proceedings...** [S.l.: s.n.], 1997. p.303–314.

REKHI, S.; TROTTER, J. D.; LINDER, D. H. Automatic layout synthesis of leaf cells. In: ACM/IEEE CONFERENCE ON DESIGN AUTOMATION, DAC, 1995. **Proceedings...** New York: ACM Press, 1995. p.267–272.

RIEPE, M. A.; SAKALLAH, K. A. Transistor level micro-placement and routing for two-dimensional digital VLSI cell synthesis. In: PHYSICAL DESIGN, ISPD, 1999. **Proceedings...** New York: ACM Press, 1999. p.74–81.

RIEPE, M. A.; SAKALLAH, K. A. Transistor placement for noncomplementary digital VLSI cell synthesis. **ACM Trans. Des. Autom. Electron. Syst.**, New York, NY, USA, v.8, n.1, p.81–107, 2003.

ROCKETT, L. An SEU Hardened CMOS Data Latch Design. **IEEE Transactions on Nuclear Science**, [S.l.], v.NS-35, n.6, p.1682–1687, Dec. 1988.

ROY, R.; BHATTACHARYA, D.; BOPPANA, V. Transistor-Level Optimization of Digital Designs with Flex Cells. **Computer**, [S.l.], v.38, p.53–61, Feb. 2005.

SANTOS, C.; WILKE, G.; LAZZARI, C.; GUNTZEL, J. L.; REIS, R. A Transistor Sizing Method Applied to an Automatic Layout Generation Tool. In: SYMPOSIUM ON INTEGRATED CIRCUITS AND SYSTEMS DESIGN, 2003. **Proceedings...** [S.l.: s.n.], 2003. p.303–307.

SECHEN, C. Chip-planning, placement, and global routing of macro/custom cell integrated circuits using simulated annealing. In: DESIGN AUTOMATION CONFERENCE, DAC, 1988. **Proceedings...** [S.l.: s.n.], 1988. p.73–80.

SERDAR, T.; SECHEN, C. Automatic datapath tile placement and routing. In: DESIGN, AUTOMATION AND TEST IN EUROPE, 2001. **Proceedings...** [S.l.: s.n.], 2001. p.13–16.

SIA. **The National Technology Roadmap for Semiconductors**. San Jose, California, 1997.

SYNOPSYS. Available at: <http://www.synopsys.com/>. Visited on: April, 2007.

TAYU, S. A simulated annealing approach with sequence-pair encoding using a penalty function for the placement problem with boundary constraints. In: ASIA SOUTH PACIFIC DESIGN AUTOMATION, ASPDAC, 2003. **Proceedings...** New York: ACM Press, 2003. p.319–324.

VISWANATHAN, N.; CHU, C. C.-N. FastPlace: efficient analytical placement using cell shifting, iterative local refinement and a hybrid net model. In: PHYSICAL DESIGN, ISPD, 2004. **Proceedings...** New York: ACM Press, 2004. p.26–33.

VUJKOVIC, M. et al. Efficient timing closure without timing driven placement and routing. In: DESIGN AUTOMATION, DAC, 2004. **Proceedings...** New York: ACM Press, 2004. p.268–273.

WANG, Q.; LILLIS, J.; SANYAL, S. An LP-based methodology for improved timing-driven placement. In: ASIA SOUTH PACIFIC DESIGN AUTOMATION, ASP-DAC, 2005. **Proceedings...** New York: ACM Press, 2005. p.1139–1143.

WEISSTEIN, E. W. **Euler Path**. Available at: <http://mathworld.wolfram.com>. Visited on: July, 2007.

WEISSTEIN, E. W. **Bisection**. Available at: <http://mathworld.wolfram.com>. Visited on: July, 2007.

WESTE, N.; ESRAGHIAN, K. **Principles of CMOS VLSI Design - A Systems Perspective**. [S.l.]: Addison-Wesley Publishing Company, 1993.

WHITAKER, S.; CANARIS, J.; LIU, K. SEU Hardened Memory Cells for a CCSDS Reed Solomon Encoder. **IEEE Transactions on Nuclear Science**, [S.l.], v.NS-36, n.6, p.1471–1477, December 1991.

WILSON, D. C.; SMITH, R. J. An experimental comparison of force directed placement techniques. In: DESIGN AUTOMATION, DAC, 1974. **Proceedings...** Piscataway: IEEE Press, 1974. p.194–199.

WIRTH, G.; VIEIRA, M.; HENES, E.; KASTENSMIDT, F. L. Modelling the sensivity of CMOS circuits to radiation induced single event transients. **Microelectonics Reliability**, [S.l.], v.47, n.3, March 2007.

WIRTH, G.; VIEIRA, M.; KASTENSMIDT, F. L. Accurate and computer efficient modelling of single event transients in CMOS circuits. **IET Circuits, Devices & Systems**, [S.l.], v.1, p.137–142, April 2007.

ZHAO, W.; CAO, Y. Predictive technology model for nano-CMOS design exploration. **J. Emerg. Technol. Comput. Syst.**, New York, NY, USA, v.3, n.1, p.1, 2007.

ZHOU, Q.; MOHANRAM, K. Gate sizing to radiation harden combinational logic. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, [S.l.], v.25, p.155– 166, Jan. 2006.

ZIESEMER, A. M. **Geracao Automatica de Partes Operativas de Circuitos VLSI**. 2007. 88p. Dissertacão, Mestrado em Ciencia da Computacão — Instituto de Informatica, UFRGS, Porto Alegre.

APPENDIX A GERAÇÃO AUTOMÁTICA DE CIRCUITOS TOLERANTES A RADIAÇÃO NO NÍVEL DE TRANSISTORES

Tecnologias submicrônicas (DSM) têm inserido novos desafios ao projeto de circuitos devido a redução de geometrias, redução na tensão de alimentação, aumento da frequência e aumento da densidade de lógica (CONG; SARRAFZADEH, 2000). Estas características reduzem significativamente a confiabilidade dos circuitos integrados devido a suceptibilidade a efeitos como *crosstalk* e acoplamento de substrato (IEEE Design & Test Staff, 2002).

Interconexões tem apresentado uma crescente importância em tecnologias DSM. Novas tecnologias tem alterado o paradigma de projeto de processos dominados pelo atraso das portas lógicas pata processos dominados pelo atraso das conexões (CHENG, 2007).

A figura A.1 mostra a formação do atraso como função das gerações de tecnologia (SIA, 1997). As tecnologias menores de $100nm$ são estimadas. É possível notar que o atraso das conexões ultrapassa o atraso dos gates no processo de tecnologia de $250nm$ quando *Al* é usado para conexões e SiO_2 é usado como isolante.

O atraso das interconexões apresenta maior importância no processo tecnológico $0.18\mu m$ quando *Cu* é usado para as conexões e um dielétrico *low k* é usado como isolante.

A quantidade de atraso das interconexões não é o unico fator importante em tecnologias DSM. A alta densidade de lógica e o número de níveis de metal faz da previsão das características das interconexões uma tarefa muito difícil. Em uma tecnologia de $180nm$, por exemplo, a capacitância por unidade de tamanho pode variar até 35 vezes (VUJKOVIC, M. et al., 2004).

Frequentemente, a lógica de um circuito é otimizada em respeito a atraso, área e potência com capacitâncias estimadas, mas seus valores podem somente ser conhecidos depois da fase de leiaute. Esta estimativa imprecisa das interconexões impossibilita que o atraso especificado seja encontrado sem a presença real de dados do leiaute.

O advento das tecnologias submicrônicas também torna obrigatório o controle do consumo de potência estático. Projetistas tem tomado providências sobre o consumo de potência dinâmica durante toda a evolução tecnológica. Por outro lado, a potência estática não era levado em conta devido a sua baixa participação no comsumo total de potência. Entretanto, a potência estática está aumentado drasticamente em tecnologias DSM.

A figura A.2 mostra a potência estática e dinâmica em microprocessadores como função dos anos (MOORE, 2003). É projetado que a potência estática cheque ao mesmo

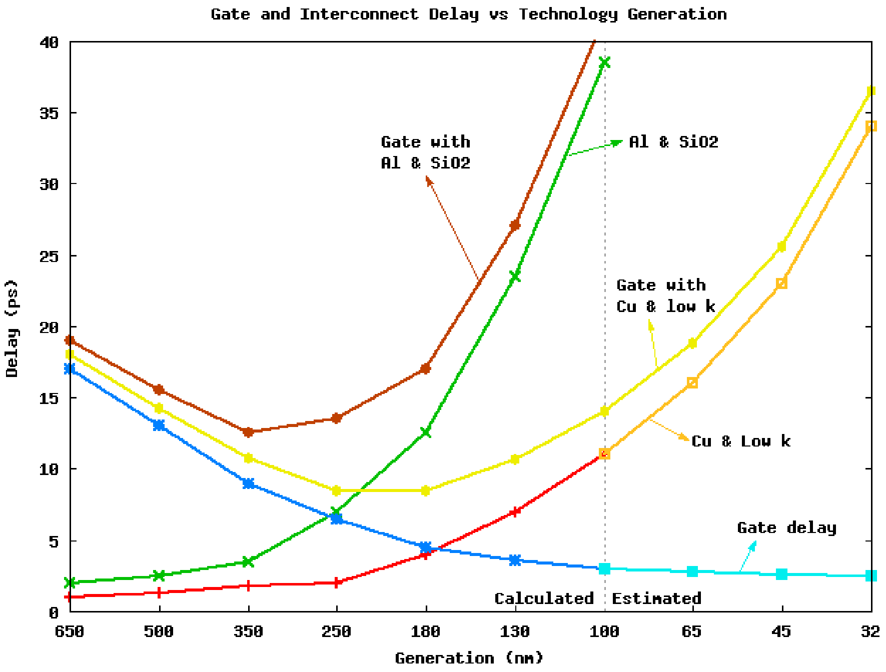


Figure A.1: Atraso das portas e das conxões como função das gerações de tecnologia (SIA, 1997). Atrasos para tecnologias menores de 100nm são estimados.

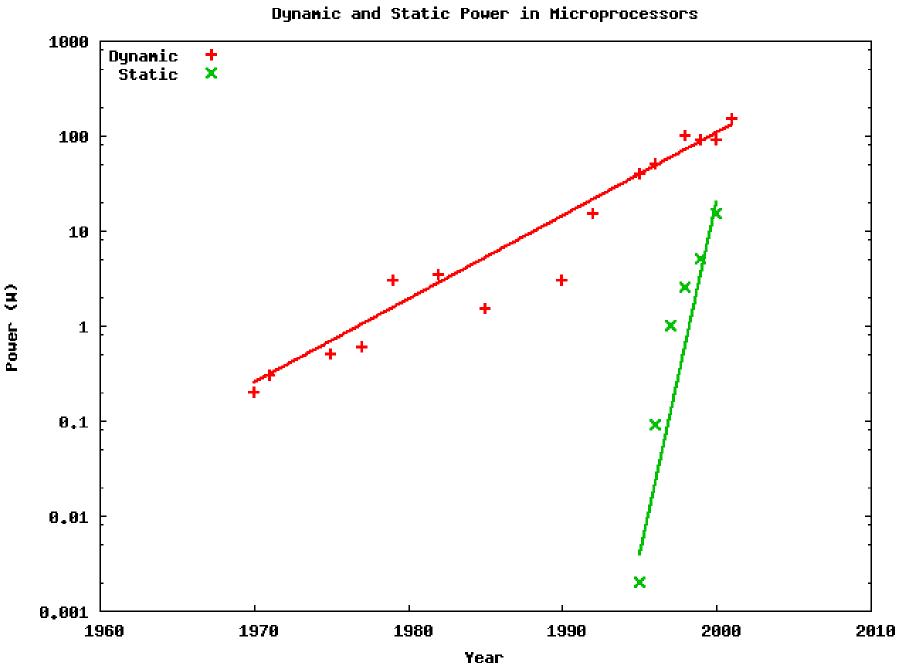


Figure A.2: Active and static power in microprocessors (MOORE, 2003).

nível da potência dinâmica nas tecnologias de $65nm$ (KIM, N. S. et al., 2003).

Tratar desses desafios requerem um cuidadoso planejamento (OTTEN; CAMPOSANO; GROENEVELD, 2002) e enfatiza a necessidade de ferramentas de automação eletrônica dos projetos (EDA) aptas a gerar e validade automaticamente circuitos integrados.

Projetos baseados em células padrão dominaram a geração de leiaute de circuitos VLSI digitais devido a algumas virtudes (GOPALAKRISHNAN; RUTENBAR, 2001). Células padrão ocultam detalhes das regras de projetos, pinos de E/S são organizados em locais acessíveis, células são geradas com relativa facilidade em blocos baseados em bandas e células podem ser pre-caracterizadas para atraso e potência.

Entretando, o vão entre aplicações de células padrão e customizadas são demostradas na literatura (ANDREW, 1998; ERIKSSON et al., 2003).

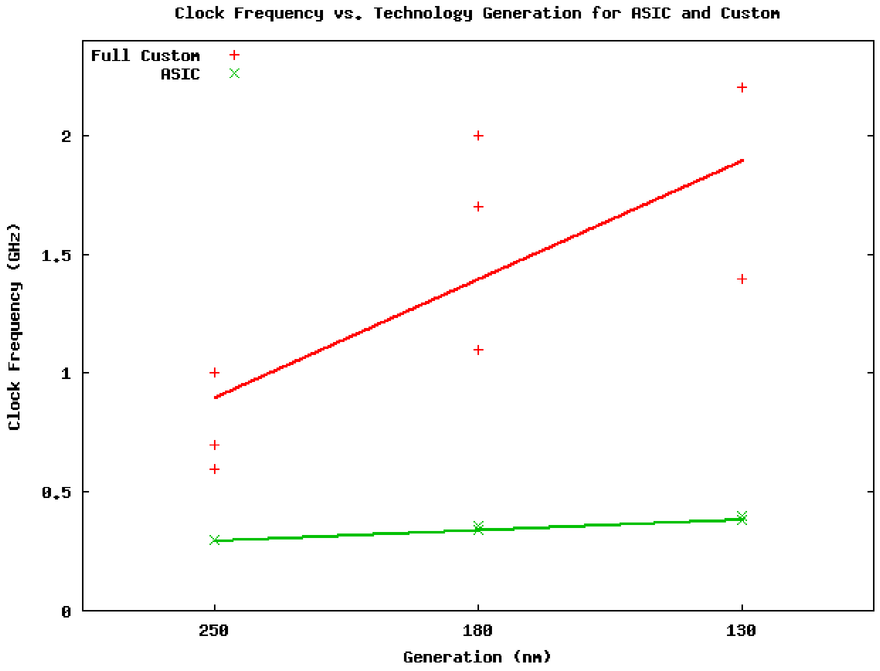


Figure A.3: Frequências de relógio de ASICs de alta frequência e processados customizados em diferentes processos de tecnologia (CHINNERY; KEUTZER, 2002).

A figura A.3 quantifica as diferenças entre a velocidade de circuitos ASIC e customizados. Chinnery e Keutzer classifica alguns processadores ASIC e customizados e avalia a frequência como função das gerações de tecnolgias (CHINNERY; KEUTZER, 2002). Alguns dos circuitos customizados são processadores Intel, AMD e IBM Power PC. Tensilica, Lexra e ARM são classificados como processadores ASIC.

A figura mostra que a diferença de frequência entre circuitos ASICs e customizados está aumentando. Autores relatam alguns dos fatores que contribuem à frequência superior em aplicações customizadas. Os principais fatores são modificações na arquitetura, eficiente projetos de árvore de relógio, fios e células e dimensionamento de transistores. Além disso, o projeto cutomizado oferece a vantagem do controle completo no tamanho e local de cada transistor para ajusto de atraso.

Apesar da utilização de células padrão ser eficiente na geração de leiaute, alguns trabalhos propoem algumas modificações no fluxo de projeto tradicional com o objetivo de reduzir a diferença existente entre projetos ASIC e customizados.

Uma otimização pós-leiaute é apresentada em (HASHIMOTO; ONODERA, 2001), na qual uma metodologia de redução no tamanho dos transistores com preservação das interconexões é proposta. 65% de redução de potência foi obtida se aumento no atraso do circuito.

Vujkovic *et al* (VUJKOVIC, M. et al., 2004) mostra que bibliotecas de células padrão não conseguem ser eficientes com a grande variação de capacitância das conexões com um número peuqno de células.

O conceito de células flexíveis é apresentado por (ROY; BHATTACHARYA; BOPPANA, 2005). Eles propoem a identificação e otimização de um mínimo número de células críticas aumentando a performance dos circuitos.

As contribuições desta tese estão divididas em duas partes. A primeira está relacionada com o desenvolvimento de uma nova metodologia apta a gerar circuitos otimizados em relação à potência consumida e o atraso. Um fluxo de projeto no cível de transistor otimiza cada porta do circiuto de acordo coma s capacitâncias envolvidas.

O fluxo proposto é baseado em ferramentas acadêmicas e comerciais. Ferramentas comerciais incluem síntese lógica, posicionamendto e roteamento, análise estática a nível de transistor e análise de consumo de potência. Ferramentas foram desenvolvidas com o objetivo de reduzir a distância entre células padrões e metodologias no cível de transistor.

Basicamente, o fluxo proposto apresenta as seguintes diferenças em comparação com a metodologia de células padrão:

1. *Geração de bibliotecas:* O leiaute das células não é gerado neste momento. Isso permite aumentar significantemente o número de funções lógicas.

2. *Otimização no nível de transistores:* Otimização de transistors permite encontrar larguras otimizadas de acordo com a capacitâncias encolcidas.

3. *Geração de leiaute:* A geração de leiaute é desenvolvida depois da otimização de transistores possibilitando um grande número de possibilidades em relação a tamanho de transistores.

4. *Um fluxo com realimentação:* Um fluxo de projeto com realimentação permite levar em conta capacitâncias extraídas durante a otimização de transistores.

Um método de redução de corrente de fuga é proposto na qual o ajuste da tamanho do canal dos transistores é aplicado para reduzir a potência estática. Esta tecnica consiste em ajustar o tamanho do canal dos transisrores em 10% para reduzir a corrente que atravessa por ele.

Todas a técnicas apresentadas permitem encontrar mais facilmente as epecificações de atraso e potência dos projetos. Comparações entre a metodologia de células padrão e a proposta mostram resultados interessantes onde a metodologia proposta apresenta 11% de melhora no atraso e 30% de redução na potência consumida.

A segunda contribuição desta tese está vinculada a aplicação do fluxo de projeto no cível de transistores na proteção de circuitos contra SEEs. Os principais aspectos estão ligados à prossibilidade de desenvolver novas metodologias de dimensionamento de transistores com o objetivo de gerar circuitos combinacionais tolerantes à radiação.

A redução no tamanho das dimensões dos circuitos podem aumentar a taxa de erros devido a SEEs. A redução no comprimento de canal dos transistores e a redução nas tensões de alimentação fazem dos circuitos mais sensíveis à particulas energéticas.

Duas contribuições estão ligadas ao desenvolvimento de circuitos tolerantes a radiação. O primeiro está ligado a inserção de redundância temporal em elementos sequenciais. A idéia principal consiste em aplicar uma técnica proposta por (ANGHEL, 2000) em flip flops. A técnica CWSP foi aplicada em microprocessadores e os resultados mostram como pior caso 85% de aumento na área ocupada e 7% de penalidade de atraso.

A segunda contribuição relacionada a metodos de tolerância a falhas está relacionada com o desenvolvimento de uma metodologia de dimensionamento de transistores. A sensibilidade dos circuitos é definida a partir da análise do mascaramento lógico e elétrico.

O mascaramento lógico é dado pela probabilidade de um pulso ser mascarado pela lógica do circuito, na qual é computado pela controlabilidade e a observabilidade. O mascaramento elétrico descreve se um pulso em um nodo não é propagado até as saídas primárias devido à atenuação elétrica.

O método de cimensionamento de transistores é baseado em um método analítico de tetecção de pulsos transientes apresentado em (WIRTH; VIEIRA; KASTENSMIDT, 2007; WIRTH et al., 2007). O modelo é baseado em parâmetros elétricos dos dispositivos. um método de propagação é usado, onde o atraso dos gates e a duração dos pulsos transientes são avaliados. O modelo de propagação é muito importante pois é assumido que a atenuação dos pulsos acontece no caminho até as saídas primárias. Assim, transistores maiores que o necessário são evitados. O modelo também considera independentemente blocos de transisrores PMOS e NMOS. Assim, somente transistores diretamente ligados à atenuação do SET são dimensionados.

A técnica de dimensionamento proposta apresentou area, atraso e consumo de potência reduzida em comparação com técnicas deTMR e CWSP, permitindo o desenvolvimento de circuitos de alta frequência, com poucas penalidade em relação à area e potência. Resultados mostram 63% de aumento médio da área ocupada, 43% de aumento médio da potência consumida e 10% de aumento médio no atraso dos circuitos.