

THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

**On DFM Considerations and Assessment
for Nanometer SoCs**

KASYAB P. SUBRAMANIYAN

Division of Computer Engineering
Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2014

On DFM Considerations and Assessment for Nanometer SoCs

Kasyab P. Subramaniyan

ISBN 978-91-7597-037-0

Copyright © Kasyab P. Subramaniyan,

Doktorsavhandlingar vid Chalmers tekniska högskola

Ny serie nr 3718

ISSN 0346-718X

Technical report 111D

Department of Computer Science and Engineering

VLSI Research Group

Division of Computer Engineering

Chalmers University of Technology

SE-412 96 GÖTEBORG, Sweden

Phone: +46 (0)31-772 10 00

Author e-mail: kasyab@chalmers.se

Cover:

A 12-inch silicon wafer. [*Source: Wikimedia Commons*]

Quotes between chapters sourced from:

Murphy's Laws and variants,

Another Set of Murphy's Laws and variants,

Quotes about science &

An Interview with Mark Twain

Printed by Chalmers Reproservice

Göteborg, Sweden 2014

On DFM Considerations and Assessment for Nanometer SoCs

Kasyab P. Subramaniyan

Division of Computer Engineering, Chalmers University of Technology

ABSTRACT

The incredible density of silicon integrated circuits has brought with it unprecedented technological advances. This is made possible through innovations at each incremental technology node. With the layout geometries of circuits approaching the physical limits of an atom, innovative enablers in manufacturing are scarce; and when they exist, increasingly expensive and/or difficult to implement. This has led to the discipline of Design for Manufacturability (DFM) becoming a mandatory consideration in the design and implementation of electronic systems.

In the nanometer era, regularity has been used extensively to combat layout issues that make the implementation of electronic systems challenging. The first part of this thesis presents a semi-custom methodology to implement layouts for datapath elements that exhibit netlist regularity. Here a novel methodology, using a domain-specific, low-level, layout-aware hardware description language, Wired, is used to create netlists for physical implementations of datapath elements such as column compression multipliers and logarithmic shifters. The netlist regularity is preserved during physical design resulting in highly regular, area efficient, yet Design Rule Check (DRC) compliant implementations.

In the second part of this thesis, the assessment of manufacturability is presented. DFM tools integrated into the traditional full-custom design environment are used to enable this. This assessment is carried out from the perspective of creating manufacturable nanometer standard-cell libraries. The metric used to assess manufacturability is Critical Feature Analysis (CFA). Counter intuitive trends indicating better manufacturability of standard cells with less regular geometries are showcased. DFM assessment extending the earlier work, and carried out on implementations of the ISCAS '89 benchmark circuits, show similar results in spite of the fact that raw implementation metrics indicate otherwise. As a final contribution, a simple model to enable early assessment of design manufacturability in System-on-Chips (SoCs) is presented. The model which is based largely on data available from the physical implementation of the design, is demonstrated on a processor implementation including a L1-cache subsystem. Various implementation aspects like floorplan and Intellectual Property (IP) inclusion are investigated in the early assessment of the DFM metric.

Keywords: DFM, Regularity, CMOS, SoC, ASIC, Multipliers, Shifters, Processor.

Preface

These contributions, published in peer reviewed conferences, are used in this monograph. The boldface text below each entry summarizes the contribution.

- **Kasyab P. Subramaniyan**, Emil Axelsson, Per Larsson-Edefors and Mary Sheeran, “Layout Exploration of Geometrically Accurate Arithmetic Circuits” in *Proc. IEEE Int. Conf. Electronics, Circuits, and Systems*, Hammamet, Tunisia, December 13-16, 2009, 795-988.
Wired Methodology applied to HPM Multiplier.
- Alen Bardizbanyan, **Kasyab P. Subramaniyan** and Per Larsson-Edefors, “Generation and Exploration of Layouts for Area-Efficient Barrel Shifters” in *Proc. IEEE Computer Society Annual Symp. on VLSI*, Lixouri Paliki Kefalonia, Greece, July 5-7, 2010, 454-455.
Wired Methodology applied to Shifters.
- **Kasyab P. Subramaniyan** and Per Larsson-Edefors, “On Regularity and Integrated DFM Metrics” in *Proc. 4th Asia Symposium on Quality Electronic Design (ASQED)*, Penang, Malaysia, July 10-11, 2012, 211-218. *Winner of the Best Paper Award*
CFA analysis of Standard-Cells.
- **Kasyab P. Subramaniyan** and Per Larsson-Edefors, “Manufacturable Nanometer Designs using Standard Cells with Regular Layout” in *Proc. 14th Int. Symp. on Quality Electronic Design (ISQED)*, Santa Clara, USA, March 4-6, 2013, 398-405.
DFM analysis of ISCAS ’89 Benchmark Circuits through CFA and implementation metrics.

The following manuscripts are under review or slated for submission but results are relevant to this work and are so included.

- **Kasyab P. Subramaniyan** and Per Larsson-Edefors, “MIDAS: Model for IP-inclusive DFM Assessment of System Manufacturability” submitted to *2014 IEEE/ACM Int. Conf. on CAD (ICCAD)*, San Jose, USA, November 3-6, 2014.
Presents the MIDAS model.

The following manuscripts have been published but are not included in this work.

- Babak Hidaji, Salar Alipour, **Kasyab P. Subramaniyan** and Per Larsson-Edefors, “Application-Specific Energy Optimization of General-Purpose Datapath Interconnect” in *Proc. IEEE Computer Society Annual Symp. on VLSI*, Chennai, India, July 4-6, 2011, 301-306.
- **Kasyab P. Subramaniyan**, Erik Ryman, Magnus Sjölander, Tung Than Hoang, Mafijul Islam and Per Larsson-Edefors, “FlexDEF: Development Framework for Processor Architecture Implementation and Evaluation”, in *Proc. 7th Conference on Ph.D Research in Microelectronics and Electronics*, Madonna di Campiglio, Italy, July 3-7, 2011, 37-40.
- Tung Than Hoang, Ulf Jälmlbrant, Erik der Hagopian, **Kasyab P. Subramaniyan**, Magnus Sjölander and Per Larsson-Edefors, “Design Space Exploration for an Embedded Processor with Flexible Datapath Interconnect”, in *Proc. IEEE Int. Conf. on Application-specific Systems, Architectures and Processors*, Rennes, France, July 7-9, 2010.
- Patrik Kimfors, Niklas Broman, Andreas Haraldsson, **Kasyab P. Subramaniyan**, Magnus Sjölander, Henrik Eriksson and Per Larsson-Edefors, “Custom Layout Strategy for Rectangle-Shaped Log-Depth Multiplier Reduction Tree”, in *Proc. IEEE Int. Conf. on Electronics, Circuits, and Systems*, Hammamet, Tunisia, December 13-16, 2009, 77-80.

Acknowledgments

Life is what happens to you while you're busy making other plans.

~John Lennon

I am now in the seventh year of my residence in Göteborg, but it feels like it was yesterday that I was wrapping up my masters thesis wondering what the future held for me. As it turns out, it has been the most fruitful and enjoyable time I have ever had. There have been some significant losses but it is the gains I gratefully remember. As my time in academia draws to a close, I would like to acknowledge all the people who have been instrumental in the process.

First of all, I would like to thank my advisor Per Larsson-Edefors for his guidance over the last six odd years. His relaxed, yet rigorous approach to guiding me during this time has resulted in my development as a researcher and as a human being. Our discussions relating to all topics have always been fun for me. The technical discussions always took a form of conversation, resulting in many interesting turns during the course of my PhD. I like to think we have some overlapping interests in films, photography and history other than the business of designing better chips. Discussions in any one of these topics were always lively and fun. Next, I would like to acknowledge my co-supervisor Lars Svensson. His inputs on my research made for better papers and the help I received from him while writing my papers was instrumental in publishing at the best venues in the area. I was also a teaching assistant in the data conversion courses that Lars was responsible for. This time was a welcome break for me to improve my knowledge of the subject while thinking about doing things better in the classroom and labs.

It would be remiss of me to not mention my opponent: Prof. Rouwaida Kanj. I am genuinely grateful that you accepted our request to be the faculty opponent for my thesis defense. I also gratefully acknowledge the other members of the committee: Prof. Ahmed Hemani, Prof. Joachim Rodrigues and Prof. Peter Enoksson. Special thanks go to Prof. Rodrigues who served as the faculty opponent in my licentiate defense and provided valuable feedback on that part of the work.

I would also like to acknowledge the ProviKing graduate school and the Swedish Foundation for Strategic Research for funding this work. Being a part of the ProViking school offered me a chance to work with researchers from disparate fields. Attending courses with these researchers was insightful to say the least. I would also like to thank Mary Sheeran for starting me out on Haskell programming. Special thanks also go to Emil Axelsson who was responsible for the development of Wired, which is used in the early part of this work. Much of the early publications would not have been possible without Emils in-depth knowledge of Haskell and the Wired system. The support from the system administrators, Rune Ljunbjörn and Peter Helander was also instrumental in keeping the CAD tools updated and their help with troubleshooting networking and other issues is gratefully acknowledged. Thanks to Gerardo Schneider, Jan Jonsson and Koen Claessen for providing me with guidance at the follow-up meetings. The administrative assistants, Tiina Rankanen and Eva Axelsson, are also gratefully acknowledged for their prompt help in dealing with administrative issues ranging from obtaining spectacle prescriptions and visa renewals to arranging for travel.

On a personal front I thank all my colleagues here at Chalmers for a fun-filled five years. Magnus, Erik and Tung, my co-conspirators in the Flexsoc implementation project, I cannot say how much I enjoyed working with you on that and also all the other times we've chatted. I would also like to thank Alen, Anurag, Angelos, Bhavi, Dmitry, Jacob, Madhavan, Mafijul and Risat for the enjoyable times I have had here. Thanks also to Vinay for the numerous competitive table tennis games and engaging conversations. Prakash, I just want to thank you for being the friend you are. I will also thank Tilak, Martin, Jan and Wiktor for the fun-filled AWs that invariably occur each month and are such

a welcome distraction from the rigors of research. I look forward to the next one. A special thank you to all my teachers, especially here at Chalmers. Every one of you were instrumental in me being where I am today. I would like to acknowledge Lars Bengtsson who was instrumental in my first experience with industry in Sweden. Your help in my stint with SAAB Space in the summer of 2008 is very deeply appreciated. To all my school and college friends, I will simply say thank you for the times we have shared. We might not be in regular contact, but irrespective of where we might be located today, I fondly remember the times we shared growing up.

To my parents, Rama and Parmesh; none of this would have been possible without your love and support. I cannot find the words to say how much you mean to me. My dearest Emma, I will say that you have been the single largest distraction, intellectually and otherwise, for the last four years and still are. I wouldn't have it any other way. Finally, I dedicate this thesis to Meenakshi Kailasam, my grandmother who passed away last February. I know how much you would've enjoyed seeing your youngest grandson get his doctorate.

Kasyab P. Subramaniyan
Göteborg, May, 2014.

Contents

Abstract	i
Preface	iii
Acknowledgments	v
Acronyms	xvii
I Introduction	1
1 Introduction	3
1.1 Background	3
1.2 Design Flows	5
1.3 Scaling and Manufacturability	8
1.4 Problem Statement and Scope of Work	11
Bibliography	13
II Placement Regularity In Semi-custom Flows	17
2 Regularity and Wired	19
2.1 Background	19
2.2 The Wired Design Environment	20
2.2.1 Logic domain	20
2.2.2 Placement	21

2.2.3	Wiring	22
2.3	Related Work	22
2.4	Methodology	24
2.5	Case Studies to verify the methodology	26
2.5.1	Barrel Shifters	26
2.5.2	Logarithmic Depth Multipliers	28
2.6	Results	30
2.6.1	Shifters	30
2.6.2	Multipliers	34
2.7	Conclusions	37
	Bibliography	37

III Manufacturability of Standard-cells and SoCs 41

3	DFM and CFA 43
3.1	Introduction 43
3.2	Regularity and Standard-cell design: Existing Literature 44
3.3	DFM Analysis - A Variability Primer 46
3.3.1	Variability Classification 46
3.3.2	Variability Analysis 48
3.4	Layout Architecture 50
3.4.1	Ultra-regular and Semi-regular Layouts 50
3.4.2	Factors affecting Analysis 52
3.5	Layout Implementation 55
3.6	A Semi-custom Design Perspective 56
3.7	Results 58
3.7.1	Cell Manufacturability Analysis 58
3.7.2	ISCAS Benchmark Circuits - Physical Implementation . 60
3.7.3	ISCAS Benchmark Circuits - CFA Results 64
3.8	Conclusions 66
	Bibliography 68

<i>CONTENTS</i>	xi
4 MIDAS	73
4.1 Introduction	74
4.2 Motivation	75
4.3 Environment and Tools	77
4.3.1 System-Level Implementation	77
4.3.2 Standard-Cell Libraries	79
4.4 MIDAS	80
4.4.1 Placement Cost	81
4.4.2 Interconnect Cost	83
4.4.3 Total DFM Cost and Normalization	85
4.5 Model Calibration	86
4.6 A Practical Test Case & Use Scenarios	89
4.7 Conclusions	91
Bibliography	91
IV Summary & Conclusions	95
5 Summary & Conclusions	97
5.1 Summary	97
5.2 Conclusion	98
Appendix	102

List of Figures

1.1	Transistor counts for integrated circuits by year [2, 3].	4
1.2	Design Flows for Analog, ASIC and FPGA methodologies.	6
1.3	Lithography source wavelengths against feature size [16].	8
1.4	Defects introduced due to the manufacturing process(Source: IMEC).	10
1.5	Relative cost of ownership of a 5000 wafer run device.[Adapted from [19]]	11
2.1	Postscript rendering of a Wired description.	21
2.2	Methodology to enforce placement regularity using Wired.	25
2.3	Barrel shifter structure.	27
2.4	Barrel shifter structures using NAND gates.	28
2.5	32-bit shifters placed in Encounter.	31
2.6	32-bit multiplexer-based shifters.	32
2.7	Metal usage for implemented 32-bit shifters in 90 nm CMOS.	33
2.8	A HPM Multiplier with a triangular PPRT.	35
2.9	A HPM Multiplier with a rectangular PPRT.	35
2.10	Total Wire length for different multiplier implementations in 90 nm CMOS.	36
3.1	Custom characterized XOR Gates.	51
3.2	Custom characterized Half adder cells.	52
3.3	Custom characterized full adder cells.	55
3.4	A full adder cell regular in Poly pitch and direction.	59
3.5	Cell count, chip density and slack plots for ISCAS benchmarks.	62
3.6	Number of vias in the ISCAS benchmark circuits after physical design.	63
3.7	Individual CFA rule contributions for the various checks.	65
4.1	CFA for placed and routed designs.	75
4.2	Implemented processor system floorplans.	78

4.3	Scatter plot of NDS values for cells in the custom libraries.	82
1	IR Drop analysis for different ISCAS'89 Benchmark circuits.	105
2	IR Drop analysis for different multiplier circuits.	106

List of Tables

2.1	Comparison of 32-bit barrel shifters in 90 nm CMOS.	31
2.2	Comparison of 64-bit barrel shifters in 90 nm CMOS.	32
2.3	Comparison of multiplier implementations in 90 nm CMOS.	34
2.4	Comparison of multiplier implementations in 65 nm CMOS.	36
3.1	Custom characterized cells in 65 nm CMOS.	51
3.2	Standard-cells implemented for the ISCAS'89 circuit tests	57
3.3	CFA Results for Ultra-regular and Semi-regular cells.	59
3.4	Physical Implementation Metrics for ISCAS'89 Benchmark Circuits .	61
3.5	Total DFM Metrics for Some Representative ISCAS'89 Benchmark Circuits	64
4.1	CFA for various sample implementations.	76
4.2	Computation of an early DFM metric for the MIPS datapath.	87
4.3	Statistics for datapath implementations considering routing blockages.	89
4.4	Computation of an early DFM metric for MIPS system.	90
1	Results for HPM implementations	103

Acronyms

ACLV	Across Chip Linewidth Variation
AOI	And-Or-Invert
BEOL	Back-End-Of-the-Line
CAA	Critical Area Analysis
CD	Critical Dimension
CFA	Critical Feature Analysis
CMOS	Complementary Metal Oxide Semiconductor
CMP	Chemical Mechanical Polishing
D2D	Die-To-Die
DEF	Design Exchange Format
DFM	Design for Manufacturability
DFY	Design for Yield
DRC	Design Rule Check
EDA	Electronic Design Automation
EUV	Extreme Ultra Violet
FA	Full Adder
FEOL	Front-End-Of-the-Line
FPGA	Field Programmable Gate Array
HA	Half Adder
HIL	High Index Liquid
HPM	High Performance Multiplier

IC	Integrated Circuit
IDV	Integrated Design Verification
ILD	Inter Layer Dielectric
IP	Intellectual Property
LER	Line Edge Roughness
LSB	Least Significant Bit/(Byte)
LVS	Layout Versus Schematic
MIDAS	Model for IP-inclusive DFM Assessment of System manufacturability
MOL	Middle-Of-the-Line
MSB	Most Significant Bit/(Byte)
NDS	Normalized DFM Score
OAI	Off-Axis Illumination
OPC	Optical Proximity Correction
P & R	Place and Route
PPG	Partial Product Generator(/Generation)
PPRT	Partial Product Reduction Tree
PLL	Phase Locked Loop
PSM	Phase Shift Masking
RDF	Random Dopant Fluctuation
RDR	Restricted Design Rule
RET	Resolution Enhancement Technique
RTL	Register Transfer Level
SoC	System-on-Chip
SRAF	Sub-Resolution Assist Feature
SRAM	Static Random Access Memory
TDM	Three Dimensional Method
VCTA	Via Configurable Transistor Array
VeSFET	Vertical Slit Field Effect Transistor
VHDL	VHSIC Hardware Description Language

W2W Wafer-To-Wafer

WDM Weighted DFM Metric

WID With-In-Die

WYSIWYG What You See Is What You Get

Part I

Introduction

Any sufficiently advanced technology is indistinguishable from magic.

~Clarke's Third Law

1

Introduction

1.1 Background

Gartner predicts worldwide semiconductor revenue figures of close to \$316 billion in 2013 [1]. This insatiable demand for high performance electronics in any number of application areas is driving further innovation in the area. Advances in Electronic Design Automation (EDA) and manufacturing techniques have resulted in the development of compact, feature rich mobile devices. The reduced device sizes that make this possible result in densities of the scale of billions of transistors per chip. This reduction, termed scaling, has continued unflinching for the last four decades, with the density doubling roughly every two years (figure 1.1) [2].

Traditional scaling now faces challenges brought on by the minute geometries that devices exhibit. The small device geometries expose second-order effects which were not dominant in nodes larger than 180 nm, causing performance penalties. The second-order effects in turn are caused by imprecise physical geometries of the manufactured circuits or impurities in the manufacturing environment. These deviations, in physical

Microprocessor Transistor Counts 1971-2011 & Moore's Law

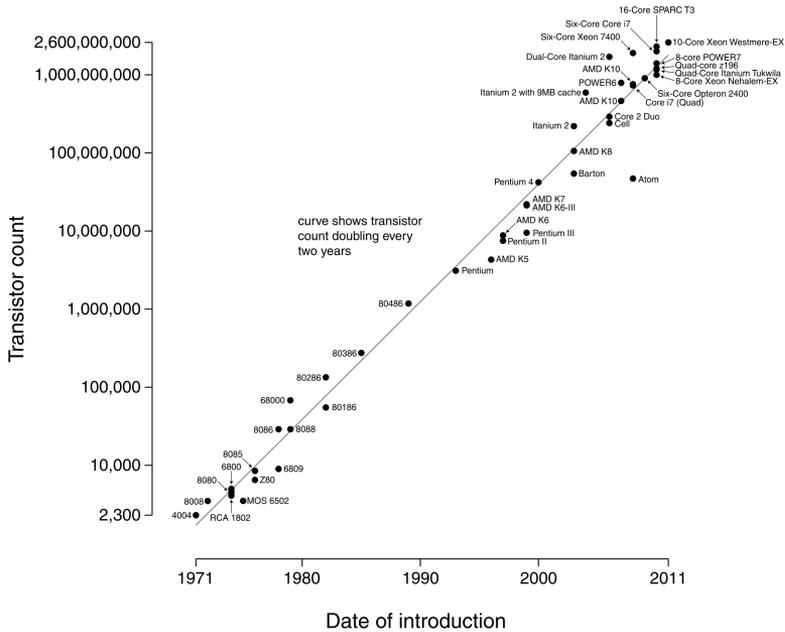


Figure 1.1: Transistor counts for integrated circuits by year [2, 3].

geometries and subsequently performance, are collectively termed as **Variability**. The most prominent of these variations has been leakage, i.e. the inability to completely turn off the transistor due to insufficient control over the channel. Furthermore, since the 180 nm technology node, Integrated Circuit (IC) manufacturing has been forced to prolong the use of sub-wavelength lithography. The rest of this chapter is largely devoted towards a brief overview of the IC design and manufacturing ecology and the problems that accompany them. The problem at hand is also introduced and the contributions of this thesis are listed to complete the introduction.

1.2 Design Flows

Scaling continues to play an important role in the way electronics are designed and manufactured. It is also worth noting that due to the complexity of modern designs, as much as 70% of the product development cycle is taken up by design verification [4]. On the system side, 52.8% of the design cost is software and the remainder, hardware [5]. While the fraction of hardware is lower, scaling has ensured that it is possible to attain densities of the order of billions of transistors per chip. The cost figures are also indicative of the steady transition towards systems with largely digital functionality. Given this trend, there may be as many as 100 Intellectual Property (IP) blocks in current designs [5]. The use of a large number of IP blocks in design is a more recent paradigm in the methodology of System-on-Chip (SoC) development. It has been brought on as a result of the integration made possible by scaling, the increased demand for functionality and the need for quick time-to-market. IP refers to functionally complete blocks such as memories or interface protocols (e.g. PCIe, USB) provided for use to the customer with verified functionality guarantees from the vendor. Such blocks ease the system integration phase of SoC development from the perspective of verification. The caveat here however, is simply that IP selection becomes exceptionally important in order to ensure that system-design related budgets such as power, timing and area are satisfied. IP integration in nanometer nodes must also satisfy the yield budgets of the overall design and thus be qualified as such. An overview of the requirements for IP may be found at [6, 7]. It is worth noting that, although recent advances have brought synthesizable analog blocks closer to reality, these are a few standard blocks like Phase Locked Loops (PLLs) [8, 9] and still not widely adopted.

In the face of such developments, traditional design flows have been adapted to accommodate the increased requirements. Figure 1.2 shows three distinct flows employed in the industry today. The analog design flow is still largely full custom in nature. This means that circuits are created from individual transistors. At nanometer geometries, these circuits are more susceptible to the effects of variability and, as such, a number of the techniques used to margin against the effects of variability were originally developed for use in the design of analog circuits. From a design perspective, the traditional divide of analog vs. digital, and full-custom vs. semi-custom design still continue to hold significance.

Analog circuit design is still largely dominated by traditional techniques involving full-custom design practices. These are modified at various stages to take into account the quantum mechanical device effects due to scaling but the circuits are still fundamentally designed at the transistor level. The most obvious impact in terms of methodology,

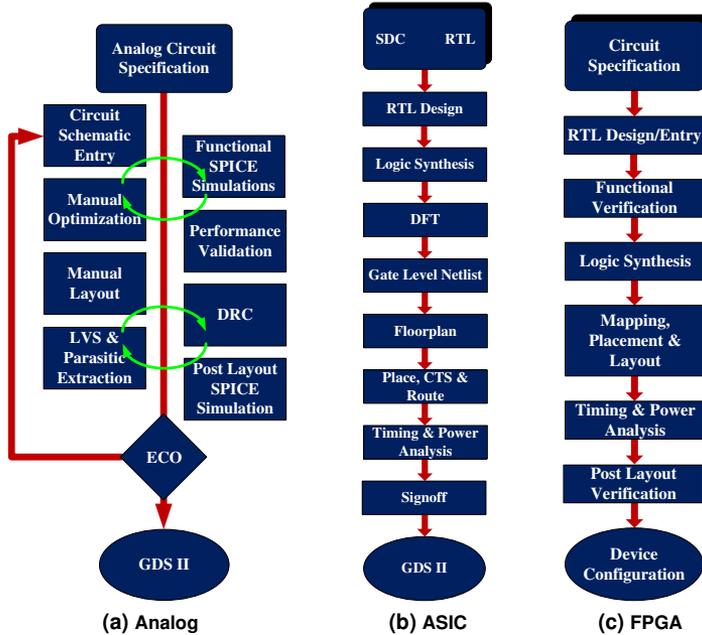


Figure 1.2: Design Flows for Analog, ASIC and FPGA methodologies.

is the number of design rules that must be fulfilled in order to be able to qualify a chip for fabrication. The number of DRC rules that must be satisfied has grown from a few hundred in the micron-scale nodes to a few thousand for the latest nanometer-scale nodes. Granted that a lot of the rules are imposed to ensure manufacturing compatibility, but nonetheless, this is one obvious aspect of design that has changed to meet the demands of scaling. In spite of the increased number of design rules to be satisfied, other forms of variability are introduced due to inaccuracies in the manufacturing process [10, 11]. Resilience to these forms of variability are addressed through more stringent verification techniques (like Monte-Carlo Simulations) prior to signoff. As geometries continue to scale further, statistical techniques depending on the process parameters are being introduced into the signoff checks. This has given rise to what is termed as Statistical Design [10] which incorporates a holistic view in an attempt to create robust designs. These techniques rely on probabilistic distributions of different variability parameters to assess the performance under different conditions. The objective is to obtain as many circuits performing as close to the performance envelope as possible in a given manufacturing lot.

Digital IC design, on the other hand, has shifted to semi-custom design techniques for all but the most high performance designs. The abstraction introduced in this type of flow, in the form of libraries of logic, clock and special (e.g power switches, level shifters etc.) cells, ensures that the designer can neglect some of the variability issues at the device level, at least in the early stages of design. The extra effort required in dealing with scaling related issues is restricted to some steps of the flow, typically in the physical design¹ stages. Compared to analog IC design however, application of design automation to large parts of the semi-custom methodology makes it easier to deal with the functional implementation aspects of a digital design. The latest EDA tools offer sophisticated techniques to perform design and verification tasks with great accuracy. With mounting costs and short turn-around-times, this sophistication eases the burden on the design and verification engineers. Passive design techniques like wire widening and via doubling ensure that variability does not cause catastrophic failure, while other tools give the designer the capability to perform statistical analyses to ensure that timing and power budgets remain unaffected.

The proliferation of Field Programmable Gate Arrays (FPGAs) and the related design tools in the last few years means that there is yet another viable option for digital designs. FPGAs today are increasingly sophisticated, building on general technology scaling. A number of complex macros are available in the high-end devices in the market today, making development of complex designs much easier and cheaper. This is at the cost of performance, but nonetheless, given the high costs involved in volume production it is an extremely competitive alternative. In the case of FPGAs, since the device itself is pre-fabricated in a given technology node, the designer relies on better design and architecture techniques to add sufficient resilience to combat the effects of some forms of parametric variability which could still affect functionality. Special EDA tools are available from the FPGA vendors with features that ensure optimal implementation. The flow is similar to a traditional digital IC flow, except in the physical design stage. Once logic synthesis is complete, mapping allocates device resources to the design. Given that the electrical parameters of the device are predictable, the remainder of the physical design flow is to ensure that the successful routing can be carried out while meeting the timing budgets. The tools then convert the solution into a bitstream that can be used to program the FPGA.

At this point it is useful to observe that, with continued scaling the effects of variability traditionally impacting analog circuits now have a prominent impact on digital

¹ Physical Synthesis is increasingly becoming relevant in the bleeding edge technologies. The interested reader may refer to [12–15].

circuits as well. Consequently a number of the techniques used to combat variability are equally applicable to analog as well as digital ICs. The limitations due to manufacturing impact the design process indirectly. The techniques used to margin against variability and enhance manufacturability is referred to as DFM or Design for Yield (DFY), where the term yield refers to the percentage of chips in a given lot that fulfill the performance criteria.

1.3 Scaling and Manufacturability

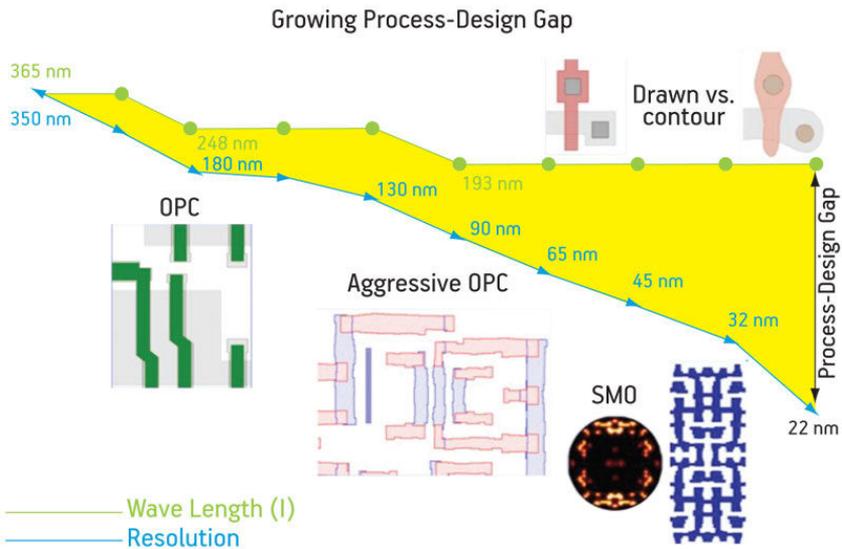


Figure 1.3: Lithography source wavelengths against feature size [16].

The 45 nm process from Intel [17, 18] introduced hafnium-based compounds as high-k dielectrics in combination with a metal gate. This results in a number of benefits, chiefly, lower leakage current in the device. These material innovations have so far kept Moore’s Law [2] on track without compromising the benefits of scaling. Concurrently, manufacturing of CMOS circuits has traditionally relied upon lithographic techniques to achieve mass production. The wavelength of the light source used to perform the lithography is an important parameter in the assessment of the fidelity of the pattern being etched on the die. The early lithography processes used 436 nm (“g-line”), 405 nm

(“h-line”) and 365 nm (“i-line”) mercury lamp based sources to achieve patterning. The development of laser based lithographic techniques revolutionized the production process and enabled continued scaling. Today 248 nm Krypton-Fluoride based and 193 nm Argon-Fluoride based excimer lasers are widely used in the process of feature patterning.

For feature sizes above the wavelength of the light source, the imaging produces patterns at high fidelity (What You See Is What You Get (WYSIWYG)). When feature sizes require sub-wavelength patterning this trend of WYSIWYG breaks down resulting in problems with the fidelity of the patterns being produced. This results in a so called Process-Design Gap (see figure 1.3), requiring expensive corrective measures to achieve the required fidelity. Mismatches between the intended pattern and the fabricated pattern primarily become visible due to insufficient lithographic accuracy for sub-wavelength patterning. Other process steps like Chemical Mechanical Polishing (CMP) and etching (used extensively in the creation of trenches and in the interconnect stack) are also difficult to control and lead to defects. These steps can directly cause open or short circuits and indirectly affect the lithographic process by creating a non-uniform patterning surface. Line end shortening, Line Edge Roughness (LER) and corner rounding are typical defects caused due to lithographic inaccuracy, in turn causing parametric variations like threshold voltage (V_{th}) variations and increased leakage currents. Dishing and erosion are typical defects of the CMP and etch process leading to open or short circuits. Other defects due to CMP and etching, like particle defects, cause variations in the resistance and capacitance of vias used to move between different interconnect layers.

A number of Resolution Enhancement Techniques (RETs) are used to avoid lithography induced defects. Optical Proximity Correction (OPC) is a technique used to improve the patterning of dense features. For complex patterns close to the resolution limit of the lithographic system, Sub-Resolution Assist Features (SRAFs) are employed by way of introducing features on the mask to make less dense areas denser. The difference between these two techniques lies in the fact that while both these techniques are employed on the masks, the SRAFs are never fabricated. It is worthwhile to note at this point that lithographers often refer to the Critical Dimension (CD) or Resolution and the Half Pitch. All of these terms refer to the geometric resolution capability of the lithographic system. The CD is defined as $CD = k1 \frac{\lambda}{NA}$ where λ is the wavelength of the lithographic source, NA is the numerical aperture of the imaging system and $k1$ is a factor indicating the aggressiveness of the lithography. The $k1$ factor under normal conditions of Rayleigh optics has a limit of 0.5 while the NA is limited to about 0.95 for systems using air as the medium to perform lithography. However, by employing techniques like Off-Axis Illumination (OAI) and Phase Shift Masking (PSM) along with

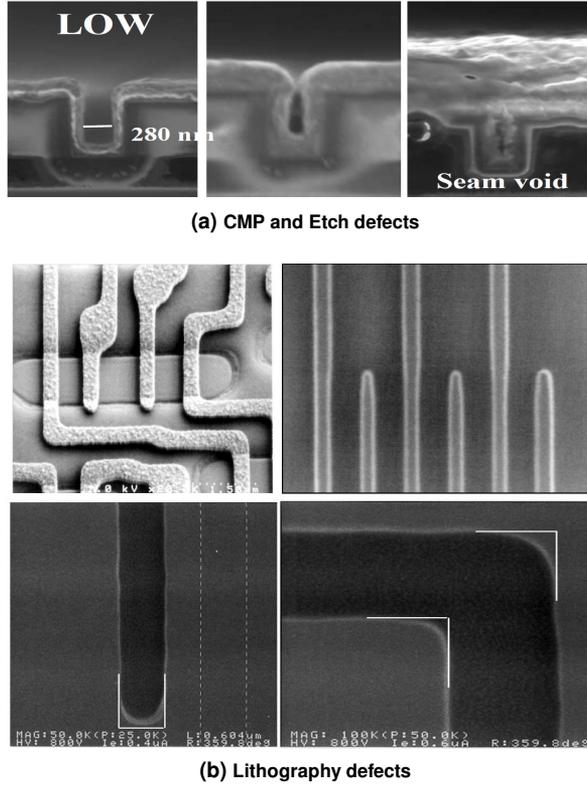


Figure 1.4: Defects introduced due to the manufacturing process (Source: IMEC).

aggressive OPC, the k_1 factor can be reduced to 0.25. Further, by using water as the medium to perform lithography the NA can be improved to 1.35 and with the use of High Index Liquids (HILs) increased to 1.65. Applying double (multiple) patterning the effective k_1 factor can be reduced to lower than the fundamental limit of 0.25. Thus with the current techniques based on 193 nm wavelength lithography a resolution of around 20 nm can be achieved before prohibitive cost prevents any further use of these techniques.

While these advances no doubt maintain the progress of Moore's Law [2], the trade-off in this case is the cost of production. Lithography using Extreme Ultra Violet (EUV) light at a wavelength of 13.5 nm shows significant promise to the continued progression of Moore's Law [2], but suffers from a number of technical challenges and cost

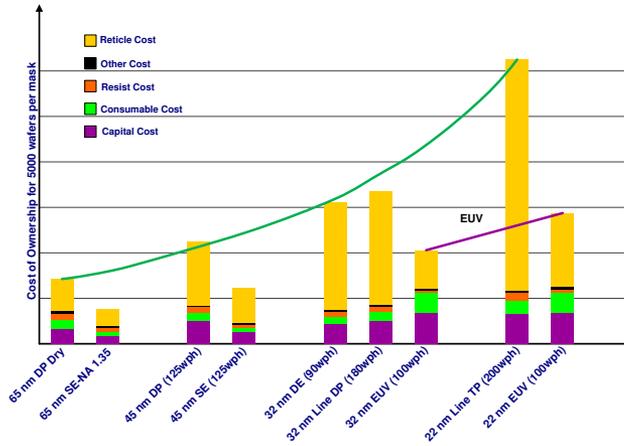


Figure 1.5: Relative cost of ownership of a 5000 wafer run device. [Adapted from [19]]

factors affecting widespread deployment². Figure 1.5 shows the cost of ownership of lithography equipment used in modern fabrication processes.

Other alternatives like maskless lithography and directed self assembly are in various stages of research but still have issues before they can be reliably deployed in production.

1.4 Problem Statement and Scope of Work

The preceding parts of this introduction have so far presented a current state of affairs relating to the design and manufacturing of circuits. However, a clear picture of the problem has not emerged. I state the problem here as follows:

In the face of increasing production cost, is there a viable means of designing variability resilient circuits and measuring manufacturability?

Given the cost constraint part of the problem, an obvious insight is that variability resilience must be conceptually built into any methodology used to develop electronic designs. The complexities of the design process alone make it obvious that methods to mitigate variability must be applicable across different levels of abstraction. Following the discussion from the previous section, it is clear that a number of the problems posed

² Given the continual delay of EUV, companies are planning the extension of 193 nm lithography until 7 nm node. See [20] for details.

by scaling are due to the geometric density of the layouts leading to patterning problems and hence parametric variation. It can then be argued that using regular patterns at regular pitches can address some of these issues. This has a direct impact on the cost as the mask creation process, now no longer requiring aggressive OPC in all steps, becomes cheaper. Noting that mask costs are a significant part of the production cost and further noting that due to the reduced number of process steps the production is quicker brings out the cost advantages of this method. However, it is also imperative that the performance advantages of scaling are not negated. Therefore, it is important to identify the contributing factors leading to complex masks and address those issues within the framework of the methodology.

Regular circuits have been proposed as candidates for variability resilient circuits since the 1990s. Early work in the area, addressing regularity of standard-cell designs, has not been re-investigated to the best of our knowledge. Owing to the fact that end goals were significantly different to the considerations today, this work was not leveraged in standard flows. At the abstraction level of semi-custom design, one can refer to placement regularity, i.e. the regular placement of standard-cells, and routing regularity. Chapter 2 deals with this aspect of regularity and a novel methodology incorporating regular placement of standard-cells is investigated. Routing in standard semi-custom flows is driven by heuristic algorithms in order to obtain a robust effort-performance tradeoff. Keeping in mind the representative gate counts of modern designs regularity related to routing is not actively investigated. Analysis of the results from this study were convincing enough for me to move to the next phase; a study of the interactions of transistor level layout regularity to methodology steps in semi-custom design methodologies. Though the adoption of geometries with minimal corners and unidirectional resources³ result in extreme device level layout regularity, there is no consensus on any quantification of regularity. When considering standard-cells, the abstraction makes regularity even harder to quantify since a tractable measure for the regular connectivity of random logic is hard to define.

Chapter 3 presents a detailed study of the factors influencing regularity, their relationship to related aspects of variability and manufacturability, and the impact of the regularity so imposed. A set of logically complete standard-cells was developed for the purpose and DFM was quantified using CFA tools from Mentor Graphics [21]. The rule-set used to check DFM was provided by the foundry. This work was extended by implementing the ISCAS '89 benchmarks using the cells developed and various fac-

³ I apply *resource* here as a term encompassing both active geometries and geometries related to gate formation and routing.

tors affecting the implementation and manufacturability were studied. The results of this study made necessary an extension of the custom library to include cells with more functionality. The new library was used to implement a test processor design incorporating a L1-cache sub-system. The cache was implemented using macros provided by the foundry. DFM was analyzed for this design using the same CFA method. Finally, with a view to predicting DFM early, a model christened Model for IP-inclusive DFM Assessment of System manufacturability (MIDAS) is developed. Chapter 4 presents this model which is based on SoC implementation statistics and the existing DFM metrics presented in Chapter 3. Standard-cell costs are computed using CFA techniques while routing costs are computed using SoC implementation statistics. IP costs are also accounted for in this model, thus allowing for its use in modern SoC designs. The metric produced by MIDAS is a weighted sum of the standard-cell, IP and routing costs. The model is demonstrated on the processor datapath and the embedded processor incorporating the L1-cache sub-system.

Due to the nature of the problem stated above, the scope of this work starts from a standard semi-custom methodology and then shifts into a lower level of abstraction in order to fully assess the contributing factors to a methodology relying on regularity to mitigate variability. That said, within the scope of this work, arithmetic functional units, the ISCAS '89 benchmark circuits and a processor design with an L1-cache sub-system are used as test vehicles. EDA tools which are standard in industrial implementation flows are used for the implementations in this thesis. Industrial tools are used to implement and characterize the custom-created standard-cell libraries that were developed for the purpose of this work. In line with the problem statement, the broader expectation of this entire thesis is to be able to predict the effect of implied constraints on manufacturability and yield.

Bibliography

- [1] J. Rivera, "Gartner Says Worldwide Semiconductor Revenue Grew 5.2 Percent in 2013," 2014, [Online Source].
- [2] G. E. Moore, "Cramming more Components onto Integrated Circuits," *Electronics*, vol. 38, no. 8, pp. 114–117, Apr. 1965.
- [3] W. G. Simon, "Transistor Counts by Year and Moore's Law - Extended to 2011," 2012, Wikimedia Commons. [Online Source].

- [4] J. Bergeron, *Writing Testbenches: Functional Verification of HDL Models, Second Edition*, Springer US, Feb. 2003.
- [5] D. Nenni, “Semiconductor Ecosystem Keynotes: ARM 2012 (SemiWiki blog article),” 2012, [Online Source].
- [6] E. Sperling, “Which ip is better?,” 2014, [Online Source].
- [7] C. Snyder, “The Integrated IP Subsystem: A Converging SoC Solution (SemiconductorEngineering article),” 2014, [Online Source].
- [8] R. A. Rutenbar, “Emerging Tools for Analog and Mixed-Signal: The Role of Synthesis and Analog Intellectual Property,” 2012, PDF Presentation. [Online Source].
- [9] N. Nandra, “Synthesizable Analog IP,” 2010, [Online Source].
- [10] M. Orshansky, S. R. Nassif, and D. Boning, *Design for Manufacturability and Statistical Design*, Springer Science+Business Media, LLC, 2008.
- [11] C. C. Chiang and J. Kawa, *Design for Manufacturability and Yield for Nano-Scale CMOS*, Springer Science+Business Media, LLC, 2007.
- [12] C.J. Alpert, C. Chu, and P.G. Villarrubia, “The coming of age of physical synthesis,” in *IEEE/ACM Int. Conf. on Computer-Aided Design (ICCAD 2007)*, Nov. 2007, pp. 246–249.
- [13] C.J. Alpert, S.K. Karandikar, Zhuo Li, Gi-Joon Nam, S.T. Quay, Haoxing Ren, C.N. Sze, P.G. Villarrubia, and M.C. Yildiz, “Techniques for fast physical synthesis,” *Proc. of the IEEE*, vol. 95, no. 3, pp. 573–599, Mar. 2007.
- [14] Kai-Hui Chang, I.L. Markov, and V. Bertacco, “Safe delay optimization for physical synthesis,” in *Asia and South Pacific Design Automation Conference (ASP-DAC 2007)*, Jan. 2007, pp. 628–633.
- [15] D. Papa, C. Alpert, C. Sze, Zhuo Li, N. Viswanathan, Gi-Joon Nam, and I.L. Markov, “Physical synthesis with clock-network optimization for large systems on chips,” *IEEE Micro*, vol. 31, no. 4, pp. 51–62, July 2011.
- [16] J. M. Brunet, M. Redford, C. Thomas, and M. Scoones, “Using DFM for competitive advantage,” 2010, [Online Source].
- [17] T. Ghani, “Challenges and Innovations in nano-CMOS Transistor Scaling,” 2010, [Online Source].
- [18] C. Webb, “Intel Design for Manufacturing and Evolution of Design Rules,” 2008, vol. 6925, p. 692503, SPIE.

- [19] International Technology Roadmap for Semiconductors 2009 Edition, “Process Integration, Devices And Structures,” 2010, [Online Source].
- [20] M. LaPedus, “What If EUV Fails? (SemiconductorEngineering article),” 2014, [Online Source].
- [21] Mentor Graphics, *YieldAnalyzer and YieldEnhancer Reference Manual*, 2010, Calibre DFM Suite Datasheet.

Part II

Placement Regularity In Semi-custom Flows

Left to themselves, things tend to go from bad to worse.

~Murphy's First Corollary

2

Regularity and Semi-custom Design

2.1 Background

In the context of the options available in the design landscape, regularity of layout has been a topic of research since the 1990s. Kutzenbausch *et al.* considered the extraction of regularity at the logic synthesis stage [1]. Jenne *et al.* question the need for layout regularity based on their experiences with traditional Place and Route (P & R) tools and standard-cell based datapath design tools [2]. More recently, work carried out by Menezes *et al.* proposes regular layouts based on a single type of cell to investigate the effects of regularity [3, 4]. Using a custom synthesis tool they show results indicating an improvement of delay at the expense of area and wire length. However, the effects of scaling along with the consideration of cost now force us to consider enforced regularity as a means of maximizing manufacturability in advanced technology nodes. Subsequent sections in this chapter detail our methodology. This methodology is based on a domain-specific, low level, layout aware hardware description language, Wired, in combination with commercial synthesis and P & R tools applied to commercial standard-cell libraries.

2.2 The Wired Design Environment

Wired is a hardware description language built on the functional programming language Haskell [5]. The primary objective is to be able to describe the following aspects of a circuit:

- Logic function that can be interfaced to standard tools as a technology mapped netlist.
- Cell placement to create built-in layout awareness.
- Some basic aspects of the wiring to allow early assessment of the quality of results.

As a result of its roots in Haskell, Wired achieves a very elegant integration of these three domains. I present the basic aspects of Wired in relation to these domains in the following sub-sections.

2.2.1 Logic domain

The logical aspect of a design is described in a simple applicative style. Consider the logical function $a + b\bar{c}$. A Wired description of this function could take the form:

```
myFunc (a,b,c) = do
  c' <- ivsvtx2 c
  bc <- an2svtx2 (b,c')
  or2svtx2 (a,bc)
```

where `ivsvtx2`, `an2svtx2` and `or2svtx2` are an inverter, an AND-gate and an OR-gate of size $X2$. Wired natively invokes simple translations of standard-cells from characterized representations. A description in Wired completely defines the resulting netlist. The system also provides different means for analyzing the netlist. One such form of analysis useful in the context of the logic domain, is Boolean simulation through built in functions like `simulate`, an example of which is seen below.

```
*Main> simulate myFunc (0,1,1)
0
*Main> simulate myFunc (0,1,0)
1
```

Wired provides some simple yet powerful mechanisms for abstraction as is common to all functional programming languages. The techniques that aid this abstraction and are most relevant in the current context are recursion and higher-order functions¹. One commonly used function which is both recursive and higher-order, is `mapM`. We can, for example, use it to define a bit multiplier:

¹ A **Higher-order Function** by definition, is a function that takes other functions as arguments or returns a function as result.

```
bitMult (a,bs) = mapM bitMult1 bs
where bitMult1 b = an2svtx2 (a,b)
```

Another useful, non-recursive, symbolic combinator is `>=>`, read as “composition”. If we wish to replace the AND-gate in `bitMult` with a NAND-gate and an inverter, one way to accomplish this operation is:

```
bitMul' (a,bs) =
mapM (bitMult1 >=> ivsvtx2) bs
where bitMult1 b = nd2svtx2 (a,b)
```

Note that we have replaced the first argument to `mapM` by the composition `(bitMult1 >=> ivsvtx2)`. This simply means that the result of each `bitMult1` will be inverted.

2.2.2 Placement



Figure 2.1: *Postscript rendering of a Wired description.*

As with any methodology used to create complex designs, in Wired too the first step involves the creation of a purely logical description, similar to the ones demonstrated above. Placement constraints are added separately without interfering too much with the original description. Wired expresses relative placement through user-provided constraints. This is especially useful in datapath circuits which lend themselves to algorithmic descriptions. Visualization is then achieved by executing the `renderWired` command after first instantiating the Wired design to a specific size. This produces a

postscript file showing the exact sizes of the corresponding library cells, like in figure 2.1. We now have a description that has both logical and geometric (layout) aspects associated with it. The Wired description can now be converted to a format that can be read by physical synthesis tools such as Cadence SoC Encounter. The *de facto* standard for such a format is the Design Exchange Format (DEF). Wired enables export of designs to this format using the `exportDEF` command. The `exportDEF` command produces a DEF file containing the netlist and absolute coordinates for each cell in the logical description described in Wired.

2.2.3 Wiring

One of the goals of the Wired system is to enable better control over performance, by providing the ability to assess the effects of the imposed placement constraints taking into account the routing. This is primarily achieved in Wired through wire-aware performance analysis enabled through a timing analyzer that takes estimated wire loads into account. In order to assess the delay of a circuit we apply the `analyzeTimingW` command to it. This timing analysis is meant only to serve as a quick reference in the process of layout exploration and a more detailed analysis can be achieved with more refined wire load models in the downstream methodology. The combination of wire-aware performance analysis and a flexible description language enables convenient wire-aware design exploration.

The preceding sections are a very light treatment of the Wired environment, meant to be a gentle introduction to its capabilities. For details about the implementation of Wired and its complete set of capabilities please refer to [6].

2.3 Related Work

Considering the abstraction level of the work discussed in this chapter, the discussion of related work here is restricted to methodologies that provide layout aware controls. In the larger scheme of things, manufacturability-aware standard-cells have also been studied in great detail. Related work discussing this aspect will be presented along with the relevant work, in the next part of this thesis.

The TEGO design accelerator [7] is a structured design tool from Tuscany Design Automation. Structural design techniques have been used in the industry to perform design exploration in order to achieve the best performance with the least possible area. TEGO offers the designer a graphical interface to perform such micro-architectural

structural experiments to quickly assess the impact of different floor plan decisions on parameters like wire length, timing, power and area. Tuscany also provides the designer with a structural language to help port IP to new nodes and generate variants in the same node. Since the macros are treated as pre-placed instances not requiring hard macros, the tools allow a great deal of flexibility in the design exploration phase.

The Integrated Design Verification (IDV) [8] system developed at Intel provides a highly integrated design environment aimed at reducing the long verification cycles typically seen in the digital design flow. This system combines a correct-by-construction and correct-by-verification scheme along with a database of verified results allowing rapid design development with smaller verification effort. In the early stages of development high level models are reduced through algorithmic transformations to achieve a viable micro-architecture. A logical implementation is derived from this for physical implementation. The IDV environment allows a high degree of integration between the logical and physical implementation phases of a design resulting in shorter overall development cycles.

In the methodology presented here Wired provides layout awareness at a fine grained cell level. TEGO works primarily at the block level and uses cell level information to improve area utilization. Additionally, while Tuscany provides a structure language to enable parametrization, this would still rely on legacy RTL descriptions to completely leverage the advantages of the same. In comparison to this Wired provides an environment where parametrization can be applied at the time of assessment while still enforcing regularity through the placement constraints. This is made possible since Wired, being based on Haskell, treats inputs as lists. Thus, while any attempt at a physical realization requires a finite size, enforced regularity constraints may be generally applied to a description meant for layout exploration.

IDV is similar to Wired in its enablement of design space exploration. However, it encompasses a much broader scope while keeping the steps of a traditional semi-custom methodology intact. Wired directly captures placement constraints in parametrized datapath descriptions and enables Boolean simulation through built in functions. Since Wired also provides the designer with the ability to interface to standard tools, other standard verification methods may also be applied. Additionally, since native descriptions of library cells are used synthesis may be completely avoided in certain cases.

Design Compiler, a commercial synthesis tool available from Synopsys employs special algorithms to extract datapath circuits from RTL descriptions [9]. Support for context driven multipliers, adders, shifters and selectors is available with extensions for special operations such as squaring and blending. Support is also provided for special

conditions such as a decoder implemented using a shifter and robust architecture selection is provided for improving timing and power. Dhumane *et al.* [10] propose a lithography aware standard-cell placement methodology that concentrates on mitigating lithography induced cell abutment errors through the uses of Edge Placement Error (EPE) based standard-cell library characterization, placement optimization techniques such as cell re-orientation, cell swapping and placement blockage creation. SRAF characterization and insertion for the purpose of enhancing printability of features across abutting cell edges is another feature of this methodology.

Neither of Design Compiler [9] or the work by Dhumane *et al.* [10] consider regularity explicitly but rather work with different considerations from either end of the design flow. By comparison, Wired is a generalized solution applicable to any circuit and, with knowledge of the manufacturing limitations, can be applied to deal with issues such as addressed by Dhumane *et al.* RegPlace [11] is a integer linear programming based placement tool that has been proposed for placement tasks on pre-fabricated regular fabrics called Structured ASICs. Though related to the regular layouts discussed throughout this part of the thesis, this topic falls outside the scope of the present discussion.

2.4 Methodology

One of the current shortcomings of the Wired description system is the inability to accurately represent and simulate sequential logic. While the description of a sequential element may be forcibly included for placement purposes, the methodology here is built around a tenet of non-disruptive development. This implies that Wired is used for the development of regular blocks which are often combinational in most modern digital designs. This fits well with the accepted practices of synchronous digital design due to the fact that in most logic dominated circuits flip flops are used to achieve timing closure. Also this does not in any way hinder the development of a modular design using random control logic in addition to data path circuits which are more regular in nature. The methodology is based on black box integration allowing for multiple blocks to be integrated. The complete scope of the methodology is shown in figure 2.2a. The RTL description at the logic synthesis stage is meant to enable efficient black box creation and integration in the physical design stage. While figure 2.2a indicates that the standard-cell library is used by Wired, it should be emphasized that this is only symbolic. Wired uses a native version of the cell library with information relevant to its operation.

In this flow, parts developed in Wired are integrated in the physical design stage. Prior to this for logic synthesis purposes black box modules are used to represent mod-

ules developed in VHDL. The specific integration steps are shown in figure 2.2b. Care should be taken to ensure that the port descriptions are uniformly maintained throughout the flow. This also implies that the hierarchy needs to be accurately maintained. Black box descriptions are used to initialize the floorplan and partitions in the physical design stage. Pre-placement is then carried out on the physical hierarchies and black boxes. The partitions are then developed individually and then integrated. It is worthwhile to note here that the DEF produced by Wired contains placement constraints of the PLACED type. This can be problematic if proper care is not taken to ensure that the desired placement constraints of the blocks imported via Wired are not made permanent once the floorplan details have been fixed during physical design. Wired generally provides generously proportioned dies depending on the placement constraints specified, so it will often be necessary to adjust the area budgets during the floorplanning and integration stages of the flow. Once integration is completed, the remaining steps involved in setting up the power delivery network, clock tree synthesis and routing are implemented as usual. The final steps towards creating a manufacturable design involve conventional DRC checks and simulation based functional verification. The final design is written out in the GDSII format.

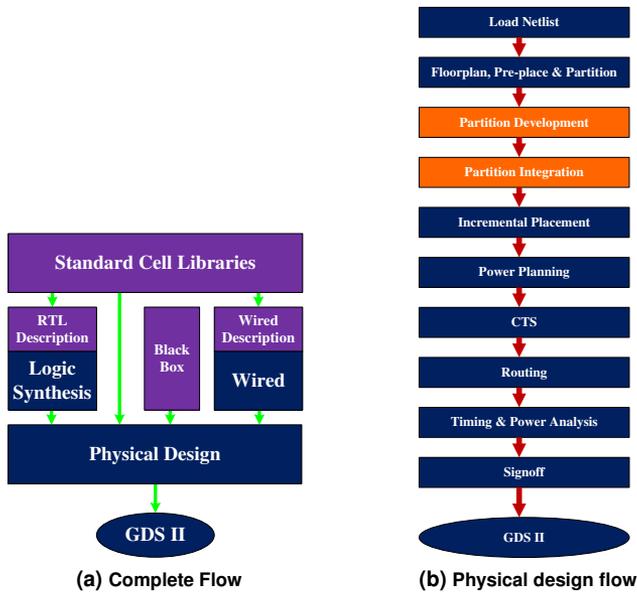


Figure 2.2: Methodology to enforce placement regularity using Wired.

Often the signoff checks occur in the full custom design environment and involve DRC checks on the polygons that make up the design. In addition, manufacturability checks as they exist today may also be implemented in this environment, on top of the DRC checks.

2.5 Case Studies to verify the methodology

While the methodology developed in the previous section is applicable to any design in which placement constraints are desired, the primary objectives were:

1. To develop a methodology to enforce regularity of placement at the standard-cell level of abstraction.
2. To assess a design implemented using such a methodology with the goal of assessing variability resilience at the least possible impact to performance and area.

In all the case studies chosen, there was some inherent regularity present making it amenable to use with Wired. Other implementations presented in the results are either variants of the regular netlist or chosen to be comparison cases. The case studies chosen for this study are presented here.

2.5.1 Barrel Shifters

Shifters are combinational circuits that shift the value on the inputs either left or right. The shift itself is accomplished by connecting the inputs to multiplexers in some fashion. When a shift of more than one bit position is required, a barrel shifter is used. Note that the barrel shifter is also used to perform a rotation when the LSB(MSB) takes the value of the MSB(LSB) when a one bit left(right) shift operation is performed. An example of a conventional barrel shifter that is capable of arithmetic and logic right shift operations is shown in figure 2.3 (see [12] for the published text). The circuit in figure 2.3 can be extended to perform both left and right shift operations, by adding some additional multiplexers on the input and output which reverse the input data set when necessary. This shifter has an 8-bit input and is capable of 7-bit shift operation.

If built using 2-to-1 multiplexers, these kind of shifters generally have $\log_2(N)$ logic depth, where N is the input size. They are capable of (N - 1)-bit shift operation. The shift type depends on the *in'* input: If *in'* = '0' a Logic Shift Right should be performed, but if *in'* = 'MSB'('in7') an Arithmetic Shift Right should be performed. This shifter can be built using different standard-cells [13]. The advantage of using multiplexer cells is that

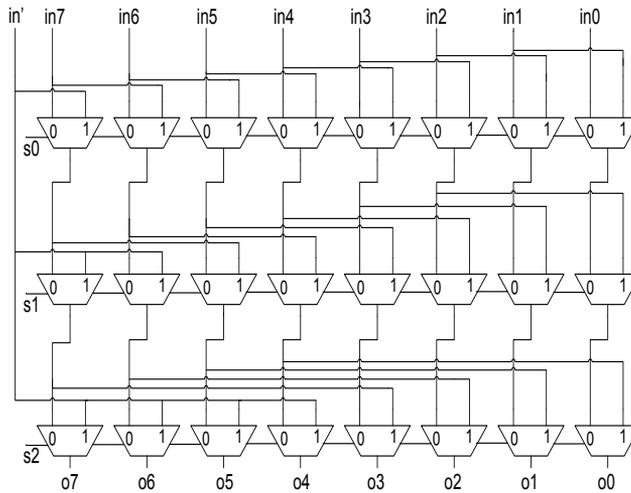
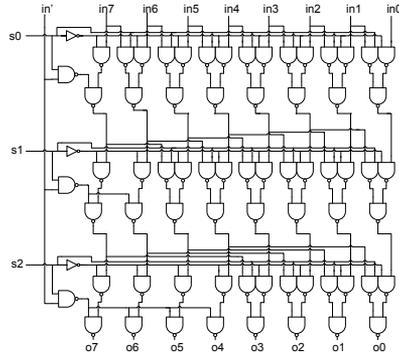


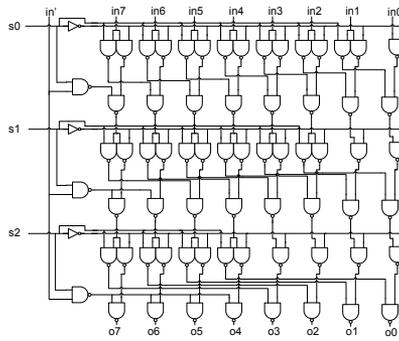
Figure 2.3: Barrel shifter structure.

the layout area is small, but faster shifters can be generated using basic logic standard-cells like NAND gates. The circuit in Fig. 2.4a shows an 8-bit shifter built using NAND gates. The even rows are in fact OR gates, while odd rows function as AND gates, but in accordance with De Morgan's Laws the circuit can be built using NAND gates only. Some of the NAND gates near the MSB side are removed as a simplification, since it is enough to create the in' signal chosen by the select signal one time for every stage.

A layout technique called fan-out splitting has been proposed for cyclic shifters [13]. The same technique can be applied to both arithmetic and logical shifters, but it is more advantageous on cyclic shifters in which wrap-around wires incur a larger wire load on the critical path [14]. The fan-out splitting technique separates the shifting and non-shifting paths. On each stage shifted and non-shifted signals are generated with a demux structure and they are collected using OR gates after every demux stage. The main advantage of the separated shifting and non-shifting paths is that the wire load on the critical path will be smaller [13]. The circuit in figure 2.4b shows an 8-bit shifter using fan-out splitting. It is also constructed using NAND gates. On the LSB side of the multiplexer based shifter there are some signals, including $in0$, which are only selected when the select signal is logic-0. This simplifies the circuit on the LSB side for this shifter, since those signals do not need a full demux structure; a single AND gate with an inverted select signal is sufficient.



(a) NAND gates



(b) NAND gates with fan-out splitting

Figure 2.4: Barrel shifter structures using NAND gates.

2.5.2 Logarithmic Depth Multipliers

Logarithmic depth multipliers are so called, because of the logarithmic relation the delay shares with the operand word length; the delay scaling as a function $O(\log_{\beta}(N))$ where N is the operand width. This class of multipliers is also called Column Compression multipliers since they rely on column-wise reduction techniques to achieve optimal timing. A number of column compression techniques have been developed over the years since they were first introduced by Wallace [15] in 1964. Common to all of these techniques is the process used to achieve the multiplication operation. The three steps in column compression namely, partial product generation, partial product reduction and final addition, lend themselves easily to architectural blocks performing the operations indicated by the name. The Partial Product Generator(/Generation) (PPG) can be simple

or use methods such as Booth or Modified-Booth when signed multipliers are desired. There are also a number of options for the final adder among the family of fast, parallel prefix adders. The different variants of log-depth multipliers arise from the different methods used to achieve partial product reduction, which is the most resource intensive portion of the multiplier.

Dadda Multiplier

The Dadda multiplier [16] is similar to the Wallace multiplier [15], displaying the same $O(\log_{\frac{3}{2}}(N))$ reduction as the Wallace multiplier. It is however different from the Wallace multiplier in its objective of achieving the multiplication using as little hardware as possible. Using only half and full adders necessary to reduce the rows of the partial product matrix corresponding that correspond to the progression [17]: $X_0 = 2, X_{i+1} = \lfloor \frac{3}{2} X_i \rfloor$.

HPM Multiplier

The High Performance Multiplier [17] scheme, developed at Chalmers University of Technology, is a variant of the Dadda algorithm. It retains the advantages of logarithmic depth that the Dadda algorithm offers but, also achieves regularity in layout by following a different order of assigning sum, carry and partial product bits to the adder cells. For each step of the HPM scheme, carry and sum bits produced at one level are consistently placed below the bits that remain to be compressed, when they are transferred to the next level meaning that they get compressed as late as possible. It is left to the outcome of implementations of this methodology to investigate in detail whether it is feasible to try to achieve regularity of routing, that is inherent to the algorithm (and implementable using full custom techniques at the expense of effort) using automated routing algorithms.

TDM Multiplier

The TDM of multiplication, developed by Oklobdzija *et al.* [18], is the fastest known multiplier implementation. This optimization of speed is achieved by algorithmically considering cell delays and sorting signal delays when assigning carry propagation in the partial product reduction stage of log depth multiplication. Thus, by assigning the shortest delays first, the overall delay achieved in column compression is near optimal for any input size globally. In the course of algorithmic multiplier creation it is necessary to differentiate fast and slow inputs, making the availability of characterized data for the constituent cells a factor for accuracy.

Stelling *et al.* [19] demonstrated the trade-off between the output carry vector and the output sum from a column. A multiplier based on a heuristic that produces a shorter sum delay and an acceptable carry vector could thus produce lower overall delay in certain cases, but matches the delay displayed by multipliers created using the original algorithm.

2.6 Results

The case studies presented above were implemented, initially using the 90 nm process node offered by ST Microelectronics and later on moving to the 65 nm technology node². Other than Wired, the tools used in this flow are Cadence RTL Compiler for logic synthesis and Cadence SoC Encounter for the physical design steps of the tool flow. I will present the results for each case study along with the test conditions.

2.6.1 Shifters

Adopting the methodology described in section 2.4, the different descriptions of the barrel shifter were annotated in Encounter. The DEF file produced by Wired was used as the initial input in each case, but annotated in three different ways [12]:

- The placement adopted in Wired was preserved entirely and the floorplan area was reduced to an extent that ensured both routability and error-free placement. Only the routing engine of Encounter was employed to complete the routing of the design. The results of such a flow are placed under the '*Wired*' column in the tables below.
- The placement adopted in Wired was abandoned entirely and the default floorplan area was used to place and route the design. From the point of view of a conventional flow, this represents the maximum freedom available to the P & R tool. The results of such a flow are placed under the '*Tool Driven*' column in the tables below.
- The third strategy allows the P & R tool a limited freedom of placement, but a complete routing freedom. This was done by employing fences to the various physical hierarchies in the netlist, to improve area efficiency. The results from such a strategy are placed under '*Fenced*' in the tables below.

² The specific implementation technology will be provided in each case.

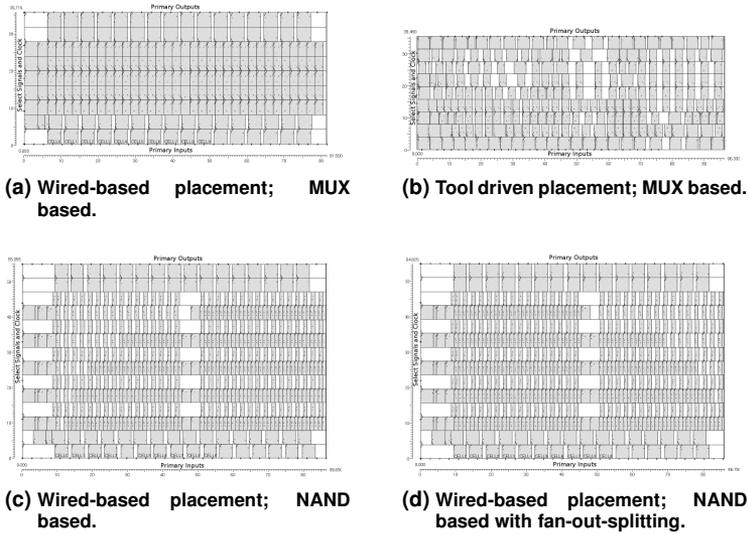


Figure 2.5: 32-bit shifters placed in Encounter.

A pictorial example of each implementation is shown in figure 2.5. In our exploration of log-depth multipliers using Wired [20], later results suggested that tightly packed cells cause some routing congestion that can be alleviated by providing "routing channels". From this experience, we estimated the best placement for the denser NAND-based shifters to be as they are; meaning that the NAND-based shifters can be shrunk further. There will be fewer rows, but the circuit will expand width wise. This causes wire lengths to grow and as a result performance decreases.

Table 2.1: Comparison of 32-bit barrel shifters in 90 nm CMOS.

Type	Slack (ps)			Core Area (mm ²)		
	Wired	Tool Driven	Fenced	Wired	Tool Driven	Fenced
Mux	248	240	235	0.002873	0.003402	0.003037
NAND	294	300	324	0.004738	0.005698	0.005414
NAND-FOS	329	302	312	0.004710	0.005698	0.004866

In order to be able to compare the quantified results, a common timing constraint of 900 ps was chosen so as to be as fast as the slowest shifter would support, for the largest word length. Table 2.1 shows the results comparing the slack and core area for 32-bit shifters implemented in 90 nm CMOS using 1.08 V as the operating voltage. Patterns

can be seen, both with respect to the various types of shifters, as well as the different strategies employed. The performance expectations of the different types are confirmed, with highly area-efficient multiplexer-based implementations and faster NAND-based implementations. Table 2.2 shows the results obtained for 64-bit implementations of the three types of shifters. The performance trends displayed for 32-bit shifters continue to be maintained here for the most part. The Wired-based placement strategy yields pre-

Table 2.2: Comparison of 64-bit barrel shifters in 90 nm CMOS.

Type	Slack (ps)			Core Area (mm ²)		
	Wired	Tool Driven	Fenced	Wired	Tool Driven	Fenced
Mux	71	33	55	0.006174	0.007453	0.006743
NAND	100	124	158	0.010565	0.012673	0.010860
NAND-FOS	180	122	148	0.010509	0.012673	0.010672

dictable performance irrespective of input word length. The tool-driven implementations show more dependence on the heuristic nature of the place and route engines making a comparison of the different types unpredictable. Even for a given type of implementation, due to the heuristics employed, a comparison of performance metrics for these strategies becomes meaningless. The Wired-based approach also shows highly compact circuits, with performance on par with circuits laid out using conventional techniques.

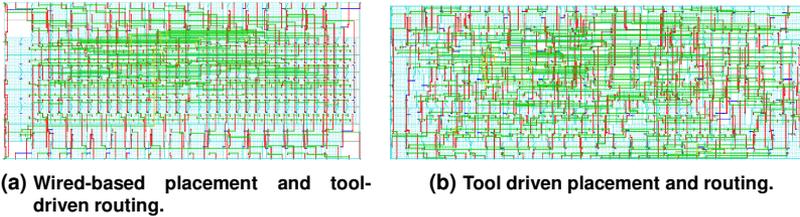
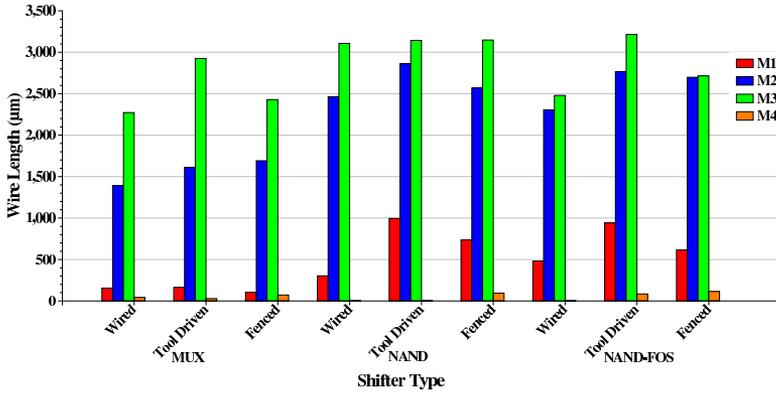
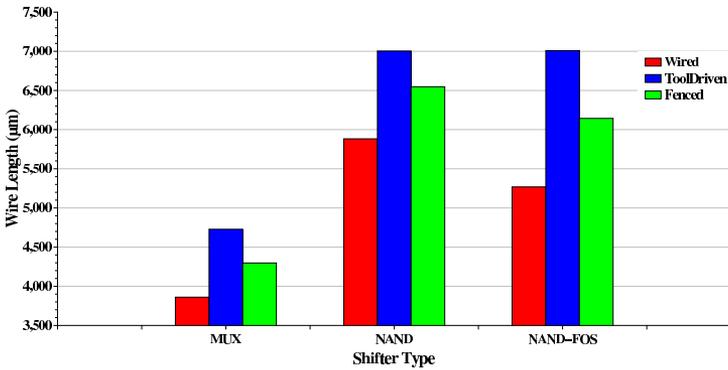


Figure 2.6: 32-bit multiplexer-based shifters.

Some simple observations can be made about the routing resulting from the exploration presented here. Figures 2.6a and 2.6b show the routing that resulted for the Wired and tool driven version of a 32-bit multiplexer-based shifter. A visual inspection shows that for an inherently regular circuit such as this one, routing becomes more regular when placement is enforced to be regular. Figure 2.7a shows the wire length distribution across the different metal layers for 32-bit shifters, for each of the placement schemes used in the study. The tool-driven implementation (figure 2.6b) is the least constrained and, as



(a) Layer-wise wire length distribution.



(b) Total wire length distribution.

Figure 2.7: Metal usage for implemented 32-bit shifters in 90 nm CMOS.

a result, the router makes use of all resources available to it, to ensure that a design that is design-rule compliant is possible. By conservatively fencing the design, the routing engine produces a design that is design-rule compliant with some reduction in total wire length (see figure 2.7b). The Wired-based placement (figure 2.6a) takes this reasoning one step further causing the router to produce a design-rule compliant design with the least resources. Figure 2.7b shows the total wire length distribution for the different types of shifters implemented and the different placement schemes used.

2.6.2 Multipliers

The HPM multiplier presented in section 2.5.2 lends itself to the Wired flow due to the inherent regularity. The results from an implementation using the HPM multiplier and the methodology presented in section 2.4 are compared against implementations of the Dadda³ and TDM multipliers.

The PPG and the PPRT for the HPM multiplier were created using Wired. While no strict placement constraints were placed on the PPG, this part was generated for the sake of completeness. Furthermore, overall area considerations were not taken into account while creating the PPG in Wired. This meant that the area for the block combining the PPG and the PPRT was overestimated. Since this could be easily corrected during the floor planning stages no effort was made to optimize this part. However, effort was spent in creating the desired shape of the PPRT. Initial implementations relied on the naturally occurring triangular shape of the PPRT (figure 2.8a). This layout style was used as a test platform to assess the impact of non-rectilinear geometries in a standard-cell flow. This implementation was compared against implementations of a Dadda multiplier and a TDM multiplier created using a standard RTL based flow. Table 2.3 shows the different configurations explored in terms of their slack and core area, for implementations at a frequency of 250 MHz and operating voltage of 1.08 V. The row showing results for Triangular 5 ML refer to an implementation constrained in Encounter to use only 5 of the available 7 metal layers. However, since non-rectilinear geometries are difficult to

Multiplier	PPRT Geometry	Slack (ns)	Area (mm ²)
HPM	Triangular	0.543	0.05064
	Rectangular	0.624	0.03719
	Triangular Channeled	0.320	0.06418
	Rectangular Channeled	0.484	0.05976
	Triangular 5 ML	0.256	0.05396
Dadda	Tool Driven	0.992	0.04703
TDM	Tool Driven	0.993	0.04406

Table 2.3: Comparison of multiplier implementations in 90 nm CMOS.

include in an implementation with any degree of efficiency in overall area (Table 2.3). Consequently, rectangular PPRTs were generated using transformations in the Wired environment. This had the effect of significantly improving the overall area utilization but resulted in severe congestion in the initial stages of the PPRT where a large number of

³ The HPM multiplier without placement constraints reduces to a Dadda and is the one considered in this exploration.

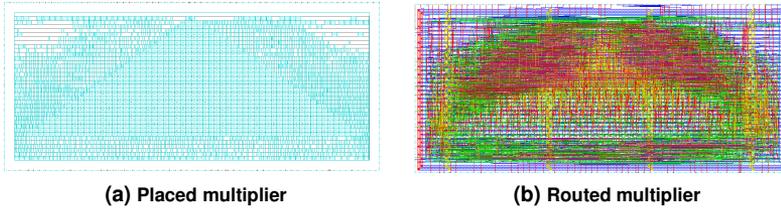


Figure 2.8: A HPM Multiplier with a triangular PPRT.

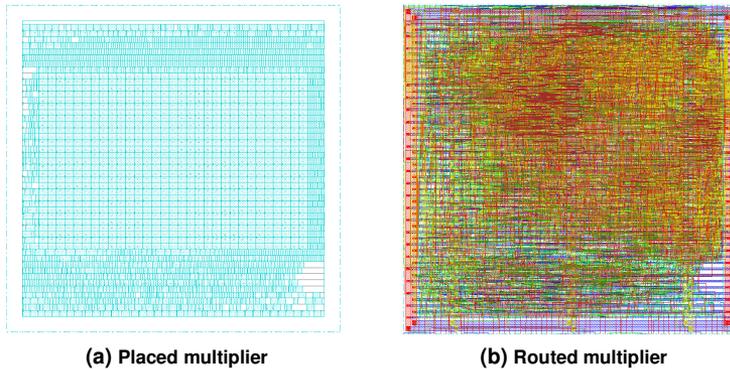


Figure 2.9: A HPM Multiplier with a rectangular PPRT.

partial products are processed. This is a qualitative visual inference and comparing figures 2.8b and 2.9b suggests that routing congestion does not affect the triangular layout as much as in the case of the highly dense rectangular layouts of the PPRT. Providing channels facilitating routing alleviates this issue somewhat but comes at a significant expense of area.

Looking at the total wire length for each of the triangular implementations and considering the Dadda as a point of reference (see figure 2.10), it can be seen that providing routing channels keeps the total length comparable to that of the Dadda (alleviating congestion at the same time), but restricting the maximum available routing layers increases the wire length significantly and also increases the area marginally.

The experience with the multiplier implementations in 90 nm CMOS proved promising enough that we continued the exploration of multiplier circuits in the 65 nm technology node. However, having established that performance does not significantly degrade due to the enforcement of regularity, we focused the effort on studying the factors

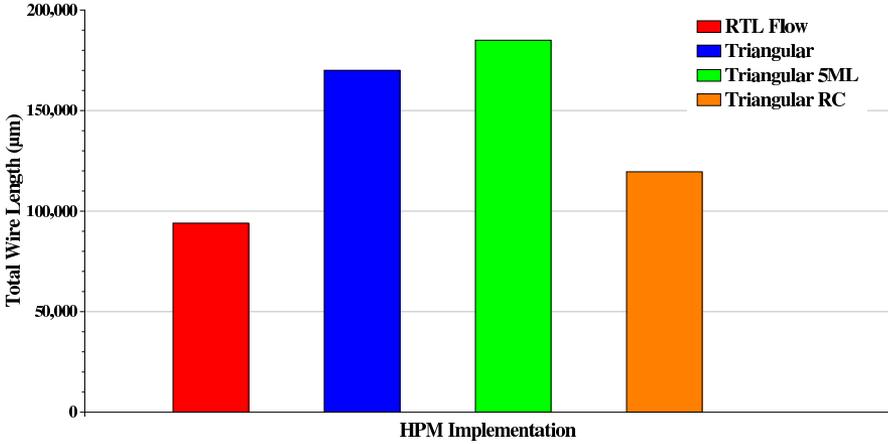


Figure 2.10: Total Wire length for different multiplier implementations in 90 nm CMOS.

impacting manufacturability the most at this level of abstraction i.e. the routing characteristics in terms of wire length and number of vias. Thus, the study of multipliers in 65 nm CMOS was restricted to different variants of the multiplier HPM multiplier, compared against a TDM multiplier implementation. The question of how to alleviate congestion while preserving area density led us to cell level regularity considerations that are presented in detail in the next part of this thesis. Table 2.4 shows the results of this exercise with the additional comparison points of Wire Length (the Length column) and the number of Vias (the NoV column).

Multiplier	PPRT Geometry	Length(μm)	NoV	Area (mm^2)	Slack (ns)
HPM	Rectangular	186004.07	35410	0.020309	0.003
	Rectangular RC-1	96183.46	22760	0.023260	0.087
	Rectangular RC-2	77650.70	19592	0.024371	0.106
TDM	Tool Driven	53574.17	15540	0.024585	0.365

Table 2.4: Comparison of multiplier implementations in 65 nm CMOS.

The original trends observed with respect to timing and area still hold in this exploration, implemented using a 400 MHz timing constraint at an operating voltage of 1.2 V. Additionally, two variants of routing channels were implemented: *RC-1* implements routing channels along the width of the design, while *RC-2* implements channels along both length and width. This allows for more routing area (and hence reduced con-

gestion) as seen from Table 2.4. It is worthwhile noticing that the TDM produces the best performance with the least routing resources, but the HPM still achieves the same timing constraint using a smaller core area.

2.7 Conclusions

From these studies it is clear that enforcing regularity at the abstraction level of standard-cell designs can produce highly area efficient implementations meeting stringent timing constraints at reduced margins (i.e. the timing constraints are satisfied but slack is lower). The flip side of this approach using foundry provided standard-cells was that there was significant impact on the routing resources required to obtain DRC compliant implementations. However the overall indications from this study were fruitful enough that the explorations were moved to the 65 nm design kit once that became available in order to keep the study up-to-date with available technology.

However, this work opened up a few questions. In dealing with the congestion issue, it is evident that congestion can be avoided by providing more area, however the regularity is destroyed. Looking into the reasons for this led us to studying the implementation of the standard-cell itself. Would standard-cells with regular layouts alleviate the issues caused by simply enforcing regularity on the abstraction layers above? Would it be possible to regularize routing by using alternate pin targets for the routing heuristics? Since the layouts of the standard-cells were not available to study these aspects, I implemented my own set of standard-cells to study the effects of regularly laid out standard-cells and also the factors which affect the creation of regular standard-cells. This work constitutes the next part of this thesis.

Bibliography

- [1] T. Kutzschebauch and L. Stok, "Regularity Driven Logic Synthesis," in *Proc. IEEE/ACM Int. Conf. on Computer Aided Design*, 2000, pp. 439–446.
- [2] P. lenne and A. Griessing, "Practical Experiences with Standard-Cell Based Datapath Design Tools. Do We Really Need Regular Layouts?," in *Proc. Design Automation Conf.*, June 1998, pp. 396–401.
- [3] C. Menezes, C. Meinhardt, R. Reis, and R. Tavares, "Design of Regular Layouts to Improve Predictability," in *Proc. 6th Int. Caribbean Conf. on Devices, Circuits and Systems*, Apr. 2006, pp. 67–72.

- [4] C. Menezes, C. Meinhard, R. Reis, and R. Tavares, "A Regular Layout Approach for ASICs," in *Proc. IEEE Computer Society Annual Symp. on Emerging VLSI Technologies and Architectures*, Mar. 2006.
- [5] ," Haskell Homepage.
- [6] Axelsson, E., *Functional Programming Enabling Flexible Hardware Design at Low Levels of Abstraction*, Ph.d. thesis, Chalmers University of Technology, 2008.
- [7] Tuscany Design Automation, ," Tuscany Homepage.
- [8] C. Seger, "Integrating Design and Verification - from Simple Idea to Practical System," in *4th ACM/IEEE Intl Conf. on Formal Methods and Models for Co-Design (MEMOCODE)*, 2006.
- [9] R. Zimmermann, "Datapath Synthesis for Standard-Cell Design," in *19th IEEE Symp. on Computer Arithmetic*, June 2009, pp. 207 –211.
- [10] N. Dhumane, S.K. Srivathsa, and S. Kundu, "Lithography Constrained Placement and Post-Placement Layout Optimization for Manufacturability," in *IEEE Computer Society Annual Symp. on VLSI*, July 2011, pp. 200 –205.
- [11] A. Chakraborty, A. Kumar, and D.Z. Pan, "RegPlace: A High Quality Open-source Placement Framework for Structured ASICs," in *46th ACM/IEEE Design Automation Conference*, July 2009, pp. 442 –447.
- [12] A. Bardizbanyan, K.P. Subramaniyan, and P. Larsson-Edefors, "Generation and Exploration of Layouts for Area-Efficient Barrel Shifters," in *Proc. IEEE Computer Society Annual Symp. on VLSI*, July 2010, pp. 454–455.
- [13] H. Zhu, Y. Zhu, C. Cheng, and D. Harris, "An Interconnect-Centric Approach to Cyclic Shifter Design Using Fanout Splitting and Cell Order Optimization," in *Asia and South Pacific Design Automation Conference*, Jan. 2007.
- [14] S. Huntzicker, M. Dayringer, J. Soprano, A. Weerasignhe, D. Harris, and D. Patil, "Energy-Delay Tradeoffs in 32-bit Static Shifter Designs," in *IEEE Int. Conf. on Computer Design*, Oct. 2008.
- [15] C. S. Wallace, "A Suggestion for a Fast Multiplier," *IEEE Transactions on Electronic Computers*, vol. 13, pp. 14–17, Feb. 1964.
- [16] L. Dadda, "Some Schemes for Parallel Multipliers," *Alta Frequenza*, vol. 34, no. 5, pp. 349–356, May 1965.
- [17] H. Eriksson, *Efficient Implementation and Analysis of CMOS Arithmetic Circuits*, Ph.d. thesis, Chalmers University of Technology, 2003.

- [18] V. G. Oklobdzija, D. Villeger, and S. S. Liu, "A Method for Speed Optimized Partial Product Reduction and Generation of Fast Parallel Multipliers Using an Algorithmic Approach," *IEEE Transactions on Computers*, vol. 45, no. 3, pp. 294–306, Mar. 1996.
- [19] P.F. Stelling, C.U. Martel, V.G. Oklobdzija, and R. Ravi, "Optimal Circuits for Parallel Multipliers," *Computers, IEEE Transactions on*, vol. 47, no. 3, pp. 273–285, Mar. 1998.
- [20] K.P. Subramaniyan, E. Axelsson, P. Larsson-Edefors, and M. Sheeran, "Layout Exploration of Geometrically Accurate Arithmetic Circuits," in *Proc. IEEE Int. Conf. Electronics, Circuits, and Systems*, Dec. 2009, pp. 795–798.

Part III

Manufacturability of Standard-cells and SoCs

Murphy was an optimist.

~O'Toole's Commentary on Murphy's Laws

3

Manufacturability of Standard-cells & SoCs

3.1 Introduction

Standard-cells have been used as a level of abstraction in the design of digital circuits. In the context of a design flow they are applied as pre-designed entities, characterized to meet certain performance goals dictated by the performance constraints of the technology node for the design (for which the cells are designed). Traditionally, the constraints involved in the design of standard-cells were related primarily to area and performance. Work presented in the previous part of this thesis showed that regular placement could create extremely area efficient designs while fulfilling stringent timing constraints. Applying such constraints *ad hoc* on foundry provided standard-cells exposed some shortcomings in the routability. The study of the tradeoffs of implementing regularity led us to study regularity in the implementation of standard-cells. With transistor geome-

tries approaching 16 nm other factors related to cost and manufacturability must also be taken into account simultaneously while designing standard-cells in these nanometer scale nodes. This chapter deals with those considerations and also deals with the impact of standard-cell architecture on the wider semi-custom design context.

Prior work in the area in the area will first be presented followed by an introduction to taxonomy and methods related to the study of variability. The sections following this, dealing with the study of the factors affecting standard-cell design in nanometer scale nodes, will look into the the considerations adopted for the study, the factors influencing those considerations and finally, the results of the study.

3.2 Regularity and Standard-cell design: Existing Literature

The impact of scaling has been studied since affordable manufacturing of electronics became a reality. The quantum mechanical effects of small geometries were studied and their effects were modeled. The impact on manufacturing due to scaling was also estimated as part of this research. As noted in section 1.3, the advent of laser based lithography changed the way the fabrication process is implemented. However, the failure to develop processes using lithography with sources less than 193 nm in wavelength has meant that the effects of scaling have been exacerbated.

Work related to regularity in standard-cell based flows has already been presented in section 2.1 and section 2.3. This section presents more recent work in this area, but concentrates more on regularity related research focusing on transistor (layout) level regularity. Research related to modeling of yield is also included in this section.

While yield modeling and defect sensitivity analysis has always been of relevance to the foundries, the study of process sensitivities on yield have also assumed importance to the design community at large since geometries were poised to enter the sub-100 nm regime. Heineken *et al.* [1] used the Poisson yield model proposed by Maly and Deszczka [2] using wafer productivity, defined as the number of working dies per wafer, as a metric to assess the manufacturability of standard-cells. Their results showed that standard-cells designed with process constraints related to device and interconnect geometries and number of vias/contacts displayed better wafer productivity.

Lavin *et al.* [3] introduced the so called “Restricted Design Rules (RDRs)” and demonstrated a flow based on circuit representation on “glyph” objects placed on a coarse grid. Their early estimates in the 65 nm technology node indicated that there

were significant benefits to restricting the layout patterns and orientations. Simultaneously, an application of RDRs by Liebmann *et al.* [4] showed that the layout restrictions had the desired effect in mitigating manufacturing induced variability. Muta *et al.* [5] demonstrated the benefits of regular gate-forming polysilicon structures on the variation of gate length¹. They explored the effect of regular gate-forming structures and single orientation and their results, supported using lithography simulations, further underline the benefits of regularity. Similar to this effort, Sunagawa *et al.* [6] study the benefits of regular layout structures on technology nodes from the 90 nm to the 45 nm technology node. Their results underscore the growing need to incorporate regular design techniques in conventional design flows as the technology nodes scale. Lin *et al.* propose a transistor level high-density layout generator for regular circuits based on Vertical Slit Field Effect Transistors (VeSFETs) [7]. The scope of this generator is limited to circuits with a few tens of transistors; however, the work also considers routing. Dal Bem *et al.* propose lithography aware regular layouts based on Via Configurable Transistor Arrays (VCTAs) [8, 9]; however, the impact on area due to the DRCs is large. Subramaniam *et al.* propose a scheme involving optimization of the design rule deck [10]. Their results indicate savings on leakage power without detrimental effects to performance.

Applicability of regularity to enhance printability has been demonstrated in the last few years based on a co-optimization approach, where the circuit, layout and the lithography are accounted for and optimized. Talalay *et al.* propose an approach to designing regular logic blocks using pre-generated layout templates [11]. Their study also proposes a possible definition for repeatable block and switch transistor logic model to describe functionality. This will be important when automated means for managing layout complexity at small geometries are desired. Similar to this effort, Ryzhenko *et al.* propose extremely regular diffusion structures extending the so called Lithographers Dream Pattern paradigm [12, 13]. Their results, carried out in the more advanced 32 nm node, features automatic cell synthesis onto the regular fabric and proposes simultaneous cell synthesis and M1 routing resulting in area advantages. Their work however, incurs a small leakage penalty.

In, by far, the most comprehensive coverage of regular fabrics, Javheri *et al.* showcase different strategies at implementing regular fabrics [14]. Their work proposes the use of logic bricks to implement commonly occurring logic functions in the design and other co-optimization techniques like pushed rules and circuit specific logic optimizations to significantly reduce the area impact in a wider design context. Their results

¹ The general variation in the variation of widths in interconnect lines is referred to as Across Chip Linewidth Variation (ACLV) when the variation is computed within the die.

indicate that adopting regularity has no significant impact on circuit performance either. This work using extremely regular patterns in layouts has been inspired by the highly dense and regular SRAM cells and the styles and the associated restrictions of the same have been migrated to logic layouts. However, co-optimization requires support from the foundry and predictive assessment has not been possible in any other simplified form. It should be noted that the density achieved in state-of-the art SRAMs is a result of highly optimized generators specifically created for this purpose by the memory manufacturers. The study I carry out here explores more generalized design techniques and methods applicable to standard industrial ASIC flows.

3.3 DFM Analysis - A Variability Primer

Manufacturability analysis is an important consideration for cost effective production of electronics. The foundries have studied the mechanisms which affect production and their relationship to cost and profitability. The cost of ownership of fabrication (see figure 1.5) equipment having becoming unaffordable to all but a few, has given rise to the *fab-less* and *fab-lite* production models. With scaling however, another phenomenon has manifested itself: the introduction of design dependent yield limitations. Traditionally, yield analysis was not an issue for a design engineer. The foundry bore the responsibility of ensuring that a design was cost-effectively manufactured. The scaling of technology nodes to the nanometer regime has changed that. In order understand the effect of scaling it is important to understand the terms variability and yield.

3.3.1 Variability Classification

Traditionally, variability analysis is classified according to where, in the process steps they take effect. Front-End-Of-the-Line (FEOL) variability refers to variability arising out of defects in the device creation steps of fabrication, while Back-End-Of-the-Line (BEOL) variability refers to the variability in the interconnect creation process [15]. Sometimes one also refers to variability in the lowest metal layers as Middle-Of-the-Line (MOL) variability. Lithography is a dominant source for FEOL variability while CMP polishing, used to planarize the metal used in interconnect at different levels, is the major contributor to BEOL variability. While interconnect variability has not been dominant in the past, it is becoming increasingly important as the devices scale and their delays become smaller.

FEOL variability primarily affects device performance, but has a critical yield impact as well. Fundamental device variability is displayed in threshold voltage variation, oxide thickness variation, energy level quantization and LER. The first three are random in nature since they depend upon the number and placement of dopant atoms. LER, the variation of the gate length along the width of the channel, however, is largely dependent on the photolithography process used to create these features. Since transistor leakage current has an exponential dependence on the gate length, the impact of LER on device performance is tremendous. This power limitation leads to large yield losses, since it occurs in high frequency bins which are also the most profit generating bins.

BEOL variability contributes directly to variation in interconnect thickness and indirectly to variation in interconnect width. Since imperfections in the CMP process cause planar defects, the lithography steps in multi-level interconnect are also affected. The insulating layer² reliability is also of concern in these steps. These effects can cause large variations in the interconnect resistance and capacitance making it more difficult to model these effects and correct for them at the physical design stage.

Another classification of variations is their nature of occurrence. Variations that are deterministic and can be modeled are termed as *Systematic Variations* while those variations that are random and cannot be modeled are called *Random Variations*. This distinction is important as some forms of variation appear to be random but are systematic in reality [15]. A good example of this is the dependence of transistor channel length on the orientation in the layout. This particular dependence arises due to shortcomings in the lithographic setup and causes a context dependence that is completely systematic. The interested reader may refer to [15] and [16] for detailed information on the techniques to study this.

Yet another classification of variability prevalent in manufacturing sector is based on the variation seen at different lots in the production line. With-In-Die (WID) variability (also known as Intra Die Variability) refers to the variation occurring within a single die. These typically are dependent on the local interactions with the reticle. On a slightly larger scale the variability depending on the relative location of a die on the wafer can also be estimated. This is termed as Die-To-Die (D2D) variability. Equipment limitations tend to contribute significantly to variations occurring between different wafers classified as Wafer-To-Wafer (W2W) variability.

² This layer is commonly referred to as Inter Layer Dielectric (ILD)

3.3.2 Variability Analysis

Over the years a number of techniques have been established in order to model and study the effects of scaling and variability. Most of the techniques applied to mitigate variability often employ statistical margins against the underlying parameters. For example, statistical simulations on a spread of gate lengths predicts the variation in leakage and so estimates the impact on performance. Typically, such simulations are used to compensate for systematic variations. Variations in the threshold voltage, V_{th} , is an interesting case since it consists of contributions that are systematic as well as random. The thickness of the oxide layer is a systematic contributor to the variation in V_{th} and can be compensated for through precise process control. V_{th} is also dependent on the doping profile of the channel. With device scaling a random phenomenon termed as Random Dopant Fluctuation (RDF) [15] is also contributing to V_{th} variations. Due to the inherently quantum mechanical nature of the problem, statistical distributions such as the Poisson model are employed to model this effect and margin against it. So far we have considered examples of variation only at the device level i.e. FEOL variation.

BEOL effects such as variations in the thickness and width of the interconnect metal and ILD also cause variations. The parametric variations can be modeled using detailed statistical techniques, but are usually compensated for during the fabrication process using dummy fills. BEOL defects such as particle defects are more critical to reliability but are random in nature and must be margined against.

During this kind of analysis a linear additive model of the form:

$$L = L_w(x, y) + L_d(x, y) + L_{wd}(x, y) + \epsilon$$

is used to account for the different contributing components of variability. ϵ depicts the random error that cannot be attributed to any component.

The techniques for dealing with random errors are all based on probabilistic estimations of defects and consequently yield. These models are based on critical area techniques and rely on defect size and density probability to compute yield under the assumed conditions. A Poisson distribution, commonly used to model such effects, takes the form:

$$Y = \exp[-D_0 A_{cr}]$$

where D_0 is the defect density and A_{cr} is the critical area function. This model is applicable when the defect distribution is uniform. When this is not the case, a negative

Binomial model expressed as:

$$Y = \left[1 + \frac{D_0 A_{cr}}{\alpha} \right]^{-\alpha}$$

is frequently used. Other models like Murphy [17], Seeds [18], Price [19] and Dingwall [20] are also applicable in such cases.

Design time analysis of manufacturability is now being employed in design flows to assess the risk due to interactions between design decisions and process dependencies. Integrated flows acting as extensions of DRCs are routinely employed to estimate the impact of contributions from the design and systematic process dependencies such as lithography and CMP. Integrated tools such as the one employed in this study (Calibre CFA), also use some kind of Critical Area Analysis (CAA) to estimate the impact of random defects. The checks are organized in the verification framework as an extension of the DRC checks and are similarly presented.

Metrics in CFA

The overall results of a DFM run using Calibre CFA for a certain design are a Weighted DFM Metric (WDM), computed on all rules, and a Normalized DFM Score (NDS). In addition, the results for individual checks are also available (see figure 3.7 later in this chapter).

The WDM is a weighted score computed on rules defined to obtain better manufacturability. The rules are categorized on criticality depending on the geometric value for the current check. The weight changes according to the criticality and is, as such, empirically assigned by the foundry. The rules are designed in such a manner that the degree of benefit is reflected and ranges from a failure to comply with the DRC to a value beyond which no further benefit is expected. This binning is again based on the experience of the foundry with those geometries. The WDM score presented is a summation of the WDM for individual rule scores averaged over the total number of checks that are run.

The Normalized DFM Score is a negative-indexed exponential of the normalized WDM score. This means that a score of 1 indicates perfect manufacturability while a value tending to 0 indicates catastrophic failure or no functionality. Equivalently, a low WDM indicates better manufacturability while a higher one indicates problematic patterns.

3.4 Standard-cell Layout Architecture

Modern standard-cell based design flows are structured in such a manner that this level of abstraction hides as many of the device level details as possible from a designer. Consequently, the design involves generation of geometry and timing related models to be used for the design of more complex functionality. Standard-cell development itself consists of all the steps involved in a full-custom design flow. Generators have been used in the past to generate layouts for standard-cells, but the legacy generators are increasingly difficult to migrate to new technology nodes. Though automatic generation is an interesting avenue for the development of standard-cells, our work does not consider it for the moment.

With scaling, a number of restrictions have been introduced by the foundries in order to maintain yield margins. Going back to the original intent of this work introduced in section 1.4, this work concentrates on standard cells incorporating different degrees of regularity.

3.4.1 Ultra-regular and Semi-regular Layouts

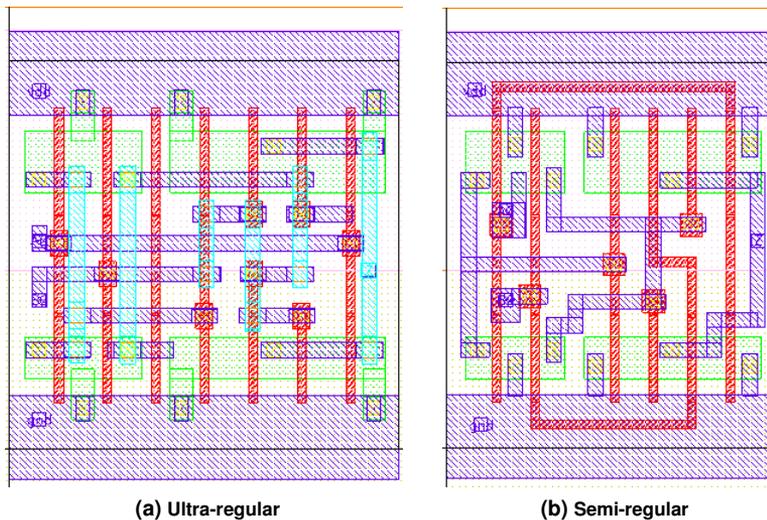
Ultra-regular layouts, as presented in this work, refer to layouts in which, in addition to maintaining a single device orientation and constant poly pitch, the directions of the local routing resources are also fixed. Widths and spacings for the layout geometries in a semi-regular layout are held as constant as allowed by area constraints but minor deviations are allowed. Poly pitch is constant across devices with multiple fingers, but routing in poly is allowed. The local routing resources are constrained in the number of layers used but not the direction.

While it is relatively easy to implement these constraints for simple two input cells at little impact to the area, it becomes increasingly difficult to do so when the complexity of the cell grows either in terms of the number of inputs or the number of devices or both. In order to analyze the tradeoffs involved in implementing regular layouts, with little or no impact on area (and performance too), it was necessary to create standard-cells with regular geometries. A basic set of eight logically complete combinational cells have been created using a commercial 65 nm process. These cells are listed in Table 3.1 and compared in terms of width to a comparable library cell. The library cells listed, especially the more complex cells, are chosen based on device sizing and performance, leading to some additional difference in the widths. The label in the parentheses, under the cell functionality column, will henceforth be used to describe the cells. Figure 3.1 shows the ultra-regular and semi-regular implementations of an AOI based two input XOR gate. In

Table 3.1: Custom characterized cells in 65 nm CMOS.

Cell Functionality	Width (μm)		
	Ultra-regular	Semi-regular	Library
And(AND)	1.6	1.4	1.0
Buffer(BUF)	1.0	1.0	0.8
Inverter(INV)	0.6	0.6	0.6
Nand(NAND)	1.0	1.0	0.8
Nor(NOR)	1.0	1.0	0.8
Exclusive-Or(XOR)	2.6	2.2	1.8
Half Adder(HA)	3.4	3.2	2.0
Full Adder(FA)	5.2	4.4	3.6

order to focus the design effort, it was decided to implement only combinational cells, which form a bulk of most digital implementations. It should be recognized here that a number of standard-cell parameters, such as cell height and width are greatly influenced by the routing requirements for sequential cells like scan enabled flip flops, which are typically denser. As another simplification of the overall implementation effort, the cus-

**Figure 3.1:** Custom characterized XOR Gates.

tom cells were implemented to have the same pitch as that of the library cells in order to focus the assessment on the less dense but more utilized combinational logic. Since these decisions also entail interactions between cells from two libraries, the widths of the power rails were also retained.

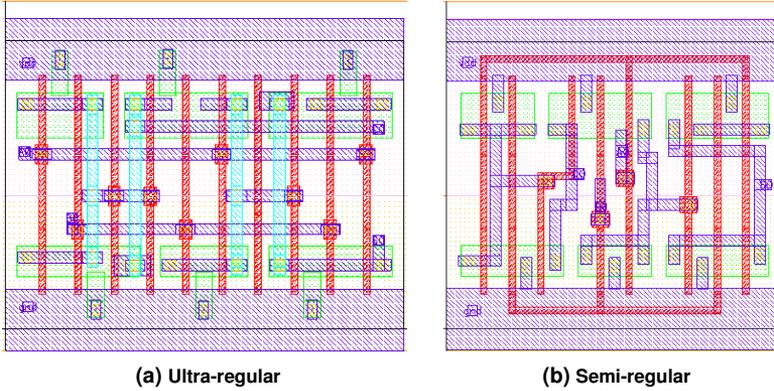


Figure 3.2: Custom characterized Half adder cells.

The layouts were checked against the standard DRC deck for the technology using the Calibre nm-DRC tool. Layout Versus Schematic (LVS) checks were also successfully carried out using the Calibre nm-LVS tool and parasitic extraction was performed using the StarRCXT tool from Synopsys. The cells are characterized for low power under standard-threshold voltage (LPSVT) conditions³ for an operating voltage of 1.2 V. In addition to the timing data, created in the *.lib* format using Cadence Encounter Library Characterizer [21], geometry abstracts (in the *.lef* format) are also created using Cadence Abstract Generator [22] for use in a industrial standard cell flow using the Cadence Encounter Digital Implementation (EDI) system [23].

3.4.2 Factors affecting Analysis

The process of manufacturing reliable electronics in the nanometer regime involves considerations across a number of levels of abstraction and requirements. Additionally, due to the complex nature of the manufacturing process, intellectual property of the different domains in design and manufacturing are also a concern. This makes it difficult to obtain data from the foundry. However, the chief concern for a physical design engineer involves the creation of a manufacturable solution under area and performance constraints. Some of the factors having a large implicit effect on the implementation of regular cell layouts are listed under the following sub-headings.

³ LPSVT describes the combination of V_{th} and physical geometries like oxide thickness which influence the threshold and results in low static power.

Gate Pitch

The gate pitch is the first stage of regularity and sets the device density for a given circuit. It affects regular measures for all other geometries directly or implicitly. Two broad definitions of gate pitch can generally be used.

The contacted gate pitch of a device can be expressed as the sum of the gate length, spacing between poly and contact and the contact width. When dummy poly is used between isolated diffusions the isolated gate pitch can be written as the sum of the poly length, contact width, poly-contact spacing, diffusion extension over contact and diffusion-poly spacing.

Assuming that upstream methodology follows the normal standard cell flow and when regular layouts are prioritized (or even mandatory) in order to keep mask costs to a minimum, a relaxed gate pitch like the isolated gate pitch will usually be preferred.

Device Pitch and Interconnect

In the past, the only consideration influencing the device pitch was the performance of the cell in question. It is usually the case for digital circuits that the minimum width is not used for performance reasons and this is advantageous when DFM considerations are taken into account.

With scaling geometries however, a big concern from a manufacturability point of view is the availability of contact redundancy. It is common knowledge within the design community that redundancy of contacts and vias increase the reliability of the fabricated circuit. However, doubling contacts for the sake of reliability alone can have detrimental effects on the performance as it necessarily means that device widths are going to be larger and thus increase diffusion capacitance.

The device pitch also influences the choice of metal routing for the local interconnect. Traditionally, alternating orthogonal directions, starting with horizontal M1 have been used. Choosing M1 perpendicular to poly makes for better local routing but decreases the availability of redundancy. Routing M1 parallel to poly is an alternate solution, eliminating the redundancy problem at the cost of diffusion width and additionally, increased M2 usage. With these considerations in mind, I chose to implement cells with M1 perpendicular to poly, without redundancy for the present discussion.

Assessing the impact of routing is more complicated due to disparate considerations like choice of architecture and choice of routing directions. Enforcing unidirectionality of routing incurs a penalty for upstream routing since it introduces blockages not seen when only M1 is used. Additionally, this measure introduces vias, which intuitively

make printability simpler but have critical manufacturability constraints. In addition to this there is also an impact to parametric yield due to the etch and CMP related defects. On the other hand allowing jogs creates problems with metal printability but poses fewer reliability concerns.

Power Supply Rails

This aspect of cell layout architecture has far reaching consequences for performance and area. Standard-cells share supply rails through abutment on successive rows. This provides significant savings in power routing and die area. Power supply rails are typically in the lower layers and are wider than normal interconnect nets in order to retain a large current carrying capacity. The width of the power rails spans 2 to 3 horizontal routing pitches.

In cells that are not routing limited the power can be supplied to the source terminals using M1 and contacts to diffusion. This allows for low RC losses in the power supply network, but takes up routing resources. Also in the context of ultra-regular layouts unidirectional routing would no longer be followed if a M1 perpendicular to Poly style is chosen. In spite of the risk of higher RC losses, in this work, I chose to implement the power supply connections through the diffusion to assess the tradeoff against routing resource availability. Alternate power supply strategies can be adopted for further enhancement [14], but are not considered in this work.

Circuit Considerations

The choice of architecture used to implement the logic function under consideration affects a number of parameters associated with enforcing regularity on layouts. It has been observed that AOI structures lend themselves more easily to regular layouts than other types of static gates (like transmission gates etc.) [14].

If the device supply connections are completed using diffusion then maintaining a spacing of one horizontal pitch yields between the supply rails and diffusion, another metal routing track that can be used for parallel device connections. Let us assume further then, that a spacing of one horizontal pitch needs to be maintained between the power supply rails and diffusion.

The overall pitch of the cell is a tradeoff between the routing requirements for densely connected logic functions, usually the scan flip-flops, and the width.

3.5 Standard-cell Layout Implementation

The previous section(section 3.4.2) highlighted the influences on creating regular layouts. This section details the specific adoption of these measure with respect to the cells considered in this study.

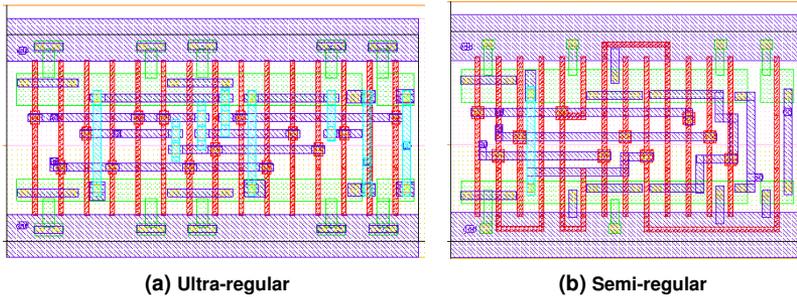


Figure 3.3: Custom characterized full adder cells.

Noting the specific problems detailed in section 3.3, the following measures were adopted for the layouts in line with the constraints introduced at the end of section 3.4.1:

- The transistor widths used here are higher than the minimum width specified by the technology.
- The traditional technique of equalizing the drives of the pull-up and pull-down networks by having a wider PMOS is still followed here. The PMOS devices are one and a half times wider than the NMOS devices.
- Regularity is maintained on a per cell basis, using single lines of diffusion as far as possible. In the case of the semi-regular layouts only the diffusion widths and poly pitch are regular (as far a possible).
- The poly layer pitch is set to the contacted gate pitch for the semi-regular cells, while this is increased to the isolated pitch for the ultra-regular cells.
- All routing layers including poly are made unidirectional for the ultra-regular layouts. This means that M2 has to be used to complete the local interconnect within the cell. Keeping the preferred directions, Poly is directed vertically, M1 horizontally and M2 vertically.
- For the semi-regular layouts, with the exception of the full adder, all layouts use

only M1 to complete internal routing⁴. Poly is used extensively in routing inputs to the gates of the transistors.

- In as many cases as possible, an effort is made to run input and output pins out to the edge of the cells for both the semi-regular and ultra-regular layouts. This is done in order to minimize extra routing within the cell during cell-to-cell routing.
- Dummy poly is employed to simplify the mask for the poly layer. In this work it is used only in the ultra-regular layouts with the observation that half-space rules are used at the cell edges. In more advanced nodes, it is mandated by DRC to employ isolated poly lines at cell edges.

3.6 A Semi-custom Design Perspective

In section 3.1, I mention that the impact of standard-cell architecture on the wider semi-custom design context is also studied. The ISCAS'89 benchmark circuit suite [24] is used as the evaluation vehicle for studying the implications of incorporating regularity into the standard-cell architecture. These benchmark circuits range from a few gates to a few thousand gates and consist of varied functionality. The thirty odd circuits that form this suite offer insights into the behavior of automated synthesis and, place and route tools. Though all the circuits are physically implemented, six of the benchmark circuits representing different sizes are chosen for the study on manufacturability metrics. The reason for this is that this work focuses on the interactions between device level geometries and the impact they have on manufacturability as indicated by integrated DFM tools, when design automation software is employed to carry out the physical implementation. This being the goal, a sample of representative circuit sizes sufficiently represents the different device level geometries and their interactions.

All the standard cells implemented in the semi-regular and ultra-regular libraries for this work are shown in Table 3.2. Note that the custom characterized libraries used in this study (as compared to the ones used in the study of the cells themselves) have been expanded to include cells with higher drive strength. The variants are noted under the *Comments* column. The libraries do not include AOI gates, but include a few inverters and buffers. Half- and full-adders are available in another drive strength (designated X4 in the *Comments* column in Table 3.2) in both libraries. For all other logic functions, cells with X4 drive strength are available only in the semi-regular library. In addition to this, the half- and full-adders in the semi-regular library have one additional variant

⁴ It is possible to complete that net without the use of M2 but it would result in obstructions. This is another tradeoff not considered explicitly in this work.

with their inputs ordered in reverse (flipped). For sequential logic, the foundry provided flip-flops are used. While it can be viewed as a shortcoming that And-Or-Invert (AOI)

Table 3.2: *Standard-cells implemented for the ISCAS'89 circuit tests*

Cell	Comments
AND	X4 available in semi-regular library only.
BUF	X4 available in semi-regular library only.
FA	X4 available in both. Variant with flipped inputs available only in the semi-regular library in both drive strengths.
HA	X4 available in both. Variant with flipped inputs available only in the semi-regular library in both drive strengths.
INV	X4 available in semi-regular library only.
NAND	X4 available in semi-regular library only.
NOR	X4 available in semi-regular library only.
XOR	X4 available in semi-regular library only.

cells are not available during implementation, this work concentrates on the impact of regular geometries. Observing that AOI gates are simply compound functions of basic gates, created to achieve area density, their absence does not in any way influence the goal of this work. AOI gates are used in the next part of this thesis in order leverage the area savings they offer.

The ISCAS'89 benchmark circuit designs are implemented using common area constraints for each variant; the constraints only specify a target utilization and row density. A common slack constraint of 750 ps is also applied to all designs during logic synthesis. This value represents a realistic target that could be fulfilled by even the largest designs in the suite. The slack constraint is primarily applied in order to obtain a realistic clock period for each design before physical implementation and is achieved by refining the clock period applied to the design during synthesis based on the slack constraint applied. Furthermore, this artificial retiming technique avoids tool-inserted registers from clouding the findings.

In the physical implementations, the metal stripes for the power rails are vertical in the implementations using semi-regular cells and horizontal in the implementations using ultra-regular cells. This style of implementing the stripes is adopted since the ultra-regular standard cells make use of M2 to complete internal routing. In the case of semi-regular layouts, with the exception of the full-adder, M1 is used exclusively to complete internal routing.

The physical implementation culminates with the GDSII stream produced by Encounter. Raw implementation statistics, such as the number of cells, number of vias, wire length, and slack, are indicative of the quality of implementation and are extracted before proceeding to the manufacturability assessment. The standard industrial flow relies on traditional full-custom DRC checks at the signoff stage. It is also at this level of abstraction that DFM checks are incorporated into the verification scheme. The results of the implementations are shown in section 3.7.2 and section 3.7.3.

3.7 Results

The cells implemented for the purpose of this study (introduced in section 3.5) have been applied on two levels of abstraction: the first is a study of manufacturability of regular standard-cell layouts using an integrated DFM analysis tool, namely Calibre CFA [25]; the second is a study of the implementation metrics of the cells applied to the ISCAS benchmarks in an industrial standard-cell based flow [26].

3.7.1 Cell Manufacturability Analysis

Only a subset of all the cells created are included in the results of this study. The layouts under consideration in this study were chosen primarily based on their utility in arithmetic circuits like adders and multipliers. Additionally, they were chosen for the layout characteristics they exhibit when only static AOI architectures are considered. I make the decision constraining the architecture based on existing knowledge related to the performance characteristics of other layout architectures [27, 28] and assertions in existing literature [14].

The XOR structure presented in figure 3.1 represents a commonly used AOI based architecture. The definition of the XOR function requires the availability of inverted versions of the inputs. The HA and FA (see figure 3.2 and figure 3.3 respectively) circuits were chosen as functional extensions of the XOR gate. While both of these circuits, by definition, depend on the XOR gate, they differ vastly in layout. Due to the fact that there is additional functionality in these circuits the number of devices is higher. There is also impact due to the different number of inputs and outputs. Commonly used AOI based architectures were implemented for these circuits as well.

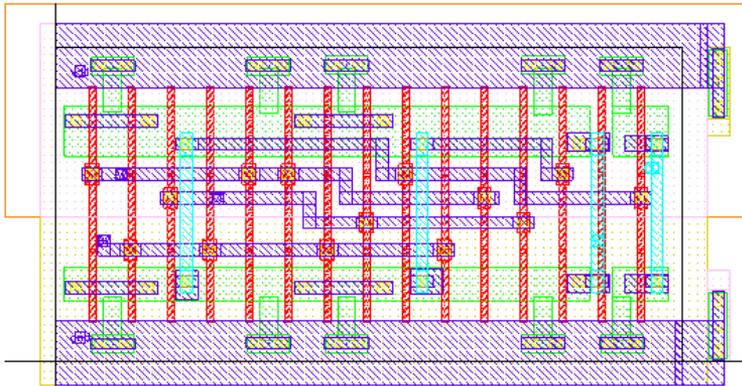
In this study, DFM checks are carried out using the CFA tool [29] using foundry-provided rule sets. This tool is integrated with other DRC and LVS tools belonging to the Calibre suite and relies on detailed rule-based checks to provide metrics on resilience

Table 3.3: CFA Results for Ultra-regular and Semi-regular cells.

Cell	Normalized DFM Score(NDS)		Normalizer
	Ultra-regular	Semi-regular	
XOR	0.58	0.74	4.14
HA	0.61	0.73	5.52
FA	0.68	0.74	9.66

to particle defects, modeling accuracy and process margins⁵. Table 3.3 shows the DFM scores for the ultra-regular and semi-regular XOR, HA and FA standard-cells developed for this work. Analyzes were run on these cells with the standard DFM deck provided by the foundry. The results indicate that the semi-regular layouts are more manufacturable than the ultra-regular ones. The fact that the layouts analyzed for DFM issues are small is highlighted by the small value of the normalizer.

All the same, a few insights can be obtained. Noting that the gate geometries are the smallest and unequivocally critical, the mask for that layer is going to have to use manufacturing techniques that are the latest-and-greatest or at least something suitably close. Given that the device diffusions are identical in both the ultra-regular and semi-regular cases, it is the choice(s) on other layers that impacts the DFM score obtained through CFA. Looking at the tradeoffs discussed in the previous sections it is clear that one of contact- and/or via-redundancy is a chief contributor. Given that the contacting

**Figure 3.4:** A full adder cell regular in Poly pitch and direction.

⁵ “Process margin” is a term indicating tolerances that layout features exhibit to defects induced due to the manufacturing process steps like lithography, optical proximity correction (OPC) and chemical-mechanical polishing (CMP).

scheme in both types of layouts are nearly identical, it is reasonable to assume that the culprits are the vias. The individual rule results (not shown here) confirm the fact that the contact and via1 related checks for the ultra-regular layouts, have a high contribution in the WDM computation and thus impact the NDS. In the case of the semi-regular layouts, the primary source of concern turns out to be the contacts followed by poly spacing rules. This indicates that using a single layer of metal to complete the internal connections rather than enforce unidirectionality of routing is a manufacturably tractable option. As a confirmation the layout for the FA was modified such that regularity is enforced in poly pitch and direction but no strict regularity of other interconnect elements are followed (figure 3.4). The CFA NDS has a value of 0.69 with the same normalizer value of 9.66. In spite of the fact that the number of layer changes is minimized, the NDS is only marginally better owing to the fact that the vias are not backed up. From figure 3.4 it is clear that back-up vias can be placed at a few locations without alteration of the routing solution. Once all the vias and as many contacts as possible are backed up, the NDS rises to 0.73. It then stands to reason that using a single metal layer for interconnect is still viable as long as the contacts are backed up. Thus, at the level of a design with a few tens of transistors, there are diminishing returns from the point of view of design effort. This may however, prove to be offset in a larger design context.

3.7.2 ISCAS Benchmark Circuits - Physical Implementation

In order to test the effect regularity at the transistor level layout has on higher levels of abstraction, the cells developed have been characterized and applied to the synthesis and physical design of the ISCAS benchmarks [30]. Since the semi-regular library, used in this part of the study, is richer in terms of drive strength and diversity of cells, three variants have been implemented. The first—designated SR—consists of the set of cells available in common with the ultra-regular library (designated UR in the implementations). The implementation designated SRX4 includes cells with higher drive strength and the flipped variants, in addition to the basic cells. The SRX4 implementation is used to assess the implications of drive-strength diversity. Both SR and SRX4 are implemented using semi-regular layout geometries. The implementation designated UR consists of all the cells with ultra-regular layout geometries. All the design variants have been implemented using the same density and aspect-ratio constraints, resulting in little (and therefore un-tabulated) variation of the core area. The clock period for the designs in the benchmark suite (after synthesis) is shown in the first column of Table 3.4. The designs in the suite range from a few gates to a few thousand gates as can be seen from Fig. 3.5a. The implementation related statistics—the wire length and the number

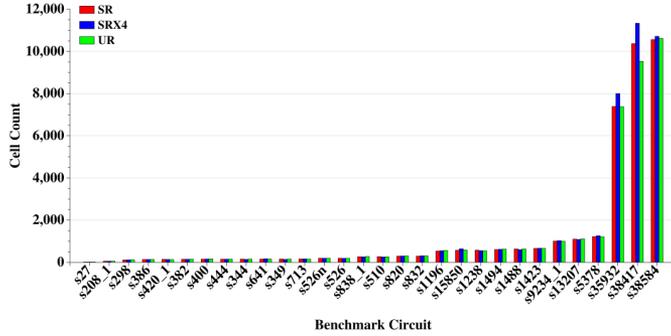
Table 3.4: Physical Implementation Metrics for ISCAS'89 Benchmark Circuits

BM	Clock Period (ns)			Slack (ns)			Wire Length (μm)			Via Count		
	SR	SRX4	UR	SR	SRX4	UR	SR	SRX4	UR	SR	SRX4	UR
s27	1.50	1.50	1.50	0.74	0.79	0.70	90.27	97.39	101.36	41	43	40
s208_1	2.00	1.75	2.00	0.73	0.31	0.52	336.05	270.40	294.02	176	160	135
s298	2.00	1.75	1.75	0.35	0.30	0.25	871.89	861.89	932.89	422	389	313
s386	2.00	2.00	2.00	0.43	0.46	0.30	924.89	908.05	989.80	475	487	344
s420_1	2.00	2.00	2.25	0.40	0.54	0.51	762.35	851.63	753.61	433	469	320
s382	2.00	1.75	2.00	0.56	0.39	0.43	922.75	952.61	933.75	499	511	401
s400	1.75	1.75	1.75	0.13	0.31	0.24	1002.35	1070.92	1028.67	540	560	410
s444	2.00	1.75	1.75	0.47	0.33	0.37	979.75	885.61	1075.72	580	567	470
s344	2.00	2.00	2.00	0.06	0.45	0.42	1209.83	962.47	986.83	561	453	330
s641	2.00	2.00	2.25	0.56	0.29	0.60	980.65	1078.12	923.87	515	582	350
s349	2.00	2.00	2.00	0.07	0.47	0.47	1142.09	900.18	1009.46	513	436	367
s713	2.00	2.00	2.00	0.56	0.44	0.48	1017.72	1062.94	867.11	533	577	334
s526n	2.00	1.75	2.00	0.24	0.25	0.32	1199.01	1279.02	1146.40	691	730	499
s526	2.00	2.00	2.00	0.23	0.40	0.30	1356.70	1238.68	1128.09	730	723	560
s838_1	2.50	3.00	2.75	0.31	0.21	0.20	1465.59	1392.78	1617.32	877	834	736
s510	2.00	2.00	2.25	0.05	0.15	0.33	2121.82	1625.75	1848.41	1078	872	740
s820	2.25	2.25	2.25	0.08	0.33	0.21	2160.01	2125.22	2263.58	1084	1123	935
s832	2.25	2.00	2.50	0.19	0.13	0.40	2127.97	2129.82	1991.57	1047	1152	803
s1196	2.75	2.50	2.75	0.24	0.10	0.24	4217.88	4262.70	4347.30	2060	2083	1655
s15850	2.50	2.25	2.50	0.39	0.39	0.51	3605.64	3844.79	3762.77	2145	2259	1730
s1238	2.75	2.50	2.75	0.22	0.18	0.27	4318.40	4647.31	4084.36	2175	2175	1650
s1494	2.75	2.25	2.50	0.00	0.07	0.07	5384.68	5836.41	6052.03	2465	2735	2051
s1488	2.50	2.50	2.50	0.04	0.06	0.05	5864.42	5474.43	6774.21	2689	2521	2266
s1423	3.50	3.00	3.25	0.12	0.28	0.09	4284.82	4177.75	4095.99	2375	2401	1839
s9234_1	2.75	2.75	2.75	0.08	0.33	0.18	7025.85	7378.69	7266.98	3639	3847	2937
s13207	2.25	2.00	2.25	0.38	0.32	0.35	6530.01	6478.76	6033.10	3893	3890	3060
s5378	2.50	2.25	2.50	0.08	0.02	0.08	11177.23	11524.20	10751.32	4773	4949	3810
s35932	6.00	5.50	6.75	0.10	0.10	-0.02	103407.10	108943.43	117733.90	30493	32742	29025
s38417	7.25	6.75	6.50	0.12	0.04	0.05	114400.34	119298.42	109440.71	41087	43979	33174
s38584	7.00	6.25	6.75	-0.32	0.05	0.48	153099.04	104280.03	112528.89	44627	39819	33680

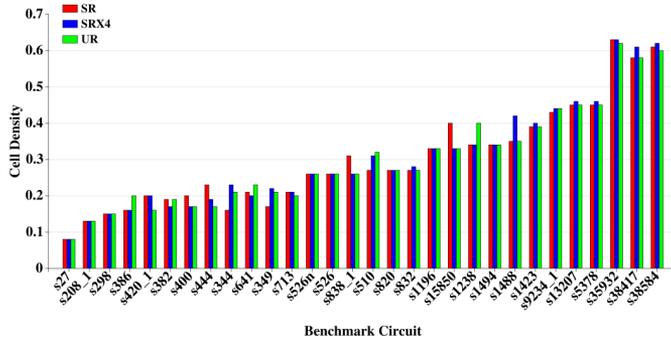
of vias—are also shown in Table 3.4 along with the slack after physical implementation.

The chip density does not show much variation across the implemented variants due to the common constraints applied (see figure 3.5b). The slack (figure 3.5c) on the other hand shows wide variation depending on the size of the design in spite of applying a synthesis slack constraint.

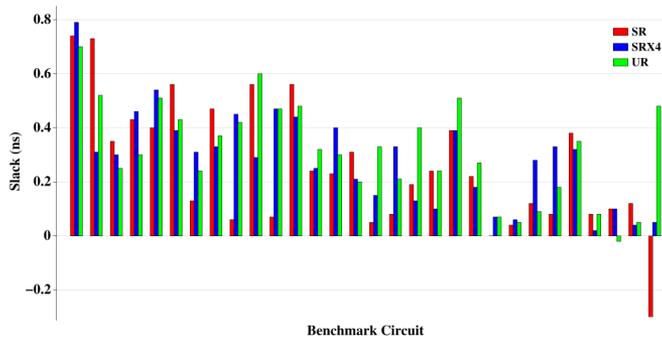
The slack shows wide variation depending on the size of the design in spite of applying a synthesis slack constraint. Looking a little more closely, Table 3.4 shows that the slack also depends on cell diversity more and more as the size of the design grows. Although it would appear that the UR and SR implementations outperform the SRX4 implementation, it should be noted that the difference in clock periods and the particular



(a) Cell count



(b) Chip Density



(c) Slack

Figure 3.5: Cell count, chip density and slack plots for ISCAS benchmarks.

physical implementation iteration influence the slack. The lack of cell and buffer diversity affects the optimization steps of the physical implementation flow negatively and this is evident in the case of the larger designs. The use of heuristics during place and route means that additional variation is introduced into the performance. The variation across the different implementations, given the constituent set of cells, is thus an inexact prediction of performance. In terms of the metal layers used to achieve DRC-compliant routing solutions, the largest designs are routed with M5 being the highest layer used. The metal usage for wiring is not excessive since the designs are not too big.

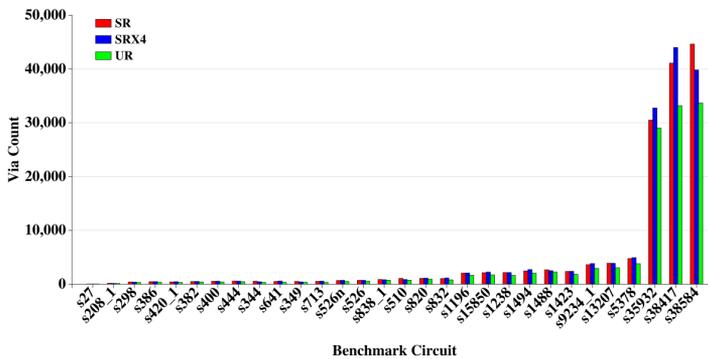


Figure 3.6: Number of vias in the ISCAS benchmark circuits after physical design.

The vias in the interconnect stack have the highest reliability concerns [31–35] and incorporating regularity at the lower levels of abstraction shows clear benefits with the UR implementations using the lowest number of vias as is evident from Fig. 3.6. This reduction in via count can be viewed as a benefit even though it could result in longer wires for the UR implementations, since vias contribute to absolute failures as well as parametric variations. Other variations in the interconnect stack such as wire width and thickness variations may be dealt with using techniques like wire spreading and wire widening, to ensure minimal impact on parametric variation. Those techniques are not considered in the present study.

A comparison of wire length of the UR variants against the SRX4 variants (for the tabulated benchmarks) shows an average increase of 0.01%. An average *decrease* of 2.9% is observed when the wire length of the UR variants are compared against the SR variants. In some individual cases, more drastic decreases of wire length can be seen indicating the impact of heuristic routing. For the other designs, however, the change in wire length varies greatly but fewer vias are still used. On average, the use of ultra-

regular layout styles results in a 22% reduction in the number of vias compared against the SR and the SRX4 variants, for the tabulated results in Table 3.4. Note that the numbers given here are the result of averaging the percentage increase of the wire length and the percentage decrease of the number of vias computed for each design.

3.7.3 ISCAS Benchmark Circuits - CFA Results

The raw implementation metrics predict better manufacturability from the point of view of the interconnect stack for the UR implementations since, on average, 22% fewer vias are used for the benchmark circuits considered in this study. This, however, says nothing about the densely packed device geometries that are typically the smallest dimensions in a layout and pose the greatest challenges to manufacturability. In order to form a complete picture of the factors impacting manufacturability, it is necessary to assess all geometries that make up the layout. This is accomplished by importing a GDSII stream produced by Encounter into the Virtuoso environment and running DFM checks on it. Having formed a rather general picture of the manufacturability at a higher level of abstraction, where the interconnect stack is prominent, only a few representative layout patterns need be assessed in order to determine the impact of ultra-regular layouts has on a standard cell-based design.

Table 3.5: Total DFM Metrics for Some Representative ISCAS'89 Benchmark Circuits

BM	SR				SRX4				UR			
	NoPC	WDM	NDS _T	NDS _L	NoPC	WDM	NDS _T	NDS _L	NoPC	WDM	NDS _T	NDS _L
s27	17	21.72	0.25	0.17	17	24.61	0.27	0.19	17	24.06	0.20	0.13
s400	150	173.21	0.42	0.32	157	187.47	0.47	0.38	163	196.59	0.37	0.28
s820	295	366.71	0.38	0.27	302	374.09	0.46	0.37	303	376.92	0.38	0.29
s5378	1219	1624.21	0.38	0.26	1261	1739.00	0.44	0.32	1207	1772.78	0.37	0.24
s35932	7388	12361.56	0.38	0.24	7998	12999.52	0.43	0.30	7378	15101.28	0.34	0.20
s38584	10559	16576.11	0.28	0.13	10712	13602.82	0.43	0.31	10616	14500.95	0.32	0.20

Table 3.5 shows the CFA metrics along with the number of physical cells (abbreviated to *NoPC* in the table). Table 3.5 shows results for a representative set of the benchmark circuits. The total WDM appears in the column following the number of physical cells. The column designated NDS_T is the total NDS resulting from the WDM in the earlier columns and the normalizer computed for the design. As noted in Sec. 3.3.2, an NDS approaching 1 is better.

Considering only the NDS as a metric of manufacturability indicates the SR and UR variants to be equally manufacturable. However, note that there is a potential weakness

Type	Group	Priority	Rule Name	Metric : Combined
Width	Defect LithoOPC	1_HC	M3X.W.1_dfm.a	531161.88
Width	Defect LithoOPC	1_HC	M2X.W.1_dfm.a	373517.92
Width	Defect LithoOPC	1_HC	M4X.W.1_dfm.a	283936.08
Width	Defect LithoOPC	1_HC	M5X.W.1_dfm.a	225788.8
Transition	Defect Impro ProcessMargin Spice	1_HC	CO_dfm.b	200160.0
Space	Defect Impro LithoOPC	1_HC	M3X.S.1_dfm.a	133371.0
Enclosure	Spice	1_HC	PO.EX.4_dfm.a	124847.64
Enclosure	Spice	1_HC	PO.EX.5_dfm.a	124847.64
Transition	Defect Impro	1_HC	VIA1X_dfm.a	120905.2
Transition	Defect Impro	1_HC	VIA2X_dfm.a	114352.0
OPC	LithoOPC ProcessMargin	2_MC	M1.OPC.R.2_dfmt	84832.92
OPC	LithoOPC ProcessMargin	2_MC	PO.OPC.R.2_dfmt	80064.6
Transition	Defect Impro Spice	1_HC	CO_dfm.a	74088.0
Width	Defect LithoOPC	1_HC	M1.W.1_dfm.a	66132.8163333
Space	LithoOPC ProcessMargin Spice	1_HC	PO.S.2_dfm.b	64326.9
Space	Defect Impro LithoOPC	1_HC	M2X.S.1_dfm.a	55857.1266667
Space	Defect Impro LithoOPC	1_HC	M4X.S.1_dfm.a	53361.6
Width	ProcessMargin	2_MC	PO.W_dfm.a	39902.4
Space	Defect Impro LithoOPC	1_HC	M5X.S.1_dfm.a	38432.16
Space	LithoOPC ProcessMargin	1_HC	M1.S.3.2_dfm.a	29497.012
Area	Defect Impro LithoOPC ProcessMargin	2_MC	M2X.A.1_dfm	27555.185625
Distance	Defect Impro ProcessMargin	2_MC	CO.D.4_dfm	25862.112
Enclosure	Defect Impro LithoOPC Spice	1_HC	M2X.EN.1_dfmt	23838.9
Enclosure	Defect Impro LithoOPC Spice	1_HC	M3X.EN.1_dfmt	22848.4
Enclosure	Defect Impro LithoOPC Spice	1_HC	VIA2X.EN.1_dfmt	22827.02
Transition	Defect Impro	1_HC	VIA3X_dfm.a	19768.0
Enclosure	Impro LithoOPC ProcessMargin Spice	1_HC	M1.EX.1_dfm.a	16908.38625
Width	LithoOPC ProcessMargin	2_MC	M4X.W_dfm.a	15689.328
Enclosure	Impro LithoOPC ProcessMargin	2_MC	CO.EX.1_dfm.a	15556.878
Space	LithoOPC ProcessMargin Spice	1_HC	PO.S.11_dfmt	13096.0
Enclosure	Impro LithoOPC ProcessMargin	2_MC	CO.EN.1.1_dfm.a	12445.272
Enclosure	Defect Impro LithoOPC Spice	1_HC	VIA1X.EN.1_dfmt	11714.14225
Width	LithoOPC ProcessMargin	2_MC	M5X.W_dfm.a	10890.72
Width	CMP ProcessMargin	2_MC	M5X.W.4_dfmt	9912.984
Enclosure	Defect Impro LithoOPC Spice	1_HC	M3X.EN.2_dfmt	9766.0
Space	Defect Impro LithoOPC	2_MC	M1.S.1_dfm.a	7958.1636
Transition	Defect Impro	1_HC	VIA4X_dfm.a	7708.0
Enclosure	Impro LithoOPC ProcessMargin Spice	1_HC	M1.EN.1_dfm.a	7509.03050001
Width	LithoOPC ProcessMargin	2_MC	M3X.W_dfm.a	7353.504
Enclosure	Defect Impro LithoOPC Spice	1_HC	M2X.EN.2_dfmt	7116.675
Enclosure	Defect Impro LithoOPC Spice	1_HC	VIA1X.EX.1_dfm.a	6893.29166667
Space	LithoOPC ProcessMargin Spice	1_HC	PO.S.2_dfm	5931.93
Enclosure	Defect Impro LithoOPC Spice	1_HC	VIA2X.EX.1_dfm.a	5825.52
Width	LithoOPC ProcessMargin	2_MC	M2X.W_dfm.a	5507.018

GUIDELINE M3X.W.1_dfm.a / Description: Width if length [L>2.0] Bin DRC = [0.0 0.1] Bin IMPACT = [0.1 0.11] Bin ADVANCED = [0.11 0.115] Bin COMFORT = [0.115 0.13]

Figure 3.7: Individual CFA rule contributions for the various checks.

in the computation. The UR variants and SR variants display similar NDS values in spite of the fact that the normalizers for the UR implementations are comparable or larger than the normalizers for the SR implementations. The explanation for this lies in the computation method itself. For a given UR implementation, a large number of low weighted scores could lead to a large WDM; however the normalization process could still result in a NDS that is comparable to the NDS of the SR implementation of the same benchmark circuit. Since weights are assigned to potential defects based on

foundry experience, one cannot interpret this data without familiarity with the specific fabrication step involved. It is worthwhile to observe also that CFA produces totals for *all* manufacturability-related checks individually. In addition to checks related to the lithography process (affected most by the layout decisions), other potential weaknesses, like SPICE accuracy, particle defects, and CMP, are also included in the various totals. This in turn influences the total NDS value computed by CFA.

NDS values for only the lithography/OPC-related checks are also presented in Table 3.5 (abbreviated to NDS_L in the table). It should be noted here that there are overlapping checks related to particle defects affecting the lithography step that are included here as well. Fig. 3.7 shows a partial screenshot of one of the CFA runs. It can be seen from Fig. 3.7 that a fair number of checks are in the defect category and check interconnect layout geometries. The NDS for the LithoOPC group of checks reveals a similar trend to the overall totals.

Quite counter-intuitively, the SRX4 implementations show the best manufacturability based on the NDS as a score, indicating that cell diversity aids manufacturability indirectly given the dominance of interconnect-related checks.

3.8 Conclusions

In this part of the thesis I explored a number of regularity measures at the cell layout level intended to improve manufacturability. These measures were applied in the implemented set of eight logically complete combinational cells testing the impact of enforced interconnect unidirectionality and the implied impact of contacts and vias. The qualitative measures were tested using the Calibre CFA from Mentor Graphics. The results show counter-intuitive trends warranting further (likely expanded) research on different aspects of the topic.

Given the counter-intuitive nature of the results, the following useful observations can be made.

- At the cell level, unidirectional interconnect routing is not beneficial. In a larger design context where interconnect issues dominate device issues, unidirectionality may have a larger impact. The results from the CFA tool, with great emphasis on via- and contact-doubling, strongly suggest this.
- The results, put into perspective, also reveal the need for different analysis methods. For the cell level checks any use of higher metal layers is penalized, but in a larger design scenario the problem posed by vias would far outstrip those posed by devices, skewing the results unfairly. While the NDS provides a tractable measure, it could

be misleading. Perhaps, methods defining DFM metrics separately for the devices and for interconnect will provide a different picture.

- As far as manufacturability is concerned, regular layouts have empirically been shown to be as good as current designs [14]. However, the need to combine regular layouts with concepts for regular routing using automated methods is still an area requiring more research. Heuristic routing yields “good enough” solutions, but when the cost of manufacturing becomes critical due to the need for mask corrections, such methods may no longer pay dividends.
- From a manufacturing point of view, a designer working with cutting edge technology must accept the fact that the smallest geometries in the design, namely the gate related geometries, need patterning at the highest fidelity. Is it then possible to reduce interconnect masking cost by any means? The answer to this is a topic of analysis in itself and we refrain from commenting further on this here.
- Careful tuning of the regular structures is an important consideration to improve timing and reduce capacitance.
- Without knowledge of reliability numbers for vias and contacts, it is difficult to assess the tradeoff to enforce routing unidirectionality, especially when feature analysis tools penalize them heavily on account of lack of redundancy. One *ad hoc* solution is to adopt relaxed gate pitches and allow limited wrong way routing at the cell level. Assessment of manufacturability for the higher layers of metal should then be carried out with a different set of rules (or patterns, when the checks are model based).

The work presented here is in many ways simpler than the works surveyed in Sec. 3.2, but the motivation for doing so comes from consideration of a wider context spanning different levels of abstraction. In order to strengthen the indications of the results of this study, the next step involved application of the cells developed in this work to a complete design.

This was done using the ISCAS’89 benchmark circuits implemented with standard-cells designed with varying degrees of regularity in a commercial 65 nm process. A standard industrial flow is adopted in order to assess the impact regularity has on the manufacturability of a digital design. On average, 22% fewer vias are used by ultra-regular implementations. The DFM metrics measured at the signoff stage using integrated DFM tools, however, indicate relatively less manufacturability for the ultra-regular implementations. The primary reason for this seems to lie in the structure of the rule deck used to carry out DFM checks. There is a dominance of defect-related checks targeting the interconnect stack in the various (overlapping) categories. The lithography-

related checks show similar trends.

An essential need, therefore, is to reconcile the estimations carried out at design time with the actual manufacturing capabilities available. In order to enable predictive manufacturability assessment it is imperative that metrics be applicable across different levels of abstraction. This presents itself as a clear avenue for future work: identifying the exact nature of the gaps in the manufacturability assessment methods applied prior to signoff. Investigations of the causes can then be incorporated into improved methods for assessing manufacturability.

Bibliography

- [1] H.T. Heineken, J. Khare, and M. d'Abreu, "Manufacturability analysis of standard cell libraries," in *Custom Integrated Circuits Conference, 1998. Proceedings of the IEEE 1998*, May 1998, pp. 321–324.
- [2] W. Maly and J. Deszczka, "Yield Estimation Model for VLSI Artwork Evaluation," *Electronics Letters*, vol. 19, no. 6, pp. 226–227, 1983.
- [3] M. Lavin, Fook-Luen Heng, and G. Northrop, "Backend CAD Flows for "Restrictive Design Rules"," in *IEEE/ACM Int. Conf. Computer Aided Design*, Nov. 2004, pp. 739–746.
- [4] L. W. Liebmann, A. E. Barish, Z. Baum, H. A. Bonges, S. J. Bukofsky, C. A. Fonseca, S. D. Halle, G. A. Northrop, S. N. Runyon, and L. Sigal, "High-performance Circuit Design for the RET-enabled 65-nm Technology Node," in *Proc. SPIE*, 2004, vol. 5379.
- [5] Hirokazu Muta and Hidetoshi Onodera, "Manufacturability-Aware Design of Standard Cells," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E90-A, no. 12, pp. 2682–2690, Dec. 2007.
- [6] H. Sunagawa, H. Terada, A. Tsuchiya, K. Kobayashi, and H. Onodera, "Effect of Regularity-enhanced Layout on Printability and Circuit Performance of Standard Cells," in *Proc. Int. Symp. on Quality of Electronic Design*, Mar. 2009, pp. 195–200.
- [7] Yi-Wei Lin, M. Marek-Sadowska, and W.P. Maly, "Layout Generator for Transistor-Level High-Density Regular Circuits," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 2, pp. 197–210, Feb. 2010.

- [8] M. Pons, F. Moll, A. Rubio, J. Abella, X. Vera, and A. González, “VCTA: A Via-Configurable Transistor Array regular fabric,” in *18th IEEE/IFIP VLSI System on Chip Conference (VLSI-SoC)*, Sept. 2010, pp. 335–340.
- [9] Vinicius Dal Bem, Paulo Butzen, Felipe S. Marranghello, Andre I. Reis, and Renato P. Ribas, “Impact and Optimization of Lithography-Aware Regular Layout in Digital Circuit Design,” in *Proc. IEEE Int. Conf. on Computer Design*, Oct. 2011, pp. 279–284.
- [10] A.R. Subramaniam, R. Singhal, Chi-Chao Wang, and Yu Cao, “Design Rule Optimization of Regular Layout for Leakage Reduction in Nanoscale Design,” in *Proc. Asia and South Pacific Design Automation Conf.*, Mar. 2008, pp. 474–479.
- [11] M. Talalay, K. Trushin, and O. Venger, “Between Standard Cells and Transistors: Layout Templates for Regular Fabrics,” in *East-West Design Test Symp.*, Sept. 2010, pp. 442–448.
- [12] W. Maly, L. Yi-Wei, and M. Marek-Sadowska, “OPC-Free and Minimally Irregular IC Design Style,” in *44th ACM/IEEE Design Automation Conference*, June 2007, pp. 954–957.
- [13] N. Ryzhenko and S. Burns, “Physical Synthesis Onto a Layout Fabric with Regular Diffusion and Polysilicon Geometries,” in *Proc. Design Automation Conf.*, June 2011, pp. 83–88.
- [14] T. Jhaveri, V. Rovner, L. Liebmann, L. Pileggi, A.J. Strojwas, and J.D. Hibbeler, “Co-Optimization of Circuits, Layout and Lithography for Predictive Technology Scaling Beyond Gratings,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 29, no. 4, pp. 509–527, Apr. 2010.
- [15] M. Orshansky, S. R. Nassif, and D. Boning, *Design for Manufacturability and Statistical Design*, Springer Science+Business Media, LLC, 2008.
- [16] C. C. Chiang and J. Kawa, *Design for Manufacturability and Yield for Nano-Scale CMOS*, Springer Science+Business Media, LLC, 2007.
- [17] B.T. Murphy, “Cost-size Optima of Monolithic Integrated Circuits,” *Proc. IEEE*, vol. 52, no. 12, pp. 1537–1545, Dec. 1964.
- [18] R.B. Seeds, “Yield and Cost Analysis of Bipolar LSI,” in *Int. Electron Devices Meeting*, 1967, vol. 13, p. 12.
- [19] J.E. Price, “A New Look at Yield of Integrated Circuits,” *Proc. IEEE*, vol. 58, no. 8, pp. 1290–1291, Aug. 1970.

- [20] A.G.F. Dingwall, “High-yield-processed Bipolar LSI Arrays,” *IEEE Trans. Electron Devices*, vol. 16, no. 2, pp. 246–247, Feb. 1969.
- [21] Cadence Design Systems, *Encounter[®] Library Characterizer*, v. 10.1.2, 2011.
- [22] Cadence Design Systems, *Abstract Generator User Guide*, 2007.
- [23] Cadence Design Systems, *Encounter[®] Digital Implementation System*, v. 10.1.2, 2011.
- [24] ACM/SIGDA benchmarks (NCSU resource), “ISCAS Benchmark Circuits,” 2007, [Online Source].
- [25] K.P. Subramaniyan and P. Larsson-Edefors, “On Regularity and Integrated DFM Metrics,” in *Proc. 4th Asia Symp. on Quality Electronic Design (ASQED)*, 2012, pp. 211–218.
- [26] K.P. Subramaniyan and P. Larsson-Edefors, “Manufacturable Nanometer Designs using Standard Cells with Regular Layout,” in *Proc. 14th Int. Symp. on Quality Electronic Design (ISQED)*, 2013, pp. 398–405.
- [27] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits, A Design Perspective*, chapter 6,7,11, pp. 587–589, Prentice Hall Electronics and VLSI series, 2003.
- [28] N. H. E. Weste and D. Harris, *CMOS VLSI Design: A Circuits And Systems Perspective*, chapter 6, Addison-Wesley Publishing Company, 2007.
- [29] Mentor Graphics, *YieldAnalyzer and YieldEnhancer Reference Manual*, 2010, Calibre DFM Suite Datasheet.
- [30] C. Albrecht and Cadence Research Laboratories at Berkeley, “IWLS 2005 Benchmarks,” 2007, [Online Source].
- [31] J. Gambino, J. Wynne, J. Gill, S. Mongeon, D. Meatyard, H. Bamnolker, L. Hall, N. Li, M. Hernandez, P. Little, M. Hamed, and I. Ivanov, “Yield and Reliability of Cu Capped with CoWP using a Self-Activated Process,” in *Proc. Int. Interconnect Tech. Conf.*, June 2006, pp. 30–32.
- [32] A. Cabrini, D. Cantarelli, P. Cappelletti, R. Casiraghi, A. Maurelli, M. Pasotti, P.L. Rolandi, and G. Torelli, “A Test Structure for Contact and Via Failure Analysis in Deep-Submicrometer CMOS Technologies,” *IEEE Trans. on Semiconductor Manufacturing*, vol. 19, no. 1, pp. 57–66, Feb. 2006.
- [33] D.-S. Kim, W.-J. Ho, J.-Y. Kim, E.-Y. Shin, J.-H. Kim, and H.-D. Lee, “New Failure Analysis of Tungsten Plug Corrosion in Via Process,” in *Proc. 13th Int.*

- Symp. on the Physical and Failure Analysis of Integrated Circuits*, July 2006, pp. 355–358.
- [34] J.W. McPherson, “Reliability Trends with Advanced CMOS Scaling and the Implications for Design,” in *Proc. Custom Integrated Circuits Conf.*, Sept. 2007, pp. 405–412.
- [35] F. Chen, M. Shinosky, B. Li, J. Gambino, S. Mongeon, P. Pokrinchak, J. Aitken, D. Badami, M. Angyal, R. Achanta, G. Bonilla, G. Yang, P. Liu, K. Li, J. Sudijono, Y. Tan, T.J. Tang, and C. Child, “Critical Ultra Low-k TDDDB Reliability Issues for Advanced CMOS Technologies,” in *Proc. IEEE Int. Reliability Physics Symp.*, Apr. 2009, pp. 464–475.

In any collection of data, the figure most obviously correct, beyond all need of checking, is the mistake.

~Finagle's Third Law

4

MIDAS: Model for IP-inclusive DFM Assessment of System Manufacturability

Picking up from the end of the last chapter, one of the avenues for further work was identified as the need for improved methods of assessing manufacturability. Personally, I was also fascinated by the design paradigm of IP inclusion in modern SoCs. Combining these two aspects gave rise to this contribution: an early DFM metric for SoCs that is also IP inclusive. In working towards this objective I also sensed an opportunity of building on the earlier studies carried out during the course of this thesis. Between the end of the previous study and this one I had developed a set of standard-cell libraries conforming to the architecture defined in section 3.4. These updated libraries include AOI cells with up to six inputs. However, I was unable to use newer technology nodes as the necessary foundry data to analyze DFM was unavailable for them.

4.1 Introduction

System implementations with a robust cost-effort tradeoff use standard-cells as a distinct level of abstraction in the design of digital circuits. Due to the growing complexity of design management, macros of sub-systems have become indispensable to handle design complexity [1]. These macros may be memories or other hard Intellectual Property (IP) functions needed in the system. Typically, the macros are provided for use to the customer as a black box, with verified functionality guarantees from the vendor. Thus, integrating such blocks into a system eases the functional and performance verification effort on the part of the system designers. However, the macros, when considered for place and route, have constraints such as routing blockages which the layout engineer must account for during the place and route stage.

Performance and cost are the primary constraints applied to the development of systems. Traditionally, the cost aspect has translated into the area occupied by the design. In nanometer technologies, this is no longer true. Manufacturing complexities, mask creation in particular, dominates the cost of production in the latest nodes [2]. Considering the widespread use of standard-cell methodologies and the ever increasing use of IP in complex yield-limited environments, it is important to consider the implications of integrating big macros alongside a collection of small standard-cells on manufacturability [1].

Manufacturability analysis of standard-cells has been carried out from the perspective of yield [3], gate length distribution [4, 5], sensitivity analysis [6], and considerations such as reliability and routing [7]. Regular cell layouts have also been proposed as a means to enhance manufacturability [8, 9]. While qualitative DFM guidelines have been the main focus of existing literature, Gomez et al. [7] explicitly propose a quantitative manufacturability metric for standard-cells. Other attempts to introduce a metric for DFM have been carried out in [8, 9]. From the perspective of IP, Aitken [10] examines existing DFM metrics and practices. He does not propose any quantitative metric specific to IP but concludes that careful attention to DFM practices is required in the face of challenges imposed by explicitly incorporating variability into testing.

In this work, I propose MIDAS: Model for IP-inclusive DFM Assessment of System manufacturability. MIDAS is an additive model to compute a simple DFM metric to enable early assessment of DFM for System-on-Chips (SoCs). “Early” in this context refers to the earliest stage where realistic physical data become available. I hypothesize that if DFM costs for the standard-cells and IP blocks can be established, then system-level routing determines the overall manufacturability of the SoC. We can view standard-cells, IP blocks and system-level routing as discrete contributors towards the

manufacturability. Critical Feature Analysis (CFA) is used to motivate this hypothesis in the next section. I subsequently demonstrate the applicability of the proposed model in early analysis of DFM using an embedded processor system. The MIDAS model builds on existing techniques and extends the ability to coarsely predict manufacturability early in the design flow.

4.2 Motivation

We quantitatively motivate the MIDAS model through traditional DFM assessment of benchmark circuits from the ISCAS'89 [11] and IWLS'05 [12] suites, and also an embedded processor system (see Section 4.3.1 for details).

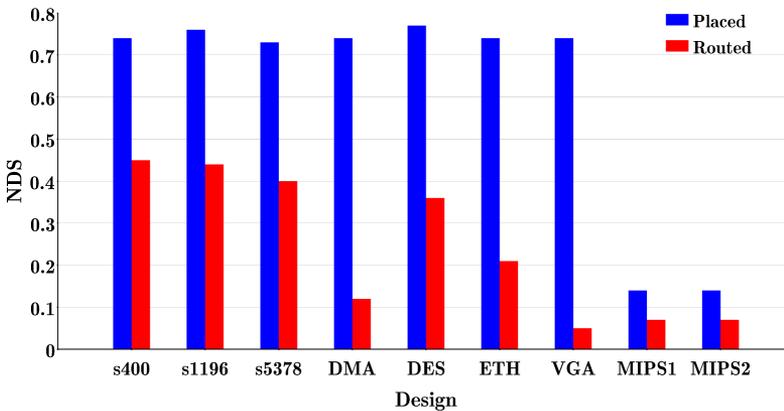


Figure 4.1: CFA for placed and routed designs.

After place and route, the implementations were imported into the Cadence Virtuoso environment for DFM assessment, which is enabled through Calibre CFA [13], using foundry-provided rule sets. This tool is a part of DRC and LVS tools belonging to the Mentor Graphics Calibre suite and relies on detailed rule- or model-based checks to provide metrics on resilience to particle defects, modeling accuracy and process margins. Scores from individual (categorized) rules are summed to form the Weighted DFM Metric (WDM) and the result is normalized to a number based on the number of devices in the design. A bound is established using the negative exponentiation of the normalized value to give the Normalized DFM Score (NDS). The WDM can have any value from 0 to infinity, while the negative exponentiation restricts the value of the NDS be-

tween 0 and 1. Being cumulative, a lower WDM is desirable for manufacturability or, conversely, a design with a NDS approaching 1 has greater resilience to process defects.

In order to accurately capture the effects of all the system-level constraints, stream data was saved for the placed design as well as the routed design so that the results of CFA could be compared. Figure 4.1 shows the results of the CFA analysis. The exact values involved are presented in Table 4.1. Here, the first column indicates the design that is implemented. All except the last two are benchmark circuits from the ISCAS'89 [11] and IWLS'05 [12] suites. The MIPS1 and MIPS2 designs are the embedded processor system with two different floorplans, details of which are outlined in Section 4.3.1. The next column is a count of the standard-cells and the number of macros (if any), followed by the status of the stream data of the design in the next column. The numbers in the next two columns are the metrics produced by CFA. The WDM shows the dependency of the cumulative cost on the design size.

Table 4.1: CFA for various sample implementations.

Design	# Cells	Status	NDS	WDM
s400	121	Placed	0.74	61.11
		Routed	0.45	175.76
s1196	348	Placed	0.76	202.48
		Routed	0.44	625.43
s5378	1005	Placed	0.73	546.33
		Routed	0.40	1720.11
DMA	24525	Placed	0.74	5611.10
		Routed	0.12	50939.78
DES	74605	Placed	0.77	37939.55
		Routed	0.36	148671.49
ETH	27748	Placed	0.74	11815.62
		Routed	0.21	62310.10
VGA	41886	Placed	0.74	16170.13
		Routed	0.05	164527.74
MIPS1	16402 + 14	Placed	0.14	145551.56
		Routed	0.07	199129.62
MIPS2	16060 + 14	Placed	0.14	145508.72
		Routed	0.07	193732.05

It is clear from Figure 4.1 and Table 4.1 that, irrespective of the design size, system-level routing affects the NDS; by as much as 70% in some cases. It must be noted here that the generally low NDS values for the MIPS designs occur as a result of the memory macros present in the design. The hard macros used in the implementations are geometrically accurate, but have the active device layers abstracted out. This results

in inaccuracies in the NDS computation, additionally so due to the area impact of the macros on the overall area. Excluding the macros from consideration during assessment increases the NDS value to match the NDS for the benchmark circuits proving that complete geometry data is necessary for accurate computation.

It can also be seen that the NDS for the placed designs is almost constant throughout (about 0.75 for the benchmark circuits and 0.14 for the MIPS designs), leading to the conclusion that the system building blocks present a base cost towards manufacturability. The fact that this value degrades to the NDS of the routed designs means that the system-level wiring alone contributes to this degradation. Thus, for a coarse estimate, the main contributions towards assessing DFM can be viewed discretely as the building blocks of the circuit and the system-level routing.

In addition to quantitatively motivating the contributors towards system manufacturability, we use this traditional DFM flow to generate base costs for the standard-cells used in this study. The cost so computed is applied in the early DFM assessment model.

Section 4.3 outlines the background, presenting the infrastructure involved at various levels of abstraction. Section 4.4 outlines the various components of the proposed model and the overall DFM metric. Validation results from the MIDAS model are presented in Section 4.5 followed by a demonstration of IP inclusion into the model in Section 4.6. Finally, the conclusions of this study are presented.

4.3 Environment and Tools

The stated objective of this work is to present a scalable, and IP-inclusive model to enable early prediction of DFM for SoCs. In order to be able to show the applicability of MIDAS, it is important to target a system that is complex enough to require different blocks (cells vs macros). The test vehicle used to achieve this is an embedded processor system. Additionally, given that the MIDAS model is based on component costs, details of the building blocks (cell or IP) are required. The following headings outline the details at various levels of abstraction along with the EDA tools required.

4.3.1 System-Level Implementation

We use a MIPS processor with a five-stage pipeline [14] and a level-one (L1) cache as the test vehicle in this work. The CPU consists of the standard pipeline units of fetch, decode, register file, ALU, and memory write-back and is augmented with a 32-bit integer multiplier. Each of the 16kB L1 data and instruction caches is implemented

with four SRAM memory macros of size 1024×32 -bit and three 128×32 -bit SRAM blocks for tags. The processor datapath has about 10K logic cells.

Additionally, we implement the processor system using two different floorplans, which utilize the memory macros in different positions in order to explore the sensitivity of the model to different system-level considerations. The floorplan, in combination with the routing blockages presented by the macros, determines the routing solution for the system. This, in combination with settings varying the row density (resulting in larger or smaller dies) and the different libraries available (see Section 4.3.2) for implementation, enables a viable number of test points to be generated. The memory macros used to implement the cache and tags are the same in all implementations and enforce routing blockages for metal layers up to M5. The macros are placed such that they lie in close proximity to the control blocks.

The first exploratory floorplan is a custom-made one referred to by the acronym “FPC” from here on. The other, a floorplan similar to those seen in industrial processor designs, is referred to by the acronym “FPI” for the rest of this work. The floorplans are laid out as shown in Fig. 4.2 for implementation with the different library sets. Synthesis was carried out using Cadence RTL Compiler [15], while place and route was carried out using Cadence Encounter Digital Implementation System (EDI) [16].

4.3.2 Standard-Cell Libraries

One of the most important aspects involved in MIDAS is to be able to assign base costs to standard-cells. To this end, we develop the standard-cells that are used in the implementations. This allows us to have complete control over the data generation process and additionally, cell libraries with distinct characteristics and for which accurate costs can be established are available for use with MIDAS. As compared to the libraries in Chapter 3, the libraries created for this work are more full fledged in terms of diversity.

The architecture of the libraries used here are in line with the architecture outlined in section 3.4. The shapes and geometries of the devices in the first of the custom libraries match those available in commercial standard-cells. In addition, routing is completed using poly wherever possible. We will refer to this library using the tag “PoR” from here on. The second library contains cells with device widths which are uniform and, additionally, unidirectional poly routing is adopted. In the case of this library, routing is completed using M2 in the vertical direction only. In order to keep the amount of M2 in the cells to a minimum, it was decided to allow small M1 jogs. We will refer to this library using the tag “M2R” from here on. A variant of the M2R library, using only M1 routing, is also available and is termed “M1R”.

The three libraries contain the same set of cells:

- A logically complete set of cells, including an inverter.
- A non-inverting buffer.
- A few variants of full- and half-adders.
- And-Or (AO), Or-And (OA), And-Or-Invert (AOI) and Or-And-Invert (OAI) cells with up to six inputs.
- Two flip-flops; one D- and one scan-enabled flip-flop with minimum drive strength.

Owing to the effort involved in creating a large number of cells, the drive strengths were restricted to minimum (X2) and twice the minimum (X4). This gives us 100 cells in each library.

The standard-cells used here were developed using an industrial 65 nm full-custom flow. Cadence Virtuoso [17, 18] was used to create the schematics and layouts. Design Rule Check (DRC) and Layout Versus Schematic (LVS) checks were carried out using the Mentor Graphics Calibre suite of tools [19]. Parasitic extraction was carried out using Synopsys StarRC [20]. Characterization was performed for 1.2 V operation at standard threshold voltage using Cadence Encounter Library Characterizer [21].

4.4 MIDAS: Model for IP-inclusive DFM Assessment of System manufacturability

The MIDAS model computes a DFM metric in a manner much the same as used to predict yield. However, instead of using defect densities alone to predict failure probabilities, the cost of each component (cell, IP and routing) is computed using costs incorporating the risk associated with the manufacturing steps including particle defects. From the motivational data presented in Section 4.2, we can identify two main components in a system-level implementation:

- The device components comprising standard-cells and IP blocks.
- The interconnect components comprising wires and vias.

The cost of standard-cells is computed using CFA in this work as indicated earlier, while IP cost can either be a pre-computed CFA metric or coarsely estimated by other means. Predicting the manufacturability of a particular routing solution requires some knowledge of the manufacturing process, but is nonetheless simple once the basis for computation is established.

Weight can be considered to be a product of the criticality of a component or geometric feature and the risk in a given geometric context. Indeed, computations of this type are applied in various risk assessment schemes such as Failure Mode and Effect Analysis (FMEA) [22, 23]. As such, both the criticality and risk values are empirically determined and assigned by the foundry. However, with some experience, for coarse estimates realistic values can be assumed. The considerations for weighting are explained for each case in the following subsections.

4.4.1 Placement Cost

The device components comprise the standard-cells and the IP, which are interconnected in some fashion to form an SoC. The cost for such blocks can be modeled using techniques such as CFA in order to obtain as accurate a value as possible. However, IPs are typically available as macros for which detailed implementation details are scarce. In such a scenario, alternate means must be employed to assess a cost for such blocks.

In an IP-inclusive scenario, the total **Placement Cost (PC)** is simply the sum of the placement costs for standard-cells and IP blocks. This is expressed as:

$$PC = PC_c + PC_m \quad (4.1)$$

Standard-Cell Cost

We begin by considering the WDM for the custom cells as a measure of placement cost for the standard-cells. The PC for standard-cells (PC_c) can then be modeled as a product of the number of instances of a given cell and its WDM:

$$PC_c = \sum_{i=c1}^{cK} N_i \times WDM_i \quad (4.2)$$

$c1$ and cK refer to the distinct types of cells in the design, N_i refers to the number of instances of a particular cell, and WDM_i is the cost associated with a single instance of the cell.

In this work, since the size of the cells in the custom libraries is limited, the spread of the NDS is also limited (Figure 4.3). The inverters in the libraries display the lowest values of NDS and represent the lower bounds of the spread. We use the product of the average WDM value of the library and the number of cells as the PC in order to ease the computational effort. A typical commercial library contains a much larger spread of drive strengths that will make it necessary to utilize accurate cost values in order to

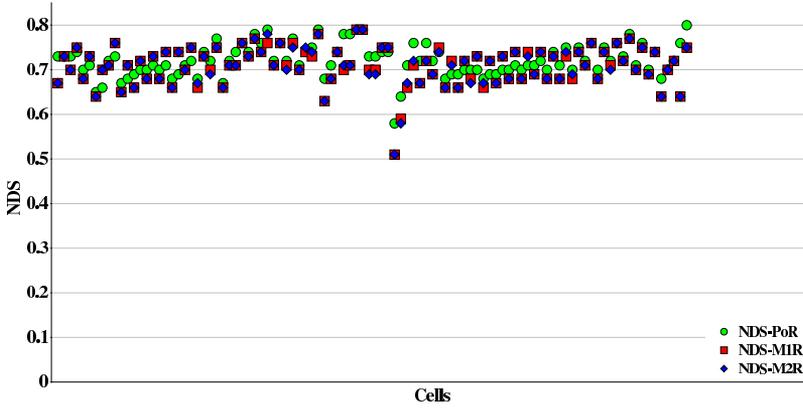


Figure 4.3: Scatter plot of NDS values for cells in the custom libraries.

accurately assess the standard-cell cost. However, with full automation of the process a much more accurate computation can be carried out in order to increase the accuracy.

IP Cost

Hard macros or IP blocks incur a placement cost in the system-wide context depending on the floorplan and the routing obstructions that the block enforces. The floorplan, influenced by the macros, also affects the core area of the SoC as well as the routing. The obstructions presented by the IPs mainly affect the routing. In the context of placement, the placement cost of incorporating IPs can be described as:

$$PC_m = \sum_{i=m1}^{mK} N_i \times C_i \quad (4.3)$$

Here $m1$ and mK refer to the distinct types of macros present in the design, N_i refers to the number of instances of each type of macro, and C_i refers to the weight of the IP block in question, be it the WDM or any other measure used.

Availability of an accurate weight certainly increases the accuracy of MIDAS and assumes great importance when the paradigm of IP-dominated designs is taken into account. However, for a coarse estimate the cost of an IP block can be approximated using known WDM values. Consider a memory macro of size $1024 \times 32b$, which is a hard macro with abstract active layers in the test implementations. It is known that each cell in the SRAM memory core consists of six devices, so if we consider the cost

per cell using the WDM of a 6-device logic gate, then the cost per memory cell can be approximated to 1.5. The total cost of the memory core can then be computed as $1024 \times 32 \times 1.5 = 49152$. Referring back to Table 4.1, we can see that the WDM for DMA (~25K cells) and ETH (~28K cells) have comparable values. Note, however, that in these cases there are a number of diverse standard-cells in the design. If the number of logic cells in the memory macro is assumed to be the same as the number of core memory cells, then this cost can be doubled to give a value of 98304. Accounting for the dense, regular nature of the macro, a conservative cost of 90000 is used for computations in subsequent sections. Similarly the $128 \times 32b$ macro is assigned a cost of 9000.

4.4.2 Interconnect Cost

Interconnect cost can be split into two distinct components, vias and wiring, each requiring individual treatment. The total **Interconnect Cost (IC)** is simply the sum of interconnect cost of vias and wiring:

$$IC = IC_v + IC_w \quad (4.4)$$

The following headings detail each of the components.

Layer Change Cost

Manufacturing limitations create risks when vias are introduced while changing layers. Long recognized as one of the yield-limiting features [24], this forms one component of the interconnect cost. A general equation to represent the via cost is:

$$IC_v = \sum_{i=vsc}^{vmc} N_i \times R_i \times C_i \quad (4.5)$$

The bounds of summation, vsc and vmc , refer to the types of vias used in the implementation. These, in order of decreasing risk, are single-cut vias and multi-cut vias. The R_i term refers to the risk for a particular type of via, while C_i refers to the criticality. The risk and criticality associated with a particular type of via is typically dependent on empirical values that the foundry determines. Thus, knowing the number of instances of each type of via enables us to weight it reasonably to compute the cost of vias of a design.

As a matter concerning accuracy, it must be noted here that further granularity can

be obtained by using instances for layer pairs with more accurate weights to ascertain this cost. The expression for the cost of vias is then modified to:

$$IC_v = \sum_{i=vs_c}^{vmc} \left[\sum_{j=lp1}^{lpK} N_{ij} \times R_{ij} \times C_{ij} \right] \quad (4.6)$$

Equation 4.5 is used exclusively in this work. Here we assign a via risk of 0.08 for single-cut vias and 0.02 for multi-cut vias. Assuming criticality of 5 and 3 for single-cut and multi-cut vias, respectively, the weight can be computed as a product of the risk and criticality. Statistics of the numbers of each type of via are obtained through the EDI command `pdi report_design`.

Wire Spacing Cost

In typical semi-custom design flows, the wire layers are directionally constrained to either be horizontal or vertical in order for heuristic routing to work. Thus, the weight due to a certain layer is limited, since the criticality for wire segments running in the same direction becomes a function of the space between them alone. Additionally, the different layers can be categorized into bins depending on the similarity of their geometries. Typically, lower layers display smaller geometries and pitches, and thus warrant a higher criticality. Risk is assigned based on the pair-wise spacing in a layer, in multiples of minimum spacing as required of DRC. A pair separated by the minimum space is more prone to defects than one with a pair with larger spacing. However, it is not critical to consider wire widths. While this is an important parameter that should be exploited to gain increased resilience to electromigration and noise immunity, the measure of wire-widening is never applied at the cost of area. Hence from an early estimation perspective, it is more critical to include meaningful spacing statistics. Thus, as alluded to earlier, layer-wise data on spacing is sufficient to compute a coarse cost of routing in order to establish a DFM metric.

Such a wire spacing cost can be represented as:

$$IC_w = \sum_{i=b1}^{bn} \left[\sum_{j=l1}^{lK} N_j \times C_j \right] \times R_i \quad (4.7)$$

As before, according to this notation, C_j represents criticality of layer j while R_i is the risk associated with bin i . In this work, we use layer-wise spacing statistics produced

using the EDI command `pdi report_dfm_metric`¹. Layers M1 through M3, in the eight layer process used for the implementations, comprise the first criticality bin and are assigned a criticality of 5. Similarly, layers M4 through M6 are assigned a criticality of 3 and the top two layers are assigned a criticality of 1. The risk for computing IC_w is assigned based on the the spacing bins: instances with minimum spacing are assigned a risk of 0.9; those with twice the minimum spacing are assigned a risk of 0.2 and instances at three times the minimum spacing are assigned a risk of 0.05. Instances having a spacing greater than this are judged to be more or less immune to the vagaries of the manufacturing process.

4.4.3 Total DFM Cost and Normalization

Sections 4.4.1 and 4.4.2 cover the components of the early DFM assessment model. The placement components are governed by Equations 4.1, 4.2 and 4.3, while the routing components are governed by Equations 4.4, 4.5 and 4.7.

The total Design Manufacturability Cost (DMC) of the design can now be expressed as:

$$\text{DMC} = \text{PC} + \text{IC} \quad (4.8)$$

This represents the overall cost of manufacturability of the design, while each of the individual components represents a measure for the manufacturability arising out the more abstract design decisions of the respective components. In order for the DMC to be useful it must be normalized. The normalization in this work is carried out against a value representing worst-case cost. This normalization cost holds little meaning in terms of a product, but is a theoretical representation of the worst-case risk indicative of a non-functional design. This value can be computed by assuming the highest criticality and worst bins for all components of the MIDAS model. For standard-cells, this is simply the product of the total number of cells and the worst WDM among them. The macro cost, if applicable, is the product of the number of macros and the cost of the macros. This cost is typically constant across the calculations, since implementation details for IP are typically unavailable. For worst-case routing cost, we consider all vias to be single cut and all the wire instances reported by `pdi report_dfm_metric` to be in the M1 layer with minimum spacing.

¹ The NanoRoute router provides both `pdi report_design` and `pdi report_dfm_metric`.

Equations 4.9, 4.10, 4.11 and 4.12 show all of the component expressions.

$$PC_{cwc} = N_{sc} \times WDM_{worst}, \quad (4.9)$$

$$PC_{mwc} = \sum_{i=m1}^{mK} N_i \times C_i, \quad (4.10)$$

$$IC_{vwc} = N_v \times R_{sc} \times C_{sc}, \quad (4.11)$$

$$IC_{wwc} = N_{wi} \times R_{MinSpace} \times C_{M1} \quad (4.12)$$

and finally, the normalizer can be expressed as:

$$\mathbf{Norm} = \mathbf{PC}_{cwc} + \mathbf{PC}_{mwc} + \mathbf{IC}_{vwc} + \mathbf{IC}_{wwc} \quad (4.13)$$

The DMC computed in Equation 4.8 can now be normalized to this value to express the fraction of the design cost to the total worst-case cost. The Design Manufacturability cost Normalized (DMN) is expressed as:

$$\mathbf{DMN} = \frac{\mathbf{DMC}}{\mathbf{Norm}} \quad (4.14)$$

A figure-of-merit (FoM) for manufacturability can then be expressed as:

$$\mathbf{FoM} = (1 - \mathbf{DMN}) \quad (4.15)$$

This value is indicative of the total risk that can be *avoided* as a result of the design decisions related to floorplanning, choice of standard-cells and IP selection.

4.5 Model Calibration

In order to test the sensitivity of the MIDAS model to various DFM considerations, we implemented the datapath portion of the MIPS system described in Section 4.3.1. Among the various considerations tested at this level were:

1. *Sensitivity to cell architecture:* Different logic libraries, described in Section 4.3.2, were employed in the implementation of the MIPS datapath to test the sensitivity of MIDAS to standard-cell architecture.
2. *Sensitivity to IP inclusion:* The ALU and multiplier which are employed in the MIPS datapath were constructed as macros to test the behavior of MIDAS in the presence of macros of different sizes.

3. *Sensitivity to IP cost*: The sensitivity to the cost of including IPs was tested using the MIPS datapath. The overall metric was computed using the WDM and again, using the cost occurring as a result of the MIDAS model.
4. *Sensitivity to routing blockages*: In order to test the MIDAS model for effects introduced by routing blockages in IP blocks, the ALU and multiplier were implemented as macros with routing blockages.

Table 4.2: *Computation of an early DFM metric for the MIPS datapath.*

Lib.	PC	IC	DMC	Normalizer	DMN	FoM	% Full	% Mod	Comment
PoR	13554.57	402756.51	416311.08	1879996.34	0.22144	0.77856	-	-	Full datapath. FoM calculated using the WDM.
M1R	14456.64	313253.98	327710.62	1659795.88	0.19744	0.80256	-	-	
M2R	14580.96	329828.95	344409.91	1738066.48	0.19816	0.80184	-	-	
PoR	48699.45	339559.49	388258.94	1661614.58	0.23366	0.76634	-1.57	-	ALU as a macro; using model for FoM computation.
M1R	46063.67	294778.73	340842.40	1604774.85	0.21239	0.78761	-1.86	-	
M2R	47626.64	308431.13	356057.77	1716607.05	0.20742	0.79258	-1.16	-	
PoR	14610.22	339559.49	354169.71	1627525.35	0.21761	0.78239	0.49	2.09	ALU as a macro; using WDM for FoM computation.
M1R	16088.33	294778.73	310867.06	1574799.51	0.19740	0.80260	0.005	1.90	
M2R	16819.87	308431.13	325251.00	1685800.28	0.19294	0.80706	0.65	1.83	
PoR	97138.78	291808.35	388947.13	1487075.54	0.26155	0.73845	-5.15	-	Multiplier as a macro; using model for FoM computation.
M1R	83123.29	267881.74	351005.03	1485928.15	0.23622	0.76378	-4.83	-	
M2R	80331.74	259595.83	339927.57	1478428.54	0.22992	0.77008	-3.96	-	
PoR	18322.87	291808.35	310131.22	1408259.63	0.22022	0.77978	0.16	5.60	Multiplier as a macro; using WDM for FoM computation.
M1R	19797.03	267881.74	287678.77	1422601.89	0.20222	0.79778	-0.60	4.45	
M2R	19464.91	259595.83	279060.74	1417561.71	0.19686	0.80314	0.16	4.29	
PoR	132095.02	242235.87	374330.89	1422552.10	0.26314	0.73686	-5.36	-	ALU and multiplier as macros; using model for FoM computation.
M1R	114435.80	230060.05	344495.85	1385292.93	0.24868	0.75132	-6.38	-	
M2R	112717.34	222045.30	334762.64	1368710.47	0.24458	0.75542	-5.79	-	
PoR	19189.88	242235.87	261425.75	1309646.96	0.19962	0.80038	2.80	8.62	ALU and multiplier as macros; using WDM for FoM computation.
M1R	21134.20	230060.05	251194.25	1291991.33	0.19442	0.80558	0.38	7.22	
M2R	21043.74	222045.30	243089.04	1277036.87	0.19035	0.80965	0.97	7.18	
PoR	132262.70	296639.77	428902.47	1575027.36	0.27231	0.72769	-6.53	-	ALU and multiplier as macros; with routing blockages; using model for FoM computation.
M1R	114574.92	264961.14	379536.06	1525187.57	0.24885	0.75115	-6.41	-	
M2R	112803.18	265044.35	377847.53	1518056.29	0.24890	0.75110	-6.33	-	
PoR	19357.56	296639.77	315997.33	1462122.22	0.21612	0.78388	0.68	7.72	ALU and multiplier as macros; with routing blockages; using WDM for FoM computation.
M1R	21273.32	264961.14	286234.46	1431885.97	0.19990	0.80010	-0.31	6.52	
M2R	21129.58	265044.35	286173.93	1426382.69	0.20063	0.79937	-0.31	6.43	

Data required for MIDAS were collected from the different implementations. The results of the FoM computation are presented in Table 4.2. Here, for each of the MIPS datapath implementations, the first column shows the logic library used in the implementation, while the last column describes the constraints of the implementation. Columns two

through seven indicate the PC, the IC, the DMC, the normalizer, the DMN, and the FoM. In the two columns following the FoM, the percentage change of the FoM is displayed for two cases: The FoM for a particular implementation compared to the “Full datapath” implementation and the FoM calculated using the WDM as compared to the FoM calculated using MIDAS. Note that the “Full datapath” implementation serves as a reference since, consisting entirely of standard-cells, the most accurate costs are available for this implementation.

A number of observations can be made in Table 4.2. The FoM values in the results here are not extremely sensitive to the cell architecture as a result of the fact that average values are used in the estimation. In reality a number of factors other than this affect the value. For example, in order to ensure power efficiency, a number of libraries with different threshold voltages are usually mixed, resulting in different costs for the cells. If instances of cells from the different libraries occur in substantial numbers, which is likely to be the case for a larger design, the effect on the accuracy will be more pronounced. Additionally, if the actual cell costs are incorporated instead of the average, the FoM will be more accurate. From these results, however, it can be said that the libraries with more regular geometries (M1R/M2R) result in a marginally better FoM than the less regular library (PoR). Note that this is the case in spite of the fact that the average WDM is worse for the M1R and M2R libraries when compared to the PoR library (1.48 vs. 1.31).

Table 4.2 also shows that when the DFM model is used to create the cost for macros, the estimation tends to be pessimistic. This can be established from the fact that when the FoM for such implementations (rows with “using model for FoM computation”) are compared against the FoM predicted for the “Full datapath” implementation (for similar libraries), smaller values are predicted. In these results up to ~7% pessimism is observed. In contrast to this, usage of WDM (rows with “using WDM for FoM computation”) for assigning macro costs is more optimistic with predictions up to ~3% higher. The FoM does not change substantially when both the ALU and multiplier are included as macros showing that the sensitivity to IP inclusion is tolerable.

On a related note, using WDM values for macros during computation of the FoM results in more optimistic prediction than using the model itself. Note that the implementation for which the manufacturability is being assessed stays the same; only the method of assigning cost for the macro changes. Up to ~9% higher values are seen in this comparison. The particular case for which this occurs is the implementation using the PoR library with both the ALU and the multiplier as macros and no routing blockages enforced. Note that, when compared against the “Full” implementation, the FoM using MIDAS is ~5% less while the FoM using WDM is ~3% more. This shows that

an accurate cost for the IP provides a better estimate for the SoC, confirming the need for accurate DFM metrics for IPs. That said, the estimation provided by MIDAS shows tolerable error considering that this is early estimation. Considering the last two rows in Table 4.2, we observe that the MIDAS model does not seem to display any sensitivity to routing blockages. This is because there is no penalty assigned to using the upper level metal layers for routing. The only consideration is a legal routing solution that is verified through traditional means.

Table 4.3: *Statistics for datapath implementations considering routing blockages.*

Design Variant	# Cells	Wire Length	# Vias
Without Routing blockages	5791	270688.23 μm	51518
With Routing blockages	5885	301446.34 μm	52594

It may be noted in Table 4.3 that the blockages affect the wire length and the number of vias. The table shows the design statistics for MIR-based datapath implementations with the ALU and multiplier implemented as macros. The first row contains the implementation with no blockages while the next row shows the implementation with blockages. The same trend is seen for implementations with the other libraries as well and this in turn will affect parametric yield and timing closure if not accounted for during later design stages.

4.6 A Practical Test Case & Use Scenarios

The results in Table 4.2 show that MIDAS provides a reasonable estimate of the manufacturability of a design. However, the effects of floorplan and cell density in an IP-limited scenario remain to be tested. For this purpose I use implementations of the MIPS system described in Section 4.3.1 along with the libraries in Section 4.3.2. The initial row density is specified during the configuration phase of the design and manual floorplanning was used to ensure that placement violations did not occur as a result of the macros. The initial cell densities used were 30%, 50% and 70%. In the last two cases, the die area generated using the default density had to be resized in order to legally accommodate all the memory macros. This shows that in an IP-dominated SoC, the cell density settings are dominated by the IP geometry. Table 4.4 shows the results of the computation of the FoM for these implementations. The initial density settings are referred to by the labels “D1”, “D2” and “D3”.

The placement cost varies very little and can as such be considered constant for a

Table 4.4: *Computation of an early DFM metric for MIPS system.*

FP	Lib	Den	PC	IC	DMC	Normalizer	DMN	FoM
FPC	PoR	D1	795168.29	651316.74	1446485.03	4851294.48	0.29816	0.70184
		D2	798311.96	734401.75	1532713.71	4998711.34	0.30662	0.69338
		D3	798340.08	757684.01	1556024.09	5127612.22	0.30346	0.69654
	M1R	D1	797968.60	603192.43	1401161.03	4887408.15	0.28669	0.71331
		D2	798260.16	675637.41	1473897.57	4997883.62	0.29490	0.70510
		D3	798274.96	666264.15	1464539.11	5086088.02	0.28795	0.71205
	M2R	D1	797765.84	589896.89	1387662.73	4858233.12	0.28563	0.71437
		D2	798159.52	681282.71	1479442.23	4903724.16	0.30170	0.69830
		D3	797968.60	709930.21	1507898.81	5201805.95	0.28988	0.71012
FPI	PoR	D1	795026.81	677368.35	1472395.16	4813556.92	0.30589	0.69411
		D2	795110.65	660922.55	1456033.20	4774847.30	0.30494	0.69506
		D3	794937.73	651644.11	1446581.84	4714415.96	0.30684	0.69316
	M1R	D1	797755.48	608188.77	1405944.25	4890182.71	0.28750	0.71250
		D2	797853.16	632915.53	1430768.69	4888054.37	0.29271	0.70729
		D3	797768.80	621955.07	1419723.87	4839055.90	0.29339	0.70661
	M2R	D1	797696.28	615930.03	1413626.31	4913579.59	0.28770	0.71230
		D2	797727.36	648237.75	1445965.11	4922484.98	0.29375	0.70625
		D3	797690.36	623483.36	1421173.72	4841975.43	0.29351	0.70649

given design with a particular library and IP set. The interconnect cost on the other hand varies quite substantially. The largest IC cost is 28.4% larger than the least, while on average the FPI floorplan yields ~5% less interconnect cost. The trends seen earlier with respect to the effect of the logic library on the FoM continues here with the M1R and M2R libraries displaying better manufacturability than the PoR library.

As a final test, with the same implementations the weights were changed: the IP costs were increased 10% and the via risks reduced by an order of magnitude. It is worth noting that under these conditions the FoM trends remained roughly the same, indicating that the MIDAS model scales mainly according to the design.

In terms of prediction, the FoM provides a scaled measure of the total risk that can be avoided with the current combination of standard-cells, IP and floorplan. The individual components—the PC and the IC—provide a design-specific measure of the risk contributed by each of the components. Splitting this down further enables more specific diagnosis; as a general rule, the greater the granularity, the greater the capability of specific diagnosis. For example, if multiple libraries are involved, then library specific sub-totals can indicate how an optimal mix of cells can be used to achieve overall yield targets. If the cost of a particular IP (ideally, internally created) is high when used in a design specific scenario, it may warrant changes to enable meeting overall goals.

Note, however, that MIDAS does not specifically pin-point DRC violations or parametric violations. These must be dealt with in other ways so as to ensure a clean hand-off

to the foundry. It is possible to set individual budgets for each of the components of MIDAS and iteratively attempt to meet the goals.

4.7 Conclusions

I have presented MIDAS; a model to enable the early prediction of DFM, built on the basis of the hypothesis that standard-cells, IP and routing components contribute discretely to manufacturability. The model uses spacing related routing statistics in addition to costs for standard-cells and IP blocks ascertained using existing DFM techniques, to determine a figure-of-merit for the manufacturability of a design. The MIDAS model is calibrated for different considerations on a MIPS datapath design and is then demonstrated on a processor system with a L1 cache. Commercial memory macros were used in the implementation of the cache. Different floorplans and custom logic libraries demonstrate the capabilities of MIDAS. From the results presented in Sections 4.5 and 4.6, it can be concluded that such a simple additive model provides useful insight into design specific yield limitations, while the FoM allows the designer to establish a normalized measure towards fulfilling the overall yield goals at very little additional effort.

Bibliography

- [1] International Technology Roadmap for Semiconductors 2011 Edition, “System Drivers,” 2012, [Online Source].
- [2] International Technology Roadmap for Semiconductors 2011 Edition, “Lithography,” 2012, [Online Source].
- [3] H.T. Heineken, J. Khare, and M. d’Abreu, “Manufacturability analysis of standard cell libraries,” in *Custom Integrated Circuits Conference, 1998. Proceedings of the IEEE 1998*, May 1998, pp. 321–324.
- [4] Hirokazu Muta and Hidetoshi Onodera, “Manufacturability-Aware Design of Standard Cells,” *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E90-A, no. 12, pp. 2682–2690, Dec. 2007.
- [5] H. Sunagawa, H. Terada, A. Tsuchiya, K. Kobayashi, and H. Onodera, “Effect of Regularity-enhanced Layout on Printability and Circuit Performance of Standard Cells,” in *Proc. Int. Symp. on Quality of Electronic Design*, Mar. 2009, pp. 195–200.

- [6] S. Sundareswaran, R. Maziasz, V. Rozenfeld, M. Sotnikov, and M. Konstantin, “A Sensitivity-aware Methodology to Improve Cell Layouts for DFM Guidelines,” in *Proc. 12th IEEE Int. Conf. on Quality Electronic Design*, Mar. 2011, pp. 1–6.
- [7] S. Gomez and F. Moll, “Evaluation of Layout Design Styles using a Quality Design Metric,” in *Proc. of IEEE Int. SOC Conference*, 2012, pp. 125–130.
- [8] T. Jhaveri, V. Rovner, L. Liebmann, L. Pileggi, A.J. Strojwas, and J.D. Hibbeler, “Co-Optimization of Circuits, Layout and Lithography for Predictive Technology Scaling Beyond Gratings,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 29, no. 4, pp. 509–527, Apr. 2010.
- [9] T. Jhaveri, L. Pileggi, V. Rovner, and A. J. Strojwas, “Maximization of layout printability/manufacturability by extreme layout regularity,” 2006, vol. 6156, pp. 615609–615609–15.
- [10] R. Aitken, “The Design and Validation of IP for DFM/DFY Assurance,” in *IEEE Int. Test Conf.*, Oct. 2006, pp. 1–7.
- [11] ACM/SIGDA benchmarks (NCSU resource), “ISCAS Benchmark Circuits,” 2007, [Online Source].
- [12] C. Albrecht and Cadence Research Laboratories at Berkeley, “IWLS 2005 Benchmarks,” 2007, [Online Source].
- [13] Mentor Graphics, *YieldAnalyzer and YieldEnhancer Reference Manual*, 2010, Calibre DFM Suite Datasheet.
- [14] David A. Patterson and John L. Hennessy, *Computer Organization & Design, The Hardware/Software Interface*, Morgan Kaufman Publishers Inc., 2nd edition, 1998.
- [15] Cadence Design Systems, *RTL Compiler*, v. 10.1, 2011.
- [16] Cadence Design Systems, *Encounter[®] Digital Implementation System*, v. 10.1.2, 2011.
- [17] Cadence Design Systems, *Virtuoso[®] Layout Editor*, v. 5.1.41, 2008.
- [18] Cadence Design Systems, *Virtuoso[®] Schematic Editor*, v. 5.1.41, 2008.
- [19] Mentor Graphics, *Calibre[®] Verification*, v. 2009.1, 2009.
- [20] Synopsys, Inc., *StarRC[®]*, v. D-2010.06, 2010.
- [21] Cadence Design Systems, *Encounter[®] Library Characterizer*, v. 10.1.2, 2011.
- [22] Dean H Stamatis, *Failure Mode and Effect Analysis: FMEA from Theory to Execution*, chapter 11,12, ASQ Press, 2003.

- [23] J.P. Bickford, J.D. Hibbeler, D. Mueller, S. Peyer, and V.S. Kumar, "Optimizing Product Yield using Manufacturing Defect Weights," in *Proc. Adv. Semiconductor Mfg. Conf.*, May 2012, pp. 16–20.
- [24] C. Hess, B.E. Stine, L.H. Weiland, T. Mitchell, M.P. Karnett, and K. Gardner, "Passive Multiplexer Test Structure for Fast and Accurate Contact and Via Failure Evaluation," *IEEE Trans.on Semiconductor Manufacturing*, vol. 16, no. 2, pp. 259–265, May 2003.

Part IV

Summary & Conclusions

Get your facts first, and then you can distort 'em as much as you please.

~Rudyard Kipling, An Interview with Mark Twain

5

Summary & Conclusions

5.1 Summary

In this thesis, I have attempted to tackle the problem of cost-effective manufacturability of ICs using design techniques, more specifically from a physical design standpoint. Regularity as a means to mitigate variability is tested at various levels of abstraction.

Chapter 3 of this thesis introduced placement regularity of standard-cells and introduced a novel methodology to implement such regularity. The results from that study, applied to different types of column compression multipliers and shifters, showed that placement regularity can be leveraged to create extremely area efficient designs. The *ad hoc* application of regularity to the placement of standard-cells leads to congestion in the routing due to the simultaneous requirement of error free routing of a large number of cells combined with the heuristic algorithms used to achieve this. The demonstrated area advantages can be leveraged if the underlying causes for congestion are identified.

In Part II of this thesis I carry out a study of the transistor level layout regularity to identify the interactions between regularity at this level of abstraction and conventional

standard-cell design flows; in particular the impact on routing characteristics with emphasis on variability related issues. In order to achieve this I created a couple of custom standard-cell libraries with different degrees of regularity. The cell level study produced counter-intuitive results for DFM assessment using integrated DFM tools. The results from this study suggested that there are limited benefits to regularity at this level of abstraction. However, implementations of the ISCAS benchmark circuits using the custom created standard-cells, analyzed using the same integrated DFM tools and compared against raw implementation statistics, suggested potential reliability benefits without any significant overhead of area and minimal performance impact. Additionally, the results from this work brought out the need for a diverse library. With this in mind, I expanded each of the custom standard-cell libraries to include one hundred cells in each library, in drive strength flavors of X2 and X4.

I conclude my contributions with an IP-inclusive model to arrive at a DFM metric for SoCs called Model for IP-inclusive DFM Assessment of System manufacturability (MIDAS). This model builds on existing techniques to additively compute a metric of manufacturability for SoCs. The metric produced by MIDAS provides a measure of the risk that can be *avoided* as a result of following DFM considerations at various levels of abstraction. This model is demonstrated on an embedded processor system including a L1-cache sub-system. The processor is implemented using the expanded custom created standard-cell libraries, while commercial memory macros are used to implement the cache and tags. Initial results of the model applied to this design show that it is scalable and versatile.

5.2 Conclusion

With CMOS technology on the verge of breaching the 10 nm limit, DFM has assumed a great deal of importance. The markets are driven by a need for highly integrated, energy efficient functionality. In such a scenario, the issue of manufacturability is deeply tied to the profitability; indeed even the survival of companies.

Judging my own work taking these external factors into account, I can confidently conclude that DFM is here to stay. At the semi-custom design abstraction level, with the need for highly integrated functionalities, there is a need for compact, area efficient layouts. Such regular layouts, if intelligently implemented, could deliver extremely competitive performance at great area advantages. The methodology followed in this work is novel, and allows for highly area efficient layouts. That said, it must also be mentioned that mainstream EDA tools are beginning to offer robust solutions with the same goals.

Furthermore, whether by design or limitations, the way forward at the abstraction level of devices is driven by regular layouts. While this is mainly an artifact of EUV lithography being delayed, the benefits to yield make it almost mandatory in the latest technology nodes. The impact of using standard-cells with regular layouts is brought out by my work as well as work carried out by others. Based on existing literature, we can also conclude that co-optimization of design and manufacturing goals goes a long way towards reducing design and verification cycles. On the assessment side, we can conclude that as technology evolves, the need for better assessment techniques also arises with it. With DFM related expertise becoming the forte of the foundries, it becomes the responsibility of the foundries to define robust assessment techniques.

Finally, as DFM assumes greater importance, the need to assess and measure manufacturability also grows; the earlier in the design cycle, the better. We have presented one such early prediction model, christened Model for IP-inclusive DFM Assessment of System manufacturability (MIDAS). MIDAS computes an additive metric based on weighted costs for standard-cells, IPs and routing and can be used at the earliest stage when physical implementation data is available. Further, since costing for standard-cells and IPs is based on existing methods this computation is a one time effort when a fully automated computation scenario is used. The cost so computed can be made available to all designs through design kit infrastructure. The design specific routing solution statistics can then be used to determine a Figure-of-Merit(FoM) for the design.

This thesis began with the goal of studying manufacturability of ICs and I can conclude by saying that, if anything, the discipline of DFM is as important as ever. My contributions at the various levels of abstraction are but a sliver of the possibilities that abound in this area. My sincere hope is that this thesis has contributed a bit more to the understanding of the challenges involved in manufacturing electronic systems.

Appendix

There's always one more bug.

~Lubarsky's Law of Cybernetic Entomology

Hits & Misses

During the course of this thesis, a number of research directions did not reach the publication stage. This appendix briefly lists some of those efforts.

Exploiting Pin Position Aliased Standard-cells

During the investigation of congestion issues with multiplier routing using the Wired methodology, we briefly raised the possibility of employing cells with aliased pin positions. The reasoning behind this was that this would provide the heuristic routing algorithms wider choice of achieving the most efficient routing.

However, once the effort into developing the cells was begun, the research related to regular layouts assumed higher priority in the ideas that I chose to follow. Although, we did publish some of the ideas related to exploiting pin-positions at a non-peer reviewed, Sweden-centric conference, this idea was not pursued further.

As far as the details of this effort go, rectangular HPM PPRTs were implemented using custom created HAs and FAs. All the implementations were carried out using a commercial 90 nm CMOS technology. The results are summarized in table 1.

Table 1: *Results for HPM implementations*

Cell	Slack	Total WL	Avg. WL	# Vias
Normal	0.259 ns	230059 μm	57.3 μm	39784
Aliased	0.158 ns	208938 μm	52.0 μm	24352
Foundry	0.804 ns	165783 μm	41.3 μm	41110

As compared to the implementation using ST library cells, though the absolute lengths are greater, the number of vias used are fewer. This can be attributed to the fact that the custom-library implementations use a larger area on account of the fact that the constituent cells are not optimized for area.

TDM Multiplier Generator

While developing the Wired methodology, there was a need to generate different types of column compression multipliers. The HPM could easily be generated using the multiplier generator developed by Magnus Sjölander, and this reduced to the Dadda once the placement constraints were removed. However, we did not have access to a generator that could provide us with the TDM PPRT.

To this end, I developed a simple Tcl-based generator that could generate the PPRT for the TDM multiplier. The script accepts the number of bits as an input and produces VHDL for the PPRT, PPG and top-level portions of the multiplier. I felt that this would be the most efficient way since infrastructure already existed for the generation of the rest of the multiplier using the in-house HPM generator. It should also be noted that cell-delays are required for the TDM algorithm. This version of the TDM generator uses variables for the HA and FA delays normalized to the delay of a XOR gate. The current version of the generator works with data extracted for a 65 nm CMOS process. However, with the correct delay data, the generator will work for other technology nodes as well.

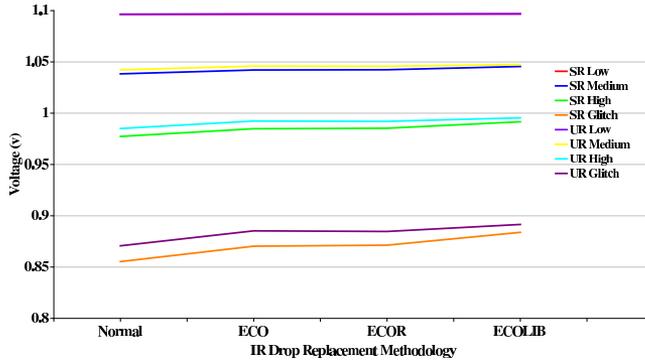
Standard-cell IR Drop Analysis

During the course of my licentiate defense, Prof. Rodrigues raised the issue of IR drop in the custom characterized cells. In order to test this, I ran static IR drop analysis on various implementations using the custom characterized standard-cells.

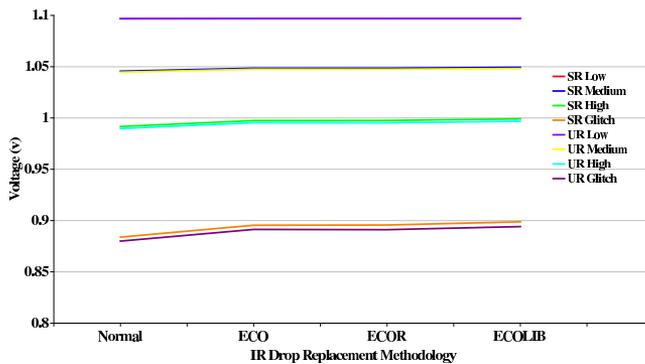
The methodology followed here is geared towards assessing the effect of the cell layout only. Thus, there are robust power distribution networks in place. The methodology has the following steps.

- Start with one kind of custom library (SR/UR). Implement the design so that synthesis achieves a slack of 750 ps. Run P & R and IR Drop analysis.
- Identify the type of cell to replace with the alternate custom cells (SR \rightarrow UR, UR \rightarrow SR). In the ISCAS benchmarks all the XOR cells are replaced, for the multipliers the FAs are replaced. ECO P & R followed by optimization is run to ensure a legal physical solution.
- Run a second ECO replacement to replace the originally replaced cell with the original cell (UR \rightarrow SR, SR \rightarrow UR). This, discounting routing noise would be expected to return the IR drop to nearly the same original value. Run IR drop analysis.

- Run an ECO flow to replace the cells with an equivalent foundry cell. Run IR Drop analysis again.



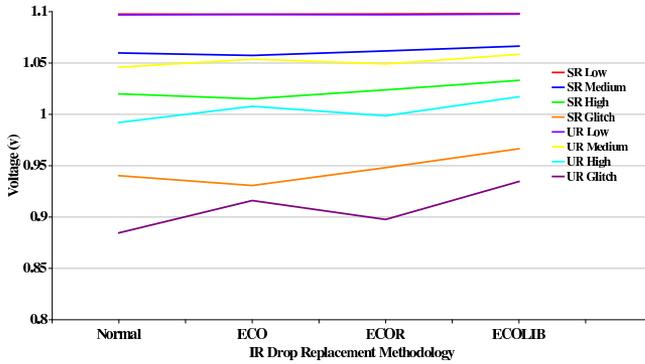
(a) s35932



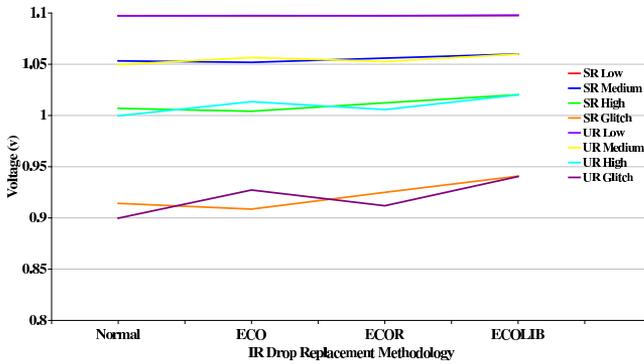
(b) s38417

Figure 1: IR Drop analysis for different ISCAS'89 Benchmark circuits.

The results are organized by type of custom library, corner and switching activity along the rows and physical implementation methodology along the columns. “Normal” refers to the original implementation, “ECO”: the first ECO step, “ECOR”: the second ECO step and “ECOLIB”: the ECO replacement with the foundry cell. In terms of switching activity, the different types tested were classified as “Low”(1% switching probability), “Medium” (20% switching probability), “High” (40% switching probability) and “Glitch” (80% switching probability). The last mechanism was implemented as a spurious means of stressing the simulation to beyond the design limits.



(a) TDM



(b) Overturned Stairs

Figure 2: IR Drop analysis for different multiplier circuits.

Figures 1 and 2 shows the worst values of IR Drop for the given conditions. The ISCAS benchmarks show an ambiguous trend with the IR Drop marginally improving/staying the same with the ECO replacements with the custom cells irrespective of type. The multipliers indicate a more consistent trend of the UR cells paying a penalty in terms of IR Drop.

Added to the fact that we did not have access to data from the foundry to run more advanced IR drop analyses, since the trends were not consistent (but acceptable all the same), this research direction was not pursued in any detail beyond this.

Automated Standard-cell Generation

Once the initial CFA analysis was carried out for the custom characterized cells, the results opened up the relevance of cell diversity. In order to completely assess the implications of cell diversity on manufacturability analysis, I needed libraries with diverse cells. Once I finalized a list of cells that would have to be created, it totalled to 100 cells per library with two library variants. Thus, I decided to script the template generation for the cells.

I created SKILL¹ scripts to enable this. I created the templates for the layouts only, using the schematics from the foundry provided libraries. The templates create the standard-cell bounding boxes, active and gate layers, and the power rails. All the geometries conform to the foundry rules so that the custom cells can be used alongside cells from the foundry library.

I completed the routing manually since the complexity to complete that though SKILL scripts was high and time-consuming. In spite of the fact that I did not pursue a full standard-cell generator, automating the creation of common cell layout elements resulted in the saving of a significant amount of time.

Standard-cell Library Migration Across Technology Nodes

In addition to the layout templates developed for the custom libraries, I also developed additional scripts to migrate libraries between technology nodes.

Initially developed to help in the migration of the custom layouts from the 65 nm node to the 45 nm, the scripts help migrate both the schematics and layouts. I wrote a combination of Tcl and SKILL scripts to accomplish this. SKILL scripts allows the transformation of layouts to an ASCII representation and also allows the transformation back into a graphical representation. The Tcl scripts are used to adjust the namespace from the old technology to the new node.

The latest design kits were rolled out in a short span of time and data required for DFM was missing from these design kits rendering much of this effort moot.

¹ SKILL is a Lisp-based language proprietary to Cadence that is extensively used in the Virtuoso design framework.

The technology community is, generally speaking, exceptionally acronym happy. For a light-hearted interpretation of some well known technology acronyms and jargon see [“The Register guide to acronyms”](#) and [“The quick guide to Register jargon”](#)