THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING

Manufacturability in the Nanometer Era: Regularity considerations in VLSI Circuits

KASYAB P. SUBRAMANIYAN

Division of Computer Engineering
Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2012

Manufacturability in the Nanometer Era: Regularity considerations in VLSI Circuits

Kasyab P. Subramaniyan

Technical report 95L
ISSN 1652-876X
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:
Collage showing design stages from concept to realization. The pictures of devices (source: Wikimedia Commons) show some of the possible uses of a realized product.

Chalmers Reproservice
Göteborg, Sweden 2012

Manufacturability in the Nanometer Era:Regularity considerations in VLSI Circuits

Kasyab P. Subramaniyan

Division of Computer Engineering, Chalmers University of Technology

ABSTRACT

Each reduction of the technology node has, along with improvements in IC fabrication technology, been the main driver in delivering the demand for function rich, integrated mobile electronics that are so prevalent. As devices keep growing smaller and geometries approach the order of a few atomic layers, it is increasingly difficult to achieve cost-effective mass production for reasons related to performance and fabrication capability. The small geometries have made visible quantum-mechanical effects that are not seen in micron scale geometries. These effects result in parametric variations and additionally, limitations in the lithographic capability means that cost effective fabrication is possible only at considerable investment. Adopting regularity in layout has been prescribed as a means to mitigate variability in small geometries. This measure however, is not widely adopted due to the lack of a structured methodology in implementing such layouts. This thesis aims to study the hurdles existing in implementing such a methodology, at different levels of abstraction.

Increasing design complexity has led to the widespread use of standard cell methodology to enable shorter design cycles. Typically, the place and route tools rely on heuristic algorithms that tradeoff run time against performance constraints. The first part of this thesis presents a novel methodology for regular layout of standard cells in a layout exploration scenario. The design flow is applied to arithmetic circuits like log-depth multipliers and shifters in order to assess various tradeoffs.

The second part of this thesis discusses regularity from a circuits perspective rather than a design perspective. The different factors affecting the implementation of a regular layout are discussed. In the latter half we discuss aspects of manufacturing, the sources of variability, assessment techniques and the impact of regularity on mitigating the negative aspects of technology scaling in the face of engineering limitations.
Finally, the studies will be summarized and the scope of future work will be presented.

Keywords: Regularity, Regular Fabrics, Bricks, CMOS, Transistor, SoC, Wired, Circuits, Arithmetic, Multipliers, Shifters, HPM, TDM, Barrel Shifter, ASIC, VLSI.

Preface

Parts, but far from all, of the contributions presented in this thesis have previously been accepted to conferences or submitted to journals.

- **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.
- 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.

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 ,"On Regularity and Integrated DFM Metrics" in *Proc. 4th Asia Symposium on Quality Electronic Design (ASQED 2012)*, Penang, Malaysia, Submitted for Publication.
- **Kasyab P. Subramaniyan** and Per Larsson-Edefors, "On Regular Standard Cell Implementations", in *Proc. Power And Timing Modeling, Optimization and Simulation (PATMOS 2012)*, Newcastle upon Tyne, United Kingdom, Submitted for Publication.

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 *Proceedings of IEEE Computer Society Annual Symposium on VLSI*, Chennai, India, July 4-6, 2011, 301-306.
- **Kasyab P. Subramaniyan**, Erik Ryman, Magnus Själander, Tung Than Hoang, Mafijul Islam and Per Larsson-Edefors, "FlexDEF: Development Framework for Processor Architecture Implementation and Evaluation", in *Proceedings of 7th Conference on Ph.D Research in Microelectronics and Electronics*, Madonna di Campiglio, Italy, July 3-7, 2011, 37-40.
- Tung Than Hoang, Ulf Jälmbrant, Erik der Hagopian, **Kasyab P. Subramaniyan**, Magnus Själander and Per Larsson-Edefors, "Design Space Exploration for an Embedded Processor with Flexible Datapath Interconnect", in *Proceedings of 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 Själander, Henrik Eriksson and Per Larsson-Edefors, "Custom Layout Strategy for Rectangle-Shaped Log-Depth Multiplier Reduction Tree", in *Proc. IEEE Int. Conf. Electronics, Circuits, and Systems*, Hammamet, Tunisia, December 13-16, 2009, 77-80.

Acknowledgments

This work was possible through help from a number of people who have my sincerest thanks.

- Professor Per Larsson-Edefors for his continuous guidance, encouragement and support for my work and my initiatives.
- Docent Lars Svensson for your support with all other aspects of my work. The parts related to my teaching hours I find especially stimulating.
- Lars Kollberg for your CAD and tool support. You will be missed.
- Rune Ljunbjörn and Peter Helander for the IT support. Thanks also to Rune now for the CAD support in recent days.
- Many thanks to the ProViking program and the Swedish Foundation for Strategic Research for their funding of this work.
- Erik Ryman for all the good times we have chatting in the shared workspace.
- Magnus Själander for being the guru of all things FlexCore and being a great motivator.
- Many thanks to the Tiina Rankanen and Eva Axelsson of the administrative staff.
- To all other colleagues in the VLSI and Computer Architecture group. Working alongside you all is a genuinely happy experience and a privilege.
- Cheers to Jan, Martin, Tilak and Wiktor for the good times we share over a beer or outside school.
- To all my other friends. I would like to thank you all even if we are not keeping in touch as you all had a part to play.

- To my family for their continuous support and encouragement. Your support makes me persevere in my endeavors.

Kasyab P. Subramaniyan
Göteborg, June 2012

Contents

List of Figures

List of Tables

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
EUV	Extreme Ultra Violet
FEOL	Front End Of the Line
FPGA	Field Programmable Gate Array
HIL	High Index Liquid
HPM	High Performance Multiplier
IC	integrated circuit
IDV	Integrated Design Verification
ILD	Inter Layer Dielectric
IP	Intellectual Property
LEF	Layout Exchange Format

LER	Line Edge Roughness
LSB	Least Significant Bit(/Byte)
LVS	Layout Versus Schematic
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
PSM	Phase Shift Masking
RDF	Random Dopant Fluctuation
RDR	Restricted Design Rule
RET	Resolution Enhancement Technique
RTL	Register Transfer Level
SIA	Semiconductor Industry Association
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

1 Introduction

1.1 Background

The Semiconductor Industry Association (SIA) published worldwide sales figures of close to $300 billion for 2011 [1]. This is the result of an insatiable demand for high performance electronics in a number of application areas. The largest market segments are the computing, mobile (including telecommunications) and consumer electronics markets. Advances in electronic design and manufacturing techniques have resulted in the development of compact, feature rich mobile devices. This in turn, has been enabled by the reduction in the device sizes resulting in greater density. This reduction, termed scaling, has continued unabated for the past four decades, with the density doubling roughly every two years(figure 1.1) [2].

The ability to create complex designs with billions of transistors is a testament to the advanced design, verification and manufacturing capabilities that have evolved with each successive reduction of the technology node. Traditional scaling is not however without its problems. As device geometries grow smaller, second-order effects begin to make their presence felt. This affects the performance of the transistors, most prominently in the form of leakage: i.e. the inability to completely turn off the transistor due

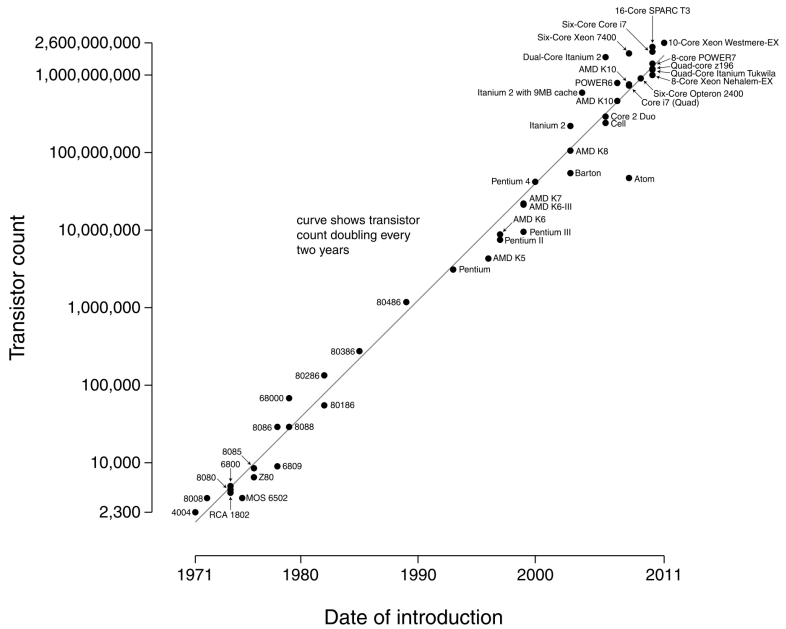


Figure 1.1: *Transistor counts for integrated circuits by year [2, 3].*

to insufficient control over the channel. This in turn impacts overall design performance. Since the 180 nm technology node, fabrication has been forced (for various reasons) to prolong the usage of sub-wavelength lithography. The remainder of this introductory chapter will provide insights into the various ways these factors affect the different aspects of designing and manufacturing a chip.

1.2 Design Flows

Scaling continues to play an extremely important role in the way electronics is designed and manufactured. Considering just the last decade, from a design perspective, the traditional divide of Analog vs. Digital and Full Custom vs. Semi Custom design still continue to hold primary 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.

Digital integrated circuit (IC) design, on the other hand, has shifted to semi-custom design techniques for all but the most high performance designs. 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. Compared to analog IC design however, application of design automation to large parts of both semi-custom and FPGA methodologies makes it easy to deal with the functional implementation aspects of a digital design. The extra effort related to dealing with scaling related issues, generally referred to as variability, is restricted to some steps of the semi-custom methodology. In the case of FPGAs the designer relies on better design and architecture techniques to ensure that the design being implemented has sufficient resilience to combat the effects of variability. These flows are shown in figure 1.2.

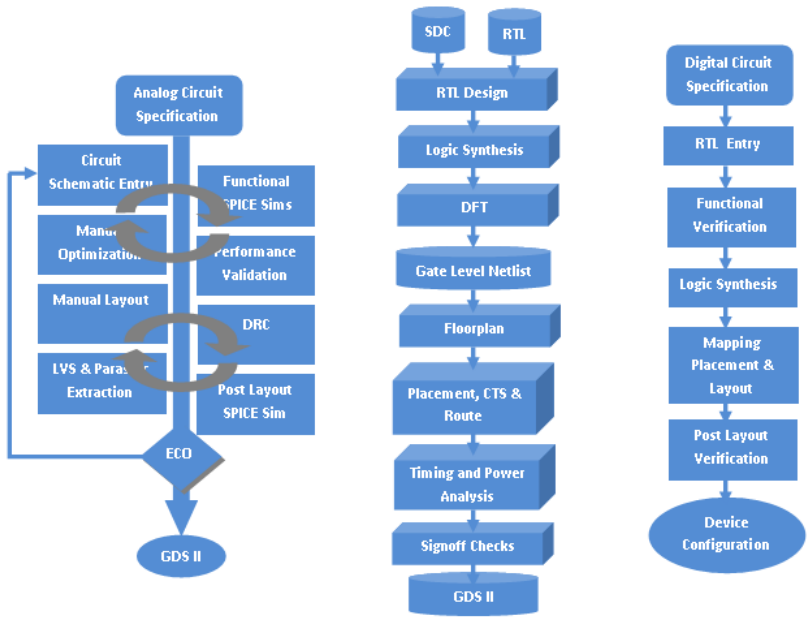


Figure 1.2: *Design Flows for Analog, Digital & FPGA methodologies.*

At this point it is useful to observe that, with continued scaling the effects of variability traditionally impacting analog circuits now have an impact on digital 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. With respect to the traditional design techniques the most obvious impact, is the number of design rules that must be fulfilled in order to be able to qualify a chip for fabrication. The number of Design Rule Checks (DRCs) that must be satisfied has grown from a few hundred rules in the micron- scale nodes to a few thousand rules 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 [4, 5]. 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 [4] 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 in an effort to obtain as many circuits performing as close to the performance envelope as possible in a given manufacturing lot. This is also referred to as Design for Yield (DfY), where the term yield refers to the percentage of chips in a given lot that fulfill the performance criteria. Another term used to describe the design flows incorporating efforts towards enhanced manufacturability is Design for Manufacturability (DfM).

1.3 Scaling and Manufacturability

The 45 nm process from Intel [6, 7] 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 kept Moore's Law [2] on track without compromising the benefits of scaling. In contrast to this, 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 pattering 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.

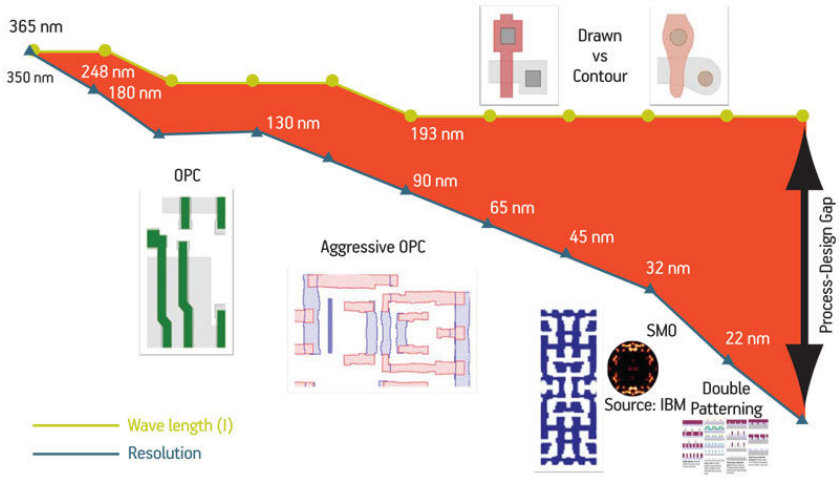
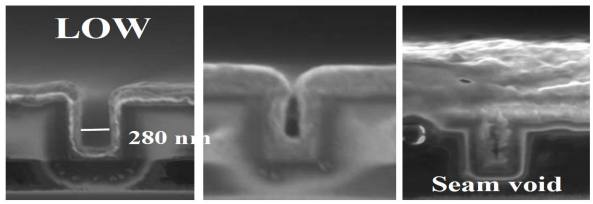


Figure 1.3: *Lithography source wavelengths against feature size [8].*

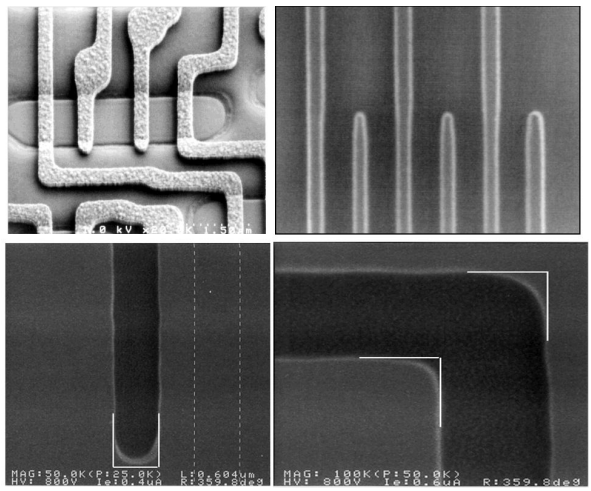
Mismatches between the intended pattern and the fabricated pattern primarily become visible due to insufficient lithographic accuracy for sub-wavelength patterning. Other process steps like Chemical Mechanical Polishing (CMP) and etching (used extensively in the creation of trenches and in the interconnect 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



(a) **CMP and Etch defects**



(b) **Lithography defects**

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

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 aggressive OPC, the $k1$ 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 $k1$ 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.

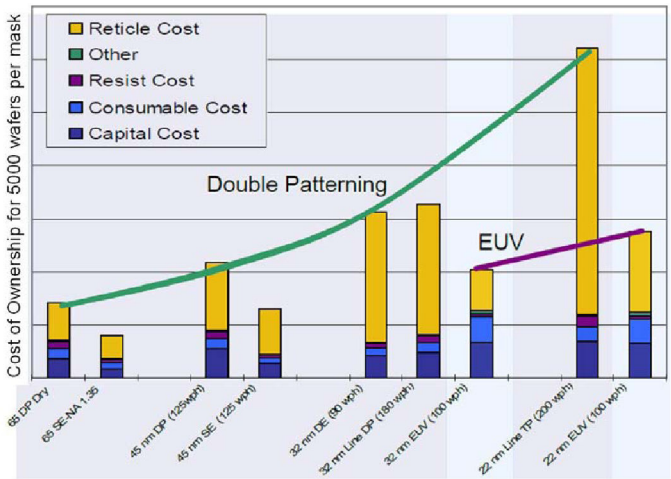


Figure 1.5: *Relative cost of ownership of a* 5000 *wafer run device [9].*

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 factors affecting widespread deployment. 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. Section 3.3 gives a much more in-depth treatment of variability and the techniques currently in use to estimate and mitigate it. This part of the thesis is simply a broad overview of the area.

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 manufacture 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, are there any viable means of designing variability resilient circuits?

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 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 we investigate a novel methodology incorporating regular placement of standard cells. 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 we do not actively investigate regularity related to routing. Analysis of the results from this study convinced us 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 resources[1] result in extreme device level layout regularity, there is no consensus on any quantification of regularity. When

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

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.

Due to the nature of the problem, 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 we will investigate regularity implemented on sample functional units performing arithmetic operations and the ISCAS benchmark circuits as and when applicable. The complete scope of this work is to establish a link between the abstract manufacturing considerations aimed at mitigating variability and the design aspects of implementing such considerations. The broader expectation is to be able to model the effect of implied constraints on manufacturability and yield.

Bibliography

[1] Kazmierski, C., "Semiconductor Industry Posts Record-Breaking Revenues Despite 2011 Challenges," SIA 2011 Revenue report. [Online].

[2] G. E. Moore, "Cramming more Components onto Integrated Circuits," *Electronics*, vol. 38, no. 8, pp. 114–117, Apr. 1965.

[3] Wgsimon, "Transistor counts by year and Moore's Law," Moore's Law and transistor counts. [Online].

[4] M. Orshansky, S. R. Nassif, and D. Boning, *Design for Manufacturability and Statistical Design*, Springer Science+Business Media, LLC, 2008.

[5] C. C. Chiang and J. Kawa, *Design for Manufacturability and Yield for Nano-Scale CMOS*, Springer Science+Business Media, LLC, 2007.

[6] Ghani, T., "Challenges and Innovations in nano-CMOS Transistor Scaling," Intel's presentation on the 45 nm technology node. [Online].

[7] C. Webb, "Intel design for manufacturing and evolution of design rules," 2008, vol. 6925, p. 692503, SPIE.

[8] Kai Peter and Reinhard März, "Quantifying returns on litho-friendly design," Illustration of Process Design Gap. [Online].

[9] International Technology Roadmap for Semiconductors 2009 Edition, ," ITRS Process Integration Devices and Structures report. [Online].

Part I

Placement Regularity In Semi-custom Flows

2

Regularity & 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]. Ienne *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 Haskell. 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. We will present the basic aspects of Wired in relation to these domains.

2.2.1 Logic domain

The logical aspect of a design is described in a simple applicative style. 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`. 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 functions. One commonly used function which is both recursive and higher-order, is `mapM`. Another useful, non-recursive, symbolic combinator is >=>, read as “composition”.

2.2.2 Placement

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

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

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 [5].

2.3 Related Work

Considering the abstraction level of the work discussed in this chapter, we keep the discussion of related work here restricted to methodologies that provide layout aware controls. In the larger scheme of things considerations towards 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 [6] 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 structure 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) [7] 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 parameterization, 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 parameterization 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 parameterized 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.

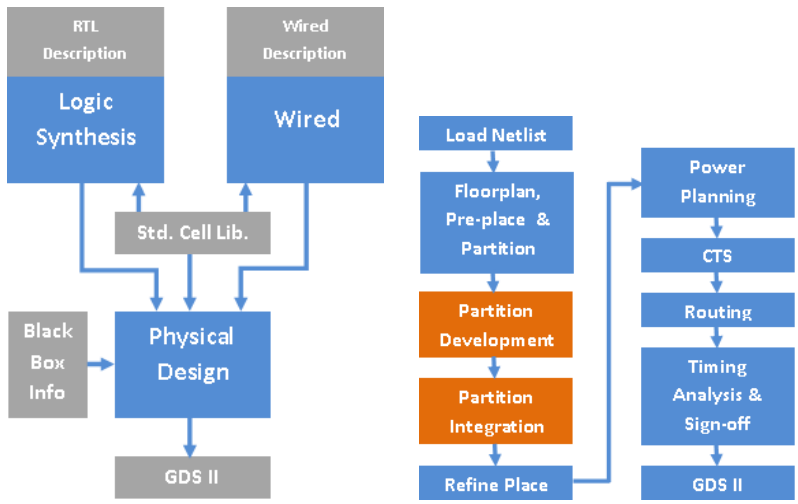
Design Complier, a commercial synthesis tool available from Synopsys employs special algorithms to extract datapath circuits from RTL descriptions [8]. Support for context driven multipliers, adders, shifters and selectors is available with extensions for special operations such as squaring and blending. 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.* [9] 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 [8] or the work by Dhumane *et al.* [9] 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 [10] 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 discussion this topic falls outside the scope of the present discussion.

2.4 Methodology

One of the 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 we will use Wired 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 in-

tegrated. The complete scope of the methodology is shown in figure 2.2(a). 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 2.2(a) 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.



(a) **Complete Flow** (b) **Physical design flow**

Figure 2.2: *Methodology to enforce placement regularity using Wired.*

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 modules developed in VHDL. The specific integration steps are shown in figure 2.2(b). 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.

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 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 [11] 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 [12] is similar to the Wallace multiplier [11], 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 [13]: $X_0 = 2, X_{i+1} = \lfloor \frac{3}{2} X_i \rfloor$.

HPM Multiplier

The High Performance Multiplier [13] 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.* [14], 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.* [15] 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 under certain cases, but matches the delay displayed by multipliers created using the original.

2.5.2 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 [16] 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 stages which reverse the input data set when it is necessary. This shifter has 8-bit input and capable of 7-bits shift operation. The shift inputs are encoded. 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.

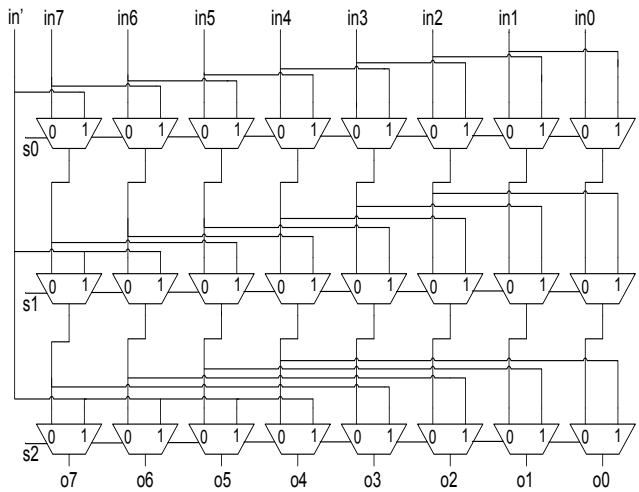
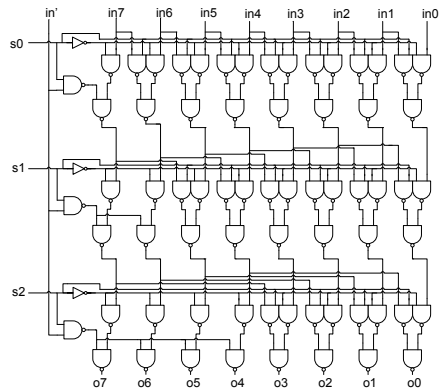
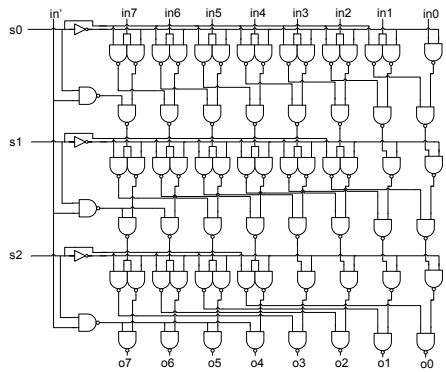


Figure 2.3: *Barrel shifter structure.*

This shifter can be built using different standard cells [17]. The advantage of using multiplexer cells is that the layout area is small, but faster shifters can be generated using basic logic standard cells like NAND gates. The circuit in Fig. 2.4(a) 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) Shifter based on NAND gates



(b) Shifter based on NAND gates and fan-out splitting

Figure 2.4: *Barrel shifter structures using NAND gates.*

A layout technique called fan-out splitting has been proposed for cyclic shifters [17]. 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 [18]. 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 [17]. The circuit in figure 2.4(b) 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.

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 node[1]. 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. We 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 [16]:

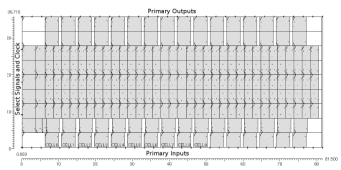
- The placement adopted in Wired was preserved entirely and the floorplan area was reduced to an extent that ensured both routability and error-free placement. Only the routing engine of Encounter was employed to complete the routing of the design. The results of such a flow are placed under the *'Wired'* 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

[1] The specific implementation technology will be provided in each case.

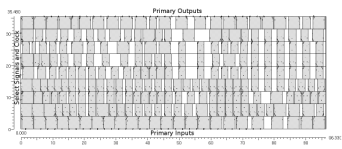
tool. The results of such a flow are placed under the *'Tool Driven'* column in the tables below.

- 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.

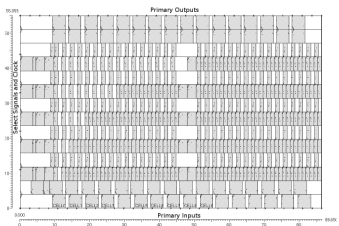
A pictorial example of each implementation is shown in figure 2.5. In our exploration of log-depth multipliers using Wired [19], later results suggested that tightly packed cells cause some routing congestion that can be alleviated by providing *"routing channels"*. From this experience, we estimated 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.



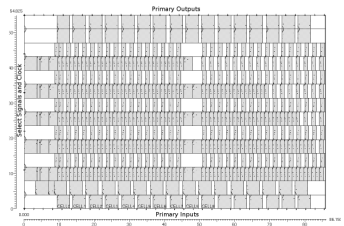
(a) **Wired-based placement; MUX based.**



(b) **Tool driven placement; MUX based.**



(c) **Wired-based placement; NAND based.**



(d) **Wired-based placement; NAND based with fan-out-splitting.**

Figure 2.5: *32-bit shifters placed in Encounter.*

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.

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

	Slack (ps)			Core Area (mm^2)		
Type	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

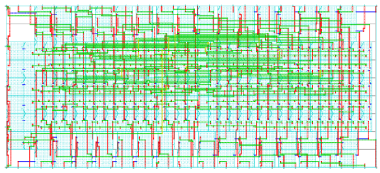
Table 2.2: *Comparison of 64-bit barrel shifters in* 90 *nm CMOS.*

	Slack (ps)			Core Area (mm^2)		
Type	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

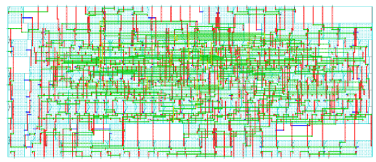
Patterns can be seen, both with respect to the various types of shifters, as well as the different 2 strategies employed. The performance expectations of the different types are confirmed, with highly area-efficient multiplexer-based implementations and faster NAND-based implementations. Table 2.2 shows 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 predictable 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.

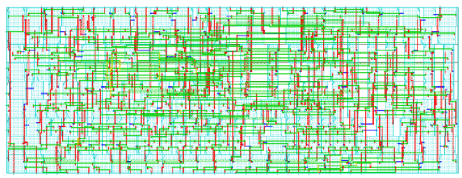
Some simple observations can be made about the routing resulting from the exploration presented here. Figures 2.6(a), 2.6(b), and 2.6(c) show the routing that resulted for a 32-bit multiplexer-based shifter. A visual inspection shows that for an inherently regular circuit such as this one, routing becomes more regular when placement is enforced to be regular. Figure 2.7(a) shows the wire length distribution across the different metal layers for a 32-bit multiplexer-based shifter for each of the placement schemes used in the study. The tool-driven implementation (figure 2.6(c)) is the least constrained and, as 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 (figure 2.6(b)), the routing engine produces a design that is design-rule compliant with some reduction in total wire length. The Wired-based placement (figure 2.6(a)) takes this reasoning one step further causing the router to produce a design-rule compliant design with the least



(a) **Wired-based placement and tool-driven routing.**

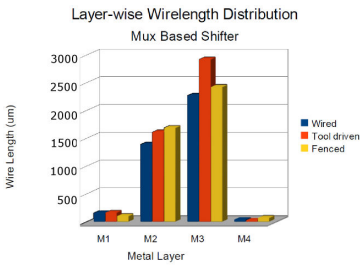


(b) **Fenced placement and tool driven routing.**

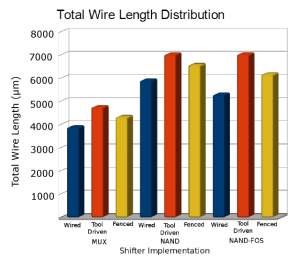


(c) **Tool driven placement and routing.**

Figure 2.6: *32-bit multiplexer-based shifters.*



(a) **Layer-wise wire-length distribution.**



(b) **Total wire-length distribution.**

Figure 2.7: *Metal usage for implemented 32-bit multiplexer-based shifters.*

resources. Figure 2.7(b) 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 2.5.1 lends itself to the Wired flow due to the inherent regularity. This multiplier is the primary test case for the methology. The results from an implementation using the HPM multiplier and the methodology presented in

section 2.4 are compared against implementations of the Dadda[2] and TDM multipliers.

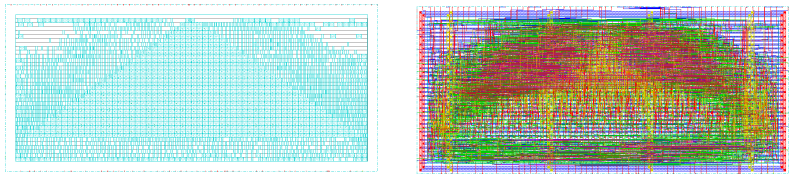
The PPG and the PPRT for the HPM multiplier were created using Wired. While no strict placement constraints were placed on the PPG, this part was generated for the sake of completeness. Further, overall area considerations were not taken into account while creating the PPG in Wired. This meant that the 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 2.8(a)). 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.

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.*

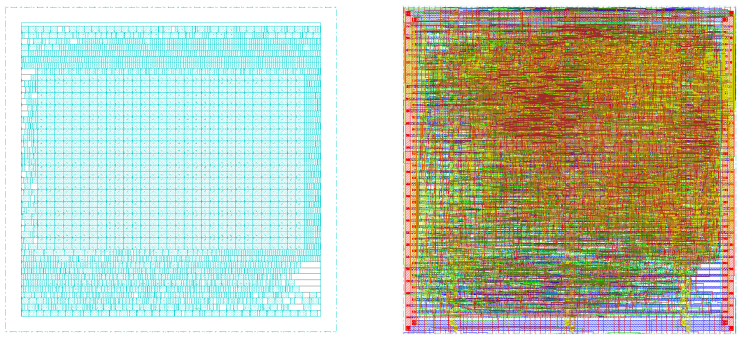
However, since non-rectilinear geometries are difficult to include in an implementation with any degree of efficiency in overall area (Table 2.3). Consequently, rectangular PPRTs were generated using transformations in the Wired environment. This had the effect of significantly improving the overall area utilization but resulted in severe congestion in the initial stages of the PPRT where a large number of partial products are processed. This is a qualitative visual inference and comparing figures 2.8(b) and 2.9(b) 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.

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



(a) **Placed multiplier** (b) **Routed multiplier**

Figure 2.8: *A HPM Multiplier with a triangular PPRT.*



(a) **Placed multiplier** (b) **Routed multiplier**

Figure 2.9: *A HPM Multiplier with a rectangular PPRT.*

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 focussed the effort on studying the factors 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

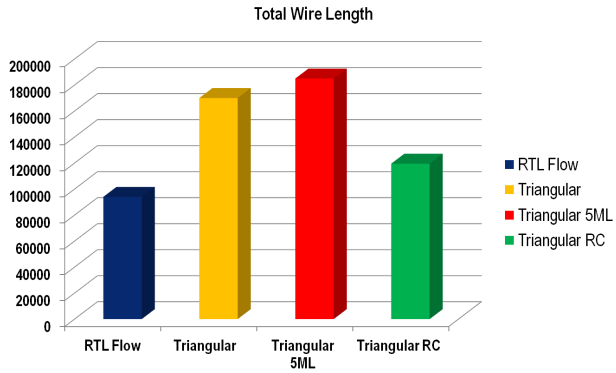


Figure 2.10: *Total Wire length for different multiplier implementations.*

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 congestion) 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.

This work opened up a few questions however. 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, we implemented our 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. Ienne 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] Axelsson, E., *Functional Programming Enabling Flexible Hardware Design at Low Levels of Abstraction*, Ph.d. thesis, Chalmers University of Technology, 2008.

[6] Tuscany Design Automation, ," Tuscany Design Automation Homepage.

[7] C. Seger, "Integrating Design and Verification - from Simple Idea to Practical System," in *Fourth ACM and IEEE Intl Conf. on Formal Methods and Models for Co-Design (MEMOCODE)*, 2006.

[8] R. Zimmermann, "Datapath synthesis for standard-cell design," in *Computer Arithmetic, 2009. ARITH 2009. 19th IEEE Symposium on*, June 2009, pp. 207 – 211.

[9] N. Dhumane, S.K. Srivathsa, and S. Kundu, "Lithography constrained placement and post-placement layout optimization for manufacturability," in *VLSI (ISVLSI), 2011 IEEE Computer Society Annual Symposium on*, July 2011, pp. 200 –205.

[10] A. Chakraborty, A. Kumar, and D.Z. Pan, "Regplace: A high quality open-source placement framework for structured asics," in *Design Automation Conference, 2009. DAC '09. 46th ACM/IEEE*, July 2009, pp. 442 –447.

[11] C. S. Wallace, "A Suggestion for a Fast Multiplier," *IEEE Transactions on Electronic Computers*, vol. 13, pp. 14–17, Feb. 1964.

[12] L. Dadda, "Some Schemes for Parallel Multipliers," *Alta Frequenza*, vol. 34, no. 5, pp. 349–356, May 1965.

[13] H. Eriksson, *Efficient Implementation and Analysis of CMOS Arithmetic Circuits*, Ph.d. thesis, Chalmers University of Technology, 2003.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] S. Huntzicker, M. Dayringer, J. Soprano, A. Weerasignhe, D. Harris, and D. Patil, "Energy-Delay Tradeoffs in 32-bit Static Shifter Designs," in *IEEE International Conference on Computer Design*, Oct. 2008.

[19] 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 II

Regularity considerations for Standard Cells

3

Regularity in Standard Cells

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 geometries 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.

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 length[1]. They explored the effect of regular gate-forming structures and single orientation and their results, supported using lithography simulations, further underline

[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.

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 standard 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 Metal1 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 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.

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 productions

and it's 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 asMiddle 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 layer[2] 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

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

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.

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} an 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 seen 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:

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

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

Design time analysis of manufacturability is now being employed by 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 Critical Feature Analysis (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 presented.

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, assigned by the foundry based on experience. 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 Implementation

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.

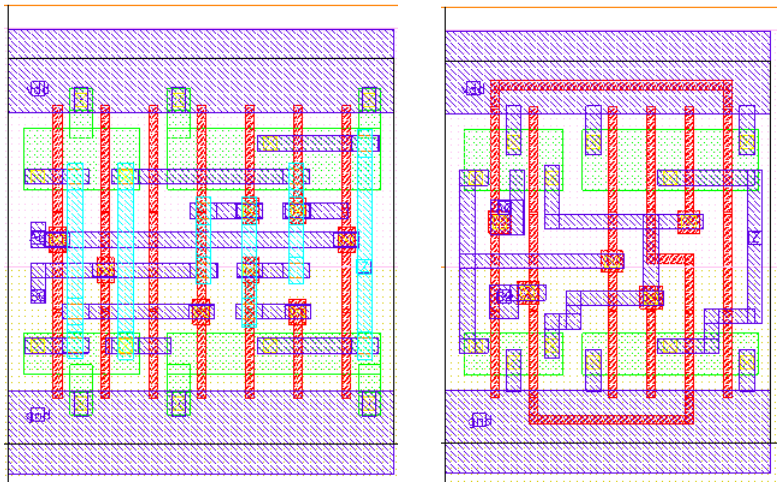
With scaling, a number of restrictions have been introduced by the foundries in order to maintain yield margins. Going back to the original intent of this work introduced in section 1.4, this work concentrates on standard cells incorporating different degrees of regularity.

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



(a) **Ultra-regular** (b) **Semi-regular**

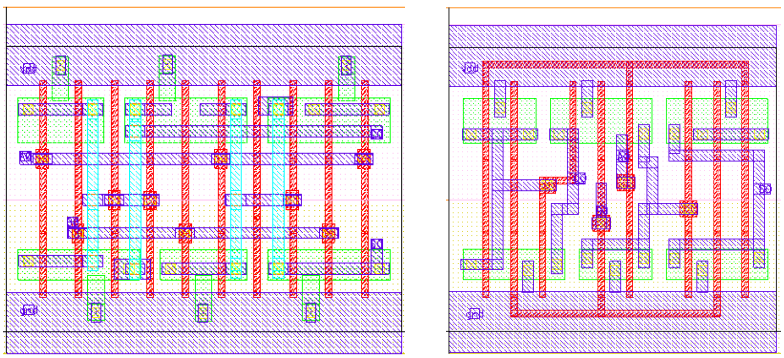
Figure 3.1: *Custom characterized XOR Gates.*

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

Table 3.1: *Custom characterized cells in* 65 *nm CMOS.*

	Width (μm)		
Cell Functionality	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

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 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 custom 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.



(a) **Ultra-regular** (b) **Semi-regular**

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. Parasitic extraction was performed using the StarRCXT tool from Synopsys. The qualified layouts were then used to extract standard cell LEF files. Timing models in the *.lib* format were extracted after characterization of the cells using Encounter Library Characterizer. The cells are characterized for low power at standard V_{th} and a nominal operating voltage of 1.2 V.

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.

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. Doubling contacts for the sake of reliability can however 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 Metal1 have been used. Choosing Metal1 perpendicular to poly makes for better local routing but decreases the availability of redundancy. Routing Metal1 parallel to poly is an alternate solution, eliminating the redundancy problem at the cost of diffusion width and additionally, increased Metal2 usage. With these considerations in mind, we chose to implement cells with Metal1 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 Metal1 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 Metal1 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 Metal1 perpendicular to Poly style is chosen. In spite of the risk of higher RC losses, in this work, we 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

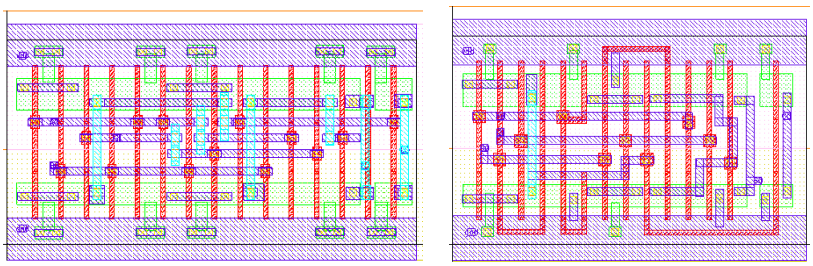
been observed that AOI structures lend themselves more easily to regular layouts than other types of static gates(like transmission gates etc.) [14].

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 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.



(a) **Ultra-regular** (b) **Semi-regular**

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 P-MOSFET is still followed here. The P-MOSFETs are one and a half times wider than the N-MOSFETs.

- 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 Metal2 has to be used to complete the local interconnect within the cell. Keeping the preferred directions, Poly is directed vertically, Metal1 horizontally and Metal2 vertically.
- For the semi-regular layouts, with the exception of the full adder, all layouts use only Metal1 to complete internal routing[3]. 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 will probably become mandatory to employ isolated poly lines at cell edges.

3.6 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; the second is a study of the implementation metrics of the cells applied to the ISCAS benchmarks in an industrial standard cell based flow.

3.6.1 Cell Manufacturability Analysis

Only a subset of 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. We make

[3] It is possible to complete that net without the use of Metal2 but it would result in obstructions. This is another tradeoff not considered explicitly in this work.

this choice constraining the architecture based on existing knowledge related to the performance characteristics of other layout architectures [21, 22] 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. Under the layout constraints of single device orientation, the routing of the inputs is qualitatively explored in this work. Given the classification of layouts in section 3.4.1, in each case we qualitatively assessed the quality of routing using the number of gate contacts as a raw measure of input routing efficiency.

The HA and FA 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.

Table 3.2: *CFA Results for Ultra-regular and Semi-regular cells.*

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

Table 3.2 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 normalizor. 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 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

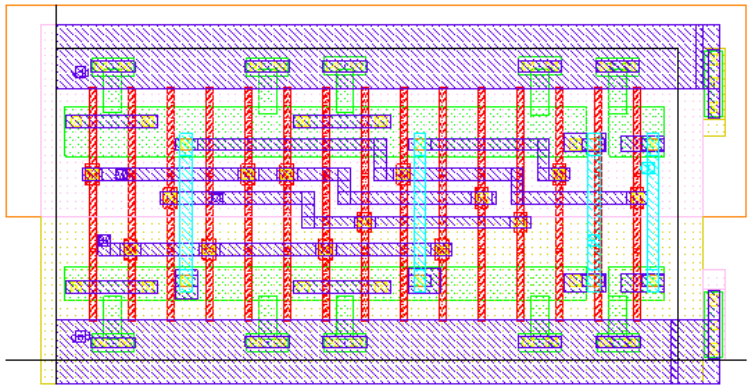


Figure 3.4: *A full adder cell regular in Poly pitch and direction.*

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 normalizor 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 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.

3.6.2 ISCAS Benchmark Circuits

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 [23]. Since the semi-regular library is richer in terms of drive strength diversity of cells at this point, 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 consists of extra cells in the semi-regular library and is used to assess the implications of drive strength diversity. Table 3.3 shows the comparison of the

Table 3.3: *Core Area, Wire Length and Slack for ISCAS benchmarks.*

ISCAS BM	Core Area (μm^2)			Wire length (μm)			Slack(ns)		
	SR	SRX4	UR	SR	SRX4	UR	SR	SRX4	UR
s1196	2307.24	2327.35	2351.21	4297.88	4377.32	4415.43	1.16	1.66	1.11
s1238	2324.03	2315.19	2356.96	4281.24	4119.65	4069.14	0.96	1.14	0.97
s13207	6543.26	6527.97	6634.62	6438.14	6394.34	6554.98	1.59	1.97	1.80
s1423	3224.00	3244.02	3388.84	3927.55	3852.58	4523.64	0.12	0.10	0.07
s1488	2404.92	2384.81	2400.06	5315.24	5358.93	5128.93	0.75	0.78	0.94
s1494	2366.46	2388.78	2396.96	5249.49	5418.99	5040.70	0.98	1.14	1.11
s15850	3669.66	3664.20	3729.72	3461.65	3821.89	3595.89	1.74	2.00	1.82
s208_1	629.92	629.92	667.13	257.67	289.78	264.55	2.45	2.50	2.51
s27	167.90	167.90	170.87	84.84	97.39	91.60	3.23	3.28	3.24
s298	1132.13	1118.83	1144.00	904.91	841.89	1087.63	2.26	2.42	2.26
s344	1243.63	1246.59	1267.34	1041.34	902.60	927.11	2.03	2.24	2.11
s349	1243.63	1246.59	1267.34	957.02	1020.22	952.57	2.07	2.22	2.19
s35932	52293.03	52231.95	55156.32	144951.23	120397.32	149571.70	-0.06	0.15	-0.10
s382	1294.80	1314.30	1333.80	909.68	939.21	999.12	2.19	2.43	2.28
s38417	57004.58	57002.40	57822.44	143171.84	121696.46	149622.25	-0.32	-0.59	-0.42
s38584	50889.90	50869.36	51628.72	127515.76	106618.21	144453.64	-0.21	0.13	-0.00
s386	929.50	932.10	934.70	899.48	952.77	908.49	2.35	2.60	2.44
s400	1138.28	1172.96	1167.19	819.06	952.65	947.71	2.19	2.41	2.29
s420_1	977.69	977.69	1036.60	566.39	517.20	620.78	1.39	1.57	1.47
s444	954.98	958.75	984.26	874.74	921.52	1047.78	2.40	2.54	2.37
s510	1040.03	1050.47	1044.04	1885.30	1887.77	2036.40	1.82	2.20	2.06
s526n	953.68	956.02	968.11	1416.15	1410.30	1223.51	1.82	2.09	2.11
s526	953.68	958.49	972.11	1305.37	1180.69	1320.43	1.83	2.04	2.09
s5378	6928.12	6887.41	7123.41	11370.80	11086.37	10382.52	1.44	1.63	1.40
s641	816.01	813.15	831.09	942.76	965.74	908.65	1.82	2.17	1.93
s713	825.37	826.41	833.04	932.81	906.77	1006.96	1.95	2.32	2.12
s820	1225.69	1184.82	1248.78	2432.01	2220.39	2251.22	1.81	2.07	2.04
s832	1230.37	1194.49	1204.00	2294.20	2207.22	2201.87	1.89	2.11	1.91
s838_1	1417.57	1362.47	1509.67	1498.50	1428.15	1328.23	0.16	0.00	0.26
s9234_1	5358.60	5356.00	5453.63	6934.24	7079.57	6631.84	0.97	1.59	1.38

"global" metrics, namely core area, wire length and slack for the different implementations. All the implementations have been implemented using the same density and aspect ratio constraints resulting in little variation of the core area. The frequency constraint for all designs in the benchmark suite was set to 250 MHz. The designs in the suite range from a few gates to a few thousand gates(figure 3.5). The implementation related statistics: the gate density, the number of cells and the number of vias, are shown in table 3.4.

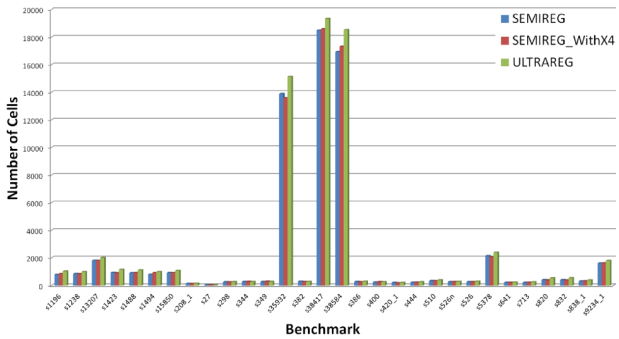


Figure 3.5: *Number of cells in the ISCAS benchmark circuits after physical design.*

The gate density does not show much variation across the implemented variants due to the common constraints applied (see figure 3.6(a)). The slack (figure 3.6(b)) on the other hand shows wide variation depending on the size of the design and the availability of diverse cell functions. Looking a little more closely, it can be seen from figure 3.7 that the slack depends on cell diversity more and more as the size of the design grows. Additionally, for the *s38417* benchmark all variants show a negative slack. This is the largest design in the benchmark suite and the lack of cell diversity affects the performance drastically. A similar effect is seen for the other large designs as well, where the lack of cell and buffer diversity impact the optimization steps of the flow negatively. The use of heuristics during routing 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. The vias making up the BEOL stack have the highest reliability concerns and incorporating regularity at the lower levels of abstraction shows clear benefits with the ultra- regular implementations using the lowest number of vias as is evident from figure 3.8. This can be viewed as a benefit even though it results in longer wire lengths for the ultra-regular implementations, since vias contribute to absolute failures as well as parametric variations. Wire width and thickness variations are 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.

With the results from the two avenues of analysis adopted for this study, it can be seen that any consideration of regularity has to be adopted bottom up. Stand-alone considerations of regularity at different levels of abstraction are clear cut only in the point of origin of the source of variability. Establishing any form of “global” estimate will need to link the different levels of design granularity while keeping the benefits of abstraction intact.

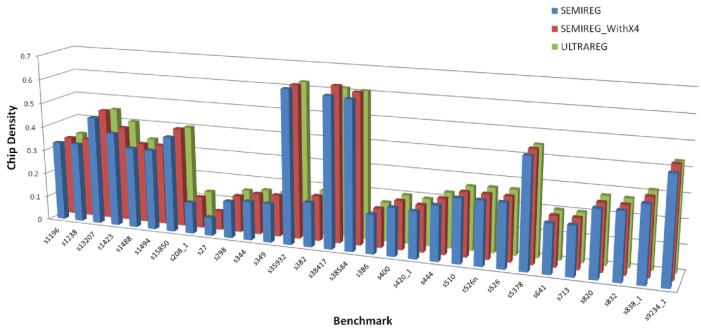
Table 3.4: *Chip Density, Number of Cells and Number of Vias for ISCAS benchmarks.*

ISCAS BM	Chip Density			Number of Cells			Number of Vias		
	SR	SRX4	UR	SR	SRX4	UR	SR	SRX4	UR
s1196	0.33061	0.33233	0.33218	799	855	1032	2087	2083	1730
s1238	0.33169	0.33305	0.33259	859	848	997	2075	2081	1685
s13207	0.45011	0.46272	0.45075	1816	1805	2011	3785	3962	3106
s1423	0.39097	0.39643	0.40392	941	904	1169	2277	2200	1903
s1488	0.33539	0.33482	0.33509	913	925	1114	2414	2422	1817
s1494	0.33411	0.33501	0.33466	814	921	998	2418	2434	1937
s15850	0.39686	0.41224	0.40025	924	920	1085	2015	2168	1656
s208_1	0.12947	0.12947	0.12992	137	126	148	150	169	133
s27	0.07608	0.07844	0.07618	37	39	32	37	43	39
s298	0.15170	0.15110	0.15211	261	253	262	392	422	385
s344	0.15996	0.16876	0.16136	276	290	269	508	447	333
s349	0.15996	0.17130	0.16136	276	305	283	433	470	364
s35932	0.63148	0.63080	0.62691	13894	13596	15131	37827	35520	34192
s382	0.18285	0.18516	0.18420	301	274	282	485	479	380
s38417	0.61695	0.64004	0.61496	18498	18587	19346	47799	46113	41275
s38584	0.61037	0.61979	0.60875	16943	17320	18522	42881	40878	38127
s386	0.16065	0.16048	0.16089	278	261	299	452	491	305
s400	0.19735	0.20066	0.19977	247	265	258	478	499	362
s420_1	0.19228	0.19228	0.19350	217	189	210	285	291	267
s444	0.22449	0.22748	0.22810	233	238	260	520	553	427
s510	0.26142	0.26203	0.26144	344	343	392	982	989	782
s526n	0.26171	0.26231	0.26511	266	272	286	771	772	555
s526	0.26171	0.26198	0.26610	267	266	302	720	674	614
s5378	0.44905	0.45436	0.45047	2153	2074	2397	4766	4740	3907
s641	0.20036	0.20409	0.20152	224	223	239	469	486	376
s713	0.20099	0.20448	0.20118	227	232	252	469	505	416
s820	0.27442	0.27299	0.27567	405	393	537	1154	1121	890
s832	0.27453	0.27321	0.27349	408	387	544	1164	1162	926
s838_1	0.30945	0.31253	0.31323	324	331	388	834	773	630
s9234_1	0.43026	0.44251	0.43160	1616	1627	1807	3577	3772	2731

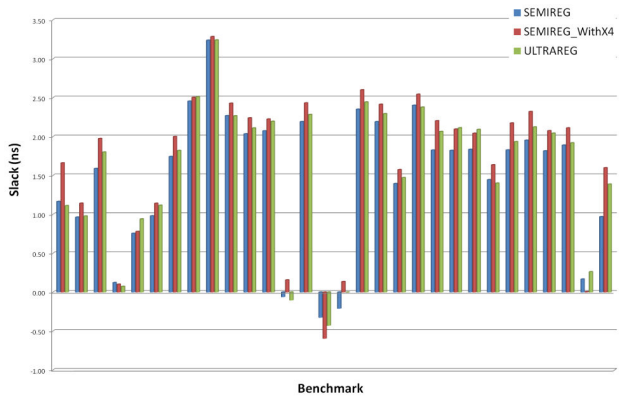
3.7 Conclusions

In this part of the study we have focused on regularity implemented at the layout level and applied to designs at a higher level of abstraction. The results provided by CFA are counter-intuitive; they suggest that regularity at the layout level is not extremely beneficial. However, implementations using such regular layouts seem to indicate otherwise. With the "soft" constraints of limited cell and buffer availability and different degrees of regularity:

- It is possible to implement designs with the same overall area at a performance penalty of reduced slack time.



(a) **Chip Density**

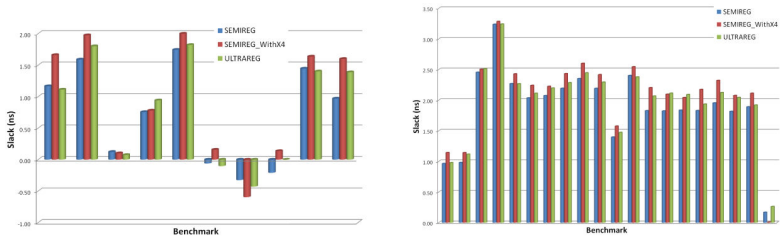


(b) **Slack**

Figure 3.6: *Chip density and slack plots for ISCAS benchmarks.*

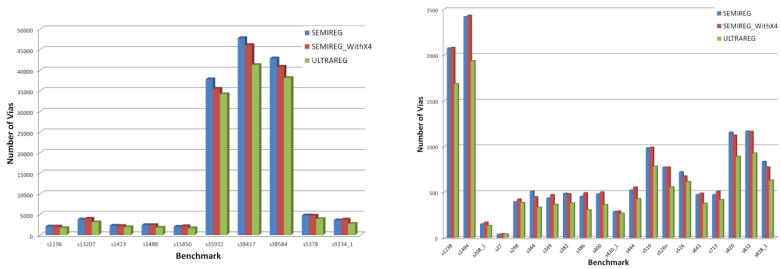
- Cell and buffer diversity is an important aspect to achieving closure on stringent timing constraints using the conventional flow.

The growing impact of manufacturing constraints on cost, performance and design/verification effort suggest that regular layouts adopted as a design requirement would serve well in reducing costs. However, to do so using conventional flows involves analysis and considerations of an ever increasing number of variables. Thus, the need for a holistic regular methodology would go a long way towards easing the burden of fabricating cost-effective electronics.



(a) **Over 1K Cells** (b) **Under 1K Cells**

Figure 3.7: *Slack plots for ISCAS benchmarks split by number of cells.*



(a) **Over 1K Cells** (b) **Under 1K Cells**

Figure 3.8: *Number of vias in the ISCAS benchmarks split by number of cells.*

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 *Computer Aided Design, 2004. ICCAD-2004. IEEE/ACM International Conference on*, 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] Muta, H. AND Onodera, H., "Manufacturability-Aware Design of Standard Cells," in *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, Dec. 2007, vol. E90-A, pp. 2682–2690.

[6] H. Sunagawa, H. Terada, A. Tsuchiya, K. Kobayashi, and H. Onodera, "Erect of regularity-enhanced layout on printability and circuit performance of standard cells," in *Quality of Electronic Design, 2009. ISQED 2009. Quality 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 *VLSI System on Chip Conference (VLSI-SoC), 2010 18th IEEE/IFIP*, 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 *Design Automation Conference, 2007. DAC '07. 44th ACM/IEEE*, 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," *Proceedings of the IEEE*, vol. 52, no. 12, pp. 1537 – 1545, dec. 1964.

[18] R.B. Seeds, "Yield and cost analysis of bipolar lsi," in *Electron Devices Meeting, 1967 International*, 1967, vol. 13, p. 12.

[19] J.E. Price, "A new look at yield of integrated circuits," *Proceedings of the IEEE*, vol. 58, no. 8, pp. 1290 – 1291, aug. 1970.

[20] A.G.F. Dingwall, "High-yield-processed bipolar lsi arrays," *Electron Devices, IEEE Transactions on*, vol. 16, no. 2, pp. 246 – 247, feb 1969.

[21] 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.

[22] N. H. E. Weste and D. Harris, *CMOS VLSI Design: A Circuits And Systems Perspective*, chapter 6, Addison-Wesley Publishing Company, 2007.

[23] Albrecht, C., ," IWLS 2005 Benchmarks Collected Benchmarks homepage. [Online].

Part III

Summary & Future Work

4

Summary & Future Work

The preceding parts of this thesis introduced regularity at different levels of abstraction to assess the impact it has on mitigating variability against the tradeoffs made in implementing regularity. Part I introduced regularity related to the placement 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.

This led to Part II of this thesis: A study of the transistor level layout regularity to identify the interaction regularity at this level of abstraction has, with conventional standard cell design flows; in particular the impact on routing characteristics with emphasis on variability related issues. The cell level study produced counter-intuitive results for DfM assessment using integrated DfM tools. The results from this study suggest that there are limited benefits to regularity at this level of abstraction. However, implementations of the ISCAS benchmark circuits using cells with varying degrees of regularity

suggest potential reliability benefits without any overhead of area and minimal performance impact.

Further research in the area would broadly progress in the direction of establishing quantifiable metrics of regularity linking the implied constraints imposed by layout regularity with the manufacturability aspects dominated by photolithography and CMP issues. Any such effort would involve the ability to model the abstract links between design decisions(requirements) and the manufacturing limitation dictating such a requirement. Such a model will prove beneficial in two ways in contrast to the current practise of iterative physical implementation tuning:

1. Design cycles for high performance circuits will be driven by performance requirements rather than manufacturing requirements since the impact of design decisions on manufacturability can be assessed early in the design cycle.
2. With relatively few models a large number of manufacturing limitations can be dealt with at early stages of the physical implementations leading to shorter product times. Additionally if the models result in simpler masks, the cost benefit is twofold.

On a more immediate note, results from Part II suggest that the diversity of cells is key to achieving performance. Relating this to regularity is essential so as to not negatively impact performance. For instance one can think of the impact regular cells with large drive strength requirements has on performance. In the case of ultra- regular layouts, since the diffusions of different devices are not individually sized and in addition single lines of diffusion are preferred, the impact of diffusion capacitance must be estimated in order to arrive at an optimal diffusion width with the same performance as cells designed with individually sized transistors.

An interesting avenue for further research is the possibility of regular routing. In the context of current design practises this is a more challenging task as it demands changes in the routing strategies currently employed. Alternately one could imagine influencing the routing towards regular solutions by using intelligently assigned pin targets.

Another possibility briefly mentioned earlier in this document (see section 3.4) but not considered in this work is automatic cell generation. While automatic cell generators have been used in the past it is increasingly difficult to create generators that are portable across the newer technology nodes due to the number of DRC considerations that must be taken into account. Any effort towards automatic generation of cells for the latest technology nodes will need to be in tune with the manufacturing capabilities available in order to obtain accurate solutions.

These are but a few of the avenues available for further work. The scope this work encompasses means that innovative solutions at different steps in the creation of an electronic product will aid the continued advance of CMOS technology.