UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Layout Regularity for Design and Manufacturability

Marc Pons Solé

Directors: Francesc Moll i Jaume Abella

Tesi presentada per obtenir el títol de Doctor per la

Universitat Politècnica de Catalunya

Programa: Enginyeria Electrònica

Barcelona, 08–07–2012

Thesis presented on the 2nd of October, 2012

To my Mother, Loles.

Contents

Acknowledgments

The thesis started thanks to the collaboration project "Variations-Aware Circuit Designs for Microprocessors" between the Intel Barcelona Research Center and the Electronic Engineering Department of the Universitat Politècnica de Catalunya. Then it was supported by the 2008 FI-B 00557 grant from the Generalitat de Catalunya and after by the FPU AP2007-04125 grant from the Spanish Ministry of Education and Science, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 248538 (Synaptic project) and by MODERN project of the Spanish Ministry of Science and Innovation (ENIAC-120003 and PLE2009-0024).

I would like also to thank the HiPICS research group (SGR 1497) from the Electronic Engineering Department and the ARCO research group (SGR 1250) from the Computer Architecture Department, both from the Universitat Politècnica de Catalunya, the Barcelona Supercomputing Center, and the Centre Suisse d'Électronique et de Microtechnique (CSEM), from Neuchâtel in Switzerland, that helped me during my thesis.

Abstract

In nowadays nanometer technology nodes, the semiconductor industry has to deal with the new challenges associated to technology scaling. On one hand, process developers face increasing manufacturing cost and variability, but also decreasing manufacturing yield. On the other hand, circuit designers and electronic design automation (EDA) developers have to reduce design turnaround time and provide the tools to cope with increasing design complexity and reduce the time-to-market. In this scenario, closer collaboration between all the actors involved is required. New approaches considering both design and manufacturing need to be explored. These are the so called design for manufacturability (DFM) techniques.

A DFM trend that is becoming dominant is to make circuit layouts more regular and repetitive. The regular layout fabrics are based on the configuration of a simplified mask set, therefore reducing the manufacturing cost. Moreover, a reduced number of layout patterns is used, allowing better process variability control and optimization. Hence, regularity reduces layout complexity and therefore design complexity, allowing faster time-to-market.

In this thesis, we explore forcing maximum layout regularity focusing on future technology nodes, with increasing design and manufacturability issues, where we expect layout regularity to be mandatory. With this objective, we have developed a new regular layout fabric called Via-Configurable Transistor Array (VCTA). The physical design is fully explained involving layout and geometrical

considerations for transistors and interconnects.

Initially, VCTA layouts developed manually have been evaluated in terms of manufacturability, but also in terms of area, energy and delay. For digital design, 32-bit binary adders designed with VCTA have been compared to standard cell layouts. For analog design, a delay-locked loop design using VCTA has been compared to its full custom version.

We have also developed a physical synthesis tool that allows us to obtain VCTA circuit layouts in an automated way. Developing our own automation tool lets us controlling all the decisions made during the physical design flow to ensure that maximum layout regularity is respected. In this case the work is based on several algorithms, for instance for routing, that we have oriented to the area optimization of the layouts.

Finally, in order to demonstrate the benefits of layout regularity, we have proposed a new layout regularity metric called Fixed Origin Corner Square Inspection (FOCSI). It is based on the geometrical inspection of the patterns in the layouts and it allows designers to compare regularity of designs but also how their regularity will impact their manufacturability. The FOCSI layout analysis tool can be used to optimize manufacturability.

Chapter 1

Introduction

Integrated circuits are more and more present in our life. From personal computers to smart-phones, or hidden electronics in cars, we are all using integrated circuits in our daily life. There are infinite examples. As layout minimum feature sizes shrink, more elements can be integrated in a single chip allowing new applications and capabilities. However, this is a challenging trend that requires enormous efforts from the semiconductor industry including manufacturers, designers and electronic design automation (EDA) developers.

In section 1.1 we first present an historical overview of technology scaling. Then, in section 1.2 we explain the integrated circuit manufacturing flow. In section 1.3 we detail the resulting challenges that have to be faced to fulfill the requirements of nowadays nanometer technologies. Finally, in section 1.4 we introduce the concept of layout regularity that is a possible solution for the semiconductor industry and that is the focus of the thesis.

1.1 History of technology scaling

Moore's Law, first proposed in 1965 by Gordon Moore, one of Intel's co-founders, predicted that every year the number of transistors in a chip will double. Ten years later, in 1975, it was modified so that from 1980 the number of transistors

will double every two years. Up to now Moore's law is still being followed. In a way it is a self-fulfilling prophecy as it is not based on fundamental arguments, but the semiconductor industry is making enormous efforts to accomplish this goal. A good example of these efforts is depicted in Figure 1.1 that shows the continuous increasing evolution of the number of transistors in Intel's processors.

In Figure 1.2 are shown the scaling challenges faced by the semiconductor industry from the manual design to nowadays. Based on a robust manufacturing process and on the appropriate EDA tools, in the first years of technology scaling, named the years of "happy" scaling, designers dealt with energy, delay and area problems. However, when reaching the deep submicron era, for technologies under 100 nm, new issues arose related to yield, defined as the percentage of good circuits over the total number manufactured. In particular, manufacturing variability is now one of the major challenges. Therefore new design for manufacturability (DFM) approaches coming both from designers and manufacturers sides, and allowed by EDA developers, are required [1, 2].

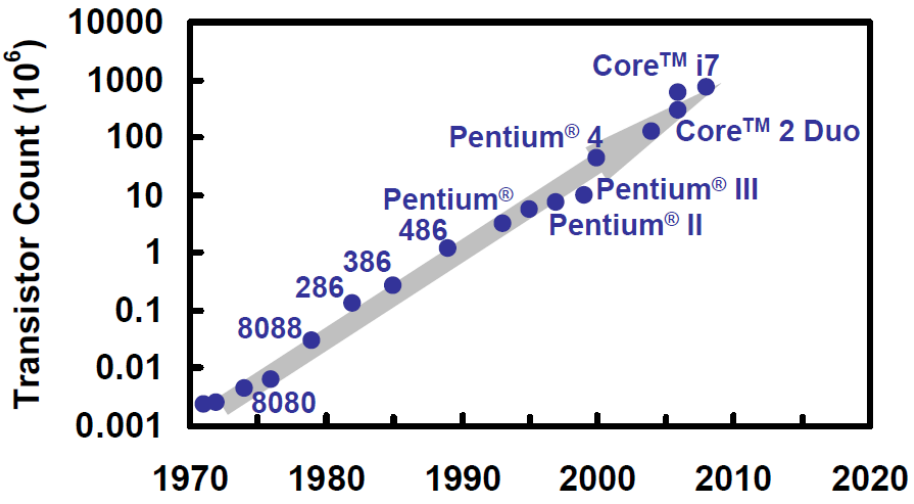


Figure 1.1: Moore's Law continues. Source: Intel.

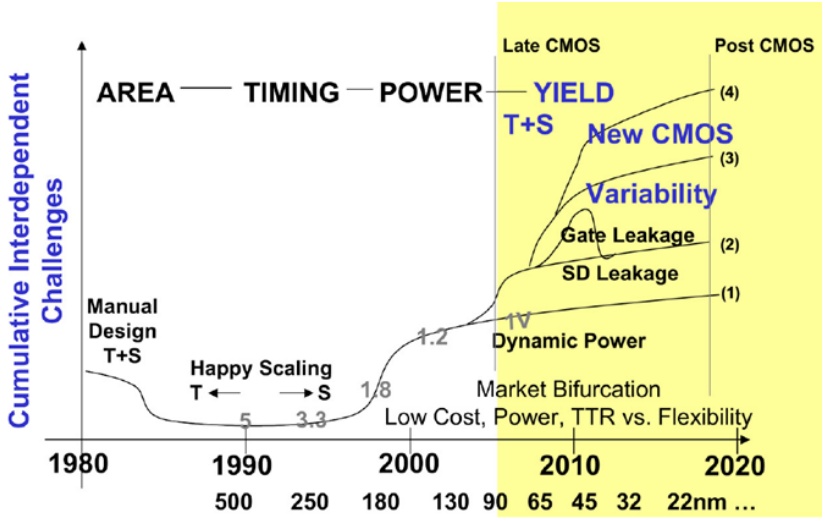


Figure 1.2: Technology scaling challenges [3].

1.2 Integrated circuit manufacturing process

1.2.1 Overview

Today's CMOS technology nodes with minimum feature sizes reaching 22 nm and below involve multi-disciplinary manufacturing processes like ingot production, lithography, ion implantation, electroplating (ECP) or chemical-mechanical polishing (CMP). Each of these steps of the process require strict manufacturing control and are very challenging for deep submicron technologies. Figure 1.3 illustrates in a schematic way several of the steps involved in the manufacturing process flow from sand to the final encapsulated chip.

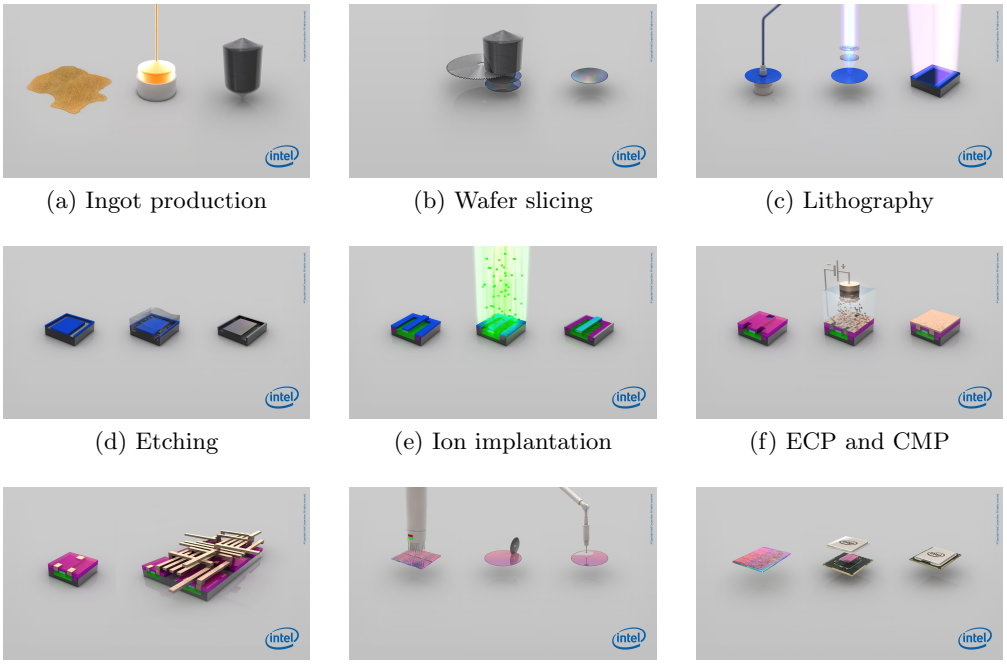
Typically, the integrated circuit manufacturing flow can be divided in three parts [4]:

- FEOL: front-end of line, for the manufacturing of transistors (PMOS and NMOS). Accurate control of transistor channel length and width, oxide thickness and dopant placement are amongst the main challenges during this step.

- MEOL: middle-end of line, for pre-metallic dielectric (PMD) and contacts

between transistors and interconnects. PMD isolates the FEOL from the back-end of line (BEOL explained next) and protect transistors. For this step the variability affects metal width and thickness as well as interlayer dielectric thickness.

- BEOL: back-end of line, for the rest of the interconnect metal layers and the vias connecting them. The goal is to create circuit functionality by interconnecting transistors and providing ground and power supplies. As for MEOL, variations appear in metal width and thickness as well as interlayer dielectric thickness, but concerning the upper metal layers.



(a) Ingot production (b) Wafer slicing (c) Lithography

(d) Etching (e) Ion implantation (f) ECP and CMP

(g) Interconnection (h) Die testing (i) Packaging

Figure 1.3: Manufacturing process overview [5].

1.2.2 Lithography

Lithography is used to obtain in the silicon wafers the small patterns of circuit layouts. It is based on projecting the pattern through a mask (including the shapes four times bigger than the desired shapes) on a photo-sensitive layer on the

top of the wafer. The photo-sensitive layer (also called resist) is then developed. If the resist is positive, the illuminated parts are eliminated. On the other hand, if the resist is negative, the non illuminated parts are eliminated. The remaining resist is then used as a protection for etching or implantation. Figure 1.4 shows the projection lithography system.

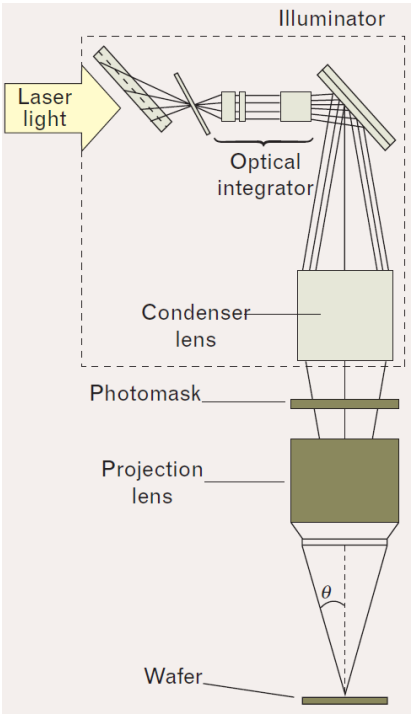


Figure 1.4: Projection lithography system [6].

The source of light is a monochrome laser treated with an illuminator to tune the angular content of light before traversing the photomask or reticle. The refracted light is then treated by a projection lens to reach the resist and form the image desired in the wafer. As traversing the reticle is equivalent to a Fourier Transform, the projection lens acts as a 2D Inverse Fourier Transform. Finally the resist detects light intensity.

The resolution of the image obtained using lithography is limited by the numerical aperture (NA) of the projection lens, by the wavelength of the laser source (λ) and by the Rayleigh constant (k_1), following equation (1.1). A 0.6

value for the k_1 parameter is required to ensure high pattern fidelity.

$$resolution = k_1 \cdot \frac{\lambda}{NA} \tag{1.1}$$

Added to this complex system, the lithography tool presents non-idealities. For instance, light intensity (also referred as dose) and distance to the resist (also referred as focus) vary. For dose, the variation observed in patterns of the layout depends on the slope of the light intensity when going from illuminated to non-illuminated zones. Ideally, the slope should be infinite at the edges of the patterns for maximum resolution. However, this is not the case for real tools and a variation in light intensity will cause a variation in the printed shapes. For the distance to the resist, there is an ideal distance, or optimum focus, where all light waves coming from different angles have the intended phase offset relative to each other and this way form a projection of the desired image on the reticle. However, when the position of the resist varies, generating defocus, the image also varies. Mask has also imperfections associated to its own manufacturing process.

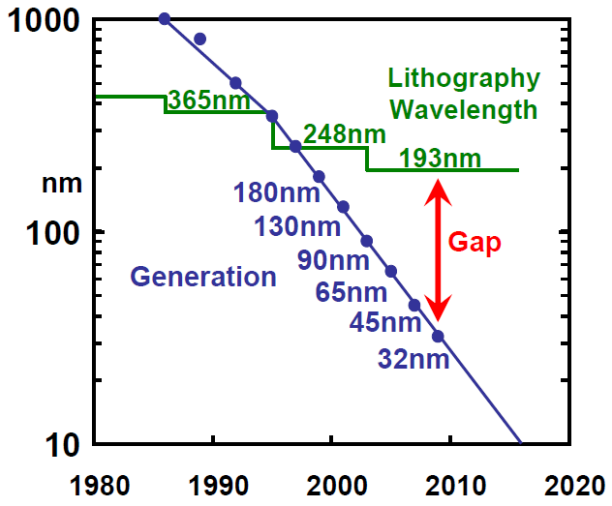


Figure 1.5: Sub-wavelength lithography gap. Source: Intel.

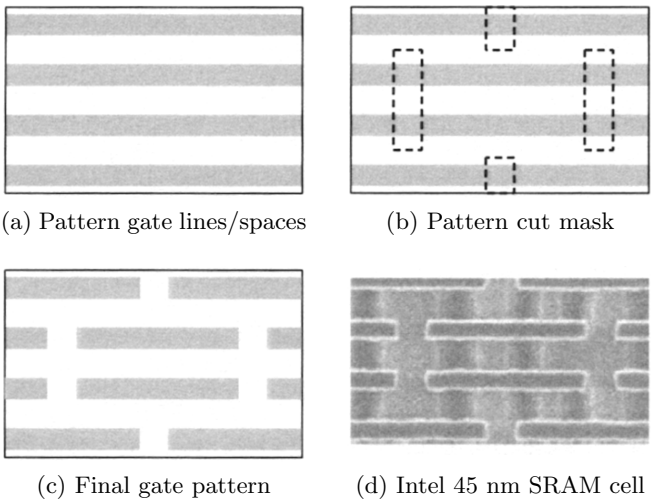
1.2.3 Resolution enhancement techniques

The conventional way to allow good resolution and, therefore, acceptable manufacturability when scaling down technologies, was to reduce the source of light wavelength (λ). However, for the last five technology nodes spanning from 90 nm to 22 nm the same 193 nm wavelength argon fluoride laser source is still being used due to the delays and challenges in developing, first, the fluorine 157 nm lithography systems, that was abandoned, and, second, the 13.4 nm extreme ultraviolet lithography (EUVL), that is expected to be available in 2013. Figure 1.5 shows the gap between lithography wavelength and critical feature sizes.

Therefore, for advanced technology nodes, these sub-wavelength lithography technologies require advanced resolution enhancement techniques (RETs) that allow reducing k_1 factor to values near 0.25 while maintaining good pattern fidelity. Amongst them there are techniques such as phase-shift mask (PSM), optical proximity correction (OPC) or double pattern technology (DPT). The principles of DPT, PSM and OPC are depicted in Figures 1.6, 1.7 and 1.8. Immersion lithography has also been introduced in last technology nodes to obtain an ultrahigh numerical aperture (NA) that also increase lithography resolution. However, these techniques are not effective for complex circuits with arbitrary layout patterns [7, 8, 9] and manufacturing variability still appears in effects like corner rounding or line-end shortening. For instance, for the widely used standard cell approach, considering a library of 1000 standard cells, there are approximately 2 million possible configurations to arrange a pair of standard cells. This large number of possible arrangements makes RETs computationally difficult.

1.2.4 Electroplating and chemical mechanical polishing

Another process steps that will influence circuit variability are electroplating (ECP) and chemical mechanical polishing (CMP), that are very correlated and are used for metal deposition and planarization.



(a) Pattern gate lines/spaces (b) Pattern cut mask

(c) Final gate pattern (d) Intel 45 nm SRAM cell

Figure 1.6: Double Patterning process flow [10].

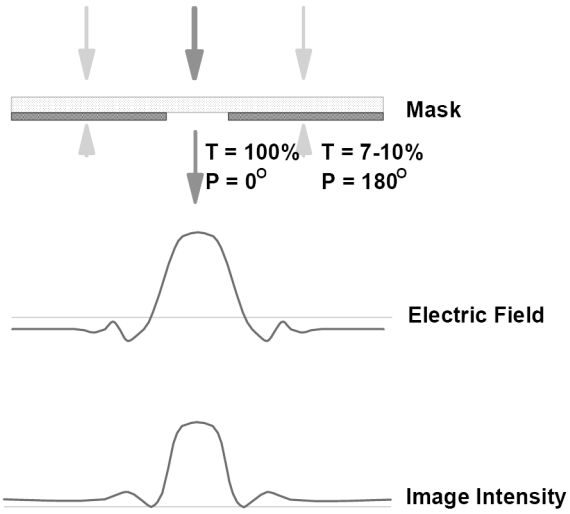


Figure 1.7: Phase Shift Mask. PSM lithography improves pattern fidelity by darkening the edges of shapes through destructive interference of light using a mildly translucent photomask. T is the amount of light that trespasses the photomask. P is its phase. [11]

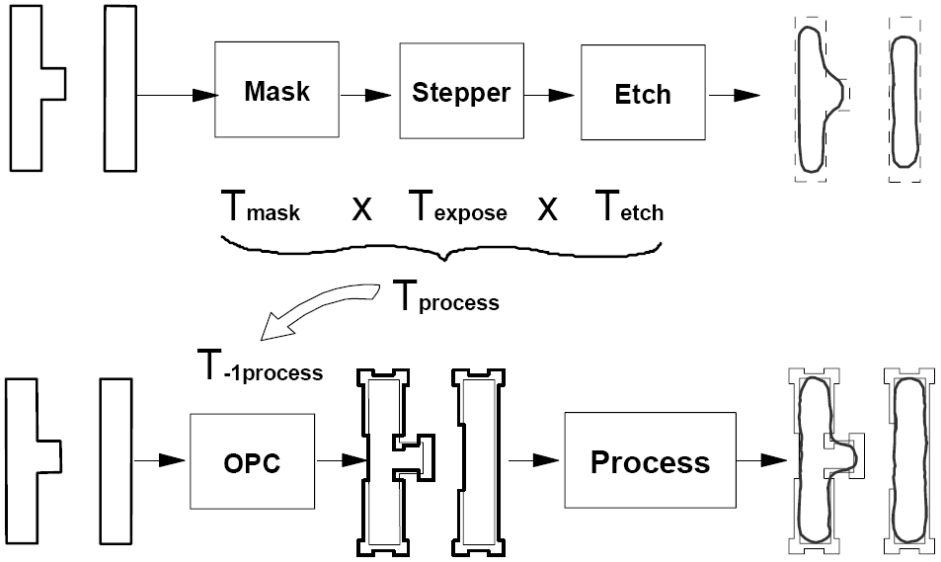


Figure 1.8: Optical Proximity Correction. OPC begins by characterizing the patterning operation and all its inaccuracies from various sources. This mathematical description of the process is used in iterative optimization routines to pre-distort the mask shapes to compensate for known, systematic, and modeled patterning inaccuracies. [11]

1.2.4.1 Electroplating

Patterned trenches for wires and vias are filled with copper using ECP. It is based on an electro-chemical copper deposition where the wafer acts as a cathode and solid copper as an anode in a plating solution containing copper ions, the electrolyte. Four chemical additives have to be adjusted to control the copper growth in the trenches [4]. These additives are:

- Suppressors, to restrain local growth.
- Chloride ions, to facilitate the suppressor adsorption.
- Accelerators, to enable local growth acceleration.
- Levelers, to reduce surface topography.

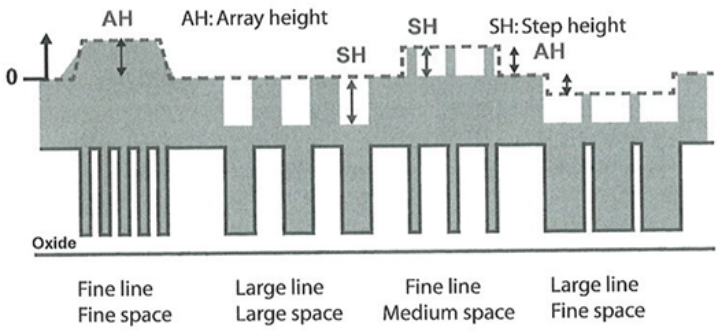
The control of these additives is critical to avoid the formation of voids in the resulting vias and wires. Moreover, different patterns of the trenches with

different metal densities systematically produce variations in the thickness of the copper grown.

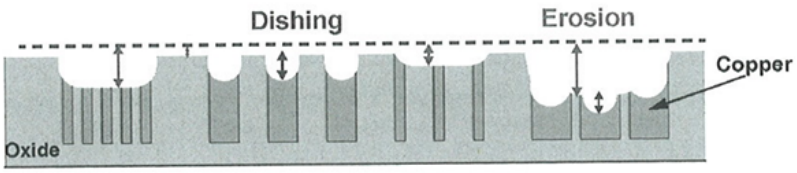
1.2.4.2 Chemical mechanical polishing

Chemical Mechanical Polishing (CMP) is used for planarization of metal layers after ECP. The excessive metal is etched away applying rotating pressure in a mechanical way with a pad while adding abrasive particles to the surface of the wafer. Therefore it is a combination of mechanical and chemical processes.

Due to surface irregularities because of different metal growth from ECP, the result of CMP is not a perfectly planar wafer (Figure 1.9). Effects like scratches, under polishing, dishing and erosion appear. Then, when applying lithography over this surface, focus variability occurs and leads to process variations in the upper metal layers. These variations will impact wires and vias behaviors.



(a) After ECP



(b) After CMP

Figure 1.9: Electroplating and Chemical Mechanical Polishing interactions [4].

1.2.4.3 Metal filling

Metal characteristics will vary substantially depending on layout and intra-layer density variations, neighborhood layout patterns and the underlying topology. That is why, to deal with variability arising from ECP–CMP interactions, dummy metal features in free spaces of the layers are added to obtain density of the materials as uniform as possible. In that way, metal growth is more uniform and so will be the resulting planarization. However as density will remain variable depending on the circuit designed, planarization variability will still occur.

1.3 Challenges of semiconductor industry

The semiconductor industry is facing increasing challenges related to the integrated circuit manufacturing process but also to the design productivity. Regarding the manufacturing process, critical issues are equipment cost, as well as mask cost, and the manufacturing variability an yield. Regarding design productivity, the problem is centered on the time-to-market and also on cost. Turnaround time required to obtain working circuits is increasing because of circuit variability, as well as is increasing the EDA tools computational effort required to deal with design complexity.

1.3.1 Manufacturing challenges

1.3.1.1 Manufacturing cost

The major contributor to manufacturing cost is equipment cost. In Figure 1.10 we can see how the wafer fabrication line cost (that includes the lithography exposure tools cost) has increased exponentially over the years of technology scaling. Nowadays, the cost can reach more than $4 billion.

Mask cost is also critical as new masks are required for each of the designs developed. Figures 1.11 and 1.12 show how the mask cost of the standard cell approach increases with every technology node, associated to the mask complexity increase, in terms of the number of mask layers. The mask cost has increased

from $102 thousand dollars, for a 350 nm design, to almost $2 million, for a 65 nm design.

To illustrate the cost challenge faced by the semiconductor industry, in Figure 1.13 we can see how design costs surpass revenues for application specific integrated circuits (ASICs). New approaches are required.

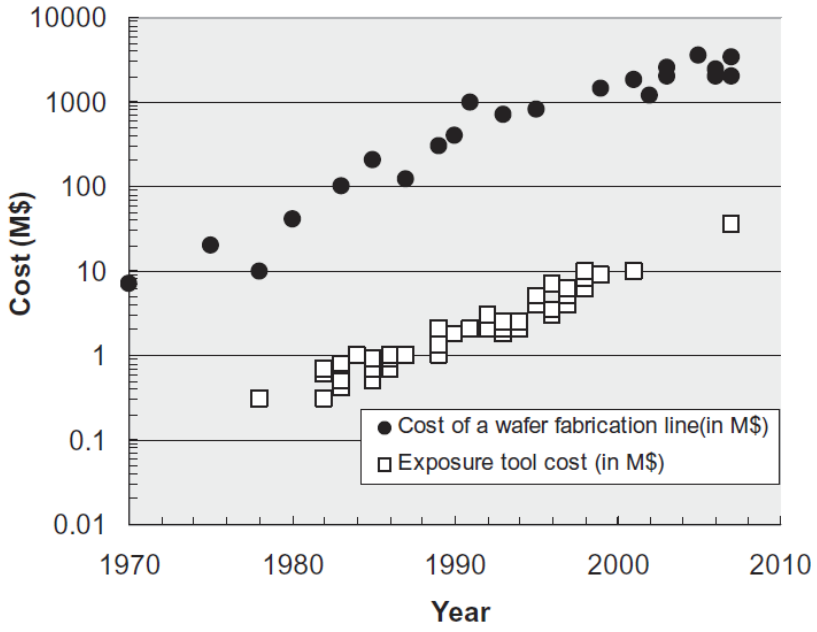


Figure 1.10: Rising cost of manufacturing. [12]

1.3.1.2 Manufacturing variability

Due to the limitations of the different steps involved in the manufacturing process, transistor and interconnect parameters vary. For integrated circuits, process variations are defined as the deviation from the expected nominal value of the characteristics of these transistors or interconnects. This deviation is calculated from a large number of samples.

Depending on the mechanism involved, we can observe different scales of process variations, such as within-die (WID), die-to-die (D2D) and wafer-to-wafer (W2W). Process variations can also be classified depending on their behavior: they can be systematic or random variations. The resulting parameter varia-

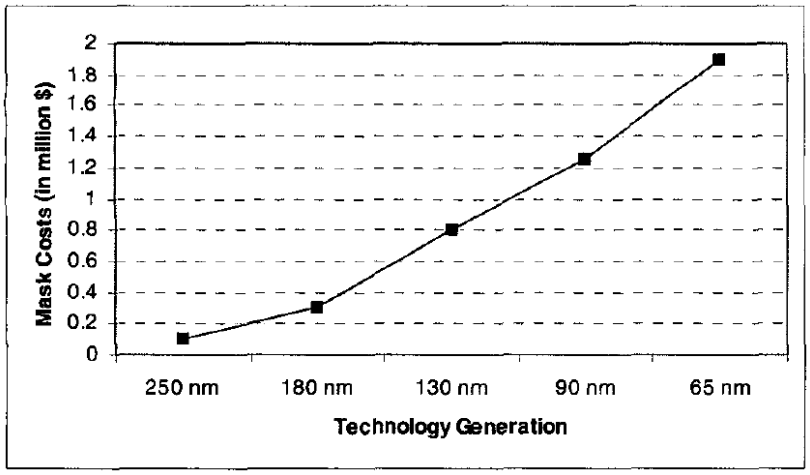


Figure 1.11: Rising cost of a CMOS standard cell mask set. [7]

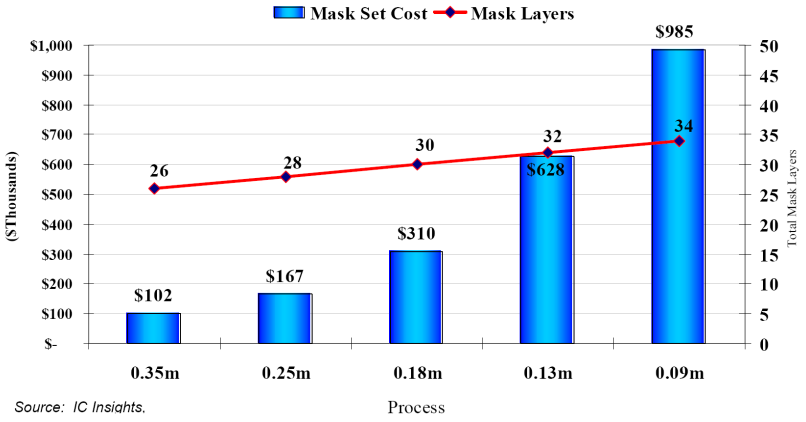


Figure 1.12: Mask layers and cost per technology node. [13]

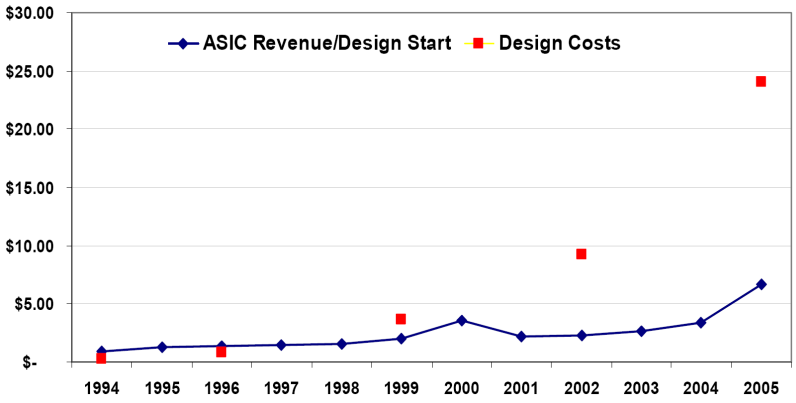


Figure 1.13: Average revenue and design costs per year. [13]

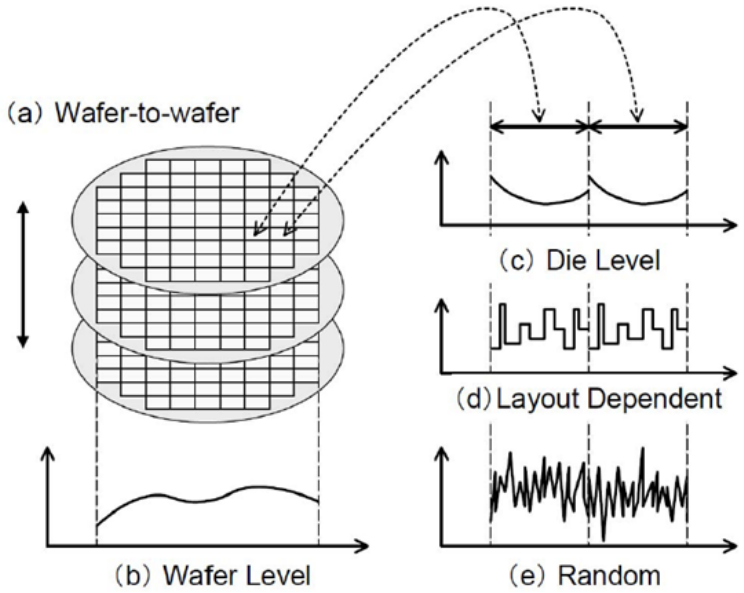


Figure 1.14: Variability classification [14].

tion from the nominal value will come from the combination of both deviations. Variability classification is depicted in Figure 1.14.

The systematic deviation component models the parameter variations that can be predicted by manufacturing environment and layout geometry (Figure 1.14 from (a) to (d)). This component is due to unintentional shifts in process conditions and to lithography tools [15].

On one hand, regarding process conditions, the systematic deviation includes the D2D and the W2W parameters variations. It also includes the WID variations that depend on the position and orientation of the device or the interconnect on the die. In general, for process conditions, the difference between the systematic deviation levels for the parameters of two devices in the same die will be smaller than the one of two devices in different dies. Similarly, the level of deviation between devices close to each other will be lower than the one of devices in separate regions of the same die.

On the other hand, due to lithography tools limitations, depending on the particular layout neighborhood, shapes will also be affected by different variations. Models for these mechanisms have been investigated and can be applied to predict the amount of variations related to them (e.g., dose and focus variability models).

The random deviation component models the parameter variations that are uncorrelated with the position on the die and that are not deterministic (Figure 1.14 (e)). This component is related to atomic-level differences between devices [15]. For instance, because of the random dopant placement in the transistor channel that shows no correlation between different devices even if they are one next to each other, threshold voltage variations can be considered random and independent for each device. In fact, the random dopant placement in small-geometry devices is especially important because it causes mismatch between neighboring transistors [16]. These variations are modeled with a normal (or Gaussian) distribution because any variable that is the sum of a large number of independent factors is likely to be normally distributed. Note that the goal of manufacturers is to find the models for these variations, nowadays considered as random, to be able to predict them and treat them as systematic variations.

1.3.1.3 Manufacturing yield

As explained before, integrated circuits yield can be defined as the percentage of good circuits manufactured over the total. Figure 1.15 shows the detailed

components of yield loss for different technologies. On one hand, defect-density related problems are caused by actual errors with the silicon, such as when a contaminating particle is introduced during fabrication. This component is not related to variability issues. However, for the other two components of yield loss, variability is a major concern. Most of the lithography based failures occur when there are defects on the masks or due to layout pattern dependent issues. In this case, systematic variability is the origin of yield loss. Parametric yield loss, on the other hand, occurs because the manufactured chip does not meet a design parameter, like frequency or power dissipation. In this case, yield loss can be caused by design errors but also because of random and systematic variability.

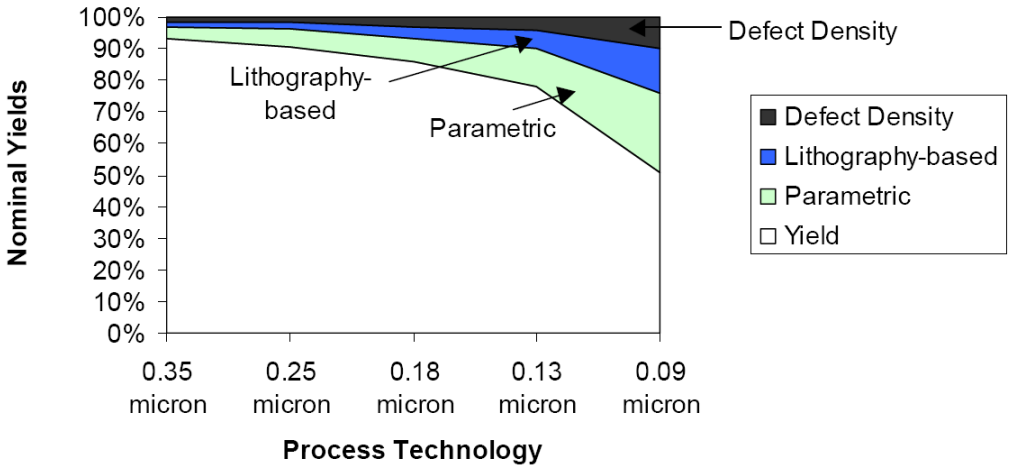


Figure 1.15: Yield factors for different process technologies. [17]

In order of importance, first, we have the parametric yield loss (25% for 90 nm). Second, we have systematic lithography based failures (15% of yield loss for 90 nm). Finally, we have random defect-density related problems (10% of yield loss for 90 nm). Chip yields are expected to drop from over 90% for 350nm to around 50% or less for 90nm. Furthermore, this trend will continue for next technology nodes [18].

1.3.2 Design challenges

1.3.2.1 Design cost

As well as manufacturing cost, design cost has also increased considerably due to technology scaling. In Figure 1.16 we can see how it is reaching $75 million for the 32 nm technology node. In fact it is directly related to the difficulties in design verification that require increasing turnaround time. Even if efforts to improve verification tools have been invested, designs are not anymore directly working on the first tapeout like for past technology nodes.

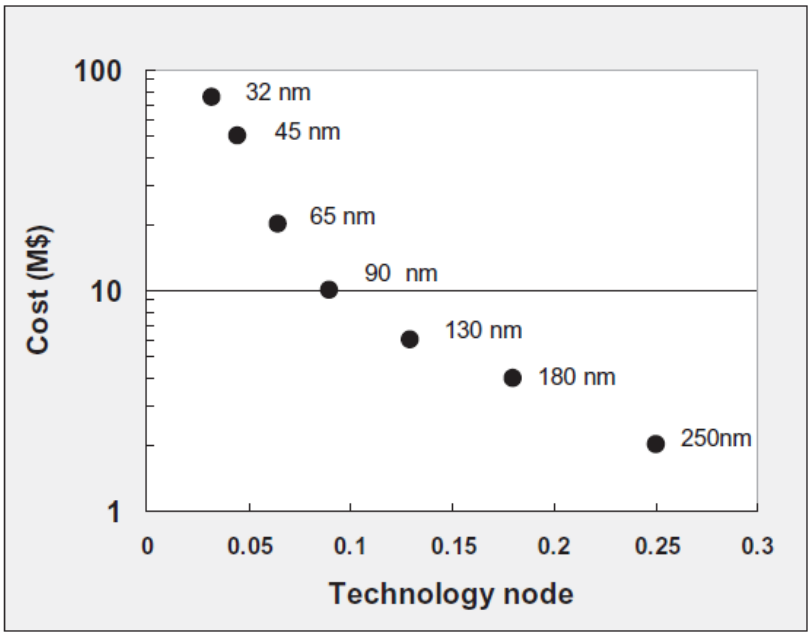


Figure 1.16: Rising cost of design. [12]

1.3.2.2 Design turnaround time

Manufacturing variability leads to variations in performance and power consumption of circuits. A large variety of numbers can be found in the literature for the impact on power and performance of process variations in different sorts of circuit. For instance, considering only transistor variations in a wafer, about 30% variation in chip frequency and 20x variation in chip leakage have been observed [19].

Other predictions show that process variations for gates and wires result in a maximum 40% circuit performance variation and a 55% circuit power dissipation variation [20]. In the field of microprocessor functional units, some results have been published for different adder implementations showing variations on both power and performance around 20% [15]. Other authors assume a 10-15% delay variation for single gates [21].

This power and performance unpredictability is taken into account by designers by adding pessimistic guard bands to ensure that the design meets the specifications. Therefore, more design iterations are required. In fact, unpredictability can cause two kinds of problems. On one hand, if variability in power and performance is overestimated, the design guard bands lead to an increase in the design effort as stated before. However, on the other hand, underestimating the variability will relax design requirements but will also require an increase in the manufacturing effort to ensure acceptable yield. In both cases, this will represent an increase in the cost and the time-to-market that can be critical for the success of the product.

1.3.2.3 Design complexity

With technology scaling, the capabilities of integration are increased. Therefore design complexity is also increased. Also, the amount of design rules in layout is increasing for advanced technology nodes due to lithography limitations. As explained before, RETs like OPC or DPT are not effective for complex layouts with arbitrary layout patterns. The verification tools show also limitations. The challenge faced by EDA developers is here to provide the tools to cope with design complexity in an efficient way to reduce the design effort and also the time-to-market and their associated costs. Important efforts have to be invested in synthesis and analysis tools to take profit of the scaling benefits.

1.4 Layout regularity solution

Layout regularity can be defined as the repetitivity of layout patterns or layouts blocks in the design. Depending on the granularity, it can be referred as micro-regularity (for repetitivity at pattern level, for small sizes of layout) or as macro-regularity (for repetitivity at block level, for higher sizes of layout). The goal is to design with a reduced set of layout bricks (e.g., patterns or blocks), making small configuration changes to generate different circuits, with the objective of tackling with manufacturing and design challenges for nanometer technologies. Regular layouts require a simplified mask set, therefore reducing the manufacturing cost and allowing better process control. Moreover, layout complexity reduction also reduces design complexity and therefore the design time.

In the 2011 edition of the International Technology Roadmap for Semiconductors (ITRS [20]), that is nowadays the reference for technology scaling needs, it is stated: **Cost-effective product manufacturing also requires continuous improvements in the area of design for manufacturability, specifically areas such as design to minimize performance/power variability, lithography-friendly designs (regular layout styles consistent with increasingly more restrictive design rules), and design for testability and reliability**.

Layout regularity is a DFM technique to address the manufacturing and design challenges in advanced technology nodes and it is the solution that we have adopted in the thesis.

Chapter 2

Related work

Layout regularity was first introduced in the early 1980s. At that time, the issue addressed was from the design side. In fact, designing integrated circuits required reducing layout complexity as no EDA tools were available. However, nowadays, layout regularity is also focused on manufacturability issues, that were not critical in the past. In section 2.1 we explain why layout regularity is becoming mandatory to tackle nowadays design and manufacturing challenges. The first regular layout fabrics and how they have evolved to nowadays fabrics are explained in section 2.2. Then, in section 2.3 we describe the existing layout analysis tools that can be used to evaluate manufacturability. Finally, in section 2.4 we justify the thesis works based on the related works presented.

2.1 The need for layout regularity

2.1.1 Regularity for manufacturing

2.1.1.1 Reducing manufacturing cost

The cost reduction due to layout regularity is based on reducing the mask cost. Regularity allows design with a reduced set of masks, that can be reused in their majority for different designs. Only the masks devoted to the configuration of the particular circuit vary from one design to the other.

2.1.1.2 Reducing manufacturing variability

Improving layout regularity is arising as a possible solution for manufacturers to reduce systematic variability [22, 12, 23, 24, 25]. Regular designs are composed by a reduced number of layout patterns in silicon and also in metal. Therefore, layout and intra-layer density variations are minimized thus reducing the manufacturing process variations associated to electroplating (ECP) and chemical mechanical polishing (CMP). Moreover, the number of possible layout neighborhoods is also reduced. The main benefit is the reduction of the amount of systematic process variations by allowing RETs (like OPC or DPT explained before) to more effectively mitigate lithography printability issues. Only a few layout patterns need to be optimized for manufacturability and therefore better process variability control can be achieved.

2.1.1.3 Improving manufacturing yield

Every two years a technology node starts at the first small circuit or transistor fabrication [26]. Then, huge investments are required in order to reach commercial chips yield. For deep sub-micron technologies the initial yield is around 15-20% and the time-to-market can last three years before reaching maximum chip yield around 50-60% [27, 28, 29].

In the first year the major part of the yield improvement takes place, then the improvement slows down every year. For the sake of illustrating this evolution we will consider that a 70% of the yield improvement occurs in the first year, 18% during second year, and 12% in the last year. In the case of regular layouts considering DFM techniques, we expect that this behavior can be compressed to one year.

Regarding the initial yield increase of regular layout proposals, we have to examine the different factors causing yield loss as explained before. For parametric yield loss (25% for 90 nm), once the design will be optimized, we hope the yield loss for regular designs will be lower than the one for standard designs because of

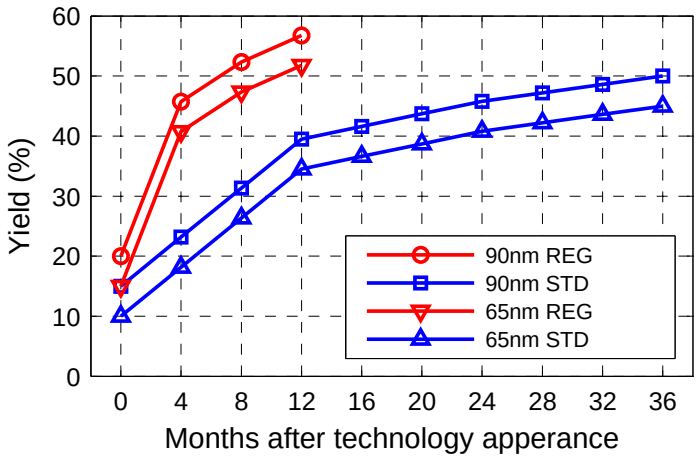


Figure 2.1: Yield predictions for standard (STD) and regular (REG) approaches. We have considered that the regular design initial yield is increased by a 5% and that yield improvement rate is increased by a 1.5 factor over a year

the manufacturing variability reduction that will increase design predictability. For systematic lithography based failures (15% of yield loss for 90 nm), regular designs are expected to perform much better than standard. In fact, by forcing layout regularity in both devices and interconnects, regular structures will reduce systematic yield losses associated to lithography tools and resolution enhancement techniques as we have explained. Finally, for random defect-density related problems (10% of yield loss for 90 nm), it is possible that regular designs perform worse than standard due to the area overhead that can be introduced by enforcing regularity. In any case, it is the less important contributor to yield loss and it is not expected to grow noticeably as opposed to the other sources of yield loss. Considering the hypothesis presented above, we have assumed that regular designs will have a little advantage in front of standard designs in terms of yield. That is why we have considered a 5% initial yield improvement in order to illustrate regular layout benefits.

We also analyze the yield improvement rate over time. The difference in speed between regular designs and standard yield evolutions is because regular is based on the repetition of a reduced set of basic blocks. The yield improvement over time is accelerated because only a reduced number of layout patterns has

to be optimized. That is why we have considered that yield improvement rate for regular designs is increased by a given factor. Note that such ratio depends strongly on the process technology and the manufacturer so it may vary a lot across different technologies and manufacturers. We have set such factor to 1.5 for the sake of illustrating potential benefits of regular designs on yield.

Based on the assumptions explained about yield evolution, initial yield level and yield improvement rate, Figure 2.1 shows predicted yield evolution for the 90 nm and 65 nm technology nodes compared to the expected yield evolution of regular layout design techniques. We can see how it is very likely that regular designs provide high yield after one year of development for a given technology node (i.e., 65 nm) when the standard approach provides acceptable yield for the previous technology node (i.e., 90 nm), that has appeared two years before but has been developed during three years. Although this is just a rough evaluation of yield, it illustrates the advantages of regular layouts with respect to standard design.

2.1.2 Regularity for design

To illustrate the benefits of layout regularity from the point of view of design, we will refer to Figure 2.2 that compares the life cycle of a design with and without including DFM techniques like improving layout regularity. In summary, we can see how DFM is predicted to reduce the time-to-market and therefore to increase the profit.

2.1.2.1 Reducing design cost

As we can see in Figure 2.2, a higher design effort is required at the beginning of the life cycle using a regular fabric, because the bricks to be repeated and the way to configure them have to be developed. However, once the fabric bricks defined, the design cost is reduced because less design re-spins are required (also referred as turnaround time, see next subsection). Furthermore, as the fabric complexity is reduced because of the reduced set of bricks, the portability of

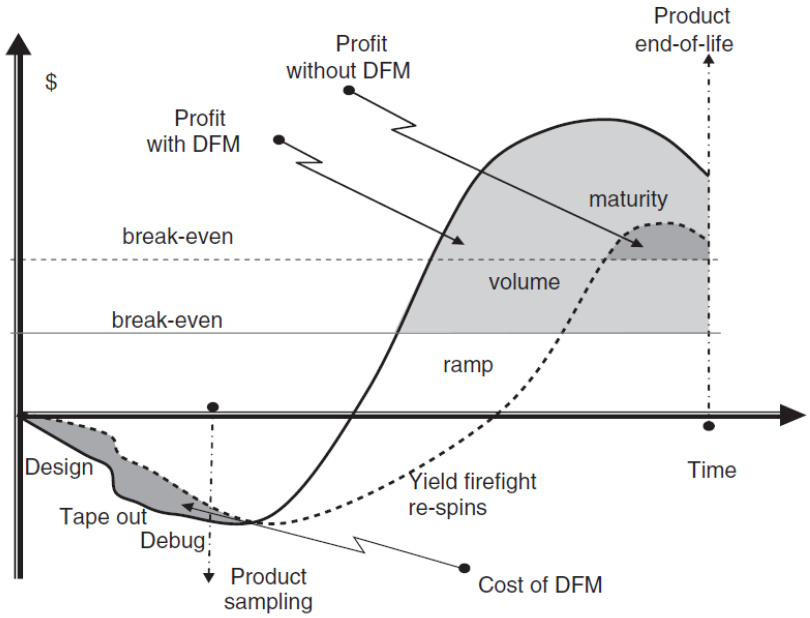


Figure 2.2: Economics of DFM. Dashed line represents the design without DFM. Solid line is for the design deploying DFM. [12]

designs is increased. For instance, when scaling down one technology, the work to develop the fabric will be also reduced.

2.1.2.2 Reducing turnaround time

Layout regularity reduces manufacturing variability. Therefore circuit unpredictability is mitigated and design guard bands can be better estimated. This is the reason why design effort and turnaround time are reduced with layout regularity. It is indicated in Figure 2.2 as the phase of yield firefight re-spins where you can see that the DFM design curve has a higher slope than the standard design curve.

2.1.2.3 Reducing complexity

Layout regularity reduces the amount of design possibilities by imposing stricter design rules and methodologies to generate the circuits. Therefore, EDA tools have to deal with less complex designs and can be optimized. For instance, as

we have explained previously, RETs become more effective in reducing manufacturing variability when improving layout regularity, but RETs also become easy and fast to achieve if compared with the resources and time spent on correcting and optimizing masks for irregular layouts [13]. Therefore the time-to-market can be reduced, such as in SRAM designs. Even if the design of the SRAM cell is critical, SRAM designs get to the market long before conventional logic designs because only one cell has to be optimized.

2.2 Regular layout fabrics

The first regular fabric proposals are gate arrays (GAs) which tackle the lack of EDA tools in the 1980s. Another set of proposals consist of standard cells which address the design of large complex integrated circuits. Finally, we introduce structured ASICs which focus on increasing regularity to mitigate manufacturing variability [30, 31].

2.2.1 Gate Arrays

2.2.1.1 Gate Matrix

Regular designs were first considered in the early 1980s as a systematic approach to chip layout when EDA tools were in development and did not have nowadays capabilities. Some of the first proposals are based on the concept of gate matrix (GM) where the regular intersection of rows and columns provide transistors and interconnections. There are two types of such structures that are shown in Figures 2.3 and 2.4. First, polysilicon oriented structures, where polysilicon columns, in which transistors are placed, are connected using metal rows to implement the function [32]. Second, there are metal oriented structures, where the intersection of metal rows, containing polysilicon gates, and diffusion columns generate transistors already connected [33].

In both cases, transistor regularity is achieved. However, interconnect regularity is not taken into account because interconnections are configured depending

on the functions synthesized.

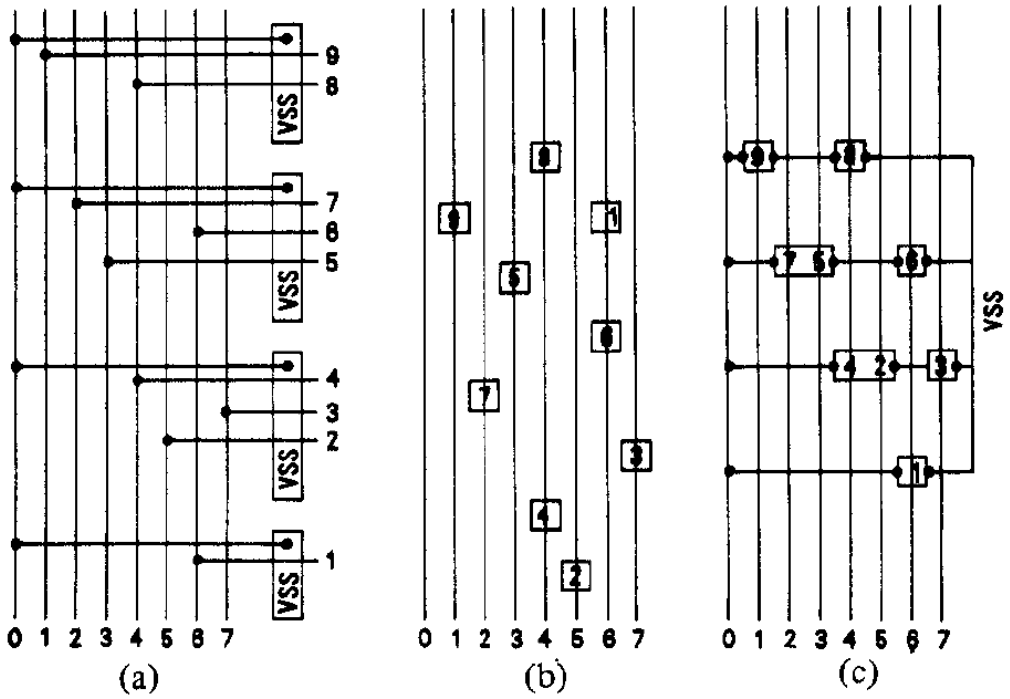


Figure 2.3: Polysilicon oriented structure. (a) Illustration of a standard cell configuration. (b)Polysilicon lines (matrix columns) which correspond to the inputs and outputs of the circuit with the transistors placed on their gating line. (c) The transistors have been grouped (matrix rows) and the interconnections are made with metal or diffusion. [32]

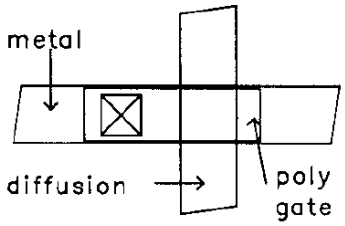


Figure 2.4: Metal oriented MOS transistor. [33]

The physical design of these GM proposals involves solving the gate matrix layout problem (GMLP) for instance using a constructive genetic algorithm (CGA) [34]. The main objective is to minimize the area of the final layout by arranging the circuit transistors in the GM. The number of metal rows used (also

referred as routing tracks) as well as the total wire length are included in the cost factors of the solution.

2.2.1.2 High-Density Gate Arrays

High-density gate arrays (HDGAs) are an evolution of GAs [35]. They are based on the optimized implementation of the structures called sea-of-transistors and sea-of-gates. The basic cells of the sea-of-gates structure proposal are gates from four to eight transistors isolated by means of oxide isolation as shown in Figure 2.5. For sea-of-transistors all transistors of the same type P or N share the same diffusion. This structure is depicted in Figure 2.6. Some circuits require transistors of different sizes and transistor gates usually share the same input. That is the reason why the common-gate HDGA structure presented on Figure 2.7 has been proposed with transistors of different sizes.

For HDGAs, transistor regularity is achieved if no different transistor sizes are considered. Interconnect regularity is not ensured because of the configuration of the wires like for GM.

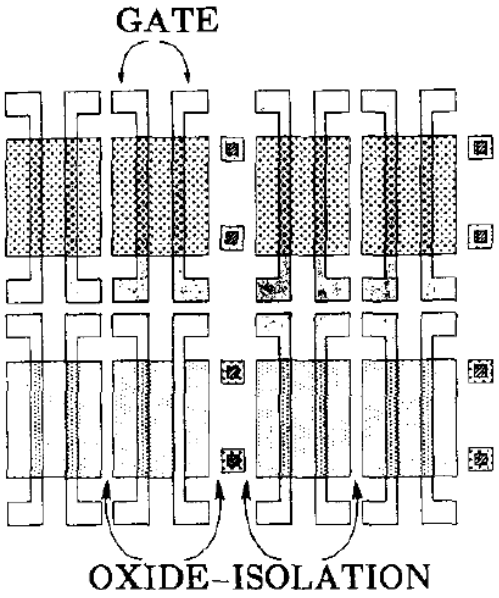


Figure 2.5: Typical example of a Sea-of-Gates architecture. [35]

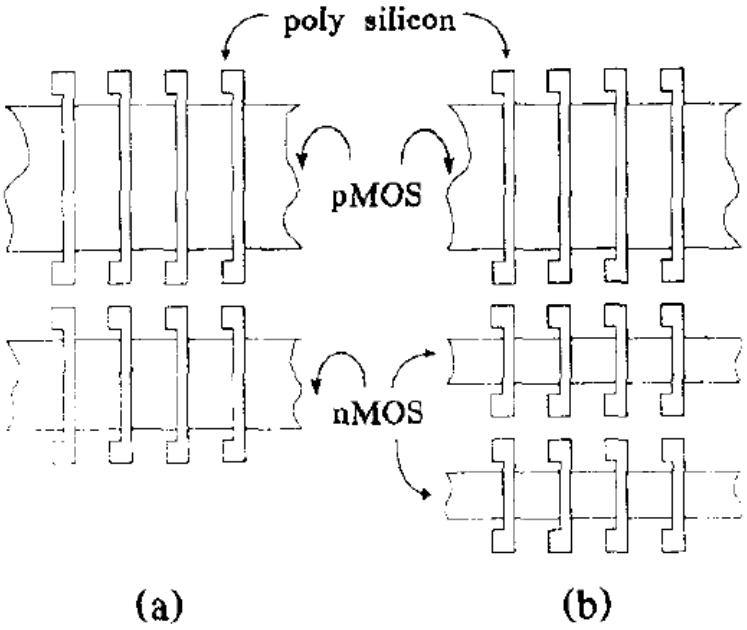


Figure 2.6: Sea-of-Transistors design. (a) Typical example of a Sea-of-Transistors architecture (b) Sea-of-Transistors architecture with multiple NMOS transistors. [35]

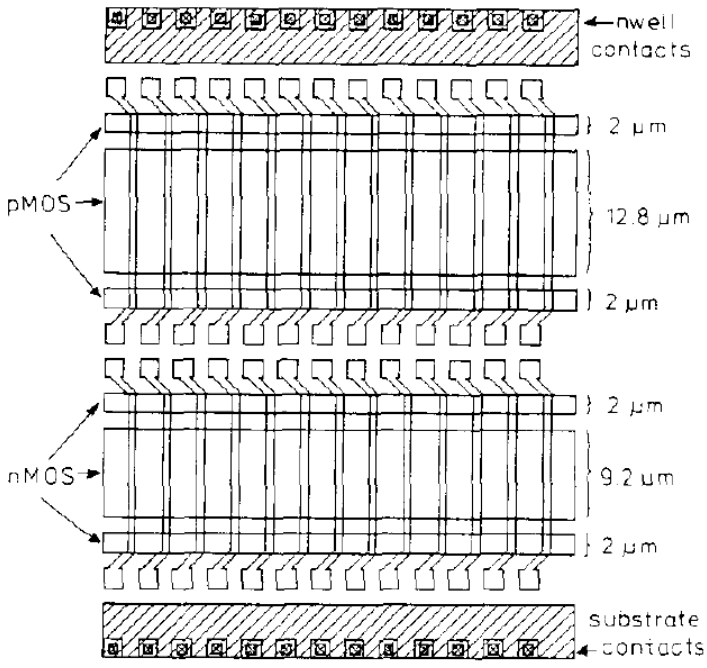


Figure 2.7: The common-gate HDGA architecture. [35]

The physical design is determined by routability requirements [36]. Routing is performed over the basic cells, therefore routing tracks positions are reserved during the cell design depending on the circuit needs. Routing algorithms are oriented to avoid routing congestion and discouraging interconnections over potential input and output positions of the cells. For placement, min-cut heuristics are used to minimize the number of interconnections that will cross the cells (to minimize routing congestion), and distances of interconnections are minimized using a quadratic metric.

2.2.1.3 Field Programmable Gate Arrays

Nowadays GAs have evolved into FPGA. The basic structure is conformed by logic blocks (LB), including Lookup Tables (LUTs) and Flip-Flops (FFs) plus the programming overhead, that are interconnected using routing blocks, consisting of connection boxes (CB) and switch boxes (SB) [37]. A possible structure is shown in Figure 2.8. The performance, power and area costs compared to the standard cell approach are in general too high because of the reconfigurability capability that it offers.

FPGA structure has a smaller degree of transistor regularity than the previous structures presented because of the use of different basic blocks with its own layout. However, interconnect regularity is achieved having prefabricated wires all along the circuit that only have to be programmed depending on the function to be implemented.

The configuration of an FPGA involves a reconfigurable computing synthesis flow that can be seen in Figure 2.9. The application description is usually described in a hardware description language (e.g., VHDL or Verilog) that is translated into a gate level netlist through the high level synthesis. Then, depending on the FPGA system, this netlist is divided into blocks that meet the FPGA resources defining the global routing that will be required between them. Then, the blocks are mapped into the FPGAs resources and are placed and routed minimizing the total design wire length of all design interconnections. The bit-

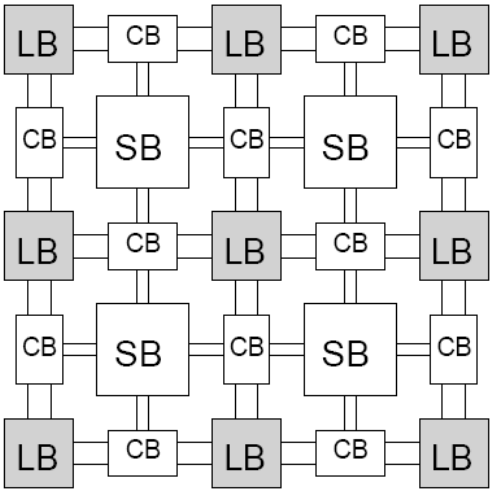


Figure 2.8: FPGA architecture example. [37]

stream is the configuration data to be loaded into the FPGA. There are also libraries of optimized circuit blocks bitstreams that can be directly loaded into the FPGA.

2.2.2 Standard Cells

With increasing circuit complexity, GAs also evolved into the standard cell design in parallel to the FPGA approach. Along with semiconductor manufacturing advances, standard cell methodology was responsible for allowing designers to scale ASICs from comparatively simple single-function integrated circuits (of several thousand gates), to complex multi-million gate devices (systems-on-chip).

A standard cell is a group of transistor and interconnect structures, which provides a boolean logic function (like NAND, NOR, XOR, inverters) or a storage function (flip-flop or latch). Ensuring that all standard cells have the same height, the resulting chip layout is composed, in a structured way, by rows of standard cells.

Nowadays, the major issue of this technique is that it is based on a too large library of standard cells. This results, like we have seen previously, on an extreme

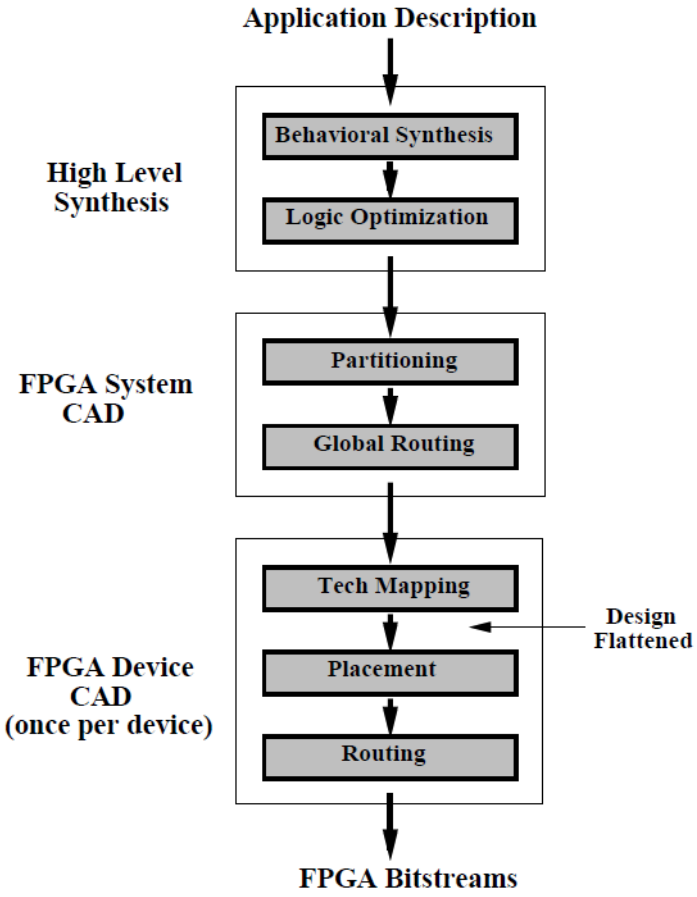


Figure 2.9: Reconfigurable Computing Synthesis Flow. [38]

layout irregularity because of the multiple ways of arranging the standard cells, and thus, on DFM and RETs inefficiencies.

2.2.2.1 Regular Logic Bricks

Reducing the number of standard cells of your library, you can have great benefits. This is the basic idea used by one of the most promising approaches to exploit regular designs that is being developed at the Carnegie Melon University. It is based on the concept of regular "Logic Bricks" [39] and obtains similar results to the standard cell approach in terms of area and performance but with a limited set of standard cells that are chosen to synthesize a given block. This way, the number of critical layout patterns are minimized and regularity is improved. The "Logic Bricks" discovery is shown in Figure 2.10. Moreover this reduced number of layout patterns inside of the bricks allows layout "pushed-rules" that are less pessimistic than common design rules leading to an area reduction [8, 40].

The experiments performed for the 65nm technology node for the implementation of an ARM9 microprocessor, using only 16 types of bricks, show a 6.67% increase in silicon area utilization and the same timing.

The transistor as well as the interconnect regularities are in this case function dependent, because they depend on the "Logic Bricks" selected to synthesize the function of the circuit. Therefore, the full chip regularity is not ensured. This kind of designs is an intermediate point between the ASIC approach and the fully regular designs. "Logic bricks" exploit the trade-off between area, power, performance and layout regularity.

2.2.2.2 Standard cells with improved regularity

The regularity of standard cell designs has been improved by Intel or AMD using polysilicon dummy features [41]. Regular structures for transistors also using dummies reduce the stress-induced performance variations [42]. Layout uniformity has also been shown to reduce yield loss associated to critical area [43]. Other works at Tela Innovations using gridded design rules have been shown to

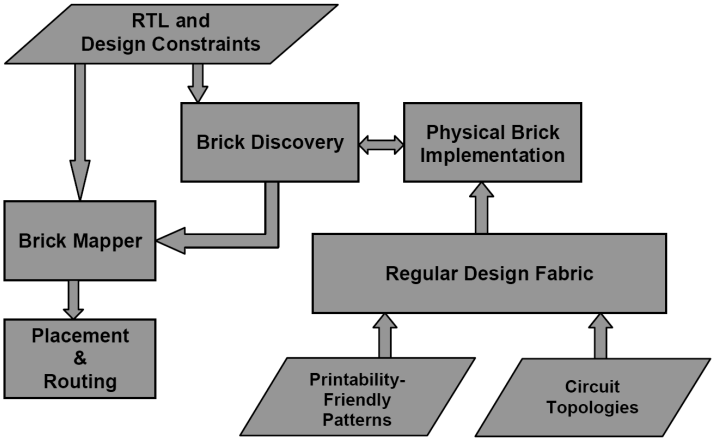


Figure 2.10: Logic Bricks Discovery. Before the physical design flow (Brick Mapper and Placement and Routing), Logic Bricks are selected depending on the circuit under study and the manufacturability requirements. [8]

reduce gate critical dimension variability by 4x to 16x by improving polysilicon regularity [44]. Similar works are being developed in our research group in the frame of the Synaptic project [45].

Regularity improvements mainly focus on the polysilicon layer because regularity also helps reducing process variations and the transistor channel length variability is dominant in the energy and delay functional yield. However interconnects still remain irregular and this leads to a larger number of different masks to be redesigned, only affordable in large productions.

2.2.2.3 Standard cell physical design flow

Including more or less regularity, all the standard cell approaches presented in this section use the same physical design flow that is the classical flow approach evolved from the 80's [46, 47]. Several tools are required to obtain the final layout of the circuit designed (a list of tools can be found in chapter 4 including tools for library characterization, logic synthesis and place and route). A summarized physical design flow is depicted in Figure 2.11.

First, designers need to choose the logic functions that will be included in the

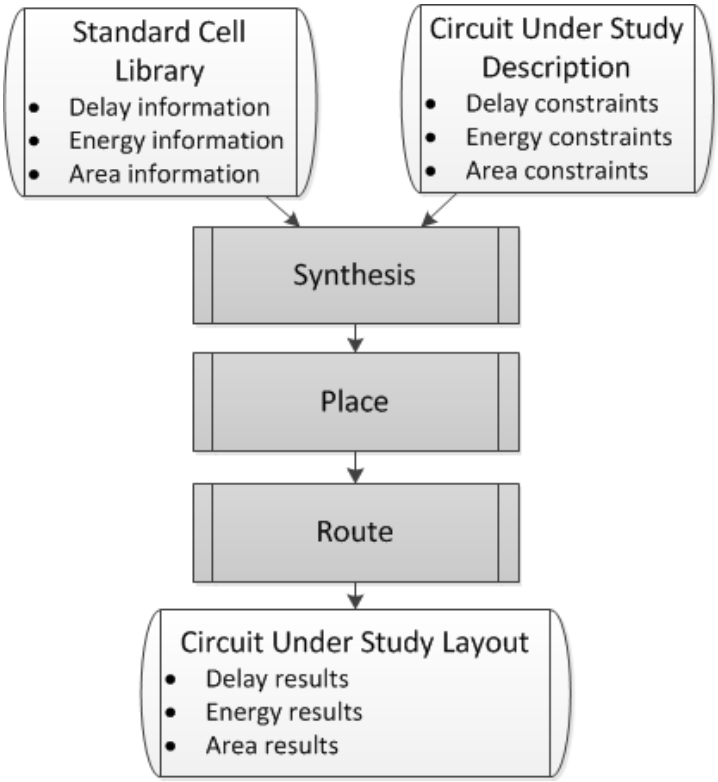


Figure 2.11: Standard cell physical design flow.

standard cell library (usually combinational and sequential cells) and also how many drives will be considered for each (the same function can be implemented with different transistor sizings so that it can be used in paths that require more or less strength). One standard cell is then associated to each of the functions with a given drive. From the logic description of these functions as well as from their cell layouts, that need to be designed one by one, the library characterization is performed in order to obtain the library files containing the geometric (including area) and also energy and delay information of each of the cells.

As explained before, note that standard cell layouts have fixed height (defined by the number of routing tracks, of height depending on the technology) and variable width, but always multiple of the technology routing pitch. All these constraints are required for later place and route. Usually, cells are divided into two zones, one containing PMOS transistors and the other containing NMOS, with polysilicon lines drawn vertically, that for compaction reasons try to be aligned respecting the inputs of the transistors to be able to share these vertical polysilicon lines between pull-up and pull-down networks. For doing that, techniques like input ordering (finding Euler Paths) and transistor folding are used (with their implications on transistor sizing).

The next step in the standard flow is logic synthesis. Having as input the circuit description (for instance in VHDL format) and the cell descriptions, the synthesizer can select the standard cells that will be used for the given circuit and generate the circuit netlist, so that the resulting circuit can meet the energy, delay and area constraints fixed by the designer.

Once the circuit netlist is synthesized, place and route is performed by taking into account the cell input and output positions (also called circuit pins) as well as energy and delay estimations for cells but also for interconnects. What is required for this step is that each of the input and output pins are on-grid, respecting the technology pitches for all layout layers so that the routing wires are always also on-grid. In a way, this is a hard constraint but it allows that each of the cells can be placed next to each other without routing overlaps, and it also diminish the

enormous amount of routing possibilities for complex circuits. Special filler cells are required to obtain the final layout, because placed standard cells can have free spaces between them that need to be filled. These spaces are also multiple of technology pitch and therefore the filler cells widths are chosen accordingly.

Finally, the layout can be evaluated to verify that energy, delay and area results meet the desired constraints for the circuit.

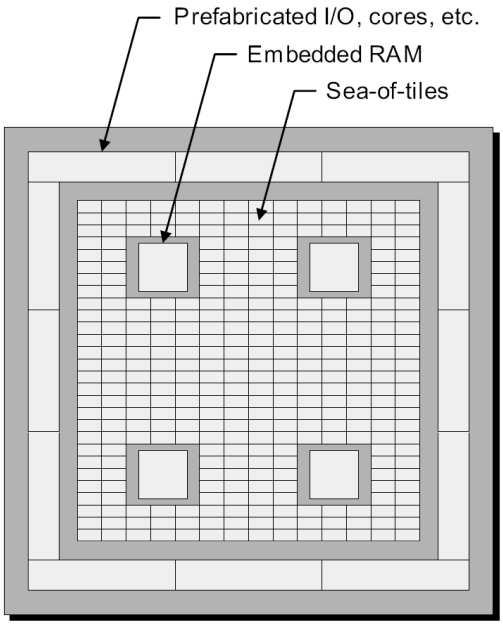


Figure 2.12: The structured ASIC concept. An array (sea) of tiles is prefabricated across the face of the chip. Structured ASICs also typically contain additional prefabricated elements, which may include configurable general-purpose I/O, microprocessor cores, gigabit transceivers, embedded (block) RAM, and so forth. [31]

2.2.3 Structured ASICs

Another class of regular layout fabrics are structured ASICs (SAs) [31]. They are constructed using an array of identical basic tiles that contain the logic (see Figure 2.12). The different types of SAs can be classified depending on their regularity granularity that is defined by the elements included in the tile. For instance, the tile can be composed by gates, multiplexers, lookup tables, buffers,

etc. The condition that has to be ensured is that all functions can be synthesized with the elements included in the tile.

2.2.3.1 NAND-based regular structure

One possibility is to consider a tile composed by a single two inputs NAND gate [48, 49] with which all functions can be implemented only configuring the interconnections.

In this proposal, transistor regularity is high, because of the use of a single basic cell (except in the zones where "dummy cells" are included). However, interconnect regularity is very poor. There are different densities of metals because of the difficulties in routability when customizing the wires for the function to be synthesized. Interconnect presents zones of higher density in "extra tracks" and "dummy cells" and zones less congested in the rest of the layout.

Regarding the physical design flow (that can be performed using the classical standard cell flow), some problems may appear in routability due to the use of the single NAND gate as basic cell. Empty "dummy cells" and "extra tracks" are possible solutions to place wires when congestion is detected. Figure 2.13 shows the resulting layout.

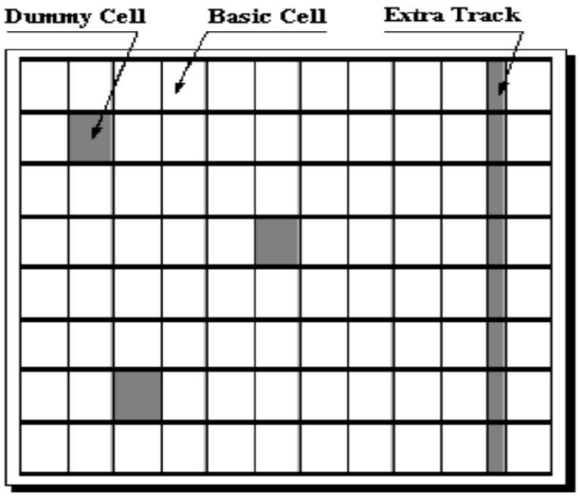


Figure 2.13: Layout with "dummy cells" and "extra tracks". [48]

2.2.3.2 Via-Configurable Logic Blocks

Another possibility is to consider a tile or basic logic element (BLE) composed by via-configurable logic blocks [50]. In particular, a BLE consists of two types of blocks: a via-configurable functional cell (VCC), containing the combinational logic needed for functions, and two via-configurable inverter arrays, containing inverters (see Figure 2.14).

Experiments performed for the 180nm technology node on 20 large benchmarks ranging from 1.5 k to 17 k nodes showed that this via-configurable design technique presents overheads in all three metrics of area, performance and power compared to standard cell design. The area increases in average by 116%. It shows 33% performance degradation and consumes 17% more power than the standard cell design.

This via-configurable logic blocks design fabric has high transistor regularity, but limited by the use of two different blocks with different compositions and layouts. Interconnect regularity is complete because all the routing channels are implemented and configured using vias.

Regarding the physical design flow, it is similar to the FPGA approach, but the configuration can be done only once. The functions are mapped to the combinational cell and the inverter cells perform the buffer connections with the surrounding tiles. In fact, the cells have prefabricated transistors, contacts and M1 wires and only the M1-M2 mask has to be configured depending on the function to be synthesized, thus reducing mask costs. In this proposal, the routing method is an important issue, as it is for all regular designs, because of the limited set of vias that can be configured [51].

2.3 Existing layout analysis tools

In this section we present the existing layout analysis tools that can be used to evaluate circuit manufacturability. We will first explain the standard DFM flow to address manufacturability issues. Then, we will give an overview of the tools

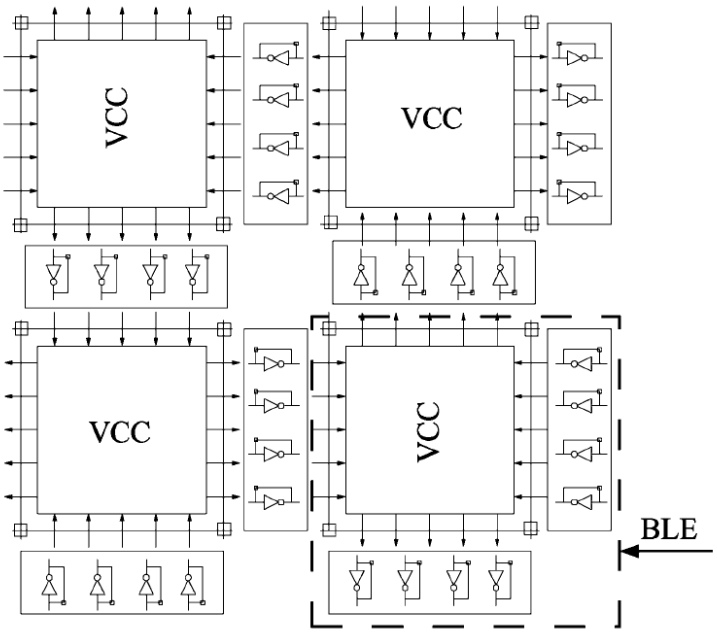


Figure 2.14: Via-configurable logic block. [50]

from Mentor Graphics, that are the most widely DFM tools used. Then, we will explain other methodologies to estimate systematic manufacturing variability, for which no commercial tools are available, and that we found in the literature. These methodologies include the evaluation of lithography proximity and coma effects variability, and also the modeling of mechanical stress variability. Finally we present how the two-dimensional Fourier Transform has been used to evaluate layout regularity.

2.3.1 Standard DFM flow

Figure 2.15 summarizes the standard DFM flow to improve designs manufacturing yield. Layout analysis tools use DFM models to estimate manufacturing yield and to identify the layout modifications required to improve manufacturability. DFM refers to the action of making modifications to the target design in order to minimize critical manufacturing operations that can cause yield losses.

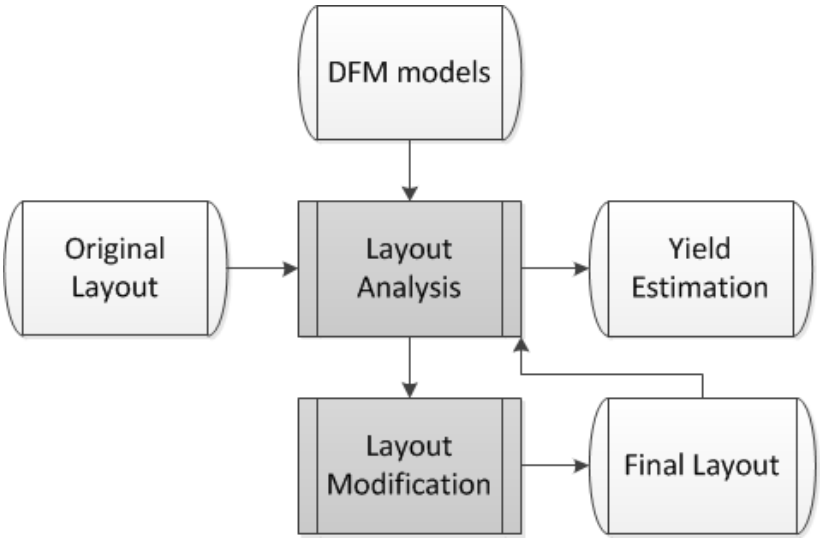


Figure 2.15: DFM standard flow.

2.3.2 Mentor Graphics DFM tools

References to Mentor Graphics tools can be found in [52]. Following the standard DFM flow, they provide a set of tools to analyze the layout, to modify it in order to improve design yield, and even to predict the printability of particular patterns by simulating lithography tools and resolution enhancement techniques.

Regarding layout analysis, Calibre YieldAnalyzer performs two kind of layout evaluations. First, the CAA (Critical Area Analysis) and then the CFA (Critical Feature Analysis). CAA is used to detect opens and shorts that can cause random yield losses associated to random particles. For this step, the distributions of random dust particles in the given process are required. On the other hand, CFA uses the design rules check (DRC) environment to check recommended rules and detect where the layout patterns can cause systematic yield losses. The set of recommended rules depends again on the particular process and does not always comprehend the whole set of possible layout patterns in a layout.

Also for layout analysis, Calibre CMPAnalyzer evaluates the impact of chemical mechanical polishing (CMP) to predict planarity variability. In that way, designers can detect thickness and resistance variability caused by decreasing

linewidths. CMP models are required for this analysis and are checked again using the DRC environment.

For layout modification, Calibre YieldEnhancer uses the results of Calibre YieldAnalyzer to modify the layout in order to improve the estimated yield results. It allows automatic via doubling, via extensions and enclosures, as well as growing polygons to a minimum size.

YieldEnhancer also includes the SmartFill algorithm which goal is to modify metal filling shapes added for CMP issues. SmartFill modifies the layout to reduce resistance variability while minimizing the number of fill shapes added.

Finally, for printability simulation, Calibre LFD (Litho-Friendly Design) in combination with Calibre Workbench is used to predict the printability of designed layouts and to find lithographic hotspots. These tools are based on models for the lithography system (lens, masks and photoresist), also including the impact of resolution enhancement techniques (like OPC) in order to simulate the resulting layout shapes. Varying particular parameters of the lithography system (like dose and focus) the tool provides a graphical representation of the final shapes in the layout giving the process variations (PV) bands. These PV bands represent the range in which the final shapes are predicted to be (see Figure 2.16). In fact, "what you see is not anymore what you get" for deep submicron technologies. However, the simulation of complex circuit layouts requires and important computational effort.

2.3.3 Systematic manufacturing variability models

Transistor channel length variations and threshold voltage variations are major sources of circuit performance unpredictability [53, 54]. We explain next the models found in the literature that can be applied to estimate their manufacturing variability.

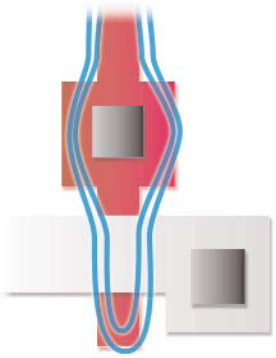


Figure 2.16: PV bands of a poly gate. Active layer in grey, poly layer in red, PVband of the poly layer in blue.

2.3.3.1 Channel length variations: proximity and coma effect

Models for systematic variations of channel length (L) variability can be found in [55] taking into account proximity and coma effects. Basically, proximity and coma effects models associate to each channel a percentage of L variation depending on the layout neighborhood on both sides of the feature to be printed. The models are based on the inspection of the layout to the left and to the right of the feature in order to define the kind of neighborhood that the channel has. They measure the distances to the first polysilicon line in each direction. Figure 2.17 depicts an example of distances n1 to the left and n2 to the right. The difference between both is that for proximity effect left side and right side distances are equivalent in their impact on variations but for coma effect they are not. The models include tables with the nominal amount of process variations for each case and the final percentage variation for L can be obtained by setting the maximum percentage range of variations. The final result is the expected L for each of the transistors on the layout. The entire circuit L distribution can then be characterized by its mean μ and its standard deviation σ.

2.3.3.2 Threshold voltage variations: mechanical stress

Models for silicon mechanical stress due to shallow trench isolation (STI) are included in the BSIM4 transistor models [56]. Transistor performance is affected

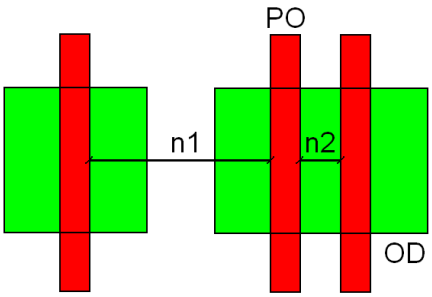


Figure 2.17: Proximity and coma effect model measurements

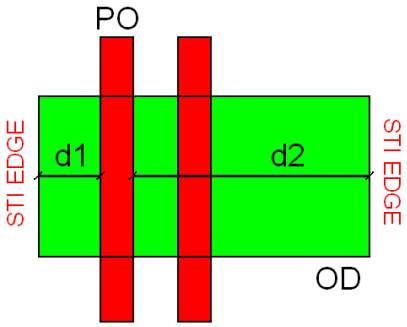


Figure 2.18: STI stress model measurements

depending on the shape of the oxide diffusion area and on the position of the device inside this area. In particular, threshold voltage (Vth) varies depending on the distances from the channel to the edge of the diffusion (where the STI begins). Figure 2.18 shows an example for the measurement of these d1 and d2 distances. The relative impact also depends on the dimensions of the transistor. Transistors with wider channel will be less affected.

2.3.4 Evaluating layout regularity

To the best of our knowledge, the only method that has already been used for this purpose is the visual comparison of a two-dimensional Fourier transform. It has been used in [8] to compare the degree of regularity of: (a) a polysilicon layer of an SRAM array, (b) logic implemented using standard cells and (c) logic implemented using a regular fabric. Since a regular layout utilizing a small number

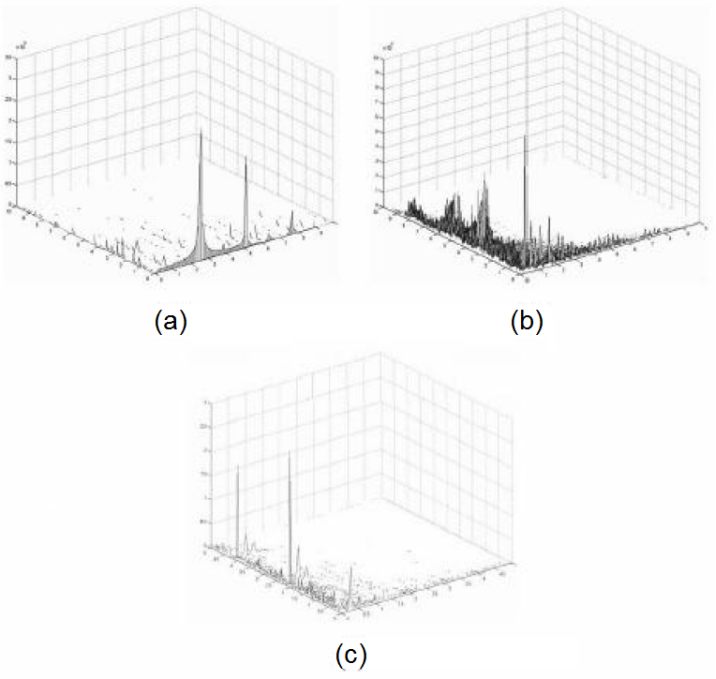


Figure 2.19: 2-D FFT spatial frequency analysis for (a) SRAM (b) standard cells (c) regular fabric [8].

of layout patterns is expected to have a finite number of frequency components the comparison is based on the number of frequency components obtained by the Fourier transform. By graphical inspection (see Figure 2.19) it can be seen that the SRAM and regular fabric layouts are more regular than the standard cells. However, the graphical inspection of the Fourier graphs does not give enough information to find out which of the two regular layouts is more regular than the other because the two frequency responses are similar.

2.4 Thesis works motivation

In this chapter we have justified the use of layout regularity to address nowadays scaling challenges from the design and also manufacturing sides. Then we have presented the existing regular layout fabrics starting from the GAs developed in the early 1980s. Finally, we have presented the way the semiconductor industry copes with manufacturing issues using different EDA tools for DFM. These are

the works that have inspired our research.

The study of the existing regular fabrics has shown how comprehensive regularity is not achieved for any of the proposals. Some are irregular at transistor level, other at interconnect level. For technologies with increasing design and manufacturing challenges, a new regular fabric with comprehensive regularity is required to maximize the regularity benefits in terms of DFM.

The study of the different physical design approaches for the existing regular fabrics has also been useful to understand the problems that have to be faced when automating this physical design. In particular, related to partitioning, placing and routing the circuits. We have seen how the physical design depends on the fabric specificities, specially on the degree of layout regularity considered. Those steps have to be addressed when developing a new regular fabric.

From the study of the different layout analysis tools, we realized that none of them measures layout regularity directly. Only the two-dimensional Fourier transform is a graphical representation giving an intuitive and qualitative measure of regularity. However it does not quantify regularity. Layout regularity can be used as a figure of merit of the design as it has been shown to improve manufacturability. Therefore there is the need of a new layout analysis tool to measure regularity.

Chapter 3

Unique contributions of the thesis

When reaching the deep submicron era, increasing manufacturing and design challenges need to be addressed. For this purpose, the thesis works are focused on the emerging DFM regular layout design techniques. The main contributions of the thesis are explained in the following subsections. First, we have proposed a new regular fabric called Via-Configurable Transistor Array (VCTA). Second, we have developed a synthesis tool to automate the VCTA regular fabric physical design. Third, we have proposed an analysis tool to measure layout regularity. The list of publications in chronological order related to these works is also given at the end of the chapter.

3.1 VCTA regular fabric

We have named our new regular fabric Via-Configurable Transistor Array (VCTA) and it is based on maximizing layout regularity focusing future technology nodes with extreme manufacturability and design issues. The objective is to maximize the regularity benefits in terms of DFM. With this purpose, VCTA uses a single basic cell containing transistors and interconnects that are configured using vias to obtain the functionality desired. Regularity constraints for the front-end and the back-end design as well as the via-configuration strategy of the fabric dictate

the VCTA basic cell design. Chapter 5 is devoted to this part of the thesis.

The first proposal with a full-adder layout was presented in the "IEEE International Workshop on Design For Manufacturability and Yield" held in conjunction of the ITC 2007 in Santa Clara, US. Then, a poster including 32-bit adders was presented in Nice, France, in the "Workshop on Process Variability: New Techniques for the Design and Test of Nanoscale Electronics" held in conjunction of the DATE 2009. In 2010, a paper including the benefits of using VCTA in terms of manufacturing variability reduction was presented in Madrid, Spain, at the "IEEE/IFIP International Conference on VLSI and System-on-Chip". In 2011, a second paper was defended in the "International Conference on Design and Test of Integrated Systems in Nanoscale Technology" in Athens, Greece, this time applying VCTA to a delay locked-loop design from an ultra wideband transceiver and demonstrating the acceleration of the design time using VCTA while maintaining circuit functionality.

3.2 VCTA automation tool

Designing with VCTA implies a specific physical design flow. Having the standard cell tools available, the first option was to adapt these tools to the physical design of the new VCTA fabric. However adapted EDA synthesis tools and algorithms were required. That is why we developed our own VCTA automation steps including transistor grouping, intra and inter-cell routing or congestion treatment. Starting from a transistor netlist, the grouping step objective is to map the transistors inside of the VCTA cells (it is equivalent to a partitioning step). Then these cells are placed and routed respecting VCTA constraints and focusing on minimizing the area of the final layouts. Chapter 6 is devoted to this part of the thesis.

For this part of the thesis, a paper was defended in 2011 in Taipei, Taiwan, in the "IEEE International SOC Conference".

3.3 FOCSI layout regularity metric tool

To satisfy the lack of an EDA tool to evaluate the benefits of layout regularity, we have developed a layout regularity metric tool named Fixed Origin Corner Square Inspection (FOCSI). No other layout regularity metrics are available. FOCSI is unique because it is able to order layouts in terms of regularity. FOCSI extracts the number of different layout generators for the selected layout layer so that, the lower this number is, the higher is the regularity of the layer. Then, the results for all the layers can be merged to calculate a complete layout regularity measure. In this way, layout designers can evaluate their layouts in terms of regularity. Finally, we have also linked FOCSI results to layout manufacturing variability to demonstrate the benefits of regular layout by proposing a variability model that makes use of the Monte Carlo analysis. FOCSI layout analysis tool can therefore be used to optimize layouts in terms of manufacturability. Chapter 7 is devoted to this part of the thesis.

For this last contribution, we have made a one year stage in the CSEM research center in Switzerland. A paper has been published for this topic in International Conference on Design, Automation & Test in Europe 2012.

3.4 Thesis dissemination

Here is the list of the contributions publications by type, indicating their citing works at the day of the presentation of the thesis. The major impact of the thesis is the VCTA regular fabric.

3.4.1 Books

- Forum 2010: M. Pons, F. Moll, J. Abella. Variations Aware Circuit Designs for Microprocessors. In Proceedings of the 2nd Barcelona Forum on Ph.D. Research in Communication, Electronics and Signal Processing, 2010, ISBN:978-84-7653-495-3.

- Forum 2009: M. Pons, F. Moll, J. Abella. Variations Aware Circuit Designs for Microprocessors. In Proceedings of the 1st Barcelona Forum on Ph.D. Research in Electronic Engineering, 2009, ISBN: 978-84-7653-398-7.

3.4.2 Conferences

- DATE 2012: M. Pons, M. Morgan, C. Piguet. Fixed Origin Corner Square Inspection Layout Regularity Metric. In International Conference on Design, Automation & Test in Europe, 2012.

- SOCC 2011: M. Pons, F. Moll, A. Rubio, J. Abella, X. Vera, A. González. Design of Complex Circuits using the Via-Configurable Transistor Array Regular Layout Fabric. In IEEE International SoC Conference, 2011.

- DTIS 2011: M. Pons, E. Barajas, D. Mateo, J.L. González, F. Moll, A. Rubio, J. Abella, X. Vera, A. González. Fast time-to-market with Via Configurable Transistor Array regular fabric: a Delay-Locked Loop design case study. In International Conference on Design & Technology of Integrated Systems in Nanoscale Era, pages 1-6, 2011.

- VLSI-SoC 2010: M.Pons, F.Moll, A.Rubio, J.Abella, X.Vera, A.González. VCTA: A Via-Configurable Transistor Array Regular Fabric. In IEEE/IFIP International Conference on VLSI and System-on-Chip, 2010, pages 335-340. Cited by:

  - DAC 2012: N.Ryzhenko, S.Burns. Standard cell routing via boolean satisfiability. In Proceedings of the 49th Annual Design Automation Conference, 2012, pages 603-612, ISBN: 978-1-4503-1199-1.

  - SBCCI 2011: V.Dal Bem, P.F.Butzen, C.E.Klock, V.Callegaro, A.I. Reis, R.P.Ribas. Area impact analysis of via-configurable regular fabric for digital integrated circuit design. In Proceedings of the 24th symposium on Integrated circuits and systems design, 2011, pages 103-108, ISBN: 978-1-4503-0828-1.

- SBCCI 2011: F.S.Marranghello, V.Dal Bem, A.I.Reis, F.Moll, R.P. Ribas. Transistor sizing in lithography-aware regular fabrics. In Proceedings of the 24th symposium on Integrated circuits and systems design, 2011, pages 97-102, ISBN: 978-1-4503-0828-1.
- ARCS 2011: F.S.Marranghello, V.Dal Bem, A.I.Reis, R.P.Ribas, F. Moll. Transistor Sizing Analysis of Regular Fabrics. In Proceedings of the 24th International Conference on Architecture of Computing Systems, 2011.
- SIM 2011: C.E.Klock, V.Callegaro, A.I.Reis, R.P. Ribas. CAD Tool for Switch Network Profiling. In 26th South Symposium on Microelectronics, 2011, pages 127-130.
- ICCD 2011: V.Dal Bem, P.F.Butzen, F.S.Marranghello, A.I.Reis, R.P. Ribas. Impact and optimization of lithography-aware regular layout in digital circuit design. In IEEE 29th International Conference on Computer Design, 2011, pages 279-284.

3.4.3 Scientific Reports

- CSEM 2011: M. Pons, C. Piguet, D. Sigg, J.L. Nagel, M. Morgan. Process variations aware standard cell libraries. In CSEM S.A. report 2011.

- FOCSI 2009: M. Pons, F. Moll, A. Rubio, J. Abella, X. Vera, A. González. FOCSI: A New Layout Regularity Metric. Internal report. (http://hdl.handle.net/2117/13385)

3.4.4 Workshops

- FETCH 2012: poster "Process Variations Aware Design", at École d'Hiver Francophone sur les Technologies de Conception des Systèmes Embarqués Hétérogènes, 2012.

- DATE Workshop 2009: poster "Addressing Process Variations with VCTA" at DATE Workshop on Process Variability: New Techniques for the Design and Test of Nanoscale Electronics, 2009.

- DFM&Y 2007: M. Pons, F. Moll, A. Rubio, J. Abella, X. Vera, A. González. Via-Configurable Transistors Array: a Regular Design Technique to Improve ICs Yield. In IEEE International Workshop on Design For Manufacturability and Yield, 2007, held in conjunction with the IEEE International Test Conference (http://hdl.handle.net/2117/1481). Cited by:

  - ARCS 2011: M.Elhoj, A.I.Reis, R.P.Ribas, F.Ferrandi, C.Pilato, F. Moll, M.Miranda, N.Woolaway, A.Grasset, P.Bonnot, G.Desoli, D. Pandini. SYNAPTIC Project: Regularity Applied to Enhance Manufacturability and Yield at Several Abstraction Levels. In Proceedings of the 24th International Conference on Architecture of Computing Systems, 2011.

  - TVLSI 2011: H.-H.Tung, R.-B.Lin, M.-C.Li, T.-H.Heish. Standard Cell Like Via-Configurable Logic Blocks for Structured ASIC in an Industrial Design Flow. In IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2011.

  - GLSVLSI 2010: Yu-Chen Chen, Hou-Yu Pang, Kuen-Wen Lin, Rung-Bin Lin, Hui-Hsiang Tung, Shih-Chieh Su. Via configurable three-input lookup-tables for structured ASICs. In Proceedings of the 20th symposium on Great lakes symposium on VLSI, 2010, pages 49-54, ISBN: 978-1-4503-0012-4.

  - ISIC 2009: Hui-Hsiang Tung, Yu-Chen Chen, Da-Wei Hsu, Shih-Jung Hsu, Sin-Yu Chen, Rung-Bin Lin. Via-configurable logic block architectures for standard cell like structured ASICs. In Proceedings of the 2009 12th International Symposium on Integrated Circuits, 2009, pages 17-20.

Chapter 4

Evaluation framework

In this chapter we summarize the evaluation framework used in the thesis. First, in section 4.1 we present the computers used for running thesis works. Then, in section 4.2 we give a list of the commercial EDA tools used as well as the methodologies and own scripts developed to interact with these tools in an automated way. Finally, in section 4.3 we detail the CMOS technologies and the standard cell libraries available as well as the circuits chosen as benchmarks and how we have evaluated them. In next chapters, circuits and layout styles explained here will be referenced.

4.1 Computation resources

The software used in this thesis have been run in several machines available thanks to the Computer Architecture and Electronic Engineering Departments at the Universitat Politècnica de Catalunya.

On one side, at the Computer Architecture Department, we have used two computer clusters. The first one, with 80 Nodes USP Xeon, each with 2 Intel Xeon processors at 2.80GHz and 2GB of RAM memory. The second one, with 73 Nodes USP Xeon Dual-Core 5148, each with 2 Intel Xeon Dual-Core processors at 2.333GHz and 12GB of RAM memory, and with 40 Nodes USP Xeon L5630

Dual, each with 2 Intel Xeon Quad-Core L5630 processors at 2.13GHz and 24GB of RAM memory.

Then, at the Electronic Engineering Department, we have used three servers. The first one, with 2 Quad Core Intel Xeon at 2.27GHz and 24GB of RAM memory. The second, with 2 Dual Core Intel Xeon at 3.20GHz and 4GB of RAM memory. The third, with an Intel Pentium D at 3.00GHz and 2GB of RAM.

Table 4.1: EDA tools

Function	Tool	Vendor
Circuit synthesis	Encounter RTL Compiler [57]	Cadence
Place and Route	Soc Encounter [58]	Cadence
Schematic generation	Virtuoso Schematic Editor [59]	Cadence
Layout generation	Virtuoso Layout Suite [60]	Cadence
Library characterization	Encounter Library Characterizer [61]	Cadence
LVS check	Calibre nmLVS [62]	Mentor Graphics
DRC check	Calibre nmDRC [63]	Mentor Graphics
Parasitic extraction	StarRC [64]	Synopsys
Circuit simulation	HSPICE [65]	Synopsys

4.2 Electronic design automation tools

4.2.1 Commercial tools

For circuit design and simulation, we have used commercial tools from different vendors including Cadence, Mentor Graphics and Synopsys (see Table 4.1). In some cases, for interacting with design tools, we learned how to use Cadence SKILL language and Mentor Graphics Standard Verification Format (SVRF). In particular, SKILL was used in the layout and schematic generation automation for VCTA (see chapter 6) and SVRF was used to define design rule checks to evaluate proximity and coma effects using Calibre nmDRC (explained in chapter 2).

4.2.2 Data treatment

For data treatment we have combined bash scripts [66] and MATLAB scripts [67]. In particular we treated HSPICE simulation outputs with bash (using its parsing capabilities) and then made calculations on them using MATLAB.

Bash scripts were also used for submitting jobs (HSPICE simulations but also our own C codes) into cluster queue nodes.

4.2.3 C programming

For developing and debugging our C codes (for VCTA automation in chapter 6 and for FOCSI metric in chapter 7) we have used gcc and gdb tools.

4.3 Benchmark circuits and evaluations

4.3.1 Circuits

In the thesis we have worked with the following circuits:

- 32-bit binary adders, a common block in digital designs
- a delay-locked loop, used for analog designs
- ISCAS'85 benchmarks, including several combinational circuits

4.3.1.1 32-bit binary adders

Binary adders structure can be divided in 3 logical blocks (Figure 4.1). First, in the Bitwise PG Logic, propagation and generation are calculated for each of the bits of the adder. Second, in the Group PG Logic, the carry calculation is performed in different ways depending on the adder considered. In our works, we have worked with 32-bit Carry-Ripple adder (CR32), Carry-Lookahead adder (CLA32) and Kogge-Stone adder (KS32). The logical cells required for this second part are shown in Figure 4.2 and Figure 4.3. Finally, in the Sum Logic, the resulting sum and carry-out are calculated [68].

4.3.1.2 Delay-Locked Loop

The Delay-Locked Loop (DLL) architecture that we have used is the one presented in Fig. 4.4. The Voltage Controlled Delay Line (VCDL) is composed of a chain of identical cells connected in series. The delay cells are based on a current

starved inverter where its delay is controlled by means of an external voltage, and a level shifter to convert the output back to rail-to-rail. The delay cell, and hence the whole VCDL, are implemented differentially to improve performance [69].

In terms of energy consumption and jitter, the dominant block of the DLL is the VCDL [70]. Careful design of the delay cells of the VCDL is needed to reduce

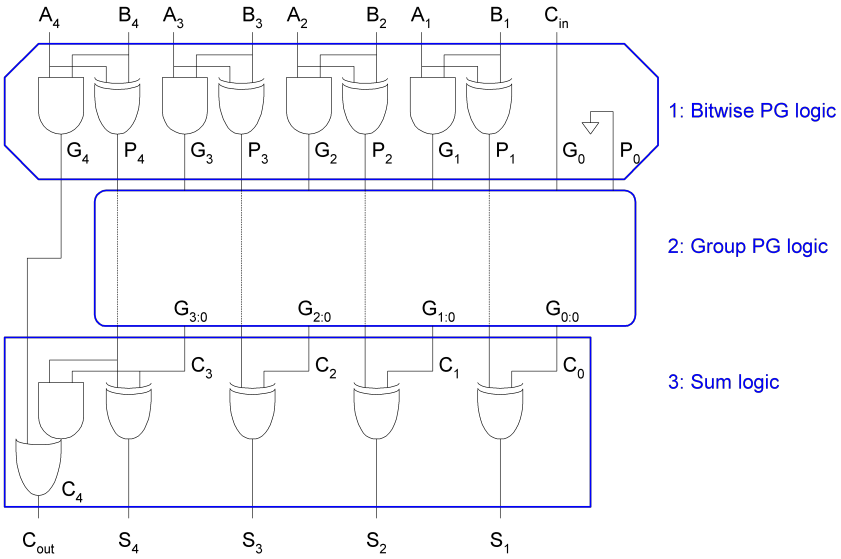


Figure 4.1: General binary adders structure. The structure for a 4-bit adder is shown to illustrate the 3 logical blocks.

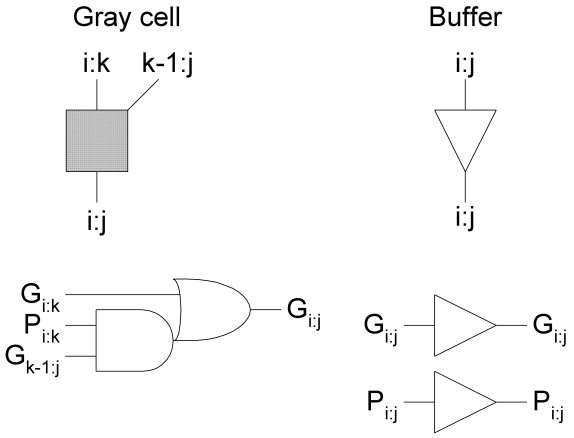


Figure 4.2: Cells needed for the Group PG Logic (CR32, CLA32 and KS32).

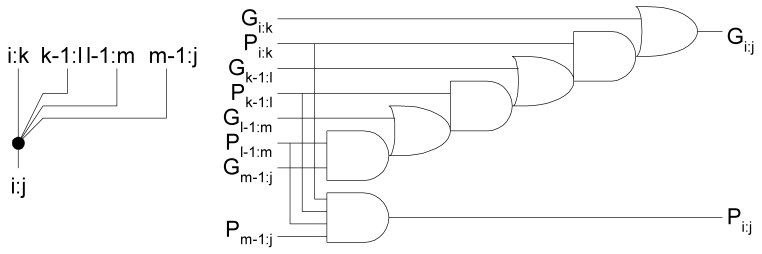


Figure 4.3: Cells needed for the Group PG Logic (CLA32).

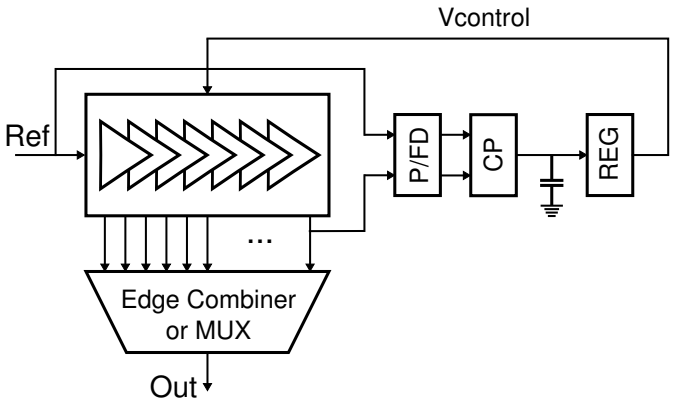


Figure 4.4: DLL architecture. P/FD = Phase/Frequency Detector, CP = Charge Pump, REG = Regulator.

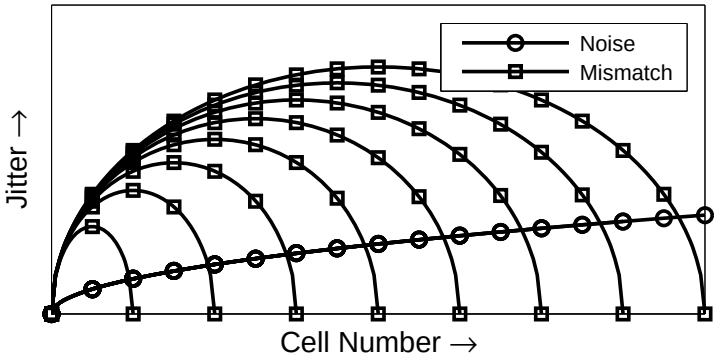


Figure 4.5: Jitter as a function of the DLL size for noise and mismatch.

the total energy consumption but also to reduce the jitter, which is mainly related to the cell mismatch (Figure 4.5). Other sources of jitter like noise or the voltage ripple in the control voltage (Vcontrol) from the regulator have less impact than mismatch.

Table 4.2: ISCAS'85 circuits description

ISCAS'85	Description
c17	6 NAND gates (test purposes)
c432	27-channel interrupt controller
c499/c1355	32-bit single-error-correcting
c880	8-bit arithmetic logic unit
c1908	16-bit single-error-correcting and double-error-detecting
c2670	16-bit arithmetic logic unit and controller
c3540	8-bit arithmetic logic unit
c5315	9-bit arithmetic logic unit
c6288	16x16 multiplier
c7552	32-bit adder/comparator

4.3.1.3 ISCAS'85 benchmarks

ISCAS'85 benchmark circuits where presented in [71]. They include several combinational circuits that are summarized in Table 4.2.

4.3.2 Technology nodes

During the thesis we have had the opportunity to work with the following technology nodes (in chronological order):

- commercial 90 nm technology available at the Electronic Engineering Department of the Universitat Politècnica de Catalunya at the beginning of the thesis
- 45 nm technology NCSU Free PDK [72]
- commercial 65 nm technology available at CSEM S.A during the last year of the thesis in Switzerland

This is the reason why, the thesis evaluations have been done in different technologies. However, the results and trends observed are not technology dependent. Therefore, we have not repeated the whole set of evaluations for each single technology.

4.3.3 Layout versions

Initially we have evaluated VCTA layouts developed manually in the 90 nm technology node:

- For digital design, we have implemented the 32-bit binary adders and compared them to their standard cell versions (STD90). For this, we have used the public standard cell layouts provided in [73] that offers a complete set of portable CMOS libraries that has been used for research projects such as the 875,000 transistors StaCS superscalar microprocessor and 400,000 transistors IEEE Gigabit HSL Router.

- For analog design, we have implemented the DLL design and compared it to its full custom version (FC). For the full custom version we have used the one designed by our colleague Enrique Barajas, from the Electronic Engineering Department at the Universitat Politècnica de Catalunya [70].

For the VCTA layouts developed later using the VCTA automation tool we have worked in the 45 nm technology node. In this case we have implemented the ISCAS'85 benchmarks and compared them to their standard cell versions using the standard cell layouts (STD45) generated from the OSU library [74].

Finally, in order to evaluate the FOCSI layout regularity metric, we have also implemented the ISCAS'85 circuits in the 65 nm technology node. In this case, we have used two different versions of standard cells:

- a commercial standard cell library (STD65) available at CSEM S.A

- a process variations aware library that we have developed at CSEM S.A. The focus of the new library has been to improve standard cells layout regularity of a library containing a reduced set of standard cells [75]. The library contains 24 cells with 15 logic gates, 6 latches, 2 flip-flops and 1 full-adder. In that way the number of possible layout neighborhoods is also reduced as fewer cell combinations can be found in the complete layout. In more detail,

all layout layers in the cells of the new library are one-dimensional to ensure better manufacturability. Moreover, to increase regularity, transistor sizing is performed using fingering so that all the individual transistors are the same size but higher drives can be obtained by connecting them in parallel. Finally, contacts are doubled when possible to increase reliability. We will refer to this new library as the Robust standard cell library (Robust65).

4.3.4 Evaluations

In Table 4.3 we summarize the circuit used in the thesis and the evaluations performed to compare the layout versions, as well as the technologies used and the chapters where they are referenced.

Table 4.3: Benchmark circuits and evaluations summary

Circuit	Layout version	Technology	Evaluations	Chapter
32-bit adders	VCTA	commercial 90nm	Delay, Energy, Area, Proximity effects	5, 6
32-bit adders	STD90	commercial 90nm	Delay, Energy, Area, Proximity effects	5, 6
DLL	VCTA	commercial 90nm	Delay, Energy, Area, Jitter	5
DLL	FC	commercial 90nm	Delay, Energy, Area, Jitter	5
ISCAS'85	Robust65	commercial 65nm	Layout Regularity and Variability	7
ISCAS'85	STD65	commercial 65nm	Layout Regularity and Variability	7
ISCAS'85	VCTA	Free PDK 45nm	Delay, Energy, Area, Layout Regularity and Variability	6, 7
ISCAS'85	STD45	Free PDK 45nm	Delay, Energy, Area, Layout Regularity and Variability	6, 7

Chapter 5

VCTA regular fabric

Existing regular fabrics presented in chapter 2 are not completely regular. In fact, some are irregular at transistor level, and others at interconnect level. In future nanometer technologies, more comprehensive regularity-based techniques will be required to deal with the increasing design and manufacturing challenges. That is why we propose a new regular layout style called Via-Configurable Transistor Array (VCTA) that maximizes regularity at both device and interconnect levels. The objective is to maximize layout regularity benefits in terms of the reduction of manufacturing and design. This is the motivation and origin of our research.

The VCTA regular fabric proposal is based on the use of a single basic cell (BC) that will be repeated along the layout. In this chapter, in section 5.1 we first explain the VCTA physical design using this single BC, detailing the BC characteristics (front-end and back-end). Then, in section 5.2 we show how the BC parameters impact the design, in particular regarding the area of the final layout and its routability, and also for energy and delay. In section 5.3 we show the evaluations of VCTA layouts developed manually for 32-bit adders as well as a Delay-Locked Loop (DLL). Finally, in section 5.4 we conclude the chapter.

5.1 VCTA physical design

5.1.1 Maximizing layout regularity

The VCTA regular fabric is based on a single basic cell layout (BC) that can synthesize different functions. To generate the VCTA layout, identical BCs are placed next to each other in rows and columns, so that all BCs have the same layout neighborhood. The only difference between BCs are the vias to configure the different functions and how the BCs are interconnected (more details are given in next sections). In that way, VCTA layout maximizes regularity at cell level, also referred as macro-regularity.

To synthesize different functions, the BC contains a via-configurable interconnect grid (VC) as well as a transistor array (TA), from which come the name of the VCTA fabric. Making use of the available transistors in the BC and configuring the vias of the interconnect grid, we can generate with the single BC the functions required for the circuits to be designed. We explain next the transistor array (BC front-end) and the interconnect grid (BC back-end). The BC layout is in this case designed to maximize regularity at pattern level, also referred as micro-regularity, so that for small areas of layout, VCTA also reduces the number of layout pattern combinations.

5.1.2 Basic cell Front-end design

The BC front-end includes PMOS and NMOS transistors to be able to synthesize the pull-up and pull-down networks of logic functions. It has the following characteristics:

- PMOS transistors are at the top of the BC, and NMOS transistors are to the bottom, aligned vertically, with polysilicon lines drawn horizontally. We took this decision to make pull-up and pull-down networks independent in terms of transistor ordering. Polysilicon lines do not need to be shared between the two networks. Doing so, we give flexibility to the BC. In

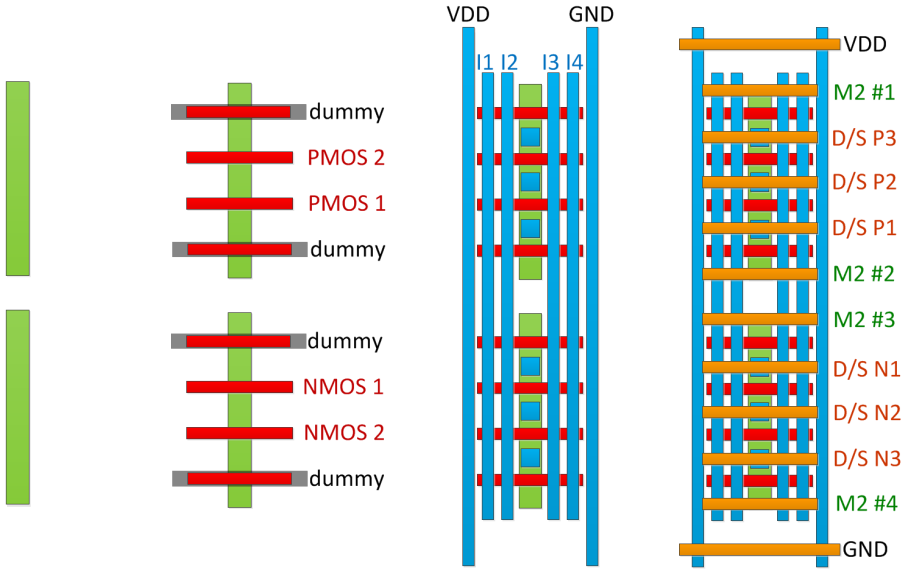
fact, pull-up and pull-down networks of different functions can be grouped together in a single BC (if having enough resources, like enough transistors available, even different functions can be grouped on a single BC).

- Transistors in each case (PMOS or NMOS) share the same oxide diffusion in order to increase transistor density and thus reduce the area of the BC. With this constraint VCTA transistors are connected in series by default. However, parallel connections can be performed by properly setting up vias using the via-configurable structure, as it will be explained in next subsections.

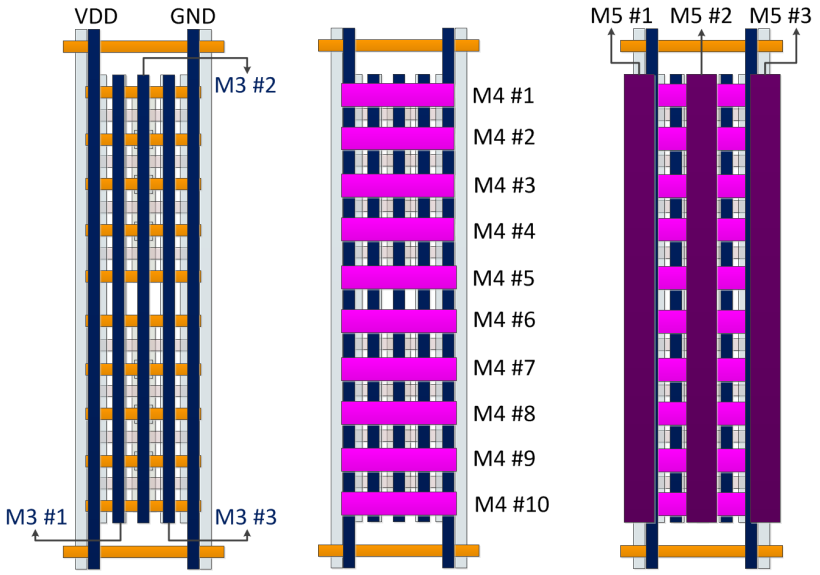
- In order to maximize regularity at transistor level, we decided to force all transistors to have the same dimensions (same width W and minimum channel length L) and also to have the same number T of PMOS and NMOS in the VCTA cell. These restrictions impose some constraints, first, on the sizing capabilities of the functions in VCTA style, and second, in the balance of the pull-up and pull-down networks, as PMOS transistors usually require higher sizes. We can obtain transistors with different number of fingers, resulting on sizings that can only be multiples of the width W of the single transistor. Available sizings goes from W to $(T \cdot W)$ in steps of W.

- To further reduce process variations, we add 2 dummy transistors (the ones on the upper and lower extremes of the PMOS and NMOS transistor arrays). In this way we avoid possible variations in drains/sources between two polysilicon gates and drains/sources at the edges with only one gate on one side.

To illustrate the front-end design of the VCTA regular fabric, Figures 5.1a and 5.1b show the transistor array where $T = 2$. In the figure there are 4 transistors of each type, but only 2 can be used (PMOS 1 and 2, and NMOS 1 and 2). The other 2 transistors are the dummy transistors (shaded in Figure 5.1b).



(a) Oxide diffusion (b) Adding poly (c) Adding metal 1 (d) Adding metal 2



(e) Adding metal 3 (f) Adding metal 4 (g) Adding metal 5

Figure 5.1: VCTA basic cell layout.

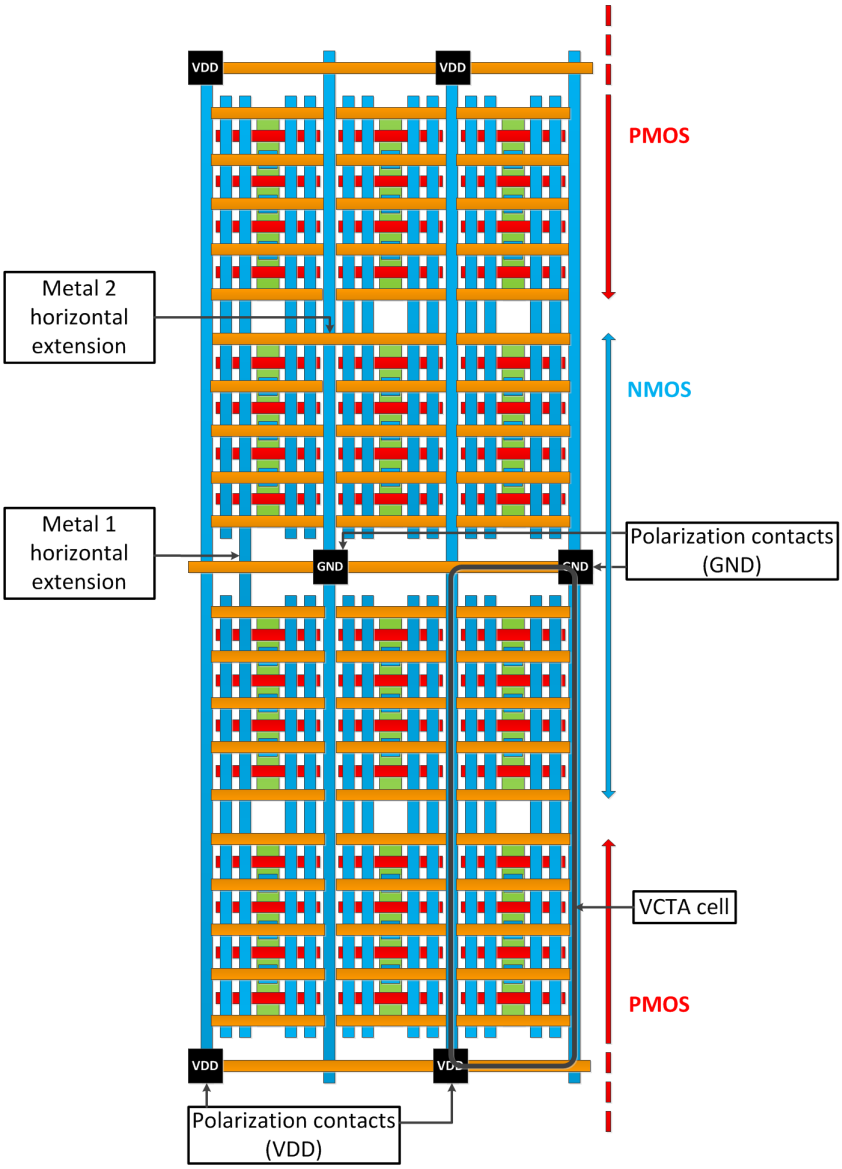


Figure 5.2: Place and interconnect grid structure for inter-cell routing and ground and power supply. In this example, we show 6 BCs, with up to 2 metal layers (vertical lines are metal 1, and horizontal lines are metal 2). Black squares are polarization contacts that are placed in the ground and power supply lines that are shared between BCs using symmetry. BCs are placed so that PMOS transistors of neighboring BCs are next to each other, and the same for NMOS transistors. P wells and N wells are shared between adjacent BCs when placing.

5.1.3 Basic cell Back-end design

The BC back-end is an interconnect grid that uses one-dimensional parallel metal lines alternating from horizontal to vertical direction from one layer to the next. In order to ensure interconnect regularity the whole grid is already in place in the BC. The configuration is done using vias (see next section).

To illustrate the back-end design of the VCTA regular fabric, Figures 5.1c to 5.1g show metal 1 to metal 5 layers of the BC. The number of layers is only limited to the available metal layers in the technology node considered.

In general, all metal layers available will be used for intra and inter-cell routing (intra-cell routing refers to the connections of the nodes inside of the BC and inter-cell routing refers to the connections from one BC to another BC). However, metal 1, metal 2 and metal 3 have other uses that we detail next.

For metal 1 layer:

- The number of metal 1 lines will determine the number of inputs that can be connected to the gates of the transistors in the BC, thus determining the maximum number of inputs that the functions mapped inside of the BC can have. In Figure 5.1c we can see how 4 of the 6 metal 1 lines can reach polysilicon lines. These will be the number of available inputs (I1 to I4 in the Figure).

- The 2 metal 1 lines at the right and left edges of the BC are shared between horizontal adjacent cells and are used to power supply (VDD) and ground (GND) distribution. In this way, we ensure that neighboring cells can be abutted horizontally. Note that these 2 metal 1 lines are also extended vertically between vertical adjacent cells forming regular VDD and GND gridded networks along all BCs (see Figure 5.2).

- Unused metal 1 lines that are not connected to gates can be used for inter-cell routing, by extending them vertically (see Figure 5.2).

- Metal 1 layer is necessary to connect drains and sources of the transistors. Contacts are already placed between metal 1 and oxide diffusion to avoid different stress effects in the transistors and also to avoid the need to configure BC contacts to synthesize the different functions (only vias need to be configured).

For metal 2 layer:

- The metal 2 lines placed over the drains and sources of transistors (D/S P1 to D/S P3 and D/S N1 to D/S N3 in Figure 5.1d) are exclusively used to configure them. That is why they are already connected to metal 1 over drains and sources using vias. These decision is made to reduce the number of vias to be configured when synthesizing a new function and because all drains and sources will always require to be connected. Even if the transistor associated is not used in the BC function, drain and source are connected together to short the transistor and avoid extra power consumption.
- Metal 2 lines at top and bottom edges are used for VDD and GND distribution (as were metal 1 lines at the right and left edges) and ensure the abutment of BCs in vertical direction.
- The rest of metal 2 lines (up to four in Figure 5.1d, named $M2\#1$ to $M2\#4$) can be used for intra-cell routing and also can be extended horizontally for inter-cell routing (see Figure 5.2).

For metal 3 layer:

- 2 metal 3 lines are connected to VDD and GND to allow the connection of drains and sources of the transistors to VDD or GND through the metal 2 lines devoted to configure drains and sources. These metal 2 lines cannot reach the right or left edges of the BC and therefore cannot be connected to metal 1 lines for VDD or GND (otherwise, metal 2 lines for drains and sources configuration will be connected to the neighbor metal 2 lines when

placing 2 BCs next to each other, forcing drains and sources of neighboring BCs in the same row to be connected to the same node). Therefore the 2 metal 3 lines devoted to VDD and GND cannot be shared in the horizontal direction between neighbor BCs. However, similarly to metal 1 lines for VDD and GND, these metal 3 lines are extended vertically forming the VDD and GND regular grid (see Figure 5.2).

- The rest of metal 3 lines (three in Figure 5.1e) are usually used for inter-cell routing by extending them vertically (like can be done for unused metal 1 lines).

Upper metal layers (metal 4 and 5 and so on) are exclusively devoted to routing and are not required for VDD and GND distribution.

5.1.4 Basic cell configuration

The BC front-end and back-end need to be configured to obtain the desired functionality. First the function is mapped in the serial transistors of the transistor array and then intra-cell routing is performed with the via-configurable interconnect structure. Finally inter-cell routing is performed by the extension of the metal lines across the borders of the BCs as explained in previous section.

To configure the transistors, metal 1 is used to reach the gates, metal 2 is used to reach the drains and sources, and metal 3 ensures the availability of VDD and GND. Then, the rest of the metal layers (including metal 1, metal 2 and metal 3 if lines are still unused) are used for intra-cell routing by configuring vias.

To illustrate the BC configuration, Figure 5.3 shows how a NAND function is synthesized using VCTA.

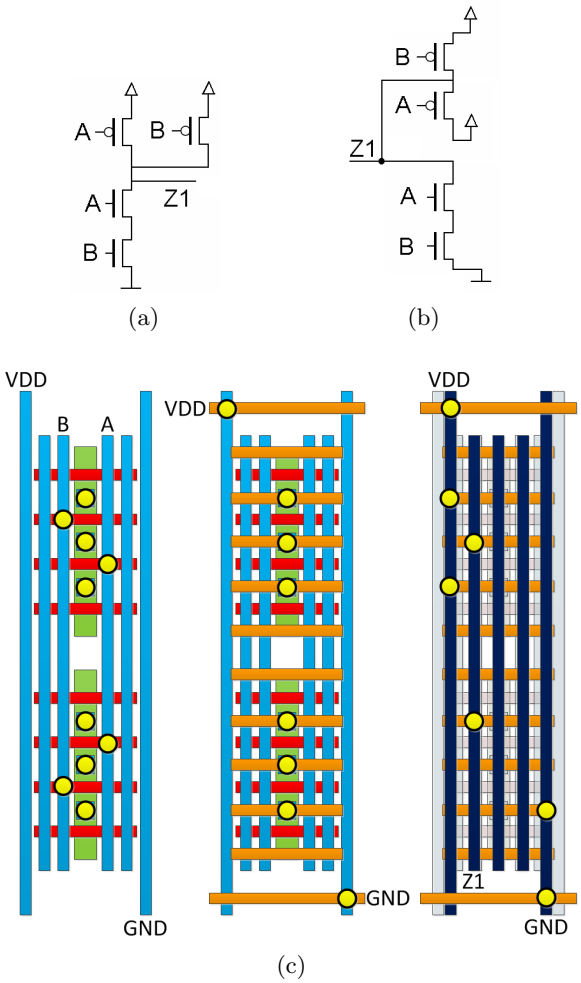


Figure 5.3: NAND: (a) Schematic (b) BC schematic with 2 PMOS and 2 NMOS transistors available (dummy transistors are not depicted) and (c) BC layout, with $T = 4$ (corresponding to 2 available PMOS and NMOS transistors), 4 possible inputs using metal 1 (2 metal 1 lines are used for ground GND and power supply VDD) and up to 3 metal layers in this case (from metal 1 to metal 3). Yellow circles are contacts and vias.

5.2 VCTA Basic cell impact on design

5.2.1 Basic cell parameters

The BC can have different resources depending on its chosen implementation that is defined by the following parameters:

- the number of transistors T for PMOS and NMOS

- the transistor channel width sizing W (channel length L is the minimum depending on the technology)
- the number of metal layers considered for routing N
- the number of metal lines in each layer M_j (e.g., the number of metal 1 lines M_1, that determines the number of inputs of the BC as transistor gates can only be accessed from this metal layer)

5.2.2 Basic cell impact on area and routability

The VCTA layout area is determined by the BC area, which depends on its parameters and on the technology design rules and contact and via geometries.

5.2.2.1 Basic cell width

The BC width W_{BC} is determined by vertical metal 1 and metal 3 lines area:

- For metal 1, the resulting BC width considering metal 1 structure W_{BCM1} follows Equation (5.1), that is graphically illustrated in Figure 5.4. $M1W$ is the metal 1 line minimum width, $M1S$ is the metal 1 line minimum spacing, CW is the metal 1 to oxide diffusion contact width, CS is the minimum distance between contacts and metal 1 lines, and $VIA1E$ is the metal 1 enclosure required for metal 1 to metal 2 via 1. As VDD and GND lines are centered at the right and left edges of the BC, only half of these lines contributes to the BC width. Moreover, when placing a via 1 to connect to the upper metal 2 lines, if extra metal 1 enclosure is required, it has also to be considered as the minimum spacing between metal 1 lines will not anymore be enough in order to remain design rule check error free when including vias for the configuration of the BC. This will depend on the particular geometries of the via 1. W_{BCM1} has to be greater or equal to the resulting calculation as this is the minimum width needed for the metal 1 structure considered. Note that CS, is ideally the sum of $M1S$ and $VIA1E$, as it is also the minimum distance between metal 1 polygons,

so that W_{BCM1} can be also written like in Equation (5.2). We distinguish the distance CS because it will be used next. For metal 1 there is a layout irregularity in the center of the BC due to contacts.

$$W_{BCM1} \geq ((M_1-1)\cdot M1W)+((M_1-2)\cdot(M1S+VIA1E))+(CW+2\cdot CS) \tag{5.1}$$

$$W_{BCM1} \geq ((M_1-1)\cdot M1W)+(M_1\cdot(M1S+VIA1E))+CW \tag{5.2}$$

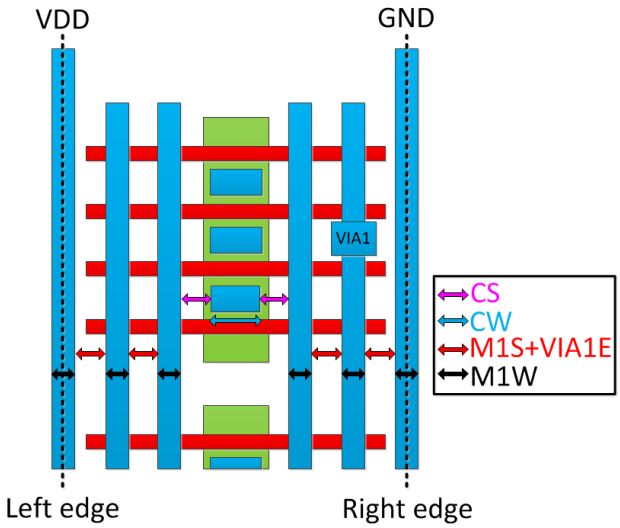


Figure 5.4: Basic cell width considering metal 1 layer. The PMOS part of the cell is zoomed to allow better visibility of spacings and widths. A possible metal 1 polygon added for the placement of a via 1 is depicted to show the need of adding the enclosure of the via 1 to the minimum spacing design rule between metal 1 lines.

- For metal 3, the width of the BC W_{BCM3} follows Equation (5.3), where $M3W$ is the metal 3 line width, $M3S$ is the metal 3 line minimum spacing, $VIA3E$ is the metal 3 enclosure required for via 3, and SH is the number of shared metal 3 lines at the right and left edges of the BC. SH can take values 0 (when no lines are shared) or 2 (when both lines are shared at each edge of the BC). Equation (5.4) and Equation (5.5) show the resulting

W_{BCM3} for the two possibilities. A value of 1 is not possible as it will not respect BC symmetry and will cause irregularities when abutting a BC to another in the horizontal direction. If it is the case, the shared metal 3 lines will act only as dummy lines, to maintain regularity when abutting a BC to another. In fact, they are not accessible from any of the BC horizontal metal lines, that can never reach the right nor left edges to avoid being shorted to horizontal lines in neighbor BCs. Figure 5.5 shows the two possibilities and illustrate Equations (5.4) and (5.5). Note that when not sharing metal lines at the edges half the space between metal lines needs to be added in each edge so that metal lines in neighbor BCs remain at the same distance. For metal 3 the pitch is therefore $M3W + M3S + VIA3E$.

$$W_{BCM3} \geq ((M_3 - (SH \cdot 0.5)) \cdot M3W) + ((M_3 - 1 + (2 - SH) \cdot 0.5) \cdot (M3S + VIA3E)) \tag{5.3}$$

$$W_{BCM3} \geq (M_3 - 1) \cdot (M3W + M3S + VIA3E) \; when \; SH = 2 \tag{5.4}$$

$$W_{BCM3} \geq M_3 \cdot (M3W + M3S + VIA3E) \; when \; SH = 0 \tag{5.5}$$

- The final BC width W_{BC} is the width fulfilling the equations from vertical metal 1 and metal 3 layers, always being multiple of the manufacturing grid (2.5 nm or 5 nm for the technology nodes available in our works). W_{BC} follows Equation (5.6). Each of the widths and spacings involved in the calculations also require to be multiple of the manufacturing grid. If these widths and spacings are the minimum design rules from the technology, this last condition is already fulfilled. However, the spacings between lines in the less restrictive metal layer need to be adapted by increasing it to reach the final W_{BC}, and respecting that they remain multiple of the manufacturing grid (an example is shown at the end of the section). Usually the distance

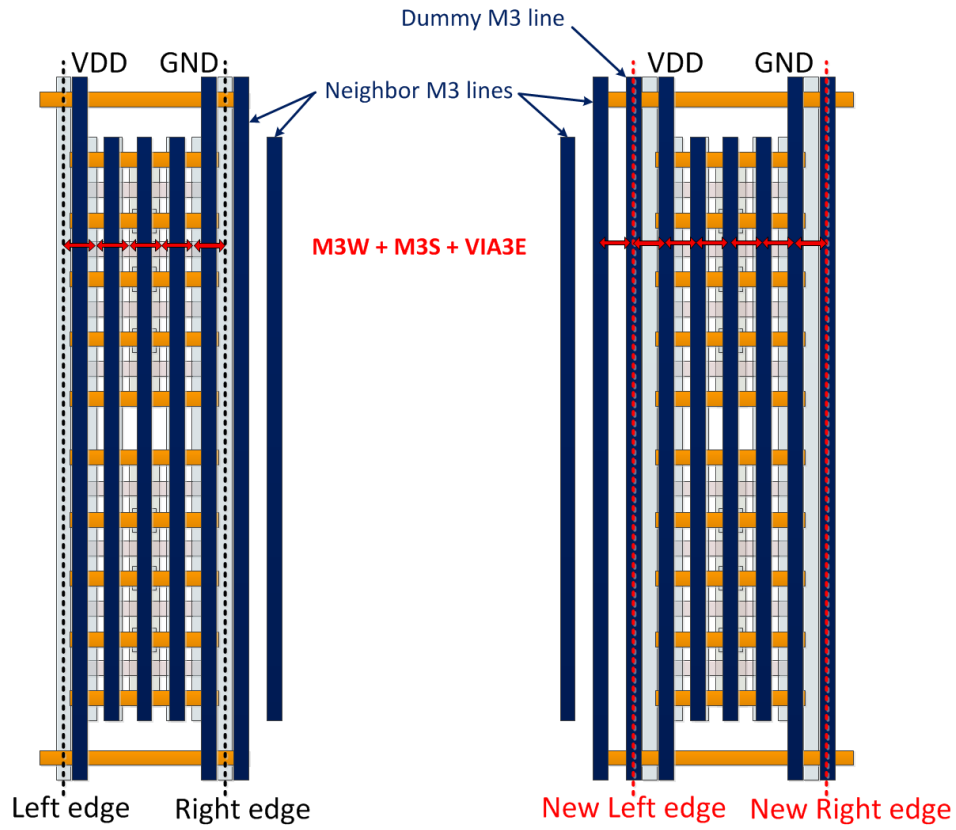


Figure 5.5: Basic cell width considering metal 3 layer. To the left is shown the metal 3 layer when lines are not shared at the edges. To the right is shown the opposite case when the distance between edges is higher. For both cases, all metal 3 lines are at the same pitch.

between metal lines will be higher than the minimal design rules because of regularity constraints.

$$W_{BC} \geq max(W_{BCM1}, W_{BCM3}) \tag{5.6}$$

- For the rest of the vertical metal layers (metal 5 and so on), we maximize the number of lines that fit in the available width fixed by metal 1 and metal 3 in order to maximize the routing capabilities without affecting the area. For this, we need the metal width and spacing design rules for those layers (an example is given next).

5.2.2.2 Basic cell height

The BC height H_{BC} is determined only by horizontal metal 2 lines area:

- For metal 2, the spacing between lines is determined by the distance between contacts to drains and sources of the transistors because metal 2 lines are placed over these contacts. This distance between contacts depends on the minimum spacing between polysilicon lines in the active zone POS and the length of the transistors L. The number of metal 2 lines M_2 depends on the number of transistors T for PMOS and NMOS, on how many metal lines are added for routing, and also adding the 2 lines for VDD and GND lines at the up and bottom edges of the BC. M_2 follows Equation (5.7) with $NM2up$, $NM2center$ and $NM2down$ as the number of metal 2 lines added for routing in the BC at the top of the PMOS transistor array, between PMOS and NMOS transistors arrays, and at the bottom of the NMOS array, respectively. The BC height H_{BC} follows Equation (5.8). In this case, as all distances considered are multiple of the manufacturing grid there is no need to verify this multiplicity. Figure 5.6 illustrates Equation (5.8). The pitch of metal 2 lines is $POS + L$.

$$M_2 = 2 \cdot (T + 1) + NM2up + NM2center + NM2down + 2 \tag{5.7}$$

$$H_{BC} = (M_2 - 1) \cdot (POS + L) \tag{5.8}$$

- The rest of horizontal metal layers are adapted to the metal 2 restrictions. We made this decision to maintain maximum transistor compaction. We do not allow the distance between transistors to be increased and therefore we do not allow the distance between metal 2 lines to be modified. The reverse calculation is done in this case. The number of metal lines for the remaining horizontal metal layers are calculated to find the maximum number of them that fits in the BC height H_{BC}. What needs to be ensured is that all metal lines are at the same distance, and that BCs can be placed next to each other without irregularities in the edges. Like for vertical metal layers starting at metal 5, for this, we need the metal width and spacing design rules for those layers (an example is given next). We maximize the number of metal lines in these layers to maximize the routing resources available in the BC.

5.2.2.3 Basic cell example

To verify the impact of BC parameters on its area (and therefore on the VCTA layout area), we detail next the width and height calculations for the VCTA BC developed for the thesis in the 45 nm technology node using the NCSU Free PDK. The design rules required are given in Table 5.1. For this BC, $H_{BC} = 4.18\mu m$ and $W_{BC} = 1.395\mu m$ and the BC parameters are the following:

- T=6 PMOS and NMOS transistors
- W=500 nm, L=50 nm for the sizing of the transistors
- N=5 (up to metal 5 layer)
- M_1=8, M_2=23, M_3=10, M_4=14, M_5=4 for the number of metal lines in each layer

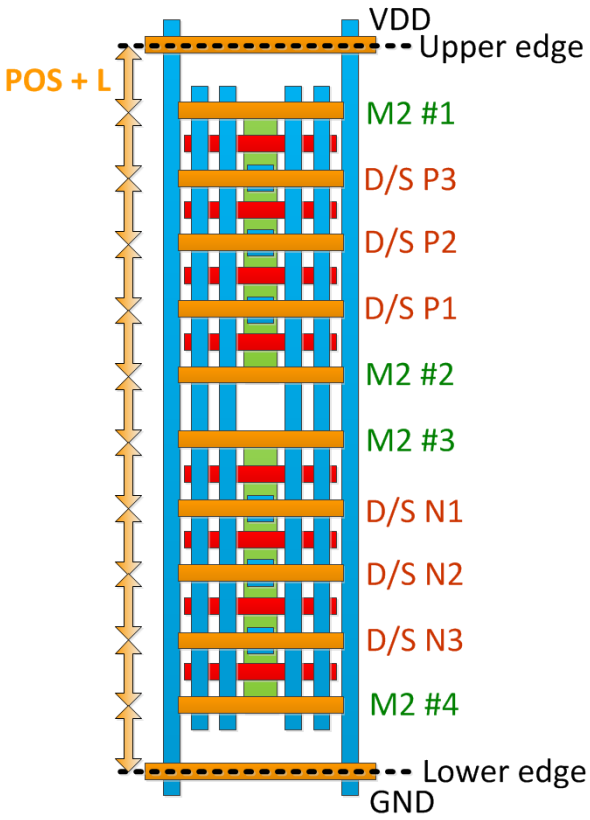


Figure 5.6: Basic cell height considering metal 2 layer. The BC with $T = 2$ transistors is shown. $M2\#1$ to $M2\#4$ are the $NM2up$, $NM2center$ and $NM2down$ added metal 2 lines for routing.

Table 5.1: NCSU Free PDK design rules

Parameter	Size (μm)
POS	0.140
M1W	0.065
M1S	0.065
VIA1E	0.035
CW	0.135
M3W	0.070
M3S	0.070
VIA3E	0.000
M4W	0.140
M4S	0.140
M5W	0.140
M5S	0.140

Regarding the width of the BC, considering metal 1 using Equation (5.2) we obtain $W_{BCM1} \geq 1.390\mu m$. For metal 3, the pitch is 0.140μm (no via 3 enclosure is required). Then, considering 10 metal lines, if sharing lines at the edges we obtain $W_{BCM3} \geq 1.260\mu m$ using Equation (5.4). Otherwise, when not sharing lines at the edges, we obtain $W_{BCM3} \geq 1.400\mu m$ using Equation (5.5). That is why, to minimize the area of the BC, the BC in the 45 nm technology node was designed using shared metal 3 lines. Therefore, following Equation (5.6), we need $W_{BC} \geq 1.390\mu m$ as the most restrictive layer is metal 1. However, $1.390\mu m$ minimum BC width requires a metal 3 pitch higher than 0.154μm, that is higher than the minimum 0.140μm pitch. Maintaining the minimum metal 3 width $M3W$, this translates into increasing the metal 3 spacing. Lines need to be more separated than the minimum spacing of 0.070μm for $M3S$. In particular, they need to be at distance higher than 0.084μm to maintain regularity. We therefore have to fix this distance to 0.085μm to be multiple of the 2.5 nm manufacturing grid, leading to a pitch of 0.155μm and to $W_{BC} = 1.395\mu m$. At his turn, this forces to adapt the metal 1 layer that we calculated to have a width of 1.390μm with the minimum design rules. The increase in width is here applied to the CS spacing, maintaining minimum spacing and width between metal 1 lines and only modifying the already irregular central part of the cell. For doing so, we need $CS = 0.1025\mu m$ (that is multiple of the manufacturing grid) and the final width of the BC is $W_{BCM} = 1.395\mu m$, verifying the value obtained for the 45 nm

layout. Then, on this final width we are able to fit up to 4 metal 5 lines of width 0.140μm at at distance 0.140μm (pitch of 0.280μm). Selecting 5 metal 5 lines will require $W_{BC} \geq 1.400\mu m$ and therefore an area overhead. This is the reason why M_5=4.

Regarding the height of the BC, that is determined by the metal 2 layer, using Equation (5.8), we obtain $H_{BC} = 4.18\mu m$, verifying the value obtained for the 45 nm layout. Included in the number of metal 2 lines M_2, note that we have selected $NM2up$=2, $NM2center$=3 and $NM2down$=2. Then, on this final height of the BC, we are able to fit up to 14 metal 4 lines of width 0.140μm at at distance 0.140μm (pitch of 0.280μm). Selecting 15 metal 4 lines will require $H_{BC} \geq 4.20\mu m$ and therefore an area overhead. This is the reason why M_4=14.

Assuming that metal 1 layer is determining the width of the BC and that metal 2 is determining its height, to further understand the impact of BC parameters on area, but also on routability, we study the effect of varying in the 45 nm technology node:

- The number of metal 1 lines M_1, that is directly related to the number of available inputs of the BC.
- The number of transistors T, that will modify the number of metal 2 lines M_2 in the BC.

We provide in Table 5.2 the calculations of the width of the BC W_{BC} when varying M_1, in Table 5.3 the calculations of the height of the BC H_{BC} when varying T, and in Table 5.4 the calculations of the area multiplying W_{BC} and H_{BC}. The number of metal 3, metal 4 and metal 5 that can be fitted in the width and height determined by metal 1 and metal 2 layers are also indicated to evaluate the routability resources of the resulting BCs. M_1 is always even to ensure BC symmetry. In fact, except the metal 1 lines devoted to VDD and GND, the rest of metal 1 lines can be devoted to access the gates of the transistors from both sides of the array (left and right). Therefore, having an odd number of

metal 1 lines will cause transistors to be irregular. For M_2, we consider the same $NM2up$=2, $NM2center$=3 and $NM2down$=2 in all cases to see the impact of varying only T. For M_3 we consider that lines are not shared at the edges to ensure that all metal 3 lines can be used for routing.

The increase in W_{BC} and H_{BC} is linear with M_1 and M_2 with increases of 0.330μm and 0.380μm per step respectively, that are the contributions of adding 2 metal 1 lines (as M_1 is even) or 2 metal 2 lines (as T is increased by one each step but for PMOS and NMOS). Therefore, the area increase is more important when increasing the BC dimensions in horizontal (adding inputs) than in vertical (adding transistors).

We observe that higher BCs allow including more routability resources (regarding metal 3, metal 4 and metal 5 lines). In that case, the increase depends on the metal layer. Upper metal layers like metal 4 or metal 5 that have wider and more separated lines, have a smaller increase (12 and 11 metal lines added for metal 4 and metal 5 in the range studied), while metal 3, with smaller width and size of the lines, benefits from a higher increase in the number of lines (21 metal 3 lines added for the range studied).

For the area calculations, we can see the combined impact of increasing M_1 and M_2. The selection of the number of inputs and transistors can be for instance traded off with the routability resources required for the particular circuit, or with the functions to be mapped into the BCs, to avoid unnecessary area overheads.

Table 5.2: Basic cell width varying M_1

M_1 lines	4	6	8	10	12	14	16	18	20	22
Inputs available	2	4	6	8	10	12	14	16	18	20
W_{BC} width (μm)	0.73	1.06	1.39	1.72	2.05	2.38	2.71	3.04	3.37	3.70
M_3 lines	5	7	9	12	14	17	19	21	24	26
M_5 lines	2	3	4	6	7	8	9	10	12	13

5.2.3 Basic cell impact on energy and delay

In this section, we will show through a possible implementation of the BC the impact of the BC parameters on energy and delay. For this, we will use the BC

Table 5.3: Basic cell height varying M_2

M_2 lines	15	17	19	21	23	25	27	29	31	33
T transistors	2	3	4	5	6	7	8	9	10	11
H_{BC} height (μm)	2.66	3.04	3.42	3.80	4.18	4.56	4.94	5.32	5.70	6.08
M_4 lines	9	10	12	13	14	16	17	19	20	21

Table 5.4: Basic cell area in (μm^2) varying the number of inputs and transistors

	Transistors									
Inputs	2	3	4	5	6	7	8	9	10	11
2	1.94	2.22	2.50	2.77	3.05	3.33	3.61	3.88	4.16	4.44
4	2.82	3.22	3.63	4.03	4.43	4.83	5.24	5.64	6.04	6.44
6	3.70	4.23	4.75	5.28	5.81	6.34	6.87	7.39	7.92	8.45
8	4.58	5.23	5.88	6.54	7.19	7.84	8.50	9.15	9.80	10.46
10	5.45	6.23	7.01	7.79	8.57	9.35	10.13	10.91	11.69	12.46
12	6.33	7.24	8.14	9.04	9.95	10.85	11.76	12.66	13.57	14.47
14	7.21	8.24	9.27	10.30	11.33	12.36	13.39	14.42	15.45	16.48
16	8.09	9.24	10.40	11.55	12.71	13.86	15.02	16.17	17.33	18.48
18	8.96	10.24	11.53	12.81	14.09	15.37	16.65	17.93	19.21	20.49
20	9.84	11.25	12.65	14.06	15.47	16.87	18.28	19.68	21.09	22.50

used in our works in the 90 technology node. The parameters of the BC are the following:

- T=6 PMOS and NMOS transistors
- W=440 nm, L=100 nm for the sizing of the transistors
- N=5 (up to metal 5 layer)
- M_1=6, M_2=23, M_3=6, M_4=23, M_5=6 for the number of metal lines in each layer

First, regarding the number of transistors, we decided to use $T = 6$. The choice of having 6 PMOS and 6 NMOS transistors in the basic cell is related to the possibility of implementing two logic branches of transistors with a maximum length of 3 serial transistors to avoid body effect and excessive serial resistance issues. In that way, we open the possibility of mapping multiple logic functions (at least two) inside of a single VCTA cell.

In what refers to transistor width W, electrical simulations were performed on a simple full-adder cell in the 90 nm technology node to decide the imple-

mentation that will later be used for the thesis. The objective was to achieve delay, energy and area results with no more than a 2X ratio overheads compared to the commercial standard cell available for the full-adder. We started using $W = 200nm$ because this was the minimum transistor channel width that ensure maximum transistor compaction when sharing the same oxide diffusion in the 90 nm node. Then, we increased W until we reached the objective. The final sizing chosen was $W = 440nm$. In Figure 5.7 the results are shown including the BC implementation with $W = 400nm$. We can see how for $W = 440nm$ the ratio overheads approach 2X for all measures (in particular, for WCD CI to CO, for AVGE and for Area, which were the most significant for the complete behavior of the full-adder). Note also that we still have room for improvement by increasing the energy and area overheads. This can indicate that further increasing W can further improve the results in delay. The decisions to be taken will depend on which of the factors need to be optimized.

Regarding the number of metal layers N, for this particular circuit, we needed only from metal 1 to metal 3 layers, that is why our first tests for VCTA used three metal layers. However, we also implemented up to metal 5 when routing congestion was found for more complex circuits.

The number of metal lines was fixed also on the needs of the full-adder circuit that requires up to four inputs in a logic branch. Therefore we chose $M_1 = 6$ to have four inputs. The rest of metal line layers where determined by regularity constraints minimizing the area increase of metals as explained before.

The optimization of the BC parameters have been left for future works as the main objective of the thesis was to illustrate the use of VCTA and to demonstrate that it can be applied to the design of complex circuits. That is why, in chapter 6, we will present the VCTA automation tool having as an input a given BC implementation. In fact, the number of transistors as well as their sizing will depend on the logic functions that have to be mapped inside of the BC and will have an important impact on the area of the BC and also on the final energy and delay results for the VCTA designs. Regarding the number of metal layers,

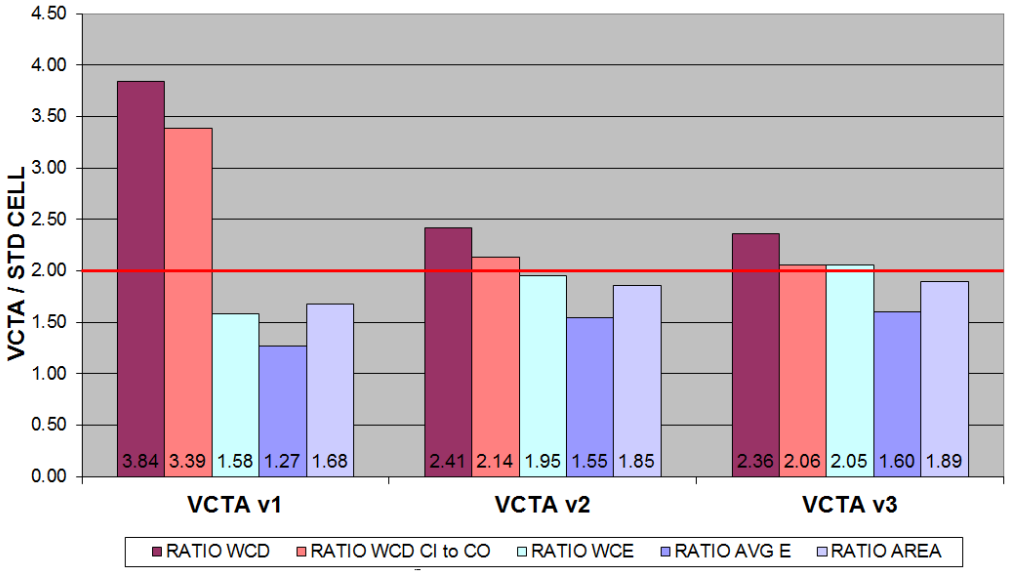


Figure 5.7: VCTA basic cell sizing. Results for a VCTA full-adder circuit indicating the ratio when compared to standard cell full-adder: WCD = worst-case delay, WCD CI to CO = worst-case delay from carry-in to carry-out (determining path delay when considering multiple bits to be added), WCE = worst-case energy, AVGE = average energy for all the possible transitions from input to output, and Area. VCTAv1 has $W = 200nm$. VCTAv2 has $W = 400nm$. VCTAv3 has $W = 440nm$.

they have to be chosen depending on the routing capabilities required. For low-congested layouts, fewer metal layers can be used. However, note that the limit is the number of metal layers of the technology and this will never suppose an extra limitation. Regarding the number of metal lines in each layer, basically, this will depend on the area of the BC, as the higher the area, the higher the number of metal lines can be fitted (always respecting the technology pitches). If more inputs are required for the logic functions, the metal 1 layer can determine the final area of the BC. In summary, the implementation of the VCTA BC has to be optimized depending on the design.

5.3 VCTA manual layouts evaluation

In order to illustrate the VCTA regular fabric we have manually implemented binary adders, a common block in digital designs, and also a DLL, used in analog

designs. In this case, we have used the commercial 90 nm technology node (see chapter 4).

5.3.1 32-bit adders evaluation

We have developed complete layouts for a 32-bit Carry-Ripple adder (CR32) for a 32-bit Carry-Lookahead adder (CLA32) and for a 32-bit Kogge-Stone adder (KS32) using the VCTA regular fabric and also the standard cell approach (STD90) to compare the area, energy, delay and manufacturing variability in both designs.

5.3.1.1 VCTA layout generation

The parameters of the BC for VCTA are the following:

- T=6 PMOS and NMOS transistors
- W=440 nm, L=100 nm for the sizing of the transistors
- N=3 (up to metal 3 layer)
- M_1=6, M_2=23, M_3=6 for the number of metal lines in each layer

Note again that we can consider many other BC implementations with different number of transistors, metal layers, etc. However, the objective of this work is to explain and demonstrate the new VCTA regular fabric.

The steps that we have followed for VCTA layout generation are: (1) find out the logic functions needed to implement the structure of the circuit, (2) map the transistors of these functions into the VCTA basic cell as we have shown in the previous section for the NAND gate in Figure 5.3, (3) manually place and route them to obtain the complete layout. The automation of the VCTA physical design flow will be explained in next chapter.

The binary adder circuits studied require 6 different types of logic functions: an inverter, an XOR, a 2-input NAND, a 4-input NAND, an AND-OR and an OR-AND [68]. We have mapped these functions into the VCTA basic cells. In

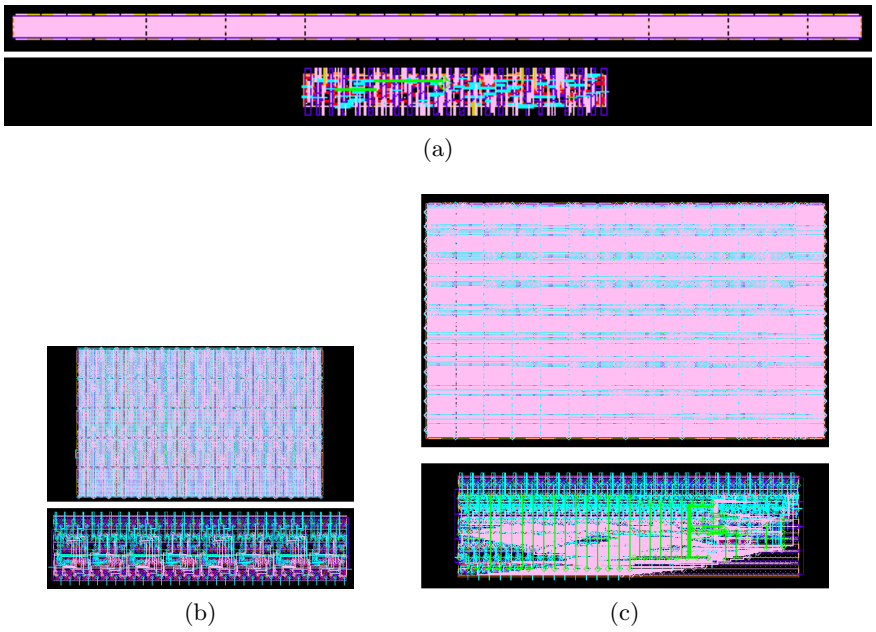


Figure 5.8: Layouts of 32-bit adders: (a) CR32 VCTA (254μm x 7μm, top) and STD90 (92μm x 10μm, bottom) (b) CLA32 VCTA (65.5μm x 40μm, top) and STD90 (80μm x 17.4μm, bottom) and (c) KS32 VCTA (110μm x 64μm, top) and STD90 (91μm x 29.5μm, bottom).

some cases we were able to implement 2 functions into a single VCTA basic cell. This can be done when the functions are next to each other in the circuit (e.g., the output of one of the functions is the input of the other one, or they share the same inputs).

As a consequence, the VCTA layouts can be composed by fewer cells than the STD90 layouts. For instance, the complete CLA32 finally required 228 standard cells and only 184 VCTA basic cells. We have manually placed and routed those VCTA cells trying to minimize the interconnect distances as well as for STD90 cells.

For illustrative purposes, the resulting complete layouts captures for VCTA and STD90 are presented for the CR32 in Figure 5.8a, for the CLA32 in Figure 5.8b and for the KS32 in Figure 5.8c.

5.3.1.2 Area, energy and delay evaluation

We have performed complete electrical simulations of the extracted layouts of CR32, CLA32 and KS32 using the HSPICE simulator. We have evaluated both the adders designed with our VCTA regular design as well as those based on standard cells (STD90) in terms of delay and energy for 10400 inputs that we have sampled from all 26 programs in the SPEC2000 benchmark suite [76]. We have measured the delay from input variation to the associated output transition considering the cross at 90% of the voltage rise or fall swings. We have also measured energy for each input combination integrating the current demand at the power supply source during the addition. Finally, we have measured the area directly from the layout. We show measurement results for worst-case delay (WCD) and average energy dissipation (AVGE) for all the inputs in Table 5.5.

Table 5.5: Adders evaluation (WCD = worst-case delay, AVGE = average energy)

	WCD(ns)	AVGE(pJ)	Area(μm^2)
CR32 STD90	2.69	0.16	920
CR32 VCTA	6.69	0.30	1778
Ratio	2.49x	1.88x	1.93x
CLA32 STD90	1.11	0.21	1394
CLA32 VCTA	2.15	0.47	2620
Ratio	1.94x	2.24x	1.88x
KS32 STD90	0.84	0.33	2684
KS32 VCTA	1.69	0.79	7046
Ratio	2.00x	2.39x	2.63x

First, this particular choice for the BC of VCTA regular design implies an increase around 2x in area (1.93x for CR32, 1.88x for CLA32 and 2.63x for KS32) when compared to the STD90 layouts. The area increase is basically due to the regularity requirements and redundancy, because all possible configurations of devices and interconnects are in place in the BC of VCTA. The BC includes dummy transistors, spare transistors and also spare interconnects which increase the total area. Moreover, layouts has been generated manually. In next chapter we will present the VCTA automation flow allowing area optimization.

In terms of WCD and AVGE, adders present also around 2x energy and delay ratios. In fact overheads introduced by VCTA when compared to STD90 are very much dependent on the function to implement. STD90 designs use different standard cells depending on the circuit optimization but VCTA always uses the same BC. Energy and delay overheads are due to the parasitics introduced by the VCTA metal grid.

Another BC parameter to optimize is transistor sizing (all the transistors have the same dimensions). With our present choice of 440 nm for width, by connecting in parallel transistors, we can only emulate wider transistors of 880 nm, 1320 nm, etc., with a width multiple of the basic transistor, and this is not always optimal.

Note also that logic functions implemented such as NAND, XOR, etc. are particularly suitable for STD90 approach, but may be suboptimal for VCTA.

5.3.1.3 Manufacturing variability evaluation

Evaluating the impact of layout regularity on manufacturing variability is key to demonstrate the usefulness of maximizing layout regularity using VCTA. We have used for this purpose the models presented in chapter 2 for proximity and coma effects and for STI mechanical stress.

A - Channel length variations: proximity and coma effect

Using those proximity and coma effect models we have measured the transistor channel length (L) systematic process variations of the adder layouts for VCTA and STD90 for the different sources of systematic variability considering 10% maximum L variations.

As all the VCTA transistors in the BC have the same layout neighborhood, with two polysilicon lines at the same distance, they are all affected by the same systematic L variations, thus showing no σ in the L distribution. This is achieved by the use of the dummy polysilicon lines at the edges of the PMOS and NMOS transistor arrays.

On the other hand, STD90 adders that use different cells with different placements present higher number of layout neighborhoods. The L statistics in terms of $3\sigma/\mu$ are presented in Table 5.6. For instance, for proximity effect, KS32 transistors see 8 neighborhoods, while for coma effect, which differentiates the sides for the distances measured, there are 10. That is why coma effect variability is higher than proximity effect variability.

Table 5.6: Channel length variations

	Proximity Effect L $3\sigma/\mu$	Coma Effect L $3\sigma/\mu$
CR32 STD90	3.53%	6.11%
CLA32 STD90	5.31%	6.19%
KS32 STD90	5.16%	6.48%

The final result is that all VCTA transistors are affected by the same L systematic variation and therefore have all the same L whereas the L variability between transistors is around 3.5-6.5% for STD90. Therefore, we can conclude that L variations for proximity effect and coma effect are minimized through VCTA regular layout designs.

Note that these results show the regularity of VCTA at two levels. First, the L variations are the same for all adders for the VCTA design whereas they depend on the particular circuit for the STD90 design. This is because VCTA uses the same BC for all adders and STD90 uses different cells. This is VCTA regularity at cell level. Second, VCTA maximizes regularity inside the BC and shows only one neighborhood for all transistors whereas STD90 shows different neighborhoods inside each of the cells. This is VCTA regularity at transistor level.

B - Threshold voltage variations: mechanical stress

Using the BSIM4 models supplied for the 90 nm technology node, we have calculated the Vth variations for PMOS and NMOS transistors in the CR32, CLA32 and the KS32 adders. The results for the VCTA and STD90 designs are shown in Table 5.7.

For VCTA transistors, there are only three different cases (for PMOS as well as for NMOS). From Figure 5.9 it can be seen that transistors 1 and 6 will have the same STI stress because the BC is symmetric. The same occurs for transistors 2 and 5 and finally for transistors 3 and 4. Furthermore, the VCTA transistors have all the same channel width and therefore will be affected similarly.

On the other hand, for STD90, there is a higher number of cases related to the different transistor neighborhoods and to the different transistor sizings. That is why for VCTA the Vth variability is around 1% and for STD90 it reaches 4% for PMOS and 7% for NMOS. The ratios for the reduction of Vth variability due to VCTA regularity are close to 0.20x.

Again, the results show VCTA regularity at two different levels. First, at cell level we can see how VCTA shows the same Vth variations independently of the circuit considered. Second, at transistor level, the number of cases for STI stress is also reduced because of transistor array regularity.

Table 5.7: Threshold Voltage variations

	PMOS Vth $3\sigma/\mu$	NMOS Vth $3\sigma/\mu$
CR32 STD90	3.21%	7.60%
CR32 VCTA	0.82%	1.24%
Ratio	0.26x	0.16x
CLA32 STD90	4.85%	6.25%
CLA32 VCTA	0.82%	1.24%
Ratio	0.17x	0.20x
KS32 STD90	4.07%	5.83%
KS32 VCTA	0.82%	1.24%
Ratio	0.20x	0.21x

5.3.2 Delay-locked loop evaluation

The application chosen for VCTA evaluation in this case is the DLL analog circuit. An analog block, that is more complex to design from the layout point of view than the digital layouts, allows more metrics of the applicability of VCTA — that were previously limited to area, energy and delay. For instance, for the

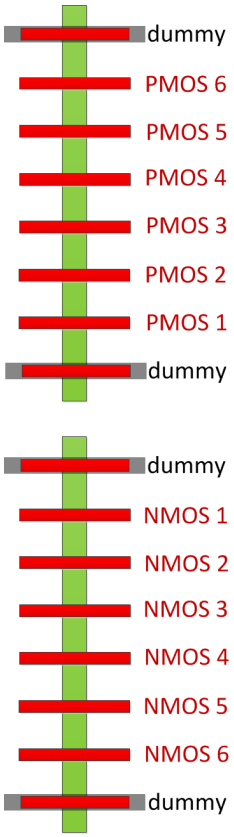


Figure 5.9: VCTA transistor array ($T = 6$).

DLL, jitter will also be evaluated. We have compared the DLL design using VCTA to its full custom version (FC).

To evaluate the performance of the FC and VCTA designs an analysis of the Voltage Controlled Delay Line (VCDL) is carried out. In fact, analyzing just one the of the delay cells of the line is sufficient to determine the behavior of the whole DLL in terms of energy consumption and jitter [70]. The VCDL must be able to compensate for process, voltage and temperature (PVT) variations and provide a constant delay, thus an analysis of the dependence of the delay, energy and jitter with the voltage control is performed for both the FC and VCTA implementations of the delay cell.

5.3.2.1 Full Custom and VCTA Layout generation

In this section we explain how FC and VCTA layouts are generated for the DLL evaluation.

A - Full custom layout

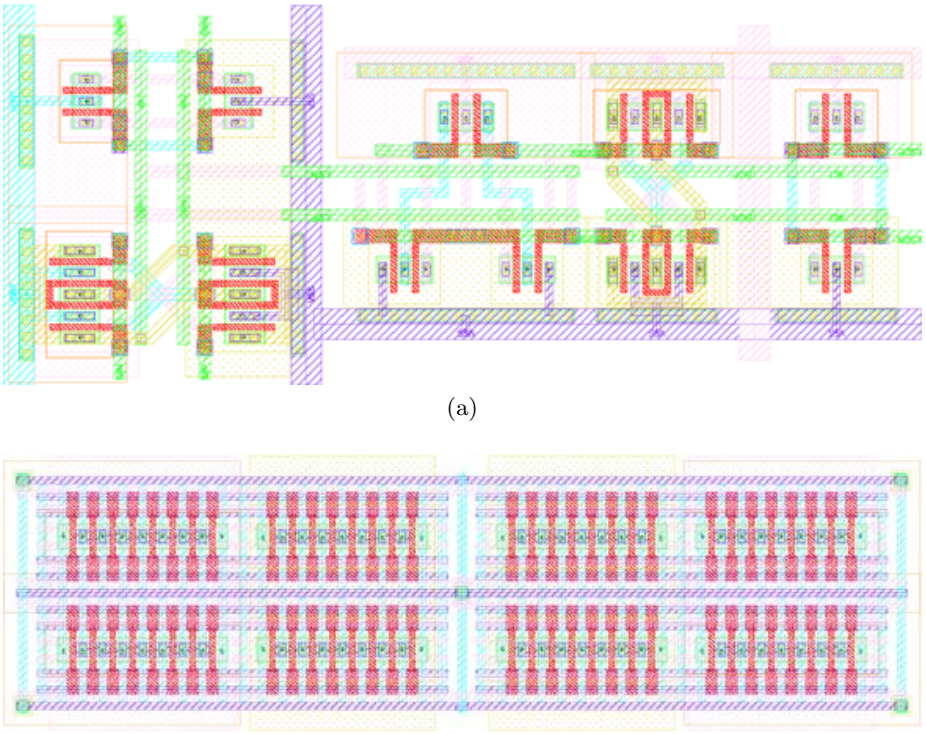
The DLL layout for FC was implemented considering the needs of the VCDL design. Therefore, an special effort was taken to ensure maximum symmetry between the two branches of the differential delay cell design. This includes not only enforcing the interconnections to have the same length, but also including the same number of vias along the signal path. The interconnections length were optimized to reduce the parasitic capacitance as much as possible to increase the system efficiency. Once the transistors sizing was determined by simulation, the NMOS and PMOS transistors were implemented using the interfingering technique. The cross-coupling technique could not be used because of the limited size of the transistors used in the design. The final full custom layout of the delay cell is depicted in Figure 5.10a.

B - VCTA layout

For the comparison of VCTA and FC designs to be fair, the operation range of the DLL (in terms of the delay needs) of both designs need to be equivalent. Ensuring that, one design can be replaced with the other maintaining the same functionality. For this, a BC transistor sizing of 560 nm was used.

The parameters of the BC for VCTA are the following:

- T=6 PMOS and NMOS transistors
- W=560 nm, L=100 nm for the sizing of the transistors
- N=3 (up to metal 3 layer)
- M_1=6, M_2=23, M_3=6 for the number of metal lines in each layer



(a)

(b)

Figure 5.10: Delay cell layouts before tiling (a) Full Custom ($79.92\mu m^2$) and (b) VCTA ($64.78\mu m^2$).

The layout of the delay cell using the VCTA structure is shown in Figure 5.10b. It is made with 4 BCs. Note that VCTA delay cell is around 19% smaller than FC delay cell. The area is slightly lower because the FC design has been optimized for performance and have to fulfill some placing and routing restrictions.

5.3.2.2 Energy, delay and jitter evaluation

In this section we present the VCDL delay cell and the DLL simulations for energy, delay and jitter.

A - Delay cell simulation

Simulations showing the VCTA and FC designs delay and energy are summarized in Tables 5.8 and 5.9 for the VCDL delay cell. Note that at schematic level,

VCTA has delay overhead when compared to FC design because of dummy and spare transistors included in the BC of VCTA. However, applying the extraction of parasitics and the metal tiling the delay overhead is compensated. In fact, the BC layout design already fulfill the metal density rules thus the tiling can be safely suppressed from the design flow, reducing design time even further.

Table 5.8: VCDL Cell delay in *ps*

Simulation	FC	VCTA
Schematic	54.7	57.7
Extracted	105.3	117.6
Ext. + Tiling	116.3	

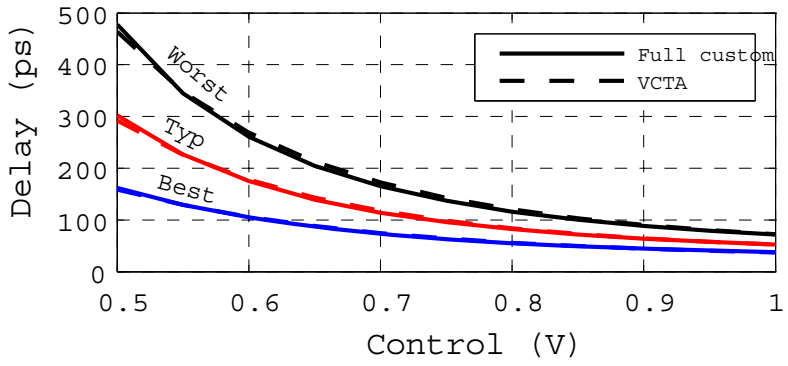
Table 5.9: VCDL Cell energy in fJ

Simulation	FC	VCTA
Schematic	42.1	58.7
Extracted	85.0	128.6
Ext. + Tiling	93.4	

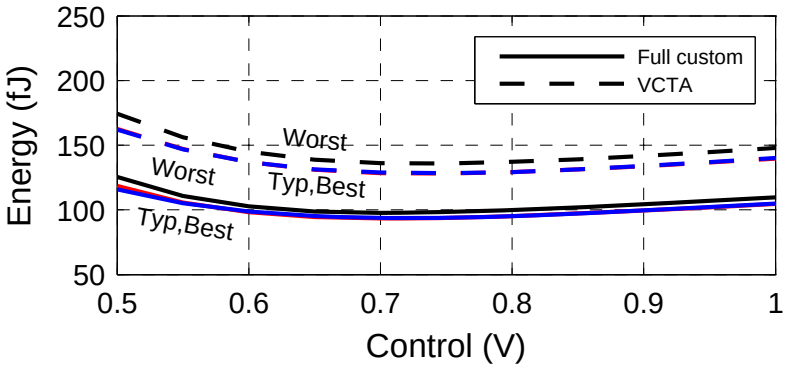
B - DLL simulation

The FC and VCTA designs have been validated by means of the DLL simulation. As explained in chapter 4, the characteristics of a delay cell are sufficient to determine the behavior of the whole DLL. PVT variation simulations are needed to evaluate the suitability of this cell. For this reason the cell was analyzed in three different cases: (1) Worst case PMOS and NMOS corner model and worst case parasitics at 353K, (2) Typical case PMOS and NMOS corner model and typical case parasitics at 298K, and (3) Best case PMOS and NMOS corner model and best case parasitics at 273K.

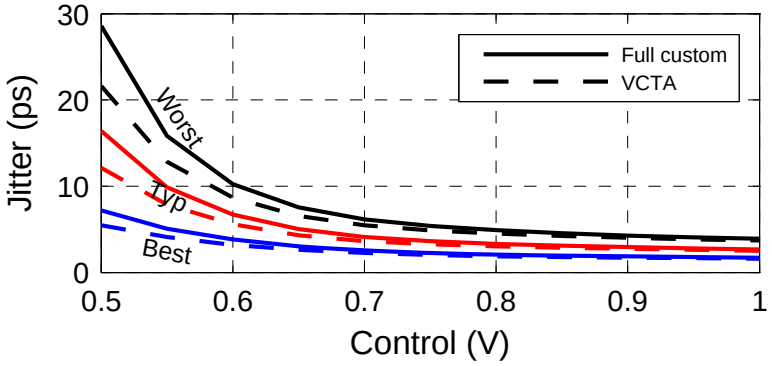
To analyze the behavior of the delay cell, simulations for its delay, energy consumption and jitter where carried while sweeping the control voltage along its range. The VCTA design needs to be able to substitute the FC design in a DLL, therefore it must have a delay versus control voltage dependence as close as possible to the original FC delay cell. The simulation results after extraction and tiling for both designs are shown in Figure 5.11.



(a)



(b)



(c)

Figure 5.11: FC and VCTA simulation results for the DLL design for the typical, best and worst cases: (a) Delay (b) Energy and (c) Jitter.

The difference between the delay of the FC design and the VCTA design is kept below ±5% for control voltages higher than 0.5V. Indeed this small difference can be compensated by means of the control voltage.

In terms of energy consumption, the VCTA design has a 35% overhead for the control voltage range of interest, as shown in Figure 5.11b. This is due to the parasitics introduced by the regular metal grid, but also due to the regularity requirements and redundancy. Indeed, all possible configurations of devices and interconnects are in place in the BC of VCTA. Moreover, the BC also includes dummy transistors, spare transistors and spare interconnects which increase the total energy consumption.

For mobile applications, where this energy consumption is critical, VCTA offers the possibility of easily scaling down the process technology. As only the layout of the BC has to be redesigned, but the same contact and via configurations can be maintained, this is not time consuming. Therefore, the energy can also be scaled down by the square of the dimensional scaling factor λ. Thus, an scaling factor of $\lambda \geq \sqrt{1.35}$ is enough to ensure the same energy consumption than full custom design. Regarding the FC design, scaling down the layout is very costly as it involves the redesign of the whole layout patterns and so it will be very time consuming.

Finally the jitter is always lower in the VCTA design than in the FC design, as depicted in Figure 5.11c. Jitter due to mismatch is dominant, thus the VCTA benefits from large transistor devices. Furthermore, as regularity is known to reduce process variations [44], the mismatch between delay cells is reduced and hence the total jitter is improved.

5.3.2.3 Regularity trade-off

The VCTA design has been demonstrated to be equivalent to the FC design in terms of delay and functionality for the VCDL of a DLL. The delay range to compensate for the PVT variation is the same, and jitter and area are smaller.

The VCTA, however, suffers from a 35% overhead in terms of energy consumption, although this can be circumvented by scaling the design to a smaller CMOS process technology which will be harder for the FC design —in particular when considering non mature technology nodes that has not been already fully optimized.

Due to its regularity, the VCTA design can be designed much faster and reach the market before full custom design. For this particular DLL design, the full custom design took roughly 6 months while the VCTA just took 1 week.

5.4 Conclusion

Based on the observation that existing regular fabrics are not considering regularity in a comprehensive way, we have developed a new regular layout design technique named Via-Configurable Transistor Array (VCTA) which aim is to maximize layout regularity at transistor and interconnect level. The objective is to fulfill the requirements imposed by future technology nodes with increasing design and manufacturing issues.

VCTA proposal maximizes regularity at cell level using a single basic cell, at transistor level with the transistor array structure, and finally at interconnect level with the via-configurable choice. We have presented the VCTA front-end and back-end, studying the VCTA basic cell parameters that will determine the area and routability of the VCTA layout as well as the energy and the delay results.

For 32-bit adders, we have seen important area, energy and delay overheads compared to the standard cell approach, but we have demonstrated how process variations can be highly reduced due to VCTA regularity. In particular, we have demonstrated that proximity and coma effects channel length variations and mechanical stress threshold voltage variations are minimized. Moreover, further optimizations regarding the VCTA cell implementation can help reducing layout regularity overheads. This is part of our future works. In the thesis, we focused

on demonstrating the applicability of VCTA regular layout fabric. Furthermore, in chapter 6 we will present the VCTA automation flow that will help us to reduce the area overheads found doing 32-bit adders layouts manually.

We also evaluated a DLL design using VCTA. In particular, the VCDL layout, that is critical for the circuit, has been implemented and compared to the full custom implementation. As VCTA maximizes regularity it has allowed the speed up of the design time and it is expected to drastically reduce the associated costs. Moreover, complete simulations have shown that VCTA layout is equivalent in terms of functionality to the full custom layout: it has the same delay range. Added to this, VCTA cell area and jitter are smaller. However, the energy consumption is higher. The trade-off between reaching the market faster and energy overhead is the key. Depending on the application, if time-to-market is more critical than energy, VCTA becomes the best choice. Furthermore, scaling down the designs using VCTA is also very easy as only the basic cell has to be redesigned. Therefore the energy overhead can be compensated. In fact, regularity is expected to be compulsory for future technologies due to design and manufacturing issues.

Chapter 6

VCTA Automation

The VCTA layouts studied in previous chapters had been developed manually. To generate VCTA layouts in an automated way, VCTA-specific synthesis tools are required. To automate the VCTA physical design we first tried to reuse the available standard cell flow and tools. However, the standard flow is library-based. The whole library of standard cells is already available and characterized for any circuit to be designed (see chapter 2). For VCTA, the cells are generated on-the-fly. VCTA is based on a single basic cell that is configured depending on the circuit under study. Moreover, the basic cell can contain multiple functions, or even parts of different functions, not as in the standard flow, where each standard cell corresponds to a function. Therefore, VCTA does not fit the standard library-based logic synthesis methodology. VCTA fabric also requires specific via and metal extensions configuration for intra and inter-cell routing. Therefore the use of the existing standard tools was not possible.

In this chapter we present a new automated flow and its algorithms for regular layout generation with VCTA. In particular, the flow focuses on reducing the area overhead associated to layout regularity. Other flows with different targets (e.g., delay, energy, etc.) can also be devised. Since the purpose of this part of the thesis is proving that a physical design flow based on VCTA is feasible, those other potential flows are left as future works.

The structure of the chapter is as follows. Section 6.1 describes the VCTA physical design flow using a full adder to illustrate each of the required steps. Section 6.2 presents the results obtained for the ISCAS'85 benchmark circuits in the 45nm technology node showing that comparable areas to the standard cell approach can be obtained. Finally section 6.3 summarizes the conclusions of this chapter.

6.1 VCTA Physical Design Flow

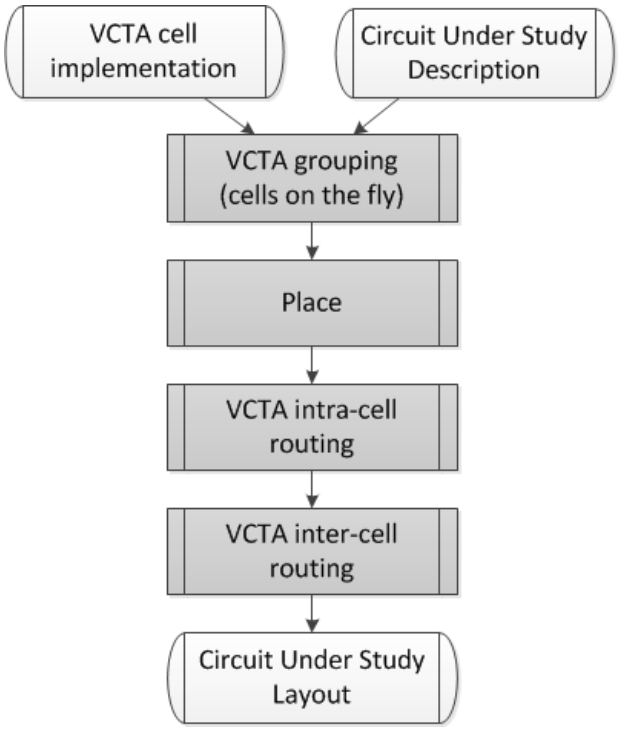


Figure 6.1: VCTA physical design flow diagram.

6.1.1 Flow overview

The VCTA physical design flow diagram is depicted in Figure 6.1. The inputs of the flow are a transistor netlist describing the circuit and a particular VCTA basic cell implementation as defined in section 5.2.1. The grouping, place and routing parts have been implemented using C code and interacting with standard tools when possible. Layout Versus Schematic (LVS) and Design Rule Check (DRC)

have also been used to verify that the circuit has been generated properly. We give details on each of these steps in the following subsections. The full adder in Figure 6.2 will be used as an example for explanations.

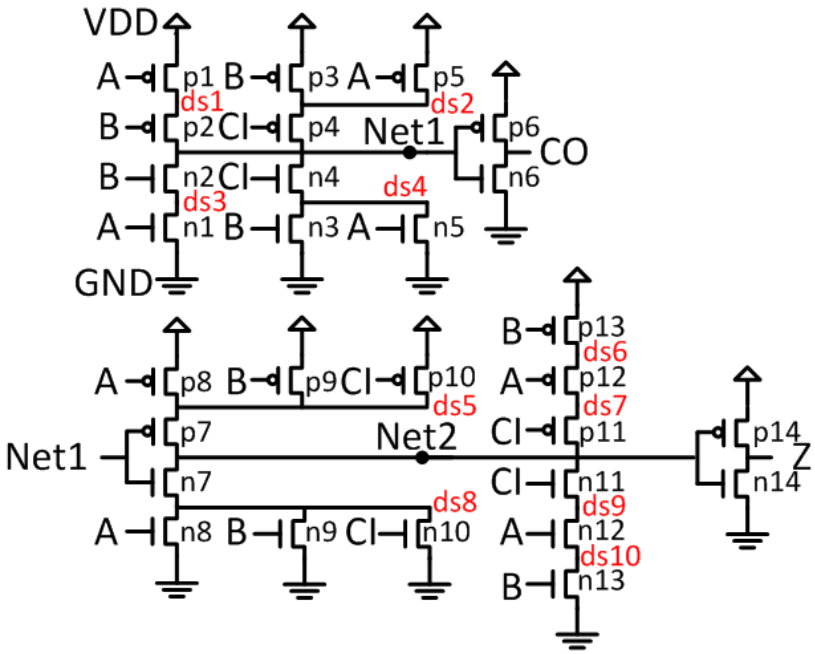


Figure 6.2: Full adder schematic.

6.1.2 VCTA Grouping

The grouping step generates the VCTA cells that will be required for the circuit under study. The objective of this step is to map the transistors of the circuit into the VCTA cells minimizing the required inter-cell connections (to simplify routing) and maximizing cell transistor occupation (to reduce the number of cells required and the final area of the layout). As a result we generate the Verilog for the circuit under study with all the VCTA cells required. This Verilog file is required for the placing step. The grouping flow is depicted in Figure 6.3. The steps involved are detailed next.

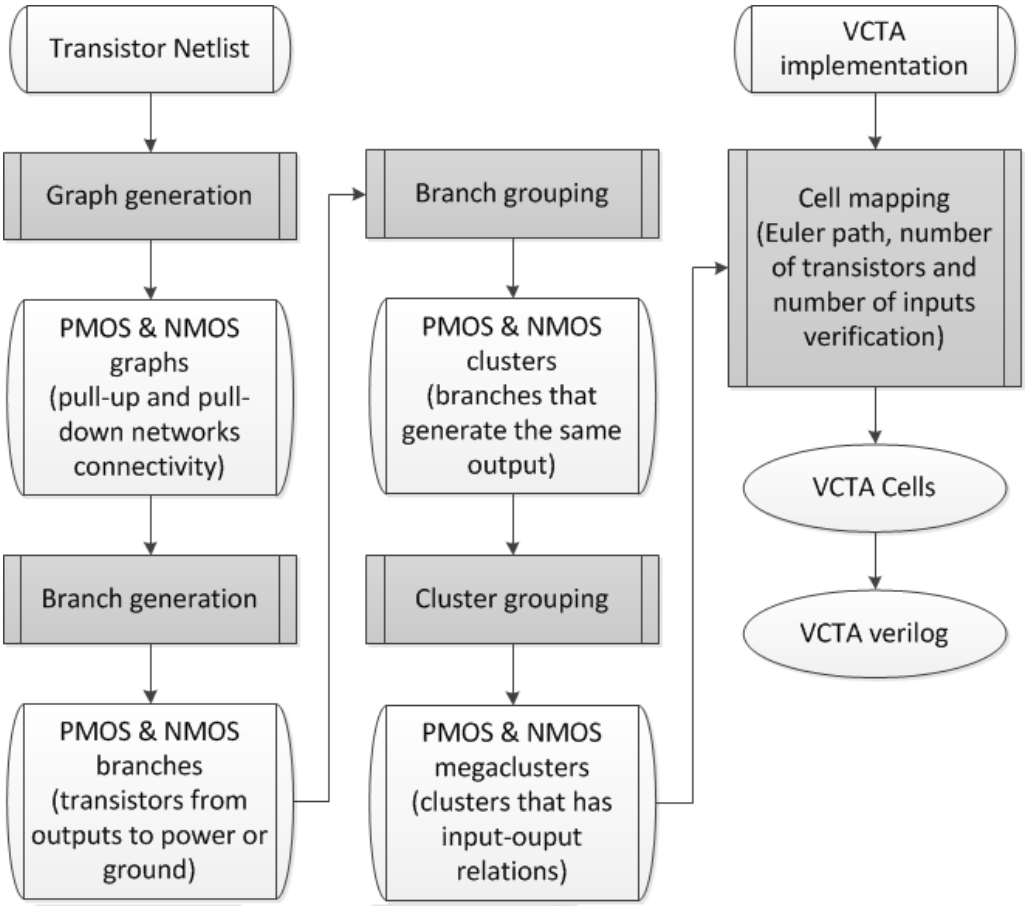
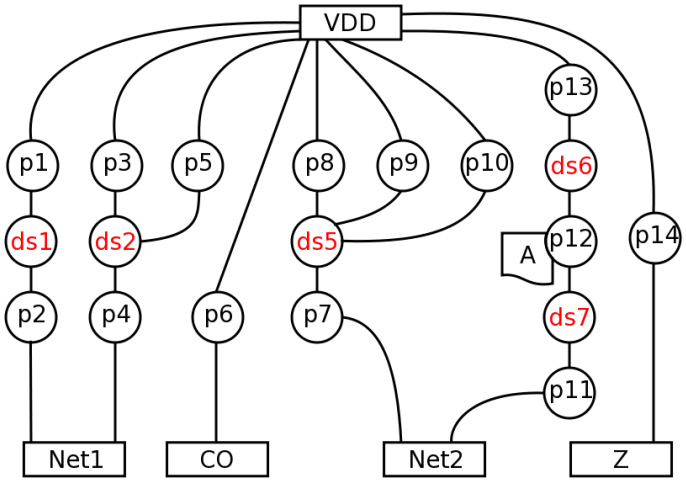
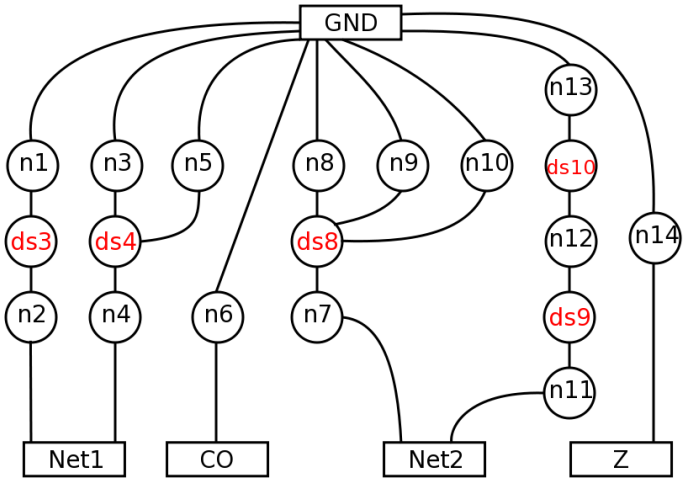


Figure 6.3: VCTA grouping flow diagram.



(a) PMOS network



(b) NMOS network

Figure 6.4: Full Adder graphs. (a) PMOS: We indicate in red the drains or sources of the transistors, except for logic function outputs (e.g., Net1) and for power supply (VDD). The gate inputs are also stored (e.g., for transistor p12 where the input of the gate is A). (b) NMOS: The common vertices with PMOS graph are Net1, CO, Net2 and Z, that are the output nodes of the logic functions in the circuit.

6.1.2.1 Netlist translation into a graph

Initially, we read the transistor netlist of the circuit (in Hspice format in this case, but any format can be adapted) to generate the corresponding graphs for PMOS and NMOS transistor networks (see Figures 6.4a and 6.4b for the full adder example). Each graph is undirected and unweighted and represents network connectivity. Vertices of the graph are: a) transistors b) drains or sources of transistors, and are updated when adding a transistor to the graph. In the case of vertices of type transistor, the gate input of the transistor is saved as a property as it will be required later (only one example is shown in Figure 6.4a for transistor p12, but all the gate inputs are saved). Sources connected to power lines are special vertices indicated as supply or ground (VDD and GND). Edges of the graph indicate if vertices are connected or not. When reading a transistor from the netlist, transistors vertices are considered connected to its drain and to its source vertices. To implement the graph, the vertices and their edges are allocated in a hash table where the hash index is a hash calculation using the characters of the name of the vertex. Doing so, we can directly access the vertices of the graph by knowing their names, without traversing the whole table.

6.1.2.2 Branches, clusters and megaclusters

To minimize the inter-cell routing, transistors are grouped together in three steps:

- Transistors that are connected from the supply lines (VDD or GND) to an output node of a logic function are grouped forming branches (of PMOS or NMOS transistors). Branch generation is done recursively and is depicted in algorithm 6.1.

- Branches that generate the same output nodes are grouped in what we call clusters. The PMOS and NMOS graphs are inspected to find their common vertices that are the common outputs defining clusters. Note that only drains or sources can be common between PMOS and NMOS graphs. Branch grouping to obtain clusters is depicted in algorithm 6.2.

Algorithm 6.1 Branch generation.

```
Graph = PMOS or NMOS network graph
V = set of vertices in Graph
for all v_i ∈ V do
    if v_i is an output vertex then
        for all v_j ∈ V connected to v_i do
            if v_j is not already in a branch then
                if (v_j ≠ supply) ∧ (v_j ≠ (output vertex)) then
                    Start a new branch b_ij containing v_i and v_j
                    FILL(b_ij, v_j)
                end if
            end if
        end for
    end if
end for
function FILL(b_f, v_f)
    for all v_k ∈ V connected to v_f do
        if v_k is not already in a branch then
            if (v_k ≠ supply) ∧ (v_k ≠ (output vertex)) then
                Add v_k to b_f
                FILL(b_f, v_k)
            end if
        end if
    end for
end function
```

Algorithm 6.2 Branch grouping.

```
GraphP = PMOS graph of the circuit
GraphN = NMOS graph of the circuit
V_out = set of output vertices common to GraphP and GraphN
B = set of PMOS or NMOS branches
for all v_i ∈ V_out do
    Start a new cluster c_i
    for all b_j ∈ B generating v_i do
        Add b_j to c_i
    end for
end for
```

- Clusters which generate the inputs for other clusters or that share inputs are grouped in megaclusters or neighbor clusters. For this we inspect the transistors in the clusters to find clusters that share transistor inputs (property that we have saved associated to this type of vertex), and we also check if the output generated by the cluster is input of any transistor in a different cluster. Megaclusters are groups of transistor branches that are neighbors and that will be mapped into the same group of VCTA cells to reduce routing complexity. The cluster grouping to obtain megaclusters is depicted in algorithm 6.3.

Algorithm 6.3 Cluster grouping.

```
1: C = set of clusters of the circuit
2: for all c_i ∈ C do
3:     for all c_j ∈ C do
4:         for all b_m branch ∈ c_i do
5:             for all b_n branch ∈ c_j do
6:                 if b_m and b_n share inputs or outputs then
7:                     c_i and c_j grouped in the same megacluster
8:                 end if
9:             end for
10:        end for
11:    end for
12: end for
```

The results for the full adder are summarized in Table 6.1, Table 6.2, Table 6.3 and Table 6.4. For instance, following algorithm 6.1 for output vertex Net1, we start PMOS branch BP1 with vertex p2, that is then filled with vertices ds1 and p1 until VDD is reached (see Figure 6.4a). Therefore PMOS branch BP1 is composed by transistors p1 and p2. Then, following algorithm 6.2, as branches BP1, BP2, BN1 and BN2 generate the same output Net1, they are all added to cluster CL1. Then, following algorithm 6.3 as Net1 is an input of branches BP3 and BN3 from cluster CL2 and also of branches BP4 and BN4 from cluster CL3, the three clusters CL1, CL2 and CL3 are grouped in megacluster MC1. Finally, as branches BP4, BP5, BN4 and BN5 from CL3 generate the output Net2, that is also input of branches BP6 and BN6 from cluster CL4, this last cluster CL4 is

also added to the megacluster MC1, that in fact is the only resulting megacluster of the full adder. Note that this is a particular case used to illustrate the design steps and that for more complex circuits several megaclusters will be found.

Table 6.1: Full adder PMOS branches.

PMOS branches	Transistors	Inputs	Output
BP1	p1 p2	A B	Net1
BP2	p3 p4 p5	A B CI	Net1
BP3	p6	Net1	CO
BP4	p7 p8 p9 p10	A B CI Net1	Net2
BP5	p11 p12 p13	A B CI	Net2
BP6	p14	Net2	Z

Table 6.2: Full adder NMOS branches.

NMOS branches	Transistors	Inputs	Output
BN1	n1 n2	A B	Net1
BN2	n3 n4 n5	A B CI	Net1
BN3	n6	Net1	CO
BN4	n7 n8 n9 n10	A B CI Net1	Net2
BN5	n11 n12 n13	A B CI	Net2
BN6	n14	Net2	Z

Table 6.3: Full adder clusters.

Clusters	Branches
CL1	BP1 BP2 BN1 BN2
CL2	BP3 BN3
CL3	BP4 BP5 BN4 BN5
CL4	BP6 BN6

Table 6.4: Full adder megaclusters.

Megaclusters	Clusters
MC1	CL1 CL2 CL3 CL4

6.1.2.3 VCTA Basic cell mapping

To maximize cell transistor occupancy and to reduce the area of the final layout, we group these neighbor branches of the megaclusters into the smallest number of

VCTA cells. The basic cell mapping for a megacluster is depicted in algorithm 6.4. First, the PMOS and NMOS branches are grouped into the smallest number of VCTA cell PMOS and NMOS transistor arrays respectively. Finally, PMOS and NMOS transistor arrays are grouped to obtain the final VCTA cells. The problem is equivalent to an unbounded knapsack problem where the objects are branches and their weight is the number of transistors in the branch. The maximum weight is determined by the maximum number of transistors T of the VCTA cell. This problem is solved using a greedy approximation algorithm sorting the branches from more to fewer transistors in order to maximize VCTA cell occupancy and minimize the final number of VCTA cells for the complete circuit (directly related to the final layout area).

However, when making decisions in the knapsack problem we also need to verify that the grouped branches can be mapped into a VCTA basic cell at two levels:

- The total number of inputs used when grouping branches has to be lower or equal to the number of available inputs that is determined by the number of metal 1 lines M_1 in the cell. As a VCTA cell is composed by the PMOS and NMOS arrays and they share the same metal 1 lines for routing inputs, both PMOS and NMOS networks are taken into consideration.
- As we have all transistors in series in the VCTA cell, we also need to check if the transistors in the branches can be organized following an Euler Path when combining them. In this case, PMOS and NMOS networks are independent. If there is no Euler Path, there will be breaks in the cell transistor array that the VCTA cell is not supporting.

Those branches that cannot be grouped occupy alone one VCTA cell. Finally, for a particular branch that does not fit inside of a VCTA cell, the branch is divided into different cells and connected later in the inter-cell routing step. The division is done splitting the branch into smaller Euler Paths that can then be fitted into the VCTA cells.

The Euler Path verification is shown in algorithm 6.5. The VCTA cell transistor array (PMOS or NMOS ta) has T serial transistors that share the same oxide diffusion. We associate to each of them an index (IND) and a gate, and upper or lower drain/source connections. The algorithm for finding the Euler Path is based on exhaustively placing the transistors (tr) to be mapped inside of the transistor array (ta) in all the possible positions and orientations. The position of the transistor is the index IND. The orientation of the transistor is where to map the drain and the source (named Up and Down connections). There are two possibilities if we consider that the transistor array is oriented vertically: (a) drain in the upper oxide diffusion zone (b) drain in the lower oxide diffusion zone. Starting from each of the transistors (a total number of $numtran$) that need to be mapped from the first position of the array ($IND = 1$) and trying all the possible combinations we ensure that if there is an Euler Path for the transistors we will find it, and vice versa. The output of the verification is whether there is or not an Euler Path for all the transistors and if there is one, what is the order of the transistors and the configuration of the resulting transistor array. When an Euler Path is found the algorithm ends even if multiple Euler Paths can be found. Note that applying this Euler Path verification algorithm, transistors from different branches can be interleaved. The transistors of a branch do not need to be adjacent in the transistor array of the cell, and this represents a degree of freedom of the VCTA fabric.

Table 6.5: Full adder VCTA cells.

Cells	Branches	Trans.	Occ.	Inputs
Cell 1	BP4 BP1 BN4 BN1	6P 6N	100%	4
Cell 2	BP2 BN5 BN2 BN5	6P 6N	100%	3
Cell 3	BP3 BP6 BN3 BN6	2P 2N	33.3%	2

The resulting VCTA cells using algorithm 6.4 for the full adder example are summarized in Table 6.5 (indicating the branches included in each cell, the number of transistors, the final transistor occupancy and the number of inputs). The different iterations for finding them are detailed next for the PMOS part (NMOS

Algorithm 6.4 Basic cell mapping. The number of transistors in the VCTA cell is T and the number of inputs available is $maxin$. A VCTA cell is composed by a PMOS transistor array (pta) and by a NMOS transistor array (nta).

```
BP = set of PMOS branches of the megacluster
BN = set of NMOS branches of the megacluster
PTA = set of PMOS transistor arrays of the VCTA cells
NTA = set of NMOS transistor arrays of the VCTA cells
Sort PMOS branches from more to fewer transistors in BP
for all bp_i ∈ BP do
    if bp_i is not already mapped in PTA then
        Start new pta_i ∈ PTA containing bp_i
        for all bp_j ∈ BP do
            if bp_j is not already mapped in PTA then
                tranP = PMOS transistors when adding bp_j to pta_i
                inP = inputs used when adding bp_j to pta_i
                euP = true if ∃ Euler Path when adding bp_j to pta_i
                if (tranP ≤ T) ∧ (inP ≤ maxin) ∧ (euP) then
                    Add bp_j to pta_i
                end if
            end if
        end for
    end if
end for
Sort NMOS branches from more to fewer transistors in BN
for all bn_i ∈ BN do
    if bn_i is not already mapped in NTA then
        Start new nta_i ∈ NTA containing bn_i
        for all bn_j ∈ BN do
            if bn_j is not already mapped in NTA then
                tranN = NMOS transistors when adding bn_j to nta_i
                inN = inputs used when adding bn_j to nta_i
                euN = true if ∃ Euler Path when adding bn_j to nta_i
                if (tranN ≤ T) ∧ (inN ≤ maxin) ∧ (euN) then
                    Add bn_j to nta_i
                end if
            end if
        end for
    end if
end for
for all pta_m ∈ PTA do
    for all nta_n ∈ NTA do
        if pta_m and nta_n are not already grouped then
            inPN = inputs used when grouping pta_m and nta_n
            if inPN ≤ maxin then
                Group pta_m and nta_n in a VCTA cell
            end if
        end if
    end for
end for
```

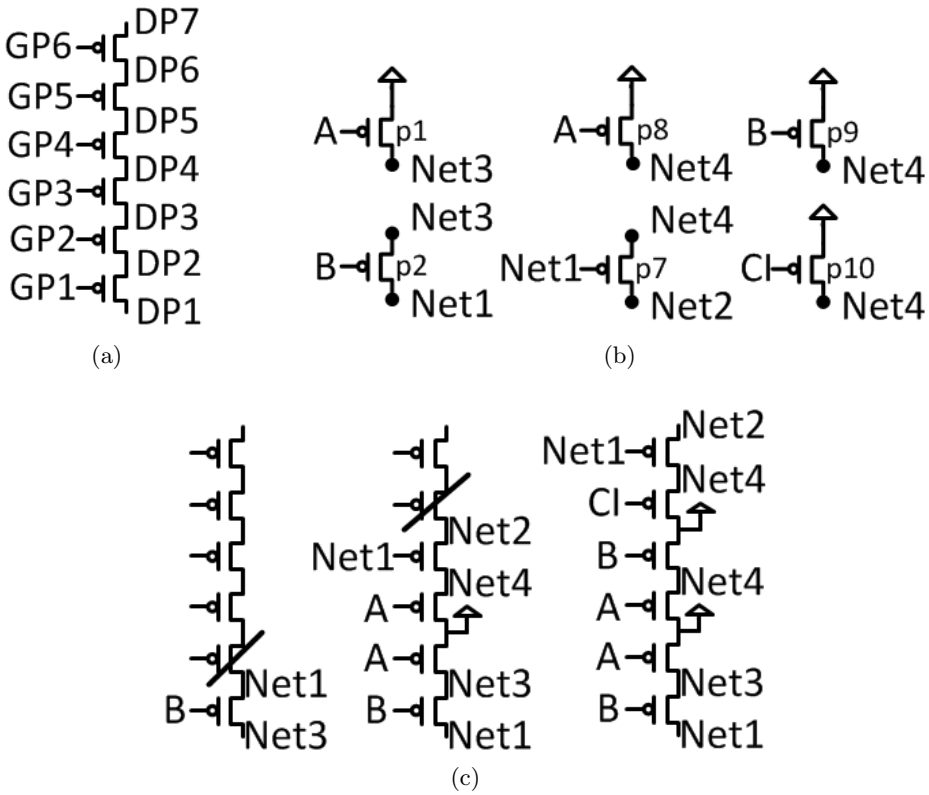
Algorithm 6.5 Euler Path verification.

```
1: ta = transistor array of T transistors to be filled
2: ta.Gate[IND] = gate of the transistor in index IND of ta
3: ta.Up[IND] = up connection of the transistor in index IND of ta
4: ta.Down[IND] = down connection of the transistor in index IND of ta
5: tr = set of numtran transistors to be fitted in ta with numtran ≤ T
6: tr.Gate[NUM] = gate of the transistor number NUM of tr
7: tr.Up[NUM] = up connection of the transistor number NUM of tr
8: tr.Down[NUM] = down connection of the transistor number NUM of tr
9: for i = 1 → numtran do Reinitialize/Empty ta
10:     ta.Gate[1] = tr.Gate[i]
11:     ta.Up[1] = tr.Up[i]
12:     ta.Down[1] = tr.Down[i]
13:     solution = COMPLETE(ta,tr,1)
14:     if solution == 0 then
15:         Reinitialize/Empty ta
16:         ta.Gate[1] = tr.Gate[i]
17:         ta.Up[1] = tr.Down[i]
18:         ta.Down[1] = tr.Up[i]
19:         solution = COMPLETE(ta,tr,1)
20:     end if
21:     if solution == numtran then
22:         Euler Path found, i = numtran to exit the for loop
23:     else
24:         No Euler Path found starting with tr number i
25:     end if
26: end for
27: function COMPLETE(ta,tr,last)
28:     solution = last
29:     if solution == numtran then return solution
30:     end if
31:     Save initial state of ta
32:     for i = 1 → numtran do
33:         if tr number i is not already placed in ta then
34:             if tr.Down[i] == ta.Up[last] then
35:                 ta.Gate[last + 1] = tr.Gate[i]
36:                 ta.Up[last + 1] = tr.Up[i]
37:                 ta.Down[last + 1] = tr.Down[i]
38:                 solution = COMPLETE(ta,tr,last + 1)
39:             else if tr.Up[i] == ta.Up[last] then
40:                 ta.Gate[transistorsplaced+1] = tr.Gate[i]
41:                 ta.Up[transistorsplaced+1] = tr.Down[i]
42:                 ta.Down[transistorsplaced+1] = tr.Up[i]
43:                 solution = COMPLETE(ta,tr,last + 1)
44:             end if
45:         end if
46:         if solution == numtran then return solution
47:         else if solution == 0 then Restore ta initial state
48:         end if
49:     end for
50:     if solution < numtran then return 0
51:     end if
52: end function
```

part is solved identically) considering T=6 transistors and 4 inputs available in the basic cell:

- Iteration 1: The branch order from more to fewer transistors is: BP4 (4 transistors), BP2 and BP5 (3 transistors), BP1 (2 transistors), BP3 and BP6 (1 transistor)
  - Try to group BP4 and BP2 (or BP5): KO because 7 transistors are required and T=6
  - Try to group BP4 and BP1: OK because only 6 transistors and 4 inputs (A, B, CI, Net1) are required, and also the Euler Path is found
  - RESULT: Cell 1 contains BP4 and BP1 (and BN4 and BN1 for the NMOS part)
- Iteration 2: Branch order: BP2 and BP5 (3 transistors), BP3 and BP6 (1 transistor)
  - Group BP2 and BP5: OK, 6 transistors, 3 inputs (A, B, CI), Euler Path found
  - RESULT: Cell 2 contains BP2 and BP5 (and BN2 and BN5 for the NMOS part)
- Iteration 3: Branch order: BP3 and BP6 (1 transistor)
  - Group BP3 and BP6: OK, 2 transistors, 2 inputs (Net1, Net2), Euler Path found
  - RESULT: Cell 3 contains BP3 and BP6 (and BN3 and BN6 for the NMOS part)
- After 3 iterations, all branches are grouped inside of 3 VCTA cells (Table 6.5)

Figure 6.5 shows the example for the PMOS part of the Cell 1 of the full adder. Assuming that the first transistor placed is p2, we first try the orientation



(a) (b) (c)

Figure 6.5: Euler Path verification: (a) VCTA schematic, T=6 serial PMOS transistors (b) Transistors in Cell 1 (c) From left to right: first try starting the path with transistor p2 in first orientation (no solution found because no other transistor is connected to Net1); second try with p2 in second orientation (after mapping the fourth transistor in the path no solution can be found because no other transistor is connected to Net2); final Euler Path found for Cell 1 (changing the choice made for the fourth transistor in the previous try).

with Net3 at the bottom and Net1 to the top (first try in Figure 6.5c). As there is no other transistor with Net1 as drain or source, this first try fails to find an Euler Path. The second try is to change Net1 at the bottom and Net3 to the top. With this orientation we can just place transistor p1 in the second transistor position sharing Net3 with p2. Then we have the choice between p8, p9 and p10 for the third transistor in the array. In the example we first chose p8. For the fourth transistor position we now can chose p7, p9 and p10. Choosing p7 leads to another fail as no other transistor has Net2 as drain or source (second try in Figure 6.5c). Therefore we change the last choice of p7 by p9. Doing so, we can then place p10 and p7 obtaining the complete Euler Path in the third try (last

try in Figure 6.5c). Other choices are also valid, but when an Euler Path is found it is chosen. The NMOS part and the rest of the cells Euler Paths are found identically.

6.1.3 VCTA Place

Once we have the resulting VCTA cells from the grouping step, the objective is to place them to minimize the final layout area. As the area can be increased when requiring extra routing tracks, we focus the placement on minimizing the inter-cell routing lengths. Such placement is beneficial given that shorter connections are likely to reduce the pressure on metal layers, thus reducing the likelihood of requiring extra rows or columns of VCTA cells to perform all connections.

Placement tools used for standard cell design flow can be used for the VCTA design flow. In particular, we have generated the required information of the VCTA basic cell to use Cadence SoC Encounter tool. For VCTA design, we only need the geometrical information for the VCTA basic cell implementation as well as the layout floorplan.

The aspect ratio of the floorplan is a parameter that can be given by the user. In our case, we have assumed a rectangular floorplan with an aspect ratio as close as possible to one, but selecting the number of rows and columns of VCTA cells to add the minimum number of spare cells to minimize the area required. For instance, for the full adder example where the grouping ends with 3 cells we can choose a floorplan with 3 rows and 1 column or with 1 row and 3 columns. A

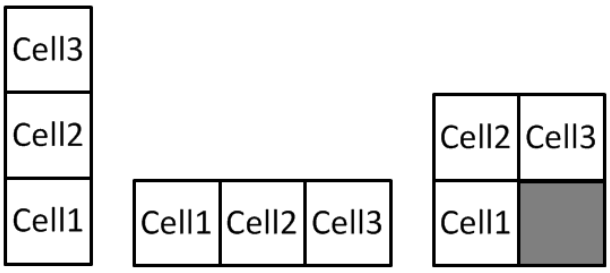


Figure 6.6: Full Adder place possibilities. Gray cell is a spare cell.

floorplan with 2 rows and 2 columns will add 1 spare cell that will represent a 25% area overhead. The 3 possibilities are shown in Fig. 6.6 including the positions where SoC Encounter places the 3 cells of the full adder. In any case, the placement can be done for any floorplan and aspect ratio determined by the user.

6.1.4 VCTA Routing

Initially we read the files generated from the previous automation steps that supplies us with the position and content of the VCTA cells required to implement the circuit. Then, the routing step is divided into two parts: intra and inter-cell routing. Optimization is finally performed using a simulated annealing algorithm to minimize the area. The routing flow is depicted in Figure 6.7.

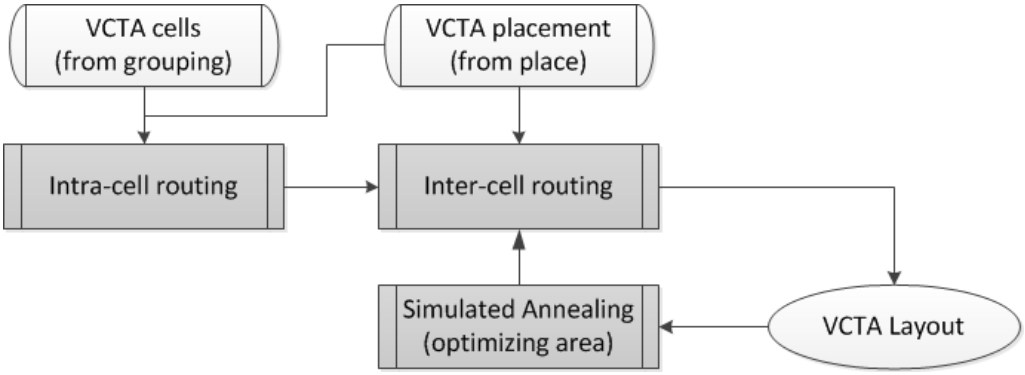


Figure 6.7: VCTA routing flow diagram.

6.1.4.1 Intra-cell routing

The intra-cell routing flow overview is shown in algorithm 6.6.

Algorithm 6.6 Intra-cell routing overview.

```
1: C = set of VCTA cells in the circuit
2: for all c_i ∈ C do
3:     Perform systematic intra-cell routing for c_i
4: end for
5: for all c_i ∈ C do
6:     Modify intra-cell routing for c_i considering neighbor cells
7: end for
```

We first use VCTA cell information coming from the VCTA grouping step to perform an initial systematic intra-cell routing. In particular, for each VCTA cell, we need the inputs of the cell and the transistors gate, drain and source connectivity in both the PMOS and NMOS transistor arrays. Knowing the VCTA basic cell resources in terms of interconnects (the number of metal lines available in each layer and all the possible positions for contacts and vias to connect them) we can then find the intra-cell configuration required for each cell (metal lines usage and contacts and vias configuration). We use algorithm 6.7 for this initial systematic intra-cell routing.

Algorithm 6.7 Systematic intra-cell routing for one VCTA cell.

```
1: Inp = set of inputs of the VCTA cell
2: Tran = set of transistors of the VCTA cell
3: for all Inp_i ∈ Inp do
4:     Assign first unused metal 1 line to Inp_i
5: end for
6: for all Tran_i ∈ Tran do
7:     Connect Tran_i gate to metal 1 line containing the required input
8: end for
9: for all Tran_i ∈ Tran do
10:    Connect Tran_i drain and source to first metal line free or to metal line
       containing the node required
11: end for
```

In more detail, we assign the cell inputs to unused metal 1 lines from left to right, following the order in which inputs are stored in the cell information. As we verify in the grouping step, we always have less inputs than metal 1 lines, therefore there is always a metal 1 line for each of the inputs. Then, we connect transistor gates to their inputs in metal 1 configuring the contacts required. By construction of the cell layout, each gate can be connected to all metal 1 lines so the connection can always be performed. Then, we connect drains and sources (that are already connected to a metal 2 line by layout construction) to inputs in metal 1 when required by configuring metal 1 to metal 2 vias. This is the case when the output of a logic function in the cell (in a drain or a source of a transistor) is the input of another transistor that is also mapped in the same cell. There is always connectivity between drains and sources in metal 2 lines and all

the metal 1 lines by construction. We also connect drains and sources to other drains and sources to perform parallel connections of transistors inside of the cell or to connect the outputs. In this case unused upper metal levels are configured. The first free metal line is used in each case starting again from left to right for vertical layers like metal 3, and from bottom to top for horizontal layers like metal 4. The outputs are therefore mapped to these metal lines by configuring the vias required. For the VCTA implementation used in the circuits presented in this work, enough resources are always found using up to metal 3, as few connections are required. In front of more congested intra-cell routing, upper metal levels can be used or the VCTA implementation has to be modified to include more metal lines (therefore increasing the cell area). Finally, we also connect drains and sources to ground and power supply when required configuring metal 2 to metal 3 vias. A metal 3 line is always reserved to ground and to power supply that is always accessible from metal 2 drains and sources.

After this first intra-cell configuration for all cells, the next step would be to perform the inter-cell routing. However, as explained in section 5.1.3, the inter-cell routing is performed extending metal lines between neighbor cells. In fact, having aligned input and output nodes in neighbor cells will allow direct extensions of metal lines for inter-cell connections without having to use extra routing resources in other metal layers. That is why, before performing the inter-cell routing step, the intra-cell routing initial decisions are modified to maximize the direct extensions of metal lines. For instance, if two neighbor cells in vertical share an input, the metal 1 line that will be chosen for this input will be the same in both cells so that the extension can be direct. This is achieved by modifying intra-cell connections cell by cell in an incremental way so that each new routed cell considers inputs/outputs of all the cells previously routed. Cells are routed from left to right and from bottom to top although any other ordering would be also valid (see algorithm 6.8). Note that when modifying a metal layer, the upper and lower metal layers also need to be reconfigured to maintain the same connectivity. This is done by moving accordingly the already placed contacts and

vias.

The initial input ordering in metal 1 for the 3 cells of the full adder example considering the place with 3 rows and 1 column is depicted in Figure 6.8a showing the 4 metal 1 lines per cell and the input associated. If no other metal layers can be used for routing these inputs, we can see how the lines can not be extended to perform the inter-cell connections and an extra column of spare cells has to be added to route the circuit. After reordering the inputs, inter-cell connections can be performed extending metal 1 lines directly (Figure 6.8b).

Algorithm 6.8 Intra-cell routing modification.

```
1: C = set of VCTA cells in the circuit
2: Sort C from cells at the bottom left corner of the layout to cells at the top
   right corner
3: for all c_i ∈ C do
4:     if c_i has a cell to the bottom then
5:         for all Vertical metal layers do
6:             Modify metal lines configuration to maximize direct extensions to
   cell to the bottom
7:             Restore connectivity with upper and lower horizontal metal layers
8:         end for
9:     end if
10:    if c_i has a cell to the left then
11:        for all Horizontal metal layers do
12:            Modify metal lines configuration to maximize direct extensions to
   cell to the left
13:            Restore connectivity with upper and lower vertical metal layers
14:        end for
15:    end if
16: end for
```

6.1.4.2 Inter-cell routing

This step performs the interconnections between cells. The input information is the list of nodes of the circuit that need to be interconnected and the cells involved in each node. To complete inter-cell routing each of the nodes of the circuit has to be connected. However which node is connected first will modify the routing decisions because our routing algorithm is greedy. For the first inter-cell routing try we choose this order randomly. As we will explain later, the order will be

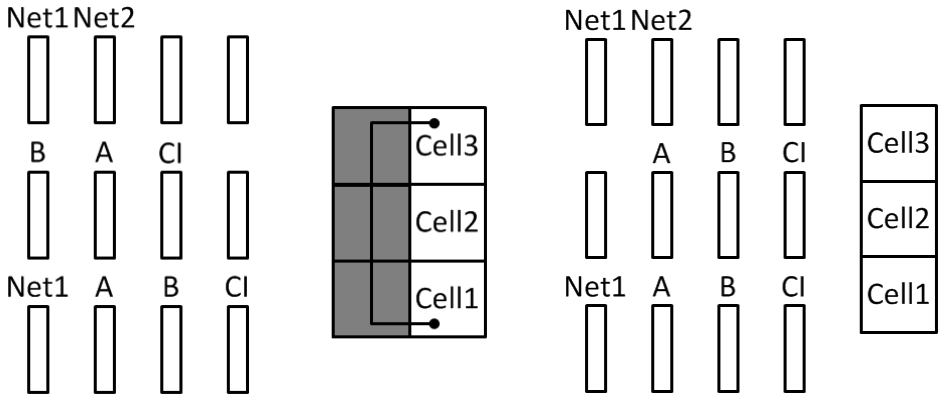


Figure 6.8: Full adder input reordering: (a) Initial input ordering: gray cells are spare cells added to perform inter-cell routing connections (b) Input reordering: no extra cells are required for inter-cell routing even for Net1 between cell 1 and cell 3.

modified to optimize the results obtained using a simulated annealing algorithm.

The connection of one node is done following the flow depicted in algorithm 6.9. First, we use Dijkstra algorithm to find the shortest path that minimizes interconnect length between each cell containing the node to be connected. If a path is found, second, there is a verification step where we check if this path can be performed using the metal grid resources available. If the path cannot be performed (due to routing congestion) we go back to find a new shortest path.

When the node connection fails even after trying all the possible shortest paths (due to routing congestion), we add routing tracks (rows and/or columns of free VCTA cells) to ensure that the node connection can be performed and we go back to the first step to find the shortest path that makes use of the new resources available. The connection can then be performed by paying an extra area overhead. However these extra cells can also be used to perform the pending connections. Therefore the connections done first will impact later connections, and that is why the order in which the nodes are connected will modify the final routing result.

To ensure that a routing track will always be available we need to verify

Algorithm 6.9 Inter-cell routing for one node in the circuit.

```
1: O = origin cell containing the node
2: D = destination cell containing the node
3: M = matrix of placed cells in the layout
4: R = routing resources in the cells of M
5: while node is not connected do
6:     path = FINDPATH(O,D,M)
7:     if path == false then
8:         ADDTRACKS(O,D,M)
9:     else
10:        routable = VERIFYPATH(O,D,R)
11:        if routable == false then
12:            Update M congestion
13:        else
14:            Connect O and D
15:            Update R resources
16:        end if
17:    end if
18: end while
19: function FINDPATH(O,D,M)
20:    Find shortest path using Dijkstra algorithm to connect O to D in M
21:    if cost == infinite then
22:        return false
23:    else
24:        return true
25:    end if
26: end function
27: function ADDTRACKS(O,D,M)
28:    Analyze O connectivity
29:    Analyze D connectivity
30:    Add corresponding rows and columns of spare cells to M
31: end function
32: function VERIFYPATH(O,D,R)
33:    P = path found by FINDPATH
34:    p_i ∈ P = cell O_i to cell D_i hops of the path
35:    for all p_i do
36:        if O_i and D_i cannot be connected using R then
37:            return false
38:        end if
39:    end for
40:    return true
41: end function
```

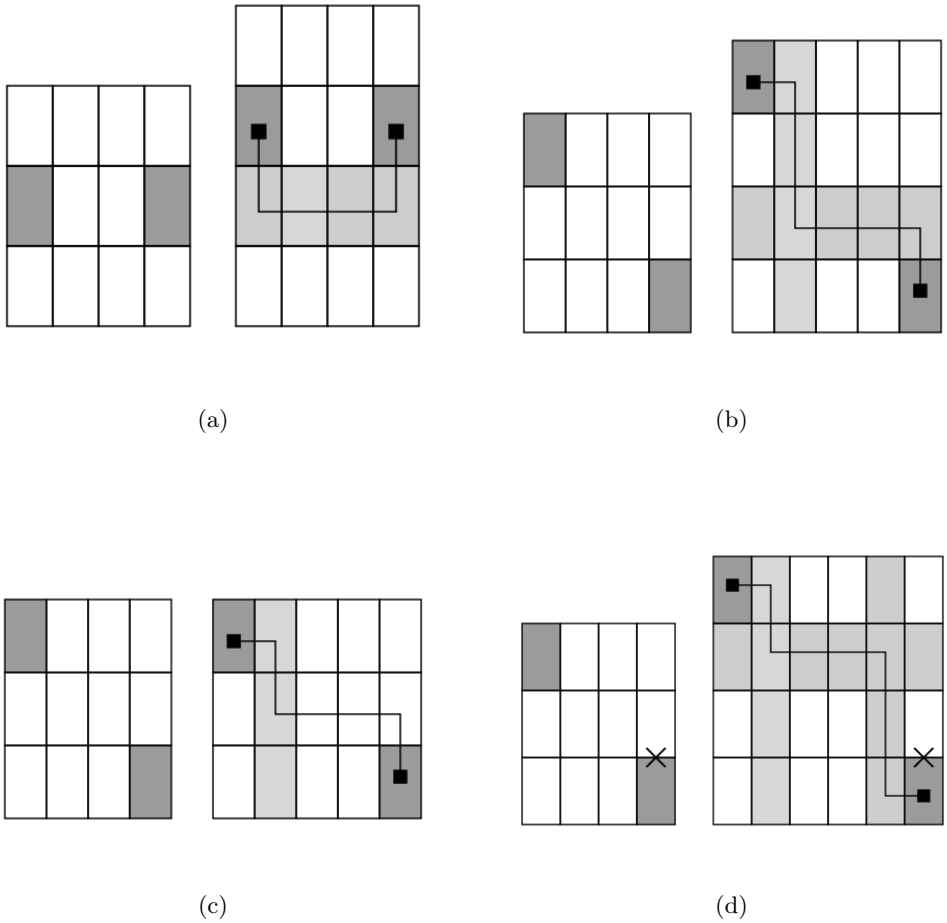


Figure 6.9: Routing congestion treatment: (a) Origin and destination cells at the same level and accessible from both vertical and horizontal directions: only one row added to ensure connectivity (b) Origin and destination cells are accessible from both vertical and horizontal directions: one row and one column added to ensure connectivity but we can try to add only the row or the column to reduce the area overhead like in Fig. 6.9c (c) Connection can be performed adding just one column: connectivity is not ensured but if the path is found we reduce the area overhead (d) One cell is not accessible from vertical direction: two columns and one row are required to ensure connectivity.

that the rows and/or columns that we add can be accessed from the origin and destination cells of the connection. That is why, we analyze these two cells connectivity to know if vertical and/or horizontal connections will be available when adding a free cell to the top or bottom or to the left or right directions respectively. In this way we can select which rows and columns to add for the

connection under study. A maximum of three rows and columns are required depending on the positions of the cells. In Figure 6.9 some of the possibilities are shown, depending on relative position of origin and destination cells and on their connectivity in vertical and horizontal. Then, using these added cells we can ensure connectivity by defining systematically the path that uses these cells. However, to minimize the number of cells added, before using the complete rows and columns found, we try if congestion is avoided by adding progressively only one of the rows or columns, then only two of them when possible. The idea is to add these rows or columns and try to find new paths that can now be possible with the new resources.

For path lookup the undirected and unweighted graph where Dijkstra is applied has as vertices the placed VCTA cells and the edges indicate whether there is connectivity or not between these cells (the M structure of algorithm 6.9). We define that adjacent cells are at distance one and the other cells are at infinite distance. Adjacent cells can also be at infinite distance if routing congestion has been updated from the feedback of the verification step.

In the second step to connect one node, we consider the metal grid resources available between each adjacent cell of the path (the R structure in algorithm 6.9). This problem is similarly solved with Dijkstra algorithm. In this case vertices are metal lines and edges values represent the distance associated to the cost of connecting metal lines. The graph is therefore undirected but weighted and in fact the edge weights are used to prioritize the use of metal inter-cell extensions and the highest levels of metal. In this way we reserve lower metal layers to local routing and use highest levels for long interconnections.

6.1.4.3 Simulated Annealing

The extra rows and columns added when routing congestion is found lead to an area increase. That is why we have included an optimization step using simulated annealing that is performed when at least one row or column has been added to the original floorplan from the place step (the initial solution). Algorithm 6.10

shows the simulated annealing flow. The cost optimized is the final number of VCTA cells required for the routing (directly related to the final layout area) and, as explained before, the solution that is chosen for each annealing's iteration is the order in which the nodes are connected. To generate the new nearby solution for each iteration we randomly modify the order for a small percent of the nodes (e.g., 5%). A solution with lower area will always be accepted and saved, but a solution with higher area can also be accepted to avoid remaining in a local minimum of the optimization. To decide if the new solution with higher area is accepted, we use constant cooling of the annealing temperature for each iteration and the acceptance probability function presented in [77] based on Boltzmann probability factor. Finally, no matter if the solution has been accepted or not, we check if the condition to end the annealing has been reached (e.g., the number of annealing iterations programmed have been done, or the routing has been done without adding rows or columns). If this is the case, the simulated annealing outputs the last solution saved. Otherwise, it starts a new iteration.

6.1.5 VCTA Layout Generation and Verification

Place and routing steps provide the position of the VCTA basic cells, the metal extensions, the contact and vias positions to configure them, and also the input and output information. Based on this information we generate the suitable scripts to create the layouts (in SKILL language for Cadence). For schematic generation the scripts are generated directly from the circuit transistor netlist description so that at the end of the flow we are able to verify our designs using LVS. DRC is error free as the VCTA basic cell is already DRC compliant by construction. The VCTA basic cell layout and schematic for the chosen VCTA implementation are the only data required.

Algorithm 6.10 Simulated annealing for inter-cell routing. The number of programmed iterations is *numiterations*.

```
1: N_min = minimum number of cells in the layout
2: N_acc = number of cells of the accepted solution
3: Iterations = number of annealing iterations performed
4: L = list of nodes to be connected
5: Sort L randomly
6: for all L_i ∈ L do
7:     Perform inter-cell routing
8:     N = number of cells of the solution
9: end for
10: Save initial L order solution
11: N_acc = N
12: Iterations = 0
13: while (N_acc > N_min) ∧ (Iterations < numiterations) do
14:     Iterations = Iterations+1
15:     Modify L order
16:     for all L_i ∈ L do
17:         Perform inter-cell routing
18:         N = number of cells of the solution
19:     end for
20:     if N < N_acc then
21:         Save L order solution
22:         N_acc = N
23:     else
24:         Use a probability function to decide if the solution is accepted
25:         if Solution is accepted then
26:             Save L order solution
27:             N_acc = N
28:         end if
29:     end if
30: end while
```

6.2 Results and Simulations

6.2.1 Manual VCTA versus Automatic VCTA Flow

We have used the proposed automatic VCTA flow to generate the layouts of the 32-bit adders (Carry-Ripple, Look-Ahead and Kogge-Stone) previously generated manually with the VCTA basic cell implementation including T=6 transistors and 4 inputs per cell (see chapter 5). The number of VCTA cells required for manual and automatic implementations are summarized in Table 6.6. The complexity of the circuits is indicated in terms of total number of transistors. As it is shown, the number of cells is equivalent for the smallest adders (from none to 4 cells reduction) and lower for the most complex adder (76 cells reduction) that has a total number of 2184 transistors. This is due to the fact that complex designs are more difficult to be generated manually. Figure 6.10 depicts manual and automatic layouts for the Kogge-Stone adder. As shown, the automatically-generated layout requires fewer cells than the manual one.

Table 6.6: 32-bit adder results.

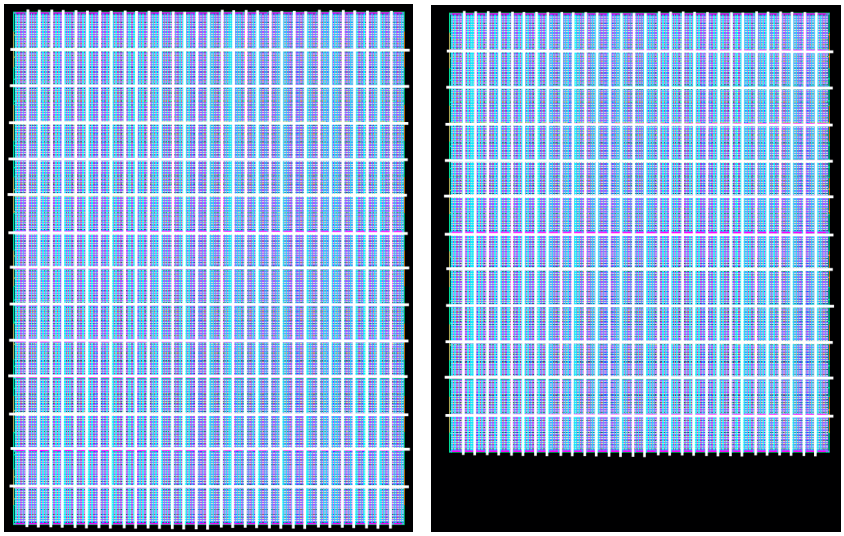
32-bit adder	Transistors	Manual	Automatic
Carry-Ripple	896	96 cells	96 cells
Look-Ahead	1432	184 cells	180 cells
Kogge-Stone	2184	448 cells	372 cells

6.2.2 Standard Flow versus VCTA Flow

We have used our automation tool to obtain the VCTA layouts for the whole set of ISCAS'85 benchmark circuits [71] using the 45nm technology node NCSU Free PDK [72]. Again the VCTA basic cell implementation has T=6 transistors and 4 possible inputs per cell. In this case, we have used standard cell layouts generated from the OSU library [74] to evaluate our tool.

6.2.2.1 Grouping

To evaluate the grouping step, we show in Table 6.7 the number of VCTA cells required to implement the circuits, as well as the resulting transistor occupancy,



(a) (b)

Figure 6.10: VCTA Kogge-Stone layouts: (a) Manual: 32 columns, 14 rows (b) Automatic: 31 columns, 12 rows. The cell grid has been superimposed for illustrative purposes.

that gives the efficiency of grouping. In average, grouping obtains around 60% occupancy. Note that the occupancy is not particularly worsened with the increase of circuit complexity as we have the minimum occupancy for c432 circuit. The total number of transistors of the different ISCAS'85 is also included as an indicator of the complexity of the circuits.

Occupancy is highly related to the VCTA implementation as it will vary when considering more or less transistors or inputs available per cell. Therefore occupancy can be improved by optimizing the choice of the VCTA cell resources. For instance, in most of the cases we use at most 4 transistors of the 6 available in the cell. Selecting T=4 may improve occupancy while reducing also the VCTA cell area. However, such change may have side effects on routing because smaller cells have fewer metal lines to route signals. Studying different VCTA implementations is part of our future work. We only study one VCTA implementation because our goal is to demonstrate that the automation is feasible.

Occupancy also depends on the logic functions present in the circuits. Note

that our starting point are those netlists generated for standard cells. A VCTA-oriented logic synthesis step could generate transistor netlists with limited number of transistors per branch, or with limited number of inputs per logic function, that would fit a particular VCTA cell implementation in terms of transistors and inputs, therefore increasing the efficiency of our proposal because the occupancy would be higher and inter-cell routing would difficultly require more added routing tracks.

Table 6.7: ISCAS'85 results for VCTA grouping.

ISCAS	Transistors	Cells Grouping	Occupancy (%)
c17	24	3	66.7
c432	578	97	49.7
c499	1646	200	68.6
c880	1310	200	54.6
c1355	1684	207	67.8
c1908	1204	149	67.3
c2670	2144	301	59.4
c3540	2936	437	56.0
c5315	4748	603	65.6
c6288	9656	1500	53.6
c7552	6406	882	60.5

6.2.2.2 Place

The input needed to perform the place step is the layout floorplan. Assuming rectangular floorplans, we select the number of VCTA columns and rows that can contain the cells obtained from the grouping step in order to minimize the cells that will not be occupied. Rows and columns selected are indicated in Table 6.8 along with the number of cells of the floorplan. Only in some cases a small number of cells is added by the place step to the grouping step.

Note, however, that the geometry of the area available is an input of the automation process. We have arbitrarily selected rectangular shapes resembling a square while minimizing the number of empty cells. Note that the VCTA cells used have a height higher than their width, and therefore layouts show more columns than rows.

Table 6.8: ISCAS'85 results VCTA place.

ISCAS	Cells Grouping	Cells Place (Rows/Cols)
c17	3	3 (1x3)
c432	97	98 (7x14)
c499	200	200 (10x20)
c880	200	200 (10x20)
c1355	207	207 (9x23)
c1908	149	150 (10x15)
c2670	301	301 (7x43)
c3540	437	437 (19x23)
c5315	603	615 (15x41)
c6288	1500	1500 (30x50)
c7552	882	882 (14x63)

6.2.2.3 Routing

The routing overhead are rows and columns added for congestion reasons. Results are shown in Table 6.9. Few rows or columns are required even for large circuits. However, the resulting overhead of this added routing tracks is strongly dependent on the aspect ratio of the floorplan. For instance, for c7552 we only need to add 2 rows, but as each row accounts for 63 added cells, the overhead is 126 cells. In these cases, the aspect ratio can be modified. However, we have considered that aspect ratio is an input of the design and we have not used it to optimize the results.

Table 6.9: ISCAS'85 results for VCTA routing.

ISCAS	Cells Place (Rows/Cols)	Cells Routing (Rows/Cols)
c17	3 (1x3)	3
c432	98 (7x14)	98
c499	200 (10x20)	200
c880	200 (10x20)	200
c1355	207 (9x23)	207
c1908	150 (10x15)	150
c2670	301 (7x43)	344 (8x43)
c3540	437 (19x23)	480 (20x24)
c5315	615 (15x41)	615
c6288	1500 (30x50)	1500
c7552	882 (14x63)	1008 (16x63)

6.2.2.4 Simulated Annealing

The annealing has been very useful particularly for congested layouts. For instance, for c7552 the initial solution required 1260 cells and we finally reached 1008 cells for a total reduction of 252 cells. In fact, most of the layouts ended

with no extra VCTA cells added for routing congestion as shown in Table 6.9.

6.2.2.5 Comparison to Standard Cells

Regarding layout area, in Table 6.10 we present the area overhead of VCTA layouts compared to the standard cell layouts using the free OSU library [74]. It reaches a maximum of 32.9% for the most complex ISCAS but overhead is in general small. Moreover, the c5315 results show that good results can be obtained for circuits with a high number of transistors. c17 is a very particular case with very few transistors where VCTA is even smaller than standard cells, but this also occurs for c1908. In general VCTA designs are slightly larger than standard cell designs despite regularity in principle imposes an area overhead.

Note that these area results are obtained for the 45 nm free OSU library of standard cells, that is an academical library. It is not an optimized commercial library like the previously used for area evaluations in the 90 nm node in chapter 5. In that case for the STD90 layouts compared to VCTA layouts, area overheads reached up to a 2.63x ratio for the KS32 adder. However, considering the reduction of the number of VCTA cells required for this KS32 adder using the automated flow (see Table 6.6), the area overhead ratio has been reduced to 2.18x, close to the ratios of 1.93x and 1.88x obtained in the 90 nm technology node for the smaller CR32 and CLA32 adders. The benefits of the automation of the VCTA physical design are not in the comparison with standard cell designs, even if in this particular case in the 45 nm node we have obtained comparable areas for VCTA layouts and standard cell layouts. The first benefit of the automation is to allow the design of complex circuits (unfeasible or requiring a tremendous amount of time if developed manually). Then, the second benefit is to obtain comparable area overheads to the ones for small VCTA layouts developed manually.

Regarding delay and energy, we have performed electrical Hspice simulations to obtain worst-case delay (WCD) and average energy consumption (AVGE) for 1,000 inputs generated with FSIM [78] for each ISCAS'85 circuit. In Table 6.10 we present the ratios obtained for VCTA layouts over standard cells. Delay is in

average 2.8 times higher for VCTA circuits and energy is 2.2 times higher. Such overheads are somehow expected given that the target of our flow is minimizing area and, even more important, proving that VCTA design flows are feasible. Thus VCTA design flows targeting other parameters such as delay or energy are part of our future work. However, we expect that the area results will be worsen.

Nevertheless, as explained in previous chapters, note that those overheads can be partially mitigated by the fact that VCTA regular circuits tolerate better process variations and that advanced technology nodes can be used for VCTA circuits earlier than for standard cells. Similarly, VCTA-oriented logic synthesis step could further mitigate overheads as well as other VCTA cell implementations which will be the focus of our future work.

Table 6.10: ISCAS'85 VCTA vs Standard Cells.

ISCAS	Area overhead (%)	WCD ratio	AVGE ratio
c17	-18.0	2.0	2.5
c432	10.8	2.8	2.5
c499	4.9	2.8	1.7
c880	8.5	2.9	2.4
c1355	5.1	2.6	1.8
c1908	-3.1	2.5	1.8
c2670	18.1	3.0	2.2
c3540	16.2	2.8	2.5
c5315	1.4	4.2	2.2
c6288	32.9	2.4	2.8
c7552	25.3	2.8	2.3

6.3 Conclusion

In this chapter we have presented the steps required for the physical design flow of the VCTA regular fabric. Compared to the standard flow, only a single VCTA cell is required instead of the whole library of cells. Then, we need two extra steps, that are the grouping step to map transistors of the circuit into VCTA, and the intra-cell routing step, to configure the VCTA cells required for the circuit (that are in this case generated on the fly, not like for predefined standard cells that cannot be optimized for the particular circuit). Finally, even if the VCTA inter-cell routing has the same function than the routing step for standard cells, in this

case, we need to consider the regularity constraints of the VCTA via-configurable structure.

The results for the automation tool developed have been demonstrated in ISCAS'85 benchmark circuits, providing evaluations for each of the steps in order to reduce the final area. The grouping step is evaluated according to VCTA cell transistor occupancy, that reaches around 60%. Place and routing are evaluated in terms of the number of cells added to generate the resulting placed and routed layout, that represent no more than 12.5% of the total number of cells. In particular, final results show that our area-oriented VCTA design flow produces layouts with comparable areas to standard cell areas using the OSU library (from -18% less area to 32.9% area overhead). Moreover we have demonstrated the feasibility of the design of complex circuits using our new VCTA automation tool. Future works will focus on reducing the energy and delay overheads starting from a VCTA-oriented logic synthesis (results presented come from the standard cell logic synthesis), then studying the impact of the VCTA cell implementation (that can also have influence on the area of the final layout), and also including energy and delay constraints in the automation flow.

Chapter 7

FOCSI Layout Regularity Metric

Usually regular techniques offer worse area, delay and energy consumption than the non-regular design approaches but, according to the degree of regularity, they reduce cost and time associated to lithography enhancement techniques and therefore systematic yield loss. Area is measured directly from the layout design, and delay and energy consumption can be predicted by simulation. However, there is not a clear method to measure layout regularity nor its impact on variability. There are tools that analyze layouts from the design for manufacturability point of view, but none of them focus on layout regularity analysis (see chapter 2). In fact, existing tools for layout analysis are very time-consuming for large layouts, in particular when having to simulate the complex lithography system and resolution enhancement techniques.

In this chapter we propose a new layout regularity metric called Fixed Origin Corner Square Inspection (FOCSI). FOCSI quantifies regularity allowing an accurate, deterministic and unambiguous comparison of layout designs. We show how layouts can be sorted based on their degree of regularity. We also provide a methodology using the Monte Carlo analysis to evaluate and understand the impact of regularity on process variability. FOCSI gives printability information from the regularity measurement in an early stage and in an easy and fast way.

In fact, as we will show next, regularity itself does not reduce process variability, but it allows further steps to optimize the manufacturing process and to obtain better printability.

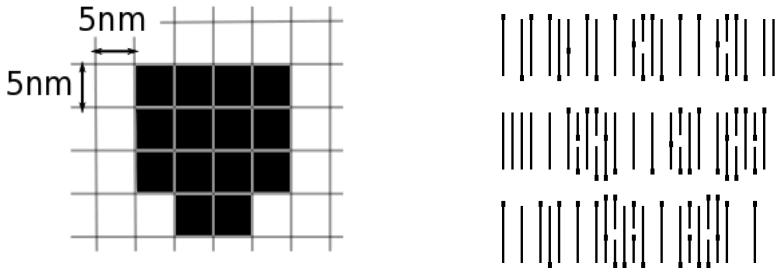
The structure of the chapter is as follows. First, in section 7.1 we present the problem addressed, we provide a definition of regularity and we propose and formulate FOCSI layout regularity metric from the single layout layer to the complete layout. Third, in section 7.2 we give FOCSI single layer measurements examples using the ISCAS'85 layouts in the 65 nm and 45 nm technology nodes presented in chapter 4. Then, in section 7.3 we give the resulting FOCSI complete layout calculation examples. Fifth, in section 7.4 we present the methodology using FOCSI results and the Monte Carlo analysis to evaluate the benefits of layout regularity in terms of variability. Finally, in section 7.5 we provide conclusions.

7.1 FOCSI formulation

7.1.1 Problem Statement

A metric is by definition a system of related measures that facilitates the quantification of some particular characteristic. In our case the characteristic to quantify is the amount of layout regularity. The metric function has to give to a layout a value indicating how much regular it is. Then, for any two layouts, it can determine which of them has higher regularity.

To the best of our knowledge, the only method that has already been used for this purpose is a visual comparison of a two-dimensional Fourier transform (see chapter 2). However, the two-dimensional Fourier transform does not quantify regularity. We give examples in next sections. It is a graphical representation giving an intuitive and qualitative measure of regularity. It can be used to compare regular versus non-regular layouts but it is difficult to use it to compare similar layouts in terms of regularity like for instance two layouts developed with regular design techniques. That is why we propose FOCSI: a new layout regularity metric that allows a deterministic and unambiguous regularity comparison for any



(a) Manufacturing grid (b) Complete layout layer

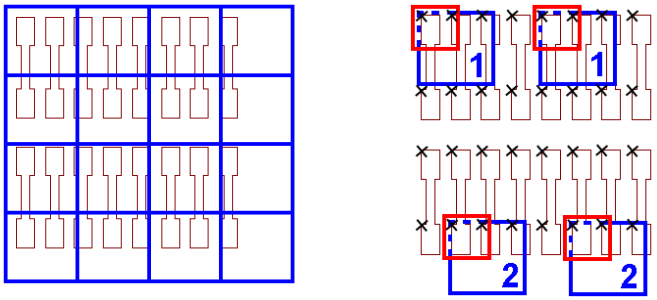
Figure 7.1: Granularity extremes.

pair of layout designs. We show in next sections that our metric can determine which of the layouts under study is more regular even if they have similar degrees of regularity. We present the broadest definition of FOCSI in order to illustrate the possibilities offered by our metric.

7.1.2 Layout Regularity Definition

We define layout regularity for a given layout layer as the property of this layer to be generated by a reduced number of different layout areas of a given shape and size (e.g., squares of 160 nm x 160 nm). We will refer to these layout areas as LAs, and to the different types of LAs in the layer as layout generators. Therefore, the lower is the number of generators that can be found amongst the LAs the higher the regularity is. The maximum regularity will be achieved when a single generator can be used to generate the whole layer by repeating it along all the layer. On the other hand, the minimum regularity will occur when all LAs are unique, and therefore, there is no repetition at all (all LAs are generators).

Regularity can be studied at different granularities depending on the size considered for LAs inspected. On one hand, the smallest LA that can be considered is defined by the manufacturing grid so that the LA considered will be a square with this manufacturing grid as both dimensions (Figure 7.1a). Possible LAs are in this case binary. We can only find in the layout two different generators: one containing the material of the layer inspected and the other containing nothing. Therefore all layouts inspected will have the same regularity and that is why we



(a) (b)

Figure 7.2: Regularity inspection: (a) Contiguous LAs (b) FOCSI methodology.

are not interested in using this extreme. On the other hand, the highest LA that can be considered is the complete layout layer (Figure 7.1b). Again, all layouts will present the same regularity being generated by a single generator. Thus, regularity must be evaluated using LAs of a size between these two extremes. The choice of the size of these LAs will be explained in next sections.

7.1.3 FOCSI Proposal

In FOCSI, we propose the use of square areas as LAs. In order to explore a layout layer to find out the number of square area generators, we need to find the LAs of sides size multiple of the manufacturing grid (e.g., side of 160 nm = 32 times the 5 nm manufacturing grid) and then to compare one to each other noting the number of different ones.

The Fixed Origin Corner Square Inspection (FOCSI) proposal first explores the layout layer in order to detect all upper left pattern corners and then considers these corners as the origins of the square LAs to be compared. The origin of the LAs can be selected differently (e.g., upper right pattern corners) but always ensuring that LAs are aligned to layout patterns. Having origins of the LAs not aligned to the patterns (e.g., dividing the layout in contiguous LAs) can lead to a situation where regularity is not captured. In Figure 7.2a it can be seen how all the LAs are different while the visual inspection of the layout shows an important degree of regularity. That is why FOCSI is pattern oriented and it inspects at

least one LA per pattern. Figure 7.2b depicts how FOCSI works. Black crosses indicate all the corners considered. Types 1 and 2 generators can be detected. Note that in this figure different LA sizes are also illustrated with red and blue squares. Once the corners are fixed, various sizings can be applied to squares in order to evaluate different granularities of regularity.

7.1.4 Single Layout Layer FOCSI

To implement the FOCSI proposal, we first transform the layout layer into a bitmap, where each minimum size square (see Figure 7.1a) is represented by a bit. The codification used assigns a 0 value to the bit when the layout is empty and a 1 value to represent the layer material (e.g., polysilicon, oxide diffusion). The whole layout layer is therefore codified as a matrix of 0's and 1's. Then, we find all upper left pattern corners from where LAs are defined. Finally, these LAs are compared sample by sample against each other in order to calculate the number of different generators the layout layer has. Two LAs are only considered identical if all their samples are identical. The result of this step of FOCSI metric is the number of different generators of the layout layer under study and for the LA sizing defined. We will refer to this number of generators as R_{layer} (e.g., R_{OD}, R_{PO} or R_{M1} for the number of generators in the oxide diffusion layer, in the polysilicon layer and in the metal 1 layer respectively). Therefore the lower R_{layer} is the higher regularity is for this layer.

7.1.5 Complete Layout FOCSI

The final step to obtain a comprehensive layout regularity value (R_{layout}) is to combine all different layout layers regularity values (R_{layer}) calculated. Defining M as the number of layout layers considered, we propose to combine these M measurements assigning weights to each one of them. In general, the layout regularity R_{layout} can be then calculated as follows:

$$R_{layout} = \sum_{j=1}^{M} \beta_j . R_{layerj} \tag{7.1}$$

where β_j are layout layers weights and R_{layerj} are the regularities measured for the M layout layers considered. In order to enable the comparison of the R_{layout} measures from different layouts we propose that the β_j parameters also fulfill the following property:

$$\sum_{j=1}^{M} \beta_j = 1 \tag{7.2}$$

Each of the M layer regularities will have a different β_j weight depending on process conditions. We give examples in next sections. R_{layout} can be considered as the final FOCSI metric result. As for R_{layer}, the lower R_{layout} is the higher regularity is. Note that this final R_{layout} is not needed if the objective is the evaluation of regularity for a concrete layer.

7.2 FOCSI for single layers

7.2.1 Granularities considered

We show single layout layer regularity measurements (R_{layer}) for LA sizings of 160 nm, 320 nm, 640 nm, 1280 nm and 2560 nm in order to see the evolution of regularity of the designs under study. The 160 nm minimum square size is chosen to ensure that at least 1 or 2 material polygons are present in the area considered so that we can measure micro-regularity. The 2560 nm maximum size have been chosen to consider macro-regularity.

7.2.2 ISCAS'85 layout results

We have used FOCSI to measure ISCAS'85 benchmarks layouts in the 45 nm and 65 nm technology nodes. For the 45 nm node, we compare VCTA layouts (VCTA45) and standard cell layouts (STD45), and for the 65 nm node, the

comparison is between the new Robust standard cell approach (Robust65) and the commercial standard cell version (STD65), all of them presented in chapter 4.

STD45 and STD65 designs are based on the reuse of layout cells with fixed height bounded by the power supplies. However they can have different widths and depending on the function implemented they can include transistors and interconnects in very dissimilar configurations. Moreover, standard cell libraries can include more than 1000 different cells, and therefore a huge number of placing and routing configurations are possible. Resulting standard cell layouts are therefore expected to have a low degree of regularity because of both the irregular internals of the cell and the irregularity across different neighbor cells.

Regarding the Robust65 library that we have developed for the thesis, to improve macro-regularity, we have chosen a reduced number of cells based on the library proposed in [75] so that the amount of placing and routing configurations compared to the classical standard cell library is reduced. Moreover, to increase micro-regularity, all layers are oriented in only one dimension, shapes are placed at constant pitch and all transistors have the same sizing. Transistor fingering is used to obtain different sizes.

Regarding VCTA, as it focuses on maximizing layout regularity (see chapter 5), we expect the higher regularity amongst the above mentioned layout styles.

Results obtained for polysilicon (PO), oxide diffusion (OD) and metal 1 (M1) layout layers are shown in Tables 7.1 to 7.3. We have chosen these three layout layers because they are the most representative of the front-end and back-end process. PO and OD define the transistor active areas, and the polysilicon gate critical dimension variation have a tremendous impact on the timing and energy consumption of digital integrated circuits. M1 layer is representative of the interconnect structure. The total number of LAs inspected is given for each of the layers to show the complexity of the circuits.

Table 7.1: ISCAS'85 FOCSI PO results

ISCAS layout	R_{PO} generators					Number of LAs
	160 nm	320 nm	640 nm	1280 nm	2560 nm	
c17 VCTA45	1	4	15	33	80	96
c432 VCTA45	1	4	15	39	120	3136
c499 VCTA45	1	4	15	39	120	6400
c880 VCTA45	1	4	15	39	120	6400
c1355 VCTA45	1	4	15	39	120	6624
c1908 VCTA45	1	4	15	39	120	4800
c2670 VCTA45	1	4	15	39	120	11008
c3540 VCTA45	1	4	15	39	120	15360
c5315 VCTA45	1	4	15	39	120	19680
c6288 VCTA45	1	4	15	39	120	48000
c7552 VCTA45	1	4	15	39	120	32256
c17 STD45	4	5	7	13	19	23
c432 STD45	18	33	78	316	529	535
c499 STD45	32	69	142	456	1477	1555
c880 STD45	37	81	180	577	1125	1166
c1355 STD45	35	70	151	488	1497	1564
c1908 STD45	38	79	158	472	1089	1140
c2670 STD45	40	83	207	830	1865	1938
c3540 STD45	41	92	241	1200	2529	2666
c5315 STD45	41	92	265	1419	3969	4292
c6288 STD45	23	47	102	588	5144	8625
c7552 STD45	42	93	313	1916	5480	5770
c17 Robust65	2	9	15	30	36	36
c432 Robust65	2	17	46	397	643	686
c499 Robust65	2	18	65	763	1821	2222
c880 Robust65	2	20	66	761	1471	1614
c1355 Robust65	2	18	71	779	1768	2182
c1908 Robust65	2	21	66	678	1340	1606
c2670 Robust65	2	25	93	1208	2544	2817
c3540 Robust65	2	25	88	1353	3121	3456
c5315 Robust65	2	25	94	2104	5546	6402
c6288 Robust65	2	19	65	1857	7880	11287
c7552 Robust65	2	25	92	2311	6529	7688
c17 STD65	5	12	21	22	22	22
c432 STD65	30	138	380	443	462	462
c499 STD65	26	113	431	889	1231	1237
c880 STD65	31	197	720	996	1093	1097
c1355 STD65	25	127	501	947	1228	1235
c1908 STD65	24	147	543	855	1008	1010
c2670 STD65	53	261	971	1559	1787	1796
c3540 STD65	55	343	1444	2241	2434	2444
c5315 STD65	67	343	1661	3576	4319	4355
c6288 STD65	28	134	1159	5680	9965	10332
c7552 STD65	71	393	2082	4359	5150	5169

Table 7.2: ISCAS'85 FOCSI OD results

ISCAS layout	R_{OD} generators					Number of LAs
	160 nm	320 nm	640 nm	1280 nm	2560 nm	
c17 VCTA45	2	2	3	3	7	10
c432 VCTA45	2	2	3	3	12	256
c499 VCTA45	2	2	3	3	12	516
c880 VCTA45	2	2	3	3	12	516
c1355 VCTA45	2	2	3	3	12	534
c1908 VCTA45	2	2	3	3	12	388
c2670 VCTA45	2	2	3	3	12	886
c3540 VCTA45	2	2	3	3	12	1223
c5315 VCTA45	2	2	3	3	12	1566
c6288 VCTA45	2	2	3	3	12	3791
c7552 VCTA45	2	2	3	3	12	2560
c17 STD45	2	2	3	7	13	15
c432 STD45	2	9	39	168	281	286
c499 STD45	2	9	36	154	512	541
c880 STD45	2	12	52	251	506	524
c1355 STD45	2	9	34	170	521	555
c1908 STD45	2	9	41	174	412	436
c2670 STD45	2	12	57	315	746	781
c3540 STD45	2	12	67	467	1029	1092
c5315 STD45	2	10	65	514	1486	1619
c6288 STD45	2	9	44	322	1737	3159
c7552 STD45	2	18	79	698	2006	2127
c17 Robust65	1	2	5	15	19	22
c432 Robust65	1	2	8	51	264	366
c499 Robust65	1	2	8	53	353	700
c880 Robust65	1	2	8	58	395	660
c1355 Robust65	1	2	8	60	363	690
c1908 Robust65	1	2	8	50	320	554
c2670 Robust65	1	2	8	68	707	1150
c3540 Robust65	1	2	8	69	908	1528
c5315 Robust65	1	2	8	69	1332	2602
c6288 Robust65	1	2	8	70	2302	5342
c7552 Robust65	1	2	8	72	1525	3000
c17 STD65	7	11	16	17	17	17
c432 STD65	23	53	171	254	269	278
c499 STD65	19	77	286	604	808	814
c880 STD65	30	71	304	554	617	617
c1355 STD65	21	90	347	679	857	859
c1908 STD65	19	58	252	476	580	581
c2670 STD65	41	116	502	884	1024	1052
c3540 STD65	47	138	567	1107	1200	1231
c5315 STD65	41	110	762	1675	2073	2085
c6288 STD65	24	76	1064	3931	6211	6504
c7552 STD65	53	194	1038	2198	2595	2647

Table 7.3: ISCAS'85 FOCSI M1 results

ISCAS layout	R_{M1} generators					Number of LAs
	160 nm	320 nm	640 nm	1280 nm	2560 nm	
c17 VCTA45	5	10	17	45	65	68
c432 VCTA45	5	10	19	81	179	2031
c499 VCTA45	5	11	19	77	171	4129
c880 VCTA45	5	11	23	119	313	4115
c1355 VCTA45	5	11	19	77	170	4280
c1908 VCTA45	5	9	18	67	135	3103
c2670 VCTA45	5	10	24	148	424	7081
c3540 VCTA45	5	11	23	153	491	9815
c5315 VCTA45	5	11	26	157	528	12582
c6288 VCTA45	5	11	23	108	271	30826
c7552 VCTA45	5	10	25	157	458	20671
c17 STD45	11	15	21	28	39	47
c432 STD45	169	377	662	982	1022	1023
c499 STD45	203	442	851	1761	2182	2192
c880 STD45	250	581	1089	1872	2044	2049
c1355 STD45	175	386	784	1800	2211	2216
c1908 STD45	201	444	789	1418	1650	1653
c2670 STD45	277	666	1342	2664	3057	3064
c3540 STD45	358	915	2043	4075	4521	4528
c5315 STD45	374	948	2200	5342	6618	6643
c6288 STD45	246	659	1581	6163	13446	13937
c7552 STD45	437	1223	2900	7373	8796	8848
c17 Robust65	2	8	34	62	76	76
c432 Robust65	8	36	162	777	1335	1356
c499 Robust65	9	56	258	1200	3249	3420
c880 Robust65	10	53	280	1387	2801	2825
c1355 Robust65	10	49	230	1103	3116	3345
c1908 Robust65	8	47	233	1103	2379	2554
c2670 Robust65	13	74	397	2363	4767	4888
c3540 Robust65	3	24	179	1986	5881	6201
c5315 Robust65	3	24	203	2680	9941	10776
c6288 Robust65	2	19	146	2026	17942	20963
c7552 Robust65	3	24	190	2665	11756	12499
c17 STD65	12	31	34	39	39	39
c432 STD65	52	324	474	532	537	543
c499 STD65	64	354	800	1327	1533	1545
c880 STD65	79	561	954	1162	1194	1199
c1355 STD65	70	386	835	1352	1499	1513
c1908 STD65	64	436	830	1146	1201	1205
c2670 STD65	112	758	1385	1824	1890	1916
c3540 STD65	122	922	1848	2530	2572	2594
c5315 STD65	120	987	2525	4238	4465	4483
c6288 STD65	60	555	2629	7589	10375	10655
c7552 STD65	154	1269	3328	5364	5642	5679

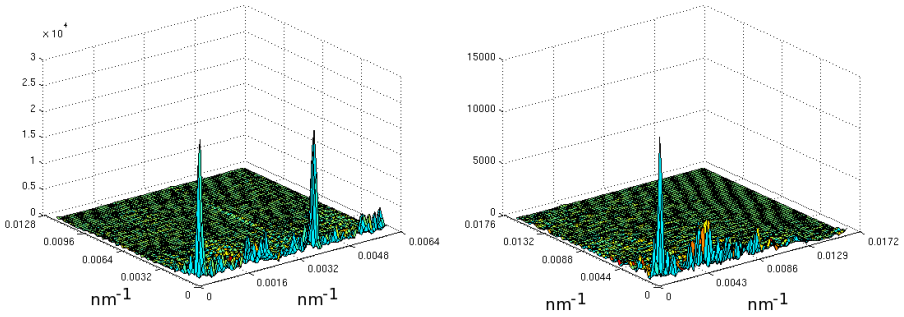
For ISCAS'85 circuits in the 45 nm node, we obtain that for all layers, VCTA designs are more regular than STD45 because for most of the cases a lower number of generators are found. c17 circuit is the only exception in some cases for PO and M1 as it is a very small circuit in which the VCTA redundancy implies a higher number of LAs and therefore more generators. Note also that the difference in regularity between VCTA and STD45 increases with LA size. For instance, if we compare the PO layer of the c2670 circuit for 160 nm and 2560 nm LA sizes, we observe that the difference in generators grows from 39 (1 vs 40) to 1745 (120 vs 1865). Note that this difference reflects the different amount of layout patterns that will need to be optimized for manufacturability when facing systematic variability issues. Micro-regularities considering small LAs are comparable but macro-regularities for big LAs are very distant because VCTA layouts are based on a single basic cell in front of a set of different cells for STD. In fact, all the ISCAS'85 circuits for VCTA (except c17) present the same number of generators in the PO and OD layers for all the LA sizes. Only the number of M1 generators vary, as M1 is used for routing configuration. We have therefore verified that layouts developed with VCTA are clearly more regular than the STD45 ones. Moreover, we have observed that regular layouts like VCTA layouts present high regularity for the whole range of regularity granularities.

For ISCAS'85 circuits in the 65 nm technology node, the higher regularity is in this case found for the new Robust65 library where layout regularity has been improved compared to the STD65 layouts. Note that for small LA size the regularity of Robust65 is even comparable to the VCTA45 regularity. In particular, for the OD layer, for 160 nm and 320 nm, VCTA45 layouts present 2 generators while Robust45 has 1 or 2 generators. However, because Robust65 layouts are still standard cell based, different combinations of cell neighborhoods are found at the macro-regularity level, and for higher LA sizes, regularity is again lower than VCTA45 regularity, and in some cases comparable to STD65.

For the sake of comparison, we have used the two-dimensional Fourier transform to evaluate regularity. This method confirms regularity results for instance

for the PO layer when comparing Robust65 and STD65 for c17 circuit (Figure 7.3). While the STD65 c17 layout spatial analysis has more representative frequential components, Robust65 c17 layout presents a clear repetition peak indicating regularity. In those comparisons where one layout is regular and the other one is not, both the two-dimensional Fourier transform and FOCSI can be used to identify the most regular layout.

However, when comparing layouts with a similar degree of regularity the two-dimensional Fourier transform is ambiguous. For instance, if we compare the PO layer for STD65 c432 layout and c499 layout (Figure 7.4), we obtain Fourier graphs that look almost the same. However, with our metric, we can see that c432 is more regular than c499 (380 versus 431 generators respectively). In fact, since the same standard cell library STD65 is used for both designs, regularity is similar but not exactly the same. Therefore, in this case, our metric is able to compare two layouts with similar regularities while the graphical inspection of the two-dimensional Fourier transform cannot.

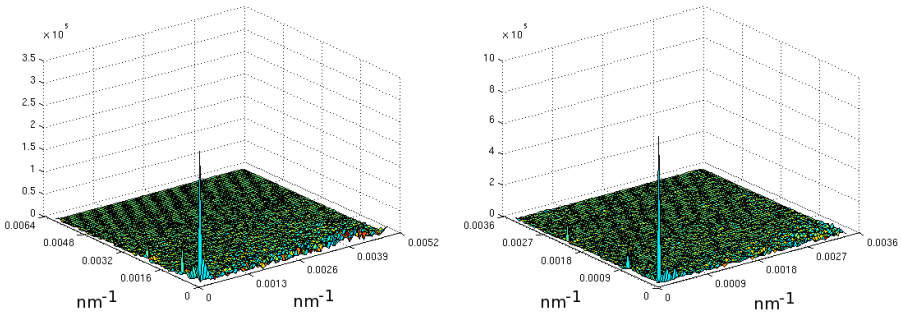


(a) Robust65 c17 (b) STD65 c17

Figure 7.3: 2D Fourier transform for polysilicon layers with different regularities

7.3 FOCSI for the complete layout

As shown in previous sections, FOCSI can measure regularity for different granularities just by varying LA sizing. However, our objective is to link regularity measurements obtained by FOCSI to the resulting systematic variability in the



(a) STD65 c432 (b) STD65 c499

Figure 7.4: 2D Fourier Transform for polysilicon layers with similar regularities

layout. That is why we first have to select the LA sizing according to the manufacturing process characteristics. In particular, focusing on lithography process variations, that is the most important source of systematic variability, the sizing will depend on the optical interaction length. Details are given next. Then, we will be able to calculate the complete layout FOCSI results.

7.3.1 FOCSI Layout Area sizing selection

Lithography enhancement techniques correct subwavelength lithography process variations taking into account a given layout area determined by the photolithography system used to manufacture the design. The corrected patterns need to be considered with their layout neighborhoods to obtain satisfactory results. Neighborhoods are bounded by the optical interaction length defined as the range of distance in which layout patterns have a non-negligible effect one on the other [79]. For our regularity measurements, in that case oriented to variability evaluation, LA sizing will be, therefore, determined by this optical interaction length. It makes no sense to consider layout regularity of areas higher than the ones defined by the optical interaction length because the regularity measured at this level will not affect the lithography enhancement techniques. Therefore, the concrete sizing of LAs for the R_{layer} measurement is defined by the optical interaction length of the manufacturing process considered.

Different works report similar values for the areas that must be considered for the optical interaction length. According to [11] the radius of influence of lithography is 5 times the minimum technology feature size. In [23] it is 500 nm for the 65 nm technology node. Finally, in [80] they have determined that an interaction radius of 220 nm for the available 193 nm lithography will take into account a 93% of the neighborhood effects. In our examples in the 65 nm and 45 nm technologies, with a 193 nm illumination source, to ensure that most of the proximity effects are considered we will use a radius of influence of approximately 320 nm that translates into 640 nm square sides sizing for FOCSI LAs.

7.3.2 ISCAS'85 layout results

For FOCSI oriented to variability evaluation the different β_j weights depend on the criticality of the layer manufacturability. Using test structures for process control to monitor and control the fabrication line, manufacturers can know which of the layout layers is the most affected by systematic subwavelength lithography based failures. Provided that these results have statistical significance, these data can be used to select the weights. Simulations of the fabrication process can also be performed taking into account different lithography enhancement techniques. For instance, if the manufacturing process is weak on M1 layer, the highest weight will be for the M1 regularity. Usually, PO layer is the most critical, because the smallest features are printed on it, like critical gate dimension.

To illustrate our regularity metric proposal in Table 7.4 we present the calculation of the complete layout regularity (R_{layout}) for the ISCAS'85 benchmarks studied in previous subsection for PO, OD and M1 layers (M = 3) with 640 nm LA sizing. Considering that the manufacturing process is PO limited we have used 0.45, 0.30 and 0.25 weights for PO, OD and M1 layers respectively. The case where OD is the most critical layer has been calculated using 0.30, 0.45 and 0.25 weights. Finally, the case where M1 is the most critical layer uses 0.30, 0.25 and 0.45 weights. Note however that our methodology is not limited to any particular set of values.

Table 7.4: ISCAS'85 complete layout FOCSI Regularity results considering PO, OD and M1 layers

	Complete layout: R_{layout}		
ISCAS layout	Layout is PO limited	Layout is OD limited	Layout is M1 limited
c17 VCTA45	11.90	10.10	12.90
c432 VCTA45	12.40	10.60	13.80
c499 VCTA45	12.40	10.60	13.80
c880 VCTA45	13.40	11.60	15.60
c1355 VCTA45	12.40	10.60	13.80
c1908 VCTA45	12.15	10.35	13.35
c2670 VCTA45	13.65	11.85	16.05
c3540 VCTA45	13.40	11.60	15.60
c5315 VCTA45	14.15	12.35	16.95
c6288 VCTA45	13.40	11.60	15.60
c7552 VCTA45	13.90	12.10	16.50
Average	**13.01**	**11.21**	**14.90**
c17 STD45	9.30	8.70	12.30
c432 STD45	212.30	206.45	331.05
c499 STD45	287.45	271.55	434.55
c880 STD45	368.85	349.65	557.05
c1355 STD45	274.15	256.60	406.60
c1908 STD45	280.65	263.10	412.70
c2670 STD45	445.75	423.25	680.25
c3540 STD45	639.30	613.20	1008.40
c5315 STD45	688.75	658.75	1085.75
c6288 STD45	454.35	445.65	753.05
c7552 STD45	889.55	854.45	1418.65
Average	**413.67**	**395.58**	**645.49**
c17 Robust65	16.75	15.25	21.05
c432 Robust65	63.60	57.90	88.70
c499 Robust65	96.15	87.60	137.60
c880 Robust65	102.10	93.40	147.80
c1355 Robust65	91.85	82.40	126.80
c1908 Robust65	90.35	81.65	126.65
c2670 Robust65	143.50	130.75	208.55
c3540 Robust65	86.75	74.75	108.95
c5315 Robust65	95.45	82.55	121.55
c6288 Robust65	68.15	59.60	87.20
c7552 Robust65	91.30	78.70	115.10
Average	**86.00**	**76.78**	**117.27**
c17 STD65	22.75	22.00	25.60
c432 STD65	340.80	309.45	370.05
c499 STD65	479.75	458.00	560.80
c880 STD65	653.70	591.30	721.30
c1355 STD65	538.30	515.20	612.80
c1908 STD65	527.45	483.80	599.40
c2670 STD65	933.80	863.45	1040.05
c3540 STD65	1281.90	1150.35	1406.55
c5315 STD65	1607.30	1472.45	1825.05
c6288 STD65	1498.00	1483.75	1796.75
c7552 STD65	2080.30	1923.70	2381.70
Average	**905.82**	**843.04**	**1030.91**

Different results are obtained in each case, with small variations because only 3 layers are considered, however, as expected, VCTA45 and Robust65 designs are more regular than STD45 and STD65 ones, with all these particular calculations and using these 3 layout layers. As shown in Table 7.4, layout regularity decreases (R_{layout} increases) when M1 is the most limiting layer because designs are more irregular in this layer than in the other ones because M1 is used for routing. The complete layout regularity value will be obtained by combining all of the layout layers involved in the designs and with more precise weighting values from the manufacturing process.

7.4 FOCSI regularity and variability

As explained before, layout regularity will help resolution enhancement techniques to become more effective, as less layout generators need to be corrected for instance by optical proximity correction, or in general, because the whole manufacturing process can be optimized for a reduced set of layout patterns. However, higher layout regularity will not imply itself lower process variations. The best example to illustrate it is a layout that can be generated by only one generator. If the printability of the generator is acceptable, the complete layout will have acceptable variations. However, if the printability of the generator is low (e.g., the patterns are placed at forbidden pitches), it will end up with a very regular layout but with a huge amount of variability. In this section we propose a methodology to understand and evaluate the impact of regularity on layout variability.

7.4.1 Variability model

To estimate variability in a layout layer for systematic sources of process variability, we propose to calculate the mean variation in the patterns of the layout. Defining N as the number of patterns in the layout and var_i the variation associated to pattern i, the mean variation in a layer can be written as:

$$\mu = \frac{\sum_{i=1}^{N} var_i}{N} \tag{7.3}$$

The systematic variation associated to each pattern will depend on the pattern itself and on the layout neighborhood inside of the radius of influence of the lithography, and this is exactly what is included in each of the LAs inspected by FOCSI. We assume that patterns with the same neighborhood will have the same variation for systematic variations associated to the manufacturing process. As each pattern of the layout will be represented at least by one FOCSI LA inspected because each pattern has at least one upper left corner, we propose to use FOCSI LAs to calculate the mean variability as follows:

$$\mu = \frac{\sum_{j=1}^{R_{layer}} n_j \cdot var_j}{N_{LA}} \tag{7.4}$$

where N_{LA} is the number of LAs inspected by FOCSI in the layer, R_{layer} is the number of generators identified by FOCSI in the layer under study, n_j is the number of occurrences of the generator j (note that this information can be easily tracked with FOCSI for each different generator identified), and var_j is the variation associated to generator j.

To solve Equation (7.4) we only need to obtain the values of var_j as all the other terms are known. Ideally, each of the FOCSI generators can be simulated in terms of lithography to obtain these values, however, this will be time consuming and the results can vary depending on the lithography models used. We propose to use the Monte Carlo method assigning random values to var_j from a distribution, that can be given by the foundries (e.g., a Gaussian distribution). This statistical methodology is already widely used for electrical simulations including process variations [23, 81].

7.4.2 ISCAS'85 layout results

We have used the Monte Carlo analysis considering a Gaussian distribution for variations of the patterns to calculate the mean variation for each layer of the ISCAS'85 layouts using VCTA45, STD45, Robust65 and STD65 layout styles. We have repeated the Monte Carlo 1082 times to calculate the mean and the standard deviation of the mean variation with a confidence level of 95% and a width for the confidence interval 5% [82]. The final distribution of variations has the same mean than the original variation distribution but the final standard deviation is reduced when compared to the original distribution deviation depending on layout regularity. The standard deviation reduction results are shown in Table 7.5 indicated as variability reduction (a higher variability reduction implies better manufacturability). They are normalized so that they are independent of the actual mean and standard deviation of the Gaussian distribution used in the Monte Carlo experiment.

On average for all STD65 layouts, the variability is only reduced 1.08% for OD, 0.39% for PO and 0.29% for M1. For STD45, the reductions are 9.90%, 2.94% and 0.68% respectively. The values are slightly higher, in particular for the OD layer, but remain comparable for the other layers. In fact, a very small reduction is obtained because both layout styles are irregular and therefore the impact of variations is similar to that of a design with maximum irregularity (where each LA is a generator). However, for Robust65 designs, the reductions are 31.35%, 3.69% and 5.95% respectively showing the OD regularity benefits (all transistors have the same size in this new Robust65 library). Finally, for VCTA45, the average reductions are 47.00%, 15.62% and 13.89% respectively. We can see how VCTA45 layouts show the higher variability reduction, reaching also important reductions for the M1 and PO layers.

In general, more regular layouts with less number of generators increase variations predictability by reducing its standard deviation (e.g., for M1 with the LA size of 640 nm, c2670 Robust65 has 397 generators and a 5.1% reduction,

Table 7.5: ISCAS'85 variability reduction results for OD, PO and M1 layers

	Variability reduction		
ISCAS layout	OD Layer	PO Layer	M1 Layer
c17 VCTA45	31.8%	12.5%	10.0%
c432 VCTA45	46.8%	15.4%	15.5%
c499 VCTA45	47.4%	15.8%	15.0%
c880 VCTA45	47.4%	15.8%	13.5%
c1355 VCTA45	47.3%	15.9%	15.0%
c1908 VCTA45	48.0%	15.5%	14.0%
c2670 VCTA45	48.6%	16.2%	14.1%
c3540 VCTA45	49.2%	15.9%	13.6%
c5315 VCTA45	49.2%	16.2%	14.4%
c6288 VCTA45	50.5%	16.3%	13.7%
c7552 VCTA45	50.8%	16.3%	14.0%
Average	**47.00%**	**15.62%**	**13.89%**
c17 STD45	41.3%	11.9%	4.4%
c432 STD45	5.4%	3.7%	0.1%
c499 STD45	7.5%	1.9%	0.3%
c880 STD45	6.3%	2.7%	0.3%
c1355 STD45	7.5%	1.8%	0.4%
c1908 STD45	6.3%	2.1%	0.4%
c2670 STD45	6.3%	1.4%	0.3%
c3540 STD45	5.8%	2.1%	0.2%
c5315 STD45	7.6%	1.0%	0.1%
c6288 STD45	8.8%	2.1%	0.6%
c7552 STD45	6.1%	1.6%	0.4%
Average	**9.90%**	**2.94%**	**0.68%**
c17 Robust65	23.6%	6.5%	4.2%
c432 Robust65	27.2%	3.4%	7.2%
c499 Robust65	34.5%	3.5%	6.3%
c880 Robust65	34.2%	4.3%	5.5%
c1355 Robust65	36.2%	3.7%	6.5%
c1908 Robust65	31.7%	3.7%	6.4%
c2670 Robust65	29.8%	2.8%	5.1%
c3540 Robust65	32.3%	3.3%	5.7%
c5315 Robust65	30.6%	2.8%	4.9%
c6288 Robust65	33.7%	3.9%	7.9%
c7552 Robust65	31.1%	2.7%	5.8%
Average	**31.15%**	**3.69%**	**5.95%**
c17 STD65	2.6%	2.0%	1.9%
c432 STD65	4.2%	0.8%	0.2%
c499 STD65	1.0%	0.7%	0.3%
c880 STD65	0.6%	0.2%	0.1%
c1355 STD65	0.9%	0.1%	0.1%
c1908 STD65	1.3%	0.1%	0.0%
c2670 STD65	0.7%	0.2%	0.1%
c3540 STD65	0.9%	0.0%	0.0%
c5315 STD65	1.0%	0.2%	0.0%
c6288 STD65	1.2%	0.6%	0.6%
c7552 STD65	0.9%	0.0%	0.0%
Average	**1.08%**	**0.39%**	**0.29%**

Table 7.6: Robust65 ISCAS'85 OD generators study

ISCAS circuit	LAs inspected	Generator maximum occurrences	Repetition percent	Variability reduction
c17	22	10	45.5%	23.6%
c432	366	206	56.3%	27.2%
c499	700	492	70.3%	34.5%
c880	660	452	68.5%	34.2%
c1355	690	504	73.0%	36.2%
c1908	554	366	66.1%	31.7%
c2670	1150	708	61.6%	29.8%
c3540	1528	1002	65.6%	32.3%
c5315	2602	1620	62.3%	30.6%
c6288	5342	3550	66.5%	33.7%
c7552	3000	1856	61.9%	31.1%

and c3540 Robust65 has 179 generators and a 5.7% reduction). However, the number of generators is not the only factor affecting variability. The number of occurrences of the generators (its distribution) has also an important influence. For instance, all Robust65 ISCAS'85 OD layers have 8 generators for the 640 nm LA size, but all have different variability reductions (c17 is a particular case with only 5 generators as it only includes 6 NAND gates, see chapter 4). In fact the weight of the most repeated generator amongst the total number of layout areas is directly related to the variability reduction result. In Table 7.6 we show, for the OD layer of the Robust65 ISCAS'85, the total number of LAs inspected, the number of occurrences of the most repeated generator for each particular circuit, which fraction of the total number of LAs correspond to such generator, and finally the resulting variability for each layout. The correlation between the percent of LAs generated by the most repeated generator and the variability reduction reaches 98.5%. Therefore, we can conclude that the degree of repetition of the most repeated generator has direct impact on the variability.

7.5 Conclusion

Layout regularity has been shown to be linked to process variability, as designers are improving the regularity of their layouts to increase their printability. However, there is no tool available that measures regularity. Existing tools to estimate variability simulate the whole complex lithography system and also in-

clude the resolution enhancement techniques effects. Therefore, these tools are complex and computationally demanding. FOCSI proposal fulfills the need of an easy and fast layout analysis that can be applied in an early stage of the design. FOCSI can calculate layout regularity for each of the layout layers and for different granularities (LA sizings) by quantifying the number of layout generators. We have shown that FOCSI provides an accurate comparison of layout layers even if their regularity is similar. Moreover, when oriented to variability evaluation (choosing the LA sizing that takes into account lithography radius of influence) FOCSI can provide the complete layout regularity evaluation by weighting the layer generators depending on layer criticality.

As expected, FOCSI results in the 45 nm and 65 nm node for ISCAS'85 benchmarks show that standard cell layouts (STD45 and STD65) are less regular than the new Robust library (Robust65) and than the proposed VCTA45 approach. In the case of VCTA layouts, we have also observed how VCTA regularity is found for the whole range of regularity granularities confirming the VCTA maximum regularity at all levels.

Then, we have linked FOCSI regularity measurements and layout variability by means of a Monte Carlo analysis showing that the decrease of the standard deviation of the mean variations in a layer depends on layout regularity in a comprehensive way, taking into account the number of generators of the layout and also its distribution. The optimization of the manufacturing process for the reduced set of layout generators can then further increase the layout printability, but this is not directly related to layout regularity. For regular layouts this will be particularly beneficial as the number of generators will be low.

An important conclusion of the chapter is that design for manufacturability techniques need to be adapted to the particular manufacturing process that will be used. When starting the thesis, the initial hypothesis was that layout regularity improves manufacturability. That is why the first works were on developing the VCTA fabric with the objective of maximizing regularity at all levels. However, as we have demonstrated, regularity itself is not at the origin of the reduction

of process variations. Regularity as we have defined it is the repetitivity in layout. Therefore on the top of regularity, to ensure good manufacturability the layout patterns that are repeated need to have themselves a good manufacturability. For instance, patterns need to avoid forbidden pitches of the lithography tools, or avoid jogs and corners using one-dimensional layout style.

In fact, regularity itself is affecting more directly the design time and the time-to-market, because the higher the repetitivity the lower the number of layout patterns that will need to be optimized and modified by the costly resolution enhancement techniques. Moreover, layout masks can be reused for different designs, for instance for VCTA designs where contacts, vias and metal extensions are different for each, but where the other masks remain unchanged like for polysilicon or oxide diffusions. In that way, we can say that regularity helps reducing manufacturing cost and can lead to reduced yield loss and improved yield ramps.

Of course, layout regularity has an impact on area, energy and delay, but a good communication between layout designers and manufacturing engineers can help reducing these overheads as regularity can be adapted to the particular process to be used, for instance allowing the use of pushed rules, that lead to lower minimum spacings between layout patterns.

FOCSI can be used to reduce layout variability at different moments of the design flow. First, applied during the place step, it can provide the information required to modify the positions of the standard cells focusing on maximizing regularity. Second, FOCSI can be used in the routing step to maximize wire regularity. In both cases, the computational cost of FOCSI recalculation is expected to be reduced as only the LAs modified will need to be taken into account to obtain the new regularity. Once all LAs and generators are identified in the first run of FOCSI, optimization algorithms can be incrementally applied to maximize regularity and therefore minimize variability.

Chapter 8

Conclusion

In this thesis we have attacked the design and manufacturing challenges in integrated circuits. In particular, we have centered our research in a future scenario for ultra-deep submicron technologies where process variations can lead to unaffordable manufacturing and design costs and increased time-to-market. The survival of integrated circuits industry requires a closer collaboration between designers, manufacturers and EDA developers.

From the design and manufacturability point of view, the solution that we have proposed is to maximize layout regularity at all levels, as it helps reducing variability by allowing the optimization of lithography tools and resolution enhancement techniques. That is why we have proposed the Via-Configurable Transistor Array (VCTA) regular fabric that maximizes regularity at device and interconnect levels.

Then, from the EDA point of view, we have automated the VCTA layout generation as existing tools do not fulfill the requirements of VCTA physical design. In this case, our contribution is a physical synthesis tool. When possible, we have adapted the VCTA flow to the available tools but most of the steps have finally been developed specifically for VCTA implying several changes from the standard design flow.

Finally, also for the EDA side, to evaluate and understand the benefits of layout regularity, we have developed the Fixed Origin Corner Square Inspection (FOCSI) metric that provides a regularity measurement that can be used to estimate the layout variability reduction that can be reached due to regularity improvement. In this case, our contribution is a layout analysis tool.

8.1 Summary of contributions

The thesis has three major contributions related to manufacturing and design of integrated circuits.

First, in this thesis we have presented a new regular layout design technique named Via-Configurable Transistor Array (VCTA). The main work of this part has been to develop the VCTA basic cell that can implement the functionality desired by configuring vias. Using lithography effect models and transistor models we have demonstrated how process variations in particular for threshold voltage and channel length can be reduced using VCTA. We have also shown how time-to-market can be reduced because of regularity as it implies better initial yield and faster yield ramp. The manufacturing cost reduction of the VCTA designs can be critical for specific designs. The reference for comparisons are always existing design techniques like the standard cell approach and the full custom layout.

Second, an important effort has also been devoted to the VCTA automation tool that is the first step to demonstrate that VCTA can be used in industry. In this case we focus on optimizing the area of the resulting layouts. Starting from a transistor netlist of the circuit, a grouping step with the objective to maximize the transistor occupancy in the VCTA cells have been developed solving a modified knapsack problem. Then, standard placement has been reused, but not standard routing. Intra and inter-cell routing algorithms have been demonstrated to manage routability congestion while considering VCTA cells limited resources. First, reordering intra-cell connections allows a decrease of the resources required

for later inter-cell routing. Then, we have worked on adding routing tracks for congestion treatment and we have used a simulated annealing algorithm to minimize the area of the routed layout. Results are successful in terms of area when compared to academical standard cells as VCTA layouts have only a 10% area increase in average. Moreover we have demonstrated the feasibility of complex circuits with VCTA (unfeasible if developed manually).

Third, a layout regularity metric (FOCSI) linked to process variations has been developed to help designers on the choice of layout style, allowing them to compare the regularity of their layouts and how it affects manufacturability. FOCSI is layout pattern oriented to be able to catch the repetitivity of shapes and it gives the number of generators of the layout. The lower this number is the higher the regularity is. Then, using a Monte Carlo analysis, we have proposed a model for systematic variability in the layout. The objective was to provide a easy and fast metric that can be used in an early stage by designers. Understanding the manufacturing process has been crucial for this step as we consider systematic variations. FOCSI results have also come to verify the improved regularity of our VCTA proposal.

8.2 Future works

VCTA regular fabric is based on a single cell that is configured to implement the different functions in the design. In this way, on the contrary of the unoptimized standard cell, we have developed a new design methodology that can generate the required cells on the fly adapted to the circuit to be designed. For standard cells, the available functions are predefined, also with a finite number of drives, and are not always optimum for all the designs. In fact, the area, energy and delay overheads found while comparing VCTA and standard cells can be due to the use of transistor netlists coming from the standard cell approach, not oriented to VCTA. In our works, we have not adapted the logic synthesis to the VCTA regular fabric. Several aspects can be improved, like the sizing of the transistors in the VCTA cells, or the number of PMOS and NMOS transistors available per

cell, as well as the number of gate inputs available. Including these constraints in the logic synthesis step can help adapting the VCTA cell implementation so that transistor occupancy can be maximized. Optimized logic synthesis can also reduce the number of transistors of the design, for instance allowing complex functions, not present amongst the standard cells, or even generating optimized functions for the design under study. The tool developed for VCTA automation is already prepared to take into account different VCTA implementations and therefore improving the logic synthesis step can directly be included in the flow. The VCTA automation flow can also be adapted to focus on energy or delay optimizations. For instance, once the grouping step is done, the resulting cells can be characterized like in the standard cell flow to be able to include energy and delay estimations. However, this approach can be much more time consuming. An intermediate possibility can be to study all the possible functions that can be mapped in a particular VCTA implementation (defining the transistors and inputs available) and then select the cells that will be used, selecting them in a way that they can generate any kind of circuits and considering timing and energy constraints. Then, these VCTA cells can be used as standard cells are used but with the VCTA particular routing.

Regarding the VCTA cell itself, it needs to be adapted to the particular manufacturing process where it will be used, in terms of design rules for space and width. As we have demonstrated, regularity itself is not at the origin of the reduction of variability. We need to optimize the layout of the VCTA cell for instance using lithography simulation tools. However, this task is relatively simple as only one cell has to be optimized. The objective is to have a variations-aware basic cell fully optimized to the particular foundry, allowing for instance minimized area by the use of pushed rules or maximized yield avoiding forbidden pitches of the process or doubling vias.

Regarding FOCSI regularity metric, other ways of combining FOCSI measurements for different layers can also be considered with new inputs for the evaluation of layouts manufacturability. For instance, the oxide diffusion and

polysilicon layers can be treated together as they define transistor shapes and can give more direct information about variability in devices. Moreover, as explained in chapter 7, FOCSI can be included in the design flow to optimize layouts in terms of regularity. In can be used for place and route algorithms to make decisions on what relative positions of cells and what interconnections positions or layers are better considering regularity. In that way FOCSI is suitable for any kind of fabric, including standard cells, which manufacturability can be also improved.

Bibliography

[1] L. Capodlieci, P. Gulpta, A.B. Kahng, D. Sylvester, and J. Yang. Toward a methodology for manufacturability-driven design rule exploration. In *Design Automation Conference, 2004. Proceedings. 41st*, pages 311–316, 2004. [cited at p. 2]

[2] Abbas El-Gamal, Ivo Bolsens, Andy Broom, Christopher Hamlin, Philippe Magarshack, Zvi Or-Bach, and Larry Pileggi. Fast, cheap and under control: the next implementation fabric. In *DAC '03: Proceedings of the 40th conference on Design automation*, pages 354–355, New York, NY, USA, 2003. ACM Press. [cited at p. 2]

[3] G. Declerck. A look into the future of nanoelectronics. In *VLSI Technology, 2005. Digest of Technical Papers. 2005 Symposium on*, pages 6 – 10, june 2005. [cited at p. 3]

[4] *IDESA Design for manufacturability flow,* `http://www.idesa-training.org/Courses.html`, 2008. [cited at p. 3, 9, 10]

[5] *From Sand to Silicon Making of a Chip Illustrations,* `http://www.intel.com/pressroom/kits/chipmaking`, 2009. [cited at p. 4]

[6] M. Rothschild, T.M. Bloomstein, T.H. Fedynyshyn, R.R. Kunz, V. Liberman, M. Switkes, N.N. Efremow, S.T. Palmacci, J.H.C. Sedlacek, D.E. Hardy, and A. Grenville. Recent Trends in Optical Lithography. *LINCOLN LABORATORY JOURNAL*, 14(3):221–236, 2003. [cited at p. 5]

[7] L. Pileggi, H. Schmit, A.J. Strojwas, P. Gopalakrishnan, V. Kheterpal, A. Koorapaty, C. Patel, V. Rovner, and K.Y. Tong. Exploring regular fabrics to optimize the performance-cost trade-off. In *Design Automation Conference, 2003. Proceedings*, pages 782–787, 2003. [cited at p. 7, 13]

[8] T. Jhaveri, L. Pileggi, V. Rovner, and A. J. Strojwas. Maximization of layout printability/manufacturability by extreme layout regularity. In *Proceedings of SPIE*, 2006. [cited at p. 7, 33, 34, 44, 45]

[9] Lei He, Andrew B. Kahng, King Ho Tam, and Jinjun Xiong. Simultaneous Buffer Insertion and Wire Sizing Considering Systematic CMP Variation and Random Leff Variation. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 26(5):845–857, 2007. [cited at p. 7]

[10] M. Bohr. Using innovation to drive Moore's Law. In *Solid-State and Integrated-Circuit Technology, 2008. ICSICT 2008. 9th International Conference on*, pages 13 –15, oct. 2008. [cited at p. 8]

[11] Lars W. Liebmann. Layout impact of resolution enhancement techniques: impediment or opportunity? In *ISPD '03: Proceedings of the 2003 international symposium on Physical design*, pages 110–117, New York, NY, USA, 2003. ACM Press. [cited at p. 8, 9, 146]

[12] B. Wong, F. Zach, V. Moroz, A. Mittal, G. Starr, and A. Kahng. *Nano-CMOS Design for Manufacturability: Robust Circuit and Physical Design for Sub-65 nm Technology Nodes*. John Wiley & Sons, 2009. [cited at p. 12, 17, 22, 25]

[13] J.D. Sawicki. DFM: magic bullet or marketing hype? In *Design and Process Integration for Microelectronic Manufacturing II. Proceedings of the SPIE.* [cited at p. 13, 14, 26]

[14] K. Takeuchi, A. Nishida, and T. Hiramoto. Random fluctuations in scaled mos devices. In *Simulation of Semiconductor Processes and Devices, 2009. SISPAD '09. International Conference on*, pages 1 –7, sept. 2009. [cited at p. 14]

[15] K. Bernstein et al. High-performance CMOS variability in the 65-nm regime and beyond. *IBM Journal of Research and Development*, 50:433–449, 2006. [cited at p. 14, 15, 18]

[16] Aseem Agarwal, David Blaauw, and Vladimir Zolotov. Statistical Clock Skew Analysis Considering Intra-Die Process Variations. In *ICCAD '03: Proceedings of the 2003 IEEE/ACM international conference on Computer-aided design*, page 914, Washington, DC, USA, 2003. IEEE Computer Society. [cited at p. 15]

[17] Handel H. Jones. A delayed 90-nm surprise. *Electronics Design Chain Magazine*, 2004. [cited at p. 16]

[18] C. Edwards. *TSMC on 40nm and 28nm yield issues,* `http://blog.shrinkingviolence.com/`. [cited at p. 16]

[19] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De. Parameter variations and impact on circuits and microarchitecture. In *Design Automation Conference, 2003. Proceedings*, pages 338–342, 2003. [cited at p. 17]

[20] *International Technology Roadmap for Semiconductors 2011,* `http://www.itrs.net/Links/2011ITRS/Home2011.htm`, 2011. [cited at p. 18, 19]

[21] A. Agarwal, V. Zolotov, and D.T. Blaauw. Statistical clock skew analysis considering intradie-process variations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 23(8):1231–1242, 2004. [cited at p. 18]

[22] J.A. Torres and F.G. Pikus. Unified process aware system for circuit layout verification. In *Design for Manufacturability through Design-Process Integration, Proceedings of the SPIE.* [cited at p. 22]

[23] M. Orshansky, D.S. Boning, and S.R. Nassif. *Design for Manufacturability and Statistical Design: A Constructive Approach.* Integrated Circuits and Systems. Springer, 2008. [cited at p. 22, 146, 149]

[24] C. Chiang and J. Kawa. *Design for Manufacturability and Yield for Nano-Scale CMOS.* Integrated Circuits and Systems. Springer, 2007. [cited at p. 22]

[25] H. Sunagawa, H. Terada, A. Tsuchiya, K. Kobayashi, and H. Onodera. Effect of regularity-enhanced layout on printability and circuit performance of standard cells. In *Quality of Electronic Design, 2009. ISQED 2009.*, pages 195–200, March 2009. [cited at p. 22]

[26] Sunil R. Shenoy and Akhilesh Daniel. Intel Architecture and Silicon Cadence: The Catalyst for Industry Innovation. *Technology@Intel Magazine*, pages 1–7, October 2006. [cited at p. 22]

[27] S. Ozdemir, D. Sinha, G. Memik, J. Adams, and Hai Zhou. Yield-Aware Cache Architectures. In *Microarchitecture, 2006. MICRO-39. 39th Annual IEEE/ACM International Symposium on*, pages 15–25, 2006. [cited at p. 22]

[28] Amit Agarwal, Bipul C. Paul, Hamid Mahmoodi, Animesh Datta, and Kaushik Roy. A Process-Tolerant Cache Architecture for Improved Yield in Nanoscale Technolo-

gies. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 13(1):27–38, January 2005. [cited at p. 22]

[29] J. A. Torres and C. N. Berglund. Integrated circuit DFM framework for deep sub-wavelength processes. In Lars W. Liebmann, editor, *Design and Process Integration for Microelectronic Manufacturing III*, volume 5756, pages 39–50. SPIE, 2005. [cited at p. 22]

[30] Deepak D. Sherlekar. Design considerations for regular fabrics. In *ISPD '04: Proceedings of the 2004 international symposium on Physical design*, pages 97–102, New York, NY, USA, 2004. ACM Press. [cited at p. 26]

[31] B. Zahiri. Structured ASICs: opportunities and challenges. In *Proceedings of 21st International Conference on Computer Design*, pages 404–409, 2003. [cited at p. 26, 37]

[32] A.D. Lopez and H.-F.S. Law. A dense gate matrix layout method for MOS VLSI. *IEEE Transactions on Electron Devices*, 27(8):1671–1675, 1980. [cited at p. 26, 27]

[33] C. Piguet, J. Zahnd, A. Stauffer, and M. Bertarionne. A metal-oriented layout structure for CMOS logic. *Solid-State Circuits, IEEE Journal of*, 19(3):425–436, 1984. [cited at p. 26, 27]

[34] A.C.M. de Oliveira and L.A.N. Lorena. A constructive genetic algorithm for gate matrix layout problems. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 21(8):969 – 974, aug 2002. [cited at p. 27]

[35] H.J.M. Veendrick, D.A.J.M. van den Elshout, D.W. Harberts, and T. Brand. An efficient and flexible architecture for high-density gate arrays. *Solid-State Circuits, IEEE Journal of*, 25(5):1153–1157, 1990. [cited at p. 28, 29]

[36] D.W. Harberts, D.A.J.M. van den Elshout, and H.J.M. Veendrick. Design for routability of a high-density gate array. In *Computer Design: VLSI in Computers and Processors, 1990. ICCD '90. Proceedings., 1990 IEEE International Conference on*, pages 56 –59, sep 1990. [cited at p. 30]

[37] Mingjie Lin and Abbas El Gamal. A routing fabric for monolithically stacked 3D-FPGA. In *Proceedings of ACM/SIGDA 15th International Symposium on Field Programmable Gate Arrays, FPGA*, pages 3–12, New York, NY, USA, 2007. ACM Press. [cited at p. 30, 31]

[38] Russell G. Tessier. *Fast Place and Route Approaches for FPGAs*. PhD thesis, 1999. [cited at p. 32]

[39] V. Kheterpal, V. Rovner, T. G. Hersan, D. Motiani, Y. Takegawa, A. J. Strojwas, and L. Pileggi. Design methodology for IC manufacturability based on regular logic-bricks. In *DAC '05: Proceedings of the 42nd annual conference on Design automation*, pages 353–358, New York, NY, USA, 2005. ACM Press. [cited at p. 33]

[40] 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*, 29(4):509 –527, april 2010. [cited at p. 33]

[41] J. Dick. Design-for-manufacturing features in nanometer processes - A reverse engineering perspective. In *ASMC*, pages 56–61, 2009. [cited at p. 33]

[42] P. G. Drennan et al. Implications of Proximity Effects for Analog Design. *CICC*, pages 169–176, 2006. [cited at p. 33]

[43] Hassan Lachkar, Olivier Rizzo, Jean-Michel Portal, and Olivier Ginez. Layout Uniformity: A metric for yield enhancement. In *Circuits and Systems (MWSCAS), 2011 IEEE 54th International Midwest Symposium on*, pages 1 –4, aug. 2011. [cited at p. 33]

[44] Michael C. Smayling, Hua yu Liu, and Lynn Cai. Low k_1 logic design using gridded design rules. volume 6925, page 69250B. SPIE, 2008. [cited at p. 34, 96]

[45] *Synaptic project, European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 248538,* www.synaptic-project.eu. [cited at p. 34]

[46] C. Mead and L. Conway. *Introduction to VLSI design*. Addison Wesley, 1980. [cited at p. 34]

[47] J. Fox. Cell-based design: a review. *Solid-State and Electron Devices, IEE Proceedings I*, 133(3):77 –82, june 1986. [cited at p. 34]

[48] C. Menezes, C. Meinhardt, R. Reis, and R. Tavares. Design of Regular Layouts to Improve Predictability. In *Devices, Circuits and Systems, Proceedings of the 6th International Caribbean Conference on*, pages 67–72, 2006. [cited at p. 38]

[49] C. Menezes, C. Meinhard, R. Reis, and R. Tavares. A regular layout approach for ASICs. In *Emerging VLSI Technologies and Architectures, 2006. IEEE Computer Society Annual Symposium on*, volume 00, pages 2 pp.–, 2006. [cited at p. 38]

[50] Y. Ran and M. Marek-Sadowska. Designing Via-Configurable Logic Blocks for Regular Fabric. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 14(1):1–14, 2006. [cited at p. 39, 40]

[51] Y. Ran and M. Marek-Sadowska. Via-Configurable Routing Architectures and Fast Design Mappability Estimation for Regular Fabrics. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 14(9):998–1009, 2006. [cited at p. 39]

[52] *Mentor Graphics Design For Manufacturing Tools,* http://www.mentor.com/products/ic_nanometer_design/design-for-manufacturing. [cited at p. 41]

[53] Munkang Choi and L. Milor. Impact on circuit performance of deterministic within-die variation in nanoscale semiconductor manufacturing. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 25(7):1350 –1367, 2006. [cited at p. 42]

[54] V. Moroz et al. Stress-aware design methodology. In *Quality of Electronic Design*, pages 807–812, 2006. [cited at p. 42]

[55] Munkang Choi and L. Milor. Diagnosis of Optical Lithography Faults With Product Test Sets. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 27(9):1657 –1669, 2008. [cited at p. 43]

[56] http://www-device.eecs.berkeley.edu/£\sim£bsim3/bsim4.html. [cited at p. 43]

[57] *Cadence Encounter RTL Compiler,* http://www.cadence.com/products/ld/rtl_compiler/pages/default.aspx. [cited at p. 54]

[58] *Cadence SoC Encounter RTL-to-GDSII System ,* http://www.cadence.com/products/di/soc_encounter/pages/default.aspx. [cited at p. 54]

[59] *Cadence Virtuoso Schematic Editor,* http://www.cadence.com/products/cic/schematic_editor/pages/default.aspx. [cited at p. 54]

[60] *Cadence Virtuoso Layout Suite,* http://www.cadence.com/products/cic/layout_suite/pages/default.aspx. [cited at p. 54]

[61] *Cadence Encounter Library Characterizer,* http://www.cadence.com/products/di/library_characterizer/pages/default.aspx. [cited at p. 54]

[62] *Mentor Graphics Calibre nmLVS,* http://www.mentor.com/products/ic_nanometer_design/verification-signoff/circuit-verification/calibre-nmlvs. [cited at p. 54]

[63] *Mentor Graphics Calibre nmDRC,* http://www.mentor.com/products/ic_nanometer_design/verification-signoff/physical-verification/calibre-nmdrc. [cited at p. 54]

[64] *Synopsys StarRC,* http://www.synopsys.com/Tools/Implementation/SignOff/Pages/StarRC-ds.aspx. [cited at p. 54]

[65] *Synopsys HSPICE,* http://www.synopsys.com/Tools/Verification/AMSVerification/CircuitSimulation/HSPICE/Pages/default.aspx. [cited at p. 54]

[66] *The GNU Bourne-Again SHell (BASH),* http://tiswww.case.edu/php/chet/bash/bashtop.html. [cited at p. 54]

[67] *MathWorks MATLAB,* http://www.mathworks.com/products/matlab/index.html. [cited at p. 54]

[68] Neil H. E. Weste and David Harris. *CMOS VLSI Design, A Circuits and Systems Perspective.* Pearson, 2005. [cited at p. 55, 85]

[69] Huang H. and Shen J. A DLL-based programmable clock generator using threshold-trigger delay element and circular edge combiner. In *Advanced System Integrated Circuits 2004. Proceedings of 2004 IEEE Asia-Pacific Conference on*, pages 76 – 79, 4-5 2004. [cited at p. 56]

[70] Barajas E., Mateo D., and Gonzlez J.L. Behavioural modelling of DLLs for fast simulation and optimisation of jitter and power consumption. In *Digital System Design 2010. DSD '10. 13th Euromicro Conference on*, september 2010. [cited at p. 56, 59, 91]

[71] F. Brglez and H. Fujiwara. A neutral netlist of 10 combinational benchmark circuits and a target translator in FORTRAN. In *IEEE International Symposium on Circuits and Systems*, 1985. [cited at p. 58, 125]

[72] *http://www.eda.ncsu.edu*. [cited at p. 58, 125]

[73] G. Petley. *VLSI and ASIC Technology Standard Cell Library Design, http://www.vlsitechnology.org*. [cited at p. 59]

[74] *http://vcag.ecen.okstate.edu/wiki*. [cited at p. 59, 125, 129]

[75] J.-M. Masgonty, S. Cserveny, C. Arm, P.-D. Pfister, and C. Piguet. Low-Power Low-Voltage Standard Cell Libraries with a Limited Number of Cells. In *PATMOS*, 2001. [cited at p. 59, 139]

[76] *http://www.spec.org/cpu2000*. [cited at p. 87]

[77] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science, Number 4598, 13 May 1983*, 220, 4598:671–680, 1983. [cited at p. 123]

[78] H.K. Lee and D.S. Ha. An efficient, forward fault simulation algorithm based on the parallel pattern single fault propagation. In *Test Conference, 1991, Proceedings., International*, page 946, oct 1991. [cited at p. 129]

[79] D.M. Pawlowski, Liang Deng, and M.D.F. Wong. Fast and Accurate OPC for Standard-Cell Layouts. *Design Automation Conference, 2007. ASP-DAC '07. Asia and South Pacific*, pages 7–12, Jan. 2007. [cited at p. 145]

[80] Daniel Morris, Kaushik Vaidyanathan, Neal Lafferty, Kafai Lai, Lars Liebmann, and Larry Pileggi. Design of embedded memory and logic based on pattern constructs. In *VLSIT*, pages 104 –105, june 2011. [cited at p. 146]

[81] A. Asenov. *Advanced Monte Carlo Techniques in the Simulation of CMOS Devices and Circuits*, volume 6046 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2011. [cited at p. 149]

[82] David S. Moore and George P. McCabe. *Introduction to the Practice of Statistics*. Freeman & Co, 1989. [cited at p. 150]

List of Figures

List of Tables