The University of Kitakyushu

Doctoral Thesis

Portability-Aware CMOS Mixed Signal VLSI Circuit Design Methodology

Author:
Yu ZHANG

Supervisor:
Prof. Shigetoshi NAKATAKE

Nakatake Laboratory
Graduate School of Environmental Engineering

August 2014

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Since the integrated circuit (IC) was introduced from 1958, it have allowed electronics to expand at an amazing rate. At first, it grew fast in the area of digital ICs. Lately, analog ICs have received more attention because the design of it need more comprehensive consideration. Nowadays, mixed-signal ICs which combines analog, digital, memory circuit in one chip is becoming the design of mainstream. Meanwhile, helping to manage IC which currently incorporate billions of transistors, electronic design automation (EDA) has now becoming an immensely successful field [1]. Combining theory and practice, EDA fostered and used theories in computation and modeling. It was one of the earliest to engage in inter-disciplinary collaboration, where the computer scientists and engineers in EDA successfully collaborated with electrical engineers, physicists, chemists, theoretical computer scientists, applied mathematics and optimization experts, and application domain specialists [2].

Being an important field in computer science and engineering, electronic design automation (EDA) of very large-scale integrated (VLSI) circuits and systems has made a significant impact on the development of information technology. For instance, it has successfully managed the exponential increase in design complexity from the first microprocessor (Intel 4004) with 2,250 transistors to the latest multi-core processor with over a billion transistors [3]. Also, EDA is one of the first fields in computer science and engineering (CS&E) that has applied the concepts and techniques of computational

modeling, computational thinking, and computational discovery to an application domain (electronic circuit design) and achieved remarkable success.

Before the IC is finally manufactured, EDA has completely transformed the design methodology by electronic engineers. Every circuit being designed today starts with a computational model (specified in an executable programming language) at a high level of abstraction. It then goes through a sequence of synthesis and optimization transformations, followed by rigorous digital simulation and prototyping, as well as formal and semi-formal verification.

IC topographies are now considered a form of intellectual property (IP). Since EDA field is one of the earliest to engage in inter-disciplinary collaboration, the computer scientists and engineers successfully collaborated with the electrical engineers to derive various levels of circuit models with application domain specialists to develop IP libraries.

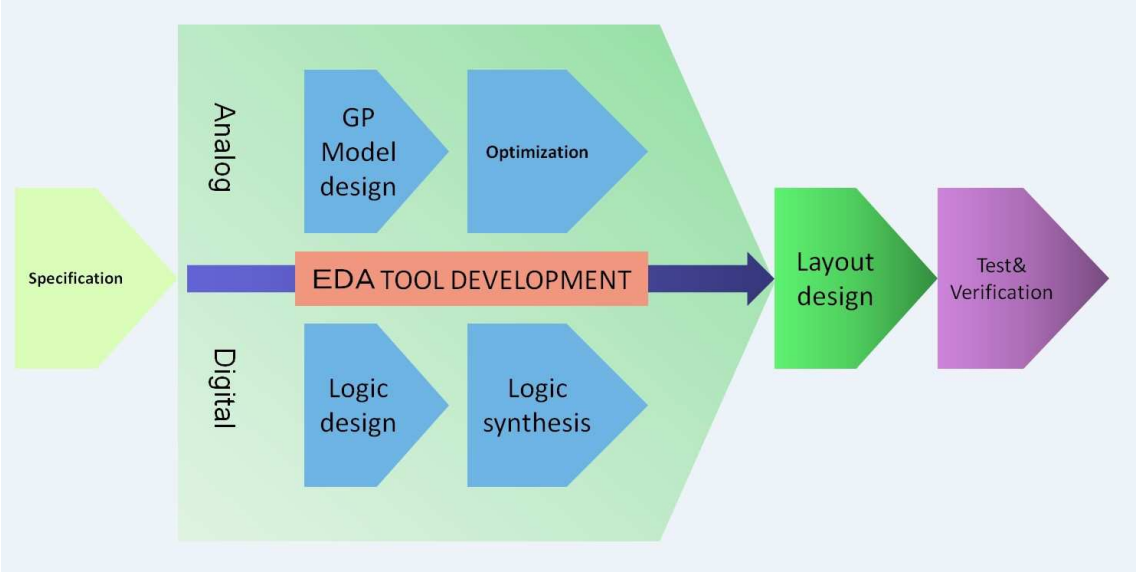

FIGURE 1.1: The design flow of our VLSI design methodology.

The definition of EDA is "the category of tools for designing and producing electronic systems ranging from printed circuit boards (PCBs) to integrated circuits [4]. Although the history of design automation algorithms such as Kernighan and Lin's bi-partitioning heuristics [5] and Kenneth Hall's r-dimensional quadratic placement procedures [6] predates the VLSI era, it was not until the early 1980s that the mystique surrounding VLSI chip design and fabrication was unveiled by Mead and Conway [7]. This led to the cultivation of VLSI education in universities and to the flourishing of research in VLSI

system design and EDA which, in turn, created automation tools for logic synthesis, layout generation, circuit simulation, design verification, reliability modeling, chip testing, debugging, and yield analysis.

Currently, the EDA field is also meeting serious challenges. For instance, nonrecurring engineering (NRE) costs associated with VLSI circuit design are skyrocketing, with estimates of over $30M per ASIC design undertaken. The rapid increase in the number of transistors available on a single chip leads to system-on-chip integration, with complex interaction between software and hardware, digital and analog, etc. Moreover, the field of applications enabled by semiconductor technology is growing at a rapid rate, ranging from very high-performance microprocessors and signal processors, to a broad array of low-power portable devices, to micro sense/communicate/actuate networks of chips that are driven by very low per-unit cost and extremely low operating power. Designers must not only create chips that function properly in conventional digital and mixed-signal operation, but must also comprehend sensors that respond to signals from many physical domains, such as pressure, temperature, chemical and optical. Meanwhile, since IC has leading to nanometer node, the layout-dependent effects (LDE) affects the circuit performance more severely.

1.2 Motivation and Objectives

As semiconductor moving to smaller process nodes especially under nanometers, the cost has become expensive when we redesign a circuit in a new and smaller process. One of the reasons is that the LDE such as shallow trench isolation (STI) stress and well proximity effect (WPE) become dominant to achieve circuit performances. Compared with digital integrated circuit, the effect to the analog circuit is even severe. In this work, we propose an automatic porting methodology considering the LDE to migrate the VLSI system to a new manufacturing process. Furthermore, we evaluate our methodology applying to several practical circuits. We introduce efficient porting methodologies for three kinds of circuit, analog, digital and memory circuit in this work. First, we propose a design methodology via geometric programming (GP) for analog circuits. We model the LDE as posynomial function of GP and then formulate an optimization problem of analog device sizing. Second, we propose a timing closure design scheme with standard cells for digital circuits. Based on this scheme the digital circuit can be easily migrated to other process.

Focusing on design of delay locked loop circuit (DLL), we demonstrate to design the standard cell along with behavior model of the circuit. Finally, we propose an automatic design tool aware scheme for memory circuits with a parameterized cell. Considering LDE, we introduce a memory cell with variable width and height. Furthermore, we show a novel structure memory array with a better trade-off between the cell size and the circuit performance such power and delay.

The portability-aware design methodology of our work is illustrated in Fig. 1.1. Our design flow starts from the specification of circuit to the test and verification stage. The design methodology we propose mainly divides in three parts: analog circuit design methodology, digital circuit design methodology, and memory circuit design methodology. These three kinds of circuits has different deign method as we will describe in the later chapters. Analog design is mainly based on circuit optimization via Geometric programing (Chapter 2). While digital part was mainly design via commercial tools such as Synopsys, etc. However, we also add some idea in designing the standard cells. We propose two kinds of digital standard cells: fixed cells and parameterized cells. The detail will be shown in Chapter 3 and Chapter 4. At same time, we also develop the EDA tools which helps placement and routing more efficiently.

Chapter 2

Analog Integrated Circuit Design

In this chapter, we mainly discuss the design methodology of analog circuit design. There are lot of design method of analog design. Compared to the conventional methods, we mainly using geometric programming to optimize our circuit. We use the circuit of op-amp to demonstrate our idea.

2.1 Introduction

A mixed signal portion has become to occupy about two third of non-memory ICs, while the design effort for analog circuits is spent three times or more than that for digital circuits. In a CMOS analog circuit, the design is composed of the circuit topology generation and the transistor sizing. In general, there could be templates of the topology for a typical circuit. For the transistor sizing, however, it spends much time for an iteration of simulations due to the complex trade-off among the circuit specifications.

As an approach to avoid the iteration of simulations in the transistor sizing, the geometric programming (GP) [9] based circuit synthesis is promising, although the applicable circuits might be limited. Solving a GP problem is very fast and efficient so that many researchers applied GP in their works. S. boyd et al. had described GP theory in [10] and they also had applied GP in many EDA problems [11]. In [12], a complete optimization framework by GP were proposed and its specifications and constraints of a wide variables were formulated as posynomial forms. They showed the design case of a CMOS op-amp where the resultant circuit generated by GP satisfied all the specifications.

In the current technology, short channel length effect can not be ignored as channel length becomes shorter and shorter. A full illustration of channel length modulation (CLM) had been described in [13]. The CLM is a shortening of the length of the inverted channel region with increase in drain bias for large drain bias. The result of CLM is an increase in current with drain bias and reduction of output resistance. The governing factor for channel length modulation is generally taken as a constant, but detail analysis shows the variation of the channel length modulation parameter λ in saturation region. It has been showed that in [14] the CLM parameter λ tends to increase dramatically as the channel length is becoming shorter.

Furthermore, as CMOS technology is scaled down, there is a lack of handling layout-dependent effects (LDE) such as the shallow trench isolation (STI) stress and the well proximity effect (WPE). Sometimes it leads to fatal defects in analog circuit designs. There are several works address the STI and WPE issues. A complete flow was proposed in [15] to characterize the influence of STI stress on performance of RF/analog circuit based on layout design and process information. Moreover, many researchers had focus on the modeling of STI stress. A model for the STI well had been presented in [16][17] contributing to the mobility, and proposes a diffusion filling to improve the timing performance, while a piecewise linear approximation model which is applicable to various STI analysis was provided in [18]. Furthermore, for taking the WPE into account, a MOSFET that is close to the edge of a well region must be characterized by the different threshold voltage and the drain current against the devices located remotely from the edge. It is demonstrated in [19] an impact of the WPE is crucial for analog designs. In [20], both STI stress and WPE had been put into consideration and demonstrated the significance on impact of analog design.

In this work, we presents LDE-aware analog circuit synthesis based on the GP. Our circuit synthesis is consistent with the layout design via the constraint generation for the LDEs. Our contributions in this work are summarized as follows;

- We formulate the STI stress and the WPE in terms of posynomials. These formulations enable us to rewrite the existing formulations for the circuit performances so as to take LDEs into account.
- We propose a novel concept of circuit synthesis for the purpose of not only optimizing transistor sizes but also generating LDE constraints in terms of LDE

variables. To our best knowledge, this is the first work in the literature to present an automatic generation of the LDEs constraints.

- We provide a design case of a two-stage op-amp, and illustrate that the synthesized circuit generated by GP with LDE constraints satisfies all the specifications through the verification of HSPICE simulation based on the BSIM model with LDE options.

The rest of this chapter is organized as follows; Section 2 describes the fundamentals of geometric programming and GP variables for the LDE. In Section 3, we introduce posynomial formulations considering LDEs with surface fitting techniques. Section 4 provides a design case of a two-stage op-amp of which the simulation results based on BSIM model with LDE-options. Section 5 concludes this work.

2.2 Preliminary

The most important feature of geometric programs is that they can be reformulated as convex optimization problems, and therefore globally optimal solutions can be computed with great efficiency, even for problems with hundreds and thousands of constraints. In this section, we first introduce the GP including the basic definitions and solving theories. Then, we describe the LDE-aware GP variables of the Op-amp design case.

2.2.1 Geometric programming problem

Since geometric programming (GP) is the main method to optimize circuit in this chapter , we would like expound relative basic definitions and GP forms in preliminary. The more detailed tutorial can be found in [10].

Let $x_1, \ldots, x_n$ denote n real positive variables, and $x = (x_1, \ldots, x_n)$ a vector with components x_i. A real valued function f of x, with the form

$$f(x) = cx_1^{a_1}x_2^{a_2}\cdots x_n^{a_n} \tag{2.1}$$

where $c > 0$ and $a_i \in R$, is called a *monomial function*. Note that the coefficients c must be nonnegative, but the exponents a_i can be any real numbers, including negative or fractional.

A sum of one or more monomials, *i.e.*, a function of the form

$$f(x) = \sum_{k=1}^{K} c_k x_1^{a_{1k}} x_2^{a_{2k}} \cdots x_n^{a_{nk}} \tag{2.2}$$

where $c_k > 0$, is called a *posynomial function*. Any monomial is also a posynomial. Posynomials are closed under addition, multiplication, and positive scaling. Monomials are closed under multiplication and division.

A *geometric program* is an optimization problem of the following form

$$\begin{array}{lll} \text{minimize} & f_0(x) & \\ \text{subject to} & f_i(x) \leq 1, & i = 1, \ldots, m, \\ & g_i(x) = 1, & i = 1, \ldots, p, \\ & x_i > 0, & i = 1, \ldots, n, \end{array} \tag{2.3}$$

where $f_0, \ldots, f_m$ are posynomial functions and $g_1, \ldots, g_p$ are monomial functions.

A geometric program can be reformulated as a *convex optimization problem*, *i.e.*, the program of minimizing a convex function subject to convex inequality constraints and linear equality constrains. This is the key to our globally and efficiently solve geometric programs. The detailed description of how GPs are solved can be found in [10].

We define new variables $y_i = \log x_i$, and take the logarithm of a posynomial f to get

$$h(y) = \log(f(e^{y_1}, \ldots, e^{y_n})) = \log(\sum_{k=1}^{t} e^{a_k^T y + b_k}) \tag{2.4}$$

where $a_k^T = [a_{1k}, \ldots, a_{nk}]$ and $b_k = \log c_k$. It can be shown that h is a *convex* function of the new variable y: for all $y, z \in \mathbf{R}^n$ and $0 \leq \lambda \leq 1$ we have

$$h(\lambda y + (1 - \lambda)z) \leq \lambda h(y) + (1 - \lambda)h(z) \tag{2.5}$$

We can convert the standard geometric program (2.3) into a convex program by expressing it as

$$\begin{array}{lll} \text{minimize} & \log(f_0(e^{y_1},\ldots,e^{y_n})) & \\ \text{subject to} & \log(f_i(e^{y_1},\ldots,e^{y_n}))\leq 0, & i=1,\ldots,m, \\ & \log(g_i(e^{y_1},\ldots,e^{y_n}))=0, & i=1,\ldots,p. \end{array} \tag{2.6}$$

This is the so-called *convex form* of the geometric program. Convexity of the convex form geometric program has several important implications: we can use efficient interior-point methods to solve them, and there is a complete and useful duality, or sensitivity theory for them.

2.2.2 Mixed-integer geometric programming

In a mixed-integer GP (MIGP), we have a GP with the additional constraint that some of the variables lie in some discrete set, such as the integers [10]:

$$\begin{array}{lll} \text{minimize} & f_0(x) & \\ \text{subject to} & f_i(x)\leq 1, & i=1,\ldots,m, \\ & g_i(x)=1, & i=1,\ldots,p, \\ & x_i\in N, & i=1,\ldots,n, \end{array} \tag{2.7}$$

where N denotes the set of natural numbers, i.e., positive integers. Mixed-integer GPs are in general very hard to solve, and all methods for solving them make some compromise, compared to methods for GP. The simplest heuristic method is to ignore the integer constraints in Eq. (2.7), and solve the resulting GP (which can always be done fast). This problem is called the GP relaxation of the MIGP, since we have relaxed the integer constraints. The optimal value of the relaxed GP is a lower bound on the optimal value of the MIGP, since when we relax we expand the feasible set (and therefore reduce the optimal value). To obtain an approximate solution of the MIGP, we simply round each of the variables $x_i,\ldots,x_n$ towards the nearest integer. Then, we fix the integer variables (i.e., treat them as constants) and solve the resulting GP.

Since Ecker's survey paper there have been several important developments, related to solving geometric programming in the exponential form. A huge improvement in computational efficiency was achieved in 1994, when Nesterov and Nemirovsky developed efficient interior point algorithms to solve a variety of nonlinear optimization problems,

including geometric programs. Recently, Kortanek et al. have shown how the most sophisticated primal-dual interior-point methods used in linear programming can be extended to geometric programming, resulting in an algorithm approaching the efficiency of current interior-point linear programming solvers. The algorithm they describe has the desirable feature of exploiting sparsity in the problem, i.e., efficiently handling problems in which each variable appears in only a few constraints.

For our purposes, the most important feature of geometric programs is that they can be globally solved with great efficiency. Problems with hundreds of variables and thousands of constraints are readily handled, on a small workstation, in minutes; the problems we encounter in this chapter, which have a few tens of variables and fewer than 100 constraints, are easily solved in under one second. To carry out the designs in this work, we implemented, in MATLAB, a simple and crude primal barrier method for solving the exponential form problem . Despite the simplicity of the algorithm (i.e., primal only, with no sparsity exploited) and the overhead of an interpreted language, the geometric programs arising in this work were all solved in approximately one to two seconds.

Perhaps even more important than the great efficiency is the fact that algorithms for geometric programming always obtain the global minimum. Infeasibility is unambiguously detected: if the problem is infeasible, then the algorithm will determine this fact, and not just fail to find a feasible point. Another benefit of the global solution is that the initial starting point is irrelevant; the same global solution is found no matter what the initial starting point is.

These properties should be compared to general methods for nonlinear optimization, such as sequential quadratic programming, which only find locally optimal solutions, and cannot unambiguously determine infeasibility. As a result, the starting point for the optimization algorithm does have an affect on the final point found. Indeed, the simplest way to lower the risk of finding a local, instead of global, optimal solution, is to run the algorithm several times from different starting points. This heuristic only reduces the risk of finding a nonglobal solution.

2.2.3 Layout-dependent Effects

Shallow trench isolation (STI), also known as Box Isolation Technique, is an integrated circuit feature which prevents electrical current leakage between adjacent semiconductor device components. STI is generally used on CMOS process technology nodes of 250 nanometers and smaller. Older CMOS technologies and non-MOS technologies commonly use isolation based on LOCOS.

STI is created early during the semiconductor device fabrication process, before transistors are formed. The key steps of the STI process involve etching a pattern of trenches in the silicon, depositing one or more dielectric materials (such as silicon dioxide) to fill the trenches, and removing the excess dielectric using a technique such as chemical-mechanical planarization.

Certain semiconductor fabrication technologies also include deep trench isolation, a related feature often found in analog integrated circuits.

The effect of the trench edge has given rise to what has recently been termed the "reverse narrow channel effect" or "inverse narrow width effect". Basically, due to the electric field enhancement at the edge, it is easier to form a conducting channel (by inversion) at a lower voltage. The threshold voltage is effectively reduced for a narrower transistor width. The main concern for electronic devices is the resulting subthreshold leakage current, which is substantially larger after the threshold voltage reduction.

Highly scaled bulk CMOS technologies make use of high energy implants to form the deep retrograde well profiles needed for latch-up protection and suppression of lateral punch-through . During the implant process, atoms can scatter laterally from the edge of the photoresist mask and become embedded in the silicon surface in the vicinity of the well edge, as illustrated in Fig. 1. The result is a well surface concentration that changes with lateral distance from the mask edge, over the range of 1um or more. This lateral non-uniforimity in well doping causes the MOSFET threshold voltages and other electrical characteristics to vary with the distance of the transitor to the well-edge. This phenomenon is commonly known as the well proximity effect (WPE).

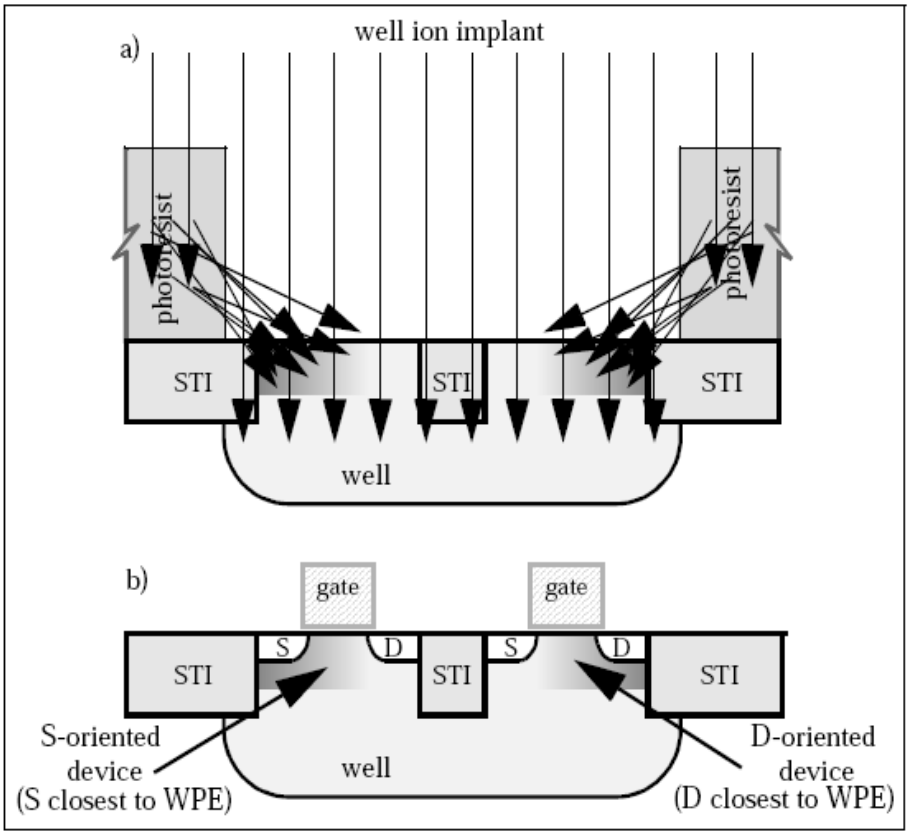

FIGURE 2.1: A depiction of the WPE.

2.3 LDE-aware GP variables

This work is mainly to optimize analog circuit considering LDE effect. While using GP, we must firstly find these LDE-aware GP variables. Although, in [12], some circuit performance variables were already defined, in this subsection, we set advanced variables and take op-amp as example to explain how we define the LDE variables.

A typical two-stage op-amp used in this work is shown in Fig. 2.2. A differential amplifier in the first stage converts the differential input voltage to differential currents. The differential currents are applied to the current-mirror load converting the differential voltage. A common-source in the second stage converts the input voltage to current. Op-amps, which are among the most widely used electronic devices today, have been used in a vast array of consumer, industrial, and scientific devices. Many standard IC op-amps cost only a few cents in moderate production volume; however some integrated or hybrid operational amplifiers with special performance specifications may cost much money in small quantities. Op-amps may be packaged as components, or used as elements of more complex integrated circuits.

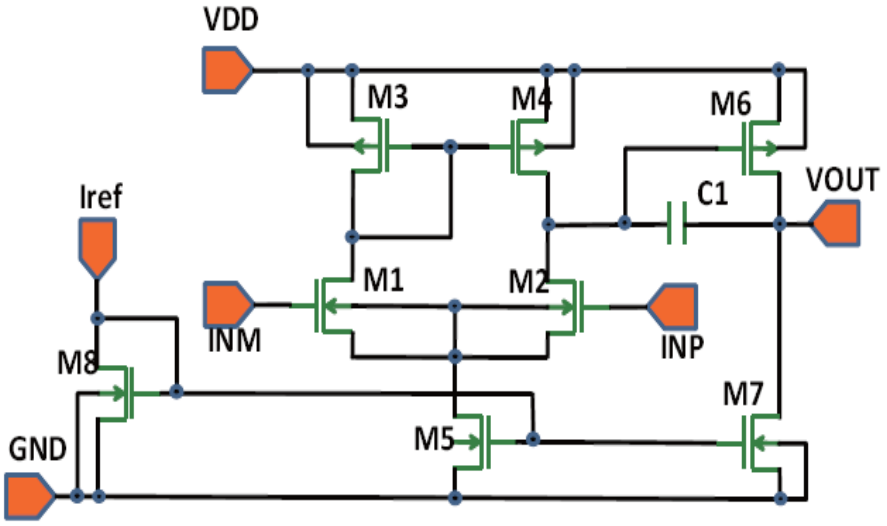

FIGURE 2.2: Two stage op-amp.

To design an op-amp, first we need to focus on those parameters that determine the circuit performance as transistor size, the bias current, etc. Those parameters include:

- The ratio of the widths and lengths for all transistors without any folding, *i.e.*, $S_1, \ldots, S_8$.
- The reference current I_{ref}.
- The value of compensation capacitor C_c.

Then, we need to characterize layout parameters considering the layout-dependent constraints. The main influence factor for STI stress is the length of diffusion (LOD) according to BSIM4 model [21][22]. In our design problem, we basically modulate the LOD through varying NF (finger number), SA and SB, as shown in Fig. 2.3. However, SD is set as constant so that the equations related to STI can be easily transformed into posynomials of GP. For WPE, SC which is defined as the closest distance from the device to a single well edge, is the main influence factor. Thus, we have the following parameter:

- The finger number of each transistor, $NF_1, \ldots, NF_8$.
- The distance between OD edge to ploy from one side, $SA_1, \ldots, SA_8$. For all transistors, we assume that $SA = SB$.

- The closest distance from the transistor to single well edge $SC_1, \ldots, SC_8$.

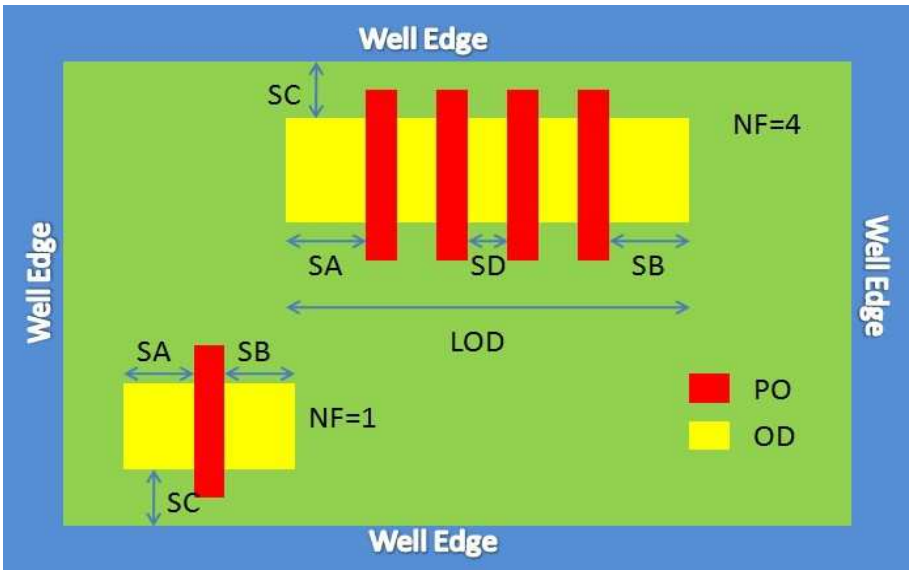

FIGURE 2.3: The variables for layout-dependent effect.

These parameters above are set as GP variables and will be optimized during the circuit synthesis via GP.

2.4 Posynomial Formulation for LDE

When we solve an optimization problem via GP, the objective and the constraints must be in the GP form of Eq. (2.3). That is, all the constraints must be in monomial or posynomial form. However lots of original equations are not monomial or posynomials. In this section, we formulate the posynomial functions for channel length, STI stress and WPE. For other constraints, we summarized them briefly in the last subsection.

2.4.1 Short Channel Effect

One of several short channel effects in MOSFET scaling, channel length modulation λ is a shortening of the length of the inverted channel region with increase in drain bias for large drain biases. The result of λ is an increase in current with drain bias and a reduction of output resistance. Channel length modulation is important because it decides the MOSFET output resistance, an important parameter in circuit design

of current mirrors and amplifiers. The hand calculation equation for the gain of the two-stage op-amp is,

$$A_v = \frac{2C_{ox}}{(\lambda_n + \lambda_p)^2}(\mu_{eff,n}\mu_{eff,p}\frac{W_2W_6}{L_2L_6I_1I_7})^{1/2} \tag{2.8}$$

where λ_n and λ_p are always set as constant. Eq. (2.8) only suit for the long channel devices, but it is different for short channel ones.

We have tested a series of data for channel length versus λ value for different kind of transistors with 90nm process [14]. The channel length of tested transistors are from 100nm to 500nm as depicted in Fig. 2.4. We have observed that as the channel length decreases, λ increases, especially beyond 180nm process. For channel length beyond 90nm, we can predict that the λ will increase pretty high. Usually, a small λ is better when we design high gain op-amp.

Using curve fitting, we approximate this trade-off curve into the following form,

$$\lambda_n = k_nL^{\tau_n}, \quad \lambda_p = k_pL^{\tau_p} \tag{2.9}$$

where $k_n = 4 \times 10^{-8}$, $k_p = 1 \times 10^{-5}$, and $\tau_n = -1.005, \tau_p = -0.630$. Replace the Eq. (2.9) into (2.8), the gain function turns to,

$$A_v = \frac{2C_{ox}}{(k_nL^{\tau_n} + k_pL^{\tau_p})^2}(\mu_{eff,n}\mu_{eff,p}\frac{S_2S_6}{I_1I_7})^{1/2} \tag{2.10}$$

where S_2, S_6 is the ratio W_2/L_2, W_6/L_6. The new gain equation Eq. (2.10) is not a posynomial function because of the denominator. Since the objective function of the GP in the op-amp design is to minimize the size of the whole circuit, we can set L as a constant and sweep it from the minimum value for generating the optimal channel length L.

Since the GP is executed again and again with L sweeping from the minimum value, the gain equation at each GP execution can be expressed as

$$A_v = \frac{2C_{ox}}{LM}(\mu_{eff,n}\mu_{eff,p}\frac{S_2S_6}{I_1I_7})^{1/2} \tag{2.11}$$

where LM equals to,

$$LM = (k_nL^{\tau_n} + k_pL^{\tau_p})^2 \tag{2.12}$$

In each execution of GP, LM is constant because of the sweeping L. Thus, Eq. (2.11) is a posynomial function.

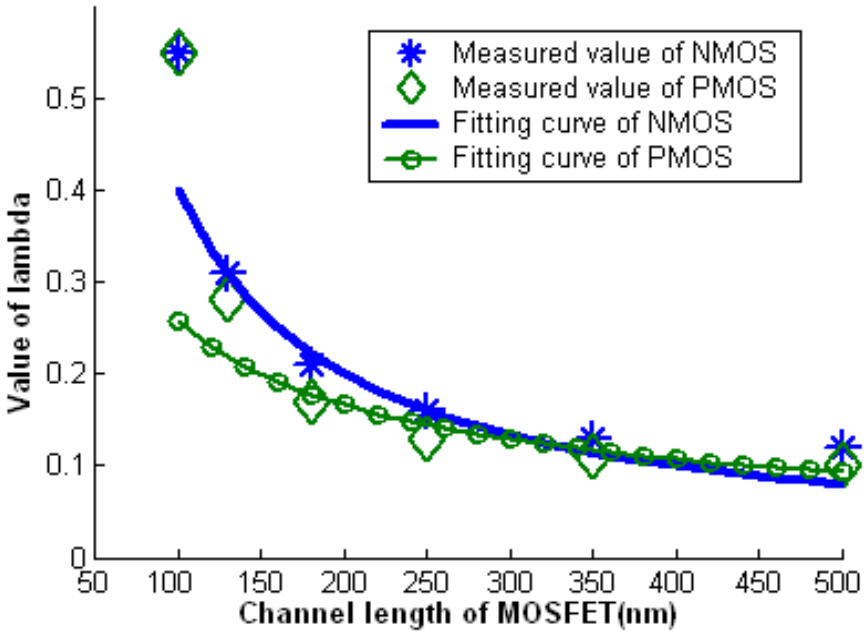

FIGURE 2.4: Curves of channel length modulation λ for NMOS and PMOS.

2.4.2 STI stress of BSIM model

As CMOS technology is scaling down, the mechanical stress, which is used to be the secondary concern of the circuit design, now becomes one of the major factors determining circuit performance. Different from other intentional mechanical stresses, STI stress, which is exerted by STI wells on active region of the device, is inevitably formed and has increasingly significant impact on device behaviour, especially in aggressively scaled-down CMOS technology.

BSIM models which developed at UC Berkeley are one of the standard models that have been used in commercial and industrial analog simulators such as HSPICE. Since the widely used HSPICE is also used in our laboratory, we aimed to create a posynomial for STI stress that matches to the BSIM models. In BSIM4 model [21][22], a STI stress model which follows the $1/LOD$ trend but reveal different L and W scaling had been developed.

TABLE 2.1: Stress Effect Model Parameters.

Parameter	Description	Value*
SA, SB	Distance between OD edge to Poly	Variable
SD†	Distance between OD neighbouring fingers	$0.2\mu m$
WLOD	Width parameter for stress effect	2×10^{-6}
KU0	Mobility degration/enhancement coefficient for strss effect	N:-4×10^{-6} P:4×10^{-6}
LKU0	Length dependence of KU0	1×10^{-6}
WKU0	Width dependence of KU0	1×10^{-6}
TKU0‡	Temperature coefficient of KU0	0
PKU0‡	Cross-term dependence of KU0	0
LLODKU0	Length parameter for U0 stress effect	1.1
WLODKU0	Width parameter for U0 stress effect	1.1
XL	Channel length offset due to mask/etch effect	0
XW	Channel width offset due to mask/etch effect	0

*These values are different for different process. In this fitting, they are set as MOS model card of [23].

†The value of SD is set as constant of the minimum value from mask layout rules.

‡For design considering temperature and cross-term effect, the value of TKU0 and PKU0 could be non-zero.

This model introduces the mechanism by adjusting the $U0$ and V_{sat} according to different W, L and OD shapes. Define mobility relative change due to stress effect as:

$$\begin{aligned} \rho_{\mu_{eff}} &= \Delta\mu_{eff}/\mu_{eff0} \\ &= (\mu_{eff} - \mu_{eff0})/\mu_{eff0} = \frac{\mu_{eff}}{\mu_{eff0}} - 1 \end{aligned} \tag{2.13}$$

then, we can obtain,

$$\mu_{eff} = \mu_{eff0}(1 + \rho_{\mu_{eff}}) \tag{2.14}$$

We assume that mobility relative change is proportional to stress distribution. It can be described as function of SA, $SB(LOD$ effect), L, W, and T dependence[21][22]:

$$\rho_{\mu_{eff}} = \frac{KU0}{K_s \cdot (Inv\_sa + Inv\_sb)} \tag{2.15}$$

We may write Inv_sa and Inv_sb in one term as

$$Inv\_sab = Inv\_sa + Inv\_sb \tag{2.16}$$

Thus, Eq. (2.15) turns to be

$$\rho_{\mu_{eff}} = \frac{KU0}{K_s \cdot Inv\_sab} \tag{2.17}$$

The calclulations of Inv_sa and Inv_sb in Eq. (2.15) according to BSIM4 model [21][22] are

$$Inv\_sa = \frac{1}{SA + 0.5 \cdot L} \tag{2.18}$$

$$Inv\_sb = \frac{1}{SB + 0.5 \cdot L} \tag{2.19}$$

For multiple finger device, the total LOD effect is the average of LOD effect to every finger as shown in Fig. 2.3. That is:

$$Inv\_sa = \frac{1}{NF} \sum_{i=0}^{NF-1} \frac{1}{SA + 0.5 \cdot L + i \cdot (SD + L)} \tag{2.20}$$

$$Inv\_sb = \frac{1}{NF} \sum_{i=0}^{NF-1} \frac{1}{SB + 0.5 \cdot L + i \cdot (SD + L)} \tag{2.21}$$

Since we assume SA and SB are the same value, Eq. (2.20) and Eq. (2.21) can be combined as

$$Inv\_sab = \frac{1}{NF} \sum_{i=0}^{NF-1} \frac{2}{SA + 0.5 \cdot L + i \cdot (SD + L)} \tag{2.22}$$

The denominator K_s in Eq. (2.15) according to BSIM4 manual [21][22] can be calculate as,

$$\begin{aligned} K_s = (1 &+ \frac{LKU0}{(L + XL)^{LLODKU0}} \\ &+ \frac{WKU0}{(W + XW)^{WLODKU0}} + \frac{PKU0}{(L + XL)^{LLODKU0}} \\ &\cdot \frac{1}{(W + XW + WLOD)^{WLODKU0}}) \\ &\cdot (1 + TKU0 \cdot (\frac{Temprature}{TNOM} - 1)) \end{aligned} \tag{2.23}$$

The meaning of parameters in Eq. (2.23) is shown in Table 2.1. As seen in Table 2.1, TKU0 and PKU0 are set as 0 as we do not consider the temperature and cross-term effect. Mask/etch effect is not the discussion issue in this work, so XL and XW are also set as 0. Thus Eq. (2.23) can be simplified as

$$K_s = 1 + \frac{LKU0}{L^{LLODKU0}} + \frac{WKU0}{W^{WLODKU0}} \tag{2.24}$$

Eq. (2.17) and Eq. (2.24) are monomial and posynomial respectively if we regard K_s

and Inv_sab as intermediate variables. However, Eq. (2.22) refers to discrete variable of NF and it is not a posynomial obviously. So we need to transform it in posynomial form. For variation of L and W in the above equations, experiment results showed that its effect is so tiny that it can be neglected. From mask design rule, we give SD a fixed value. And other model parameters can be found in BSIM manual. Now we leave SA and NF as variables. Then we observed the 3D trade-off surface of μ_{eff}/μ_{effo} versus NF and SA assuming that $SA = SB$ (see Fig. 2.5 and Fig. 2.6). The dotted point is the calculated point, while the smooth surface is obtain by using surface fitting.

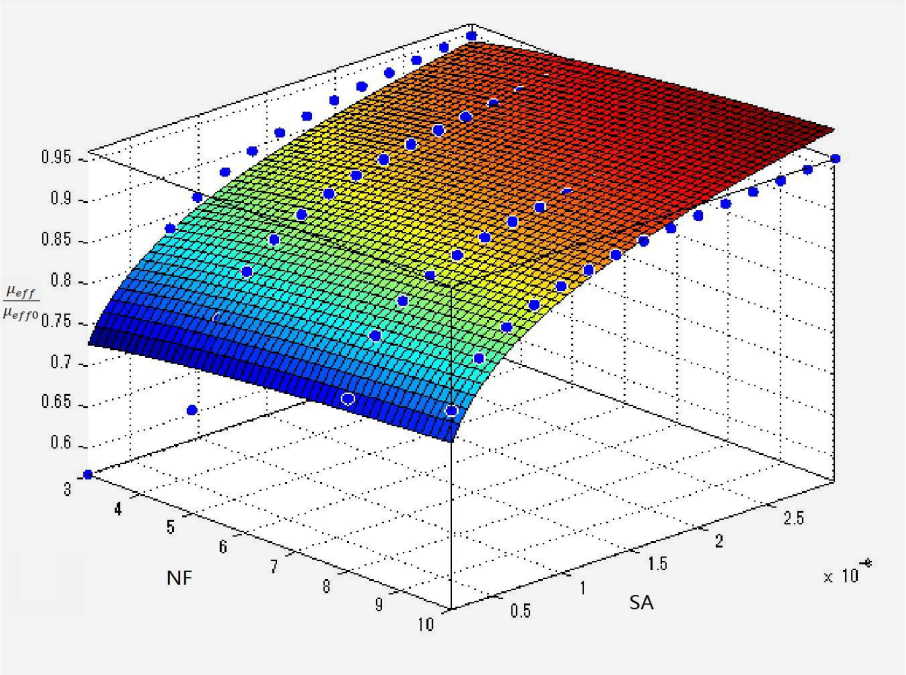

FIGURE 2.5: The 3D trade-off surface of μ_{eff}/μ_{effo} vs NF and SA for NMOS using fitting model $\mu_{eff}/\mu_{effo} = aNF^bSA^c$.

The coefficients of fitting surfaces are summed up in Table 2.2. The two equations are monomials, obviously. In order to indicates how well data points fit the surface, we introduce the coefficient of determination R^2.

The coefficient of R^2 can be calculated as,

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{2.25}$$

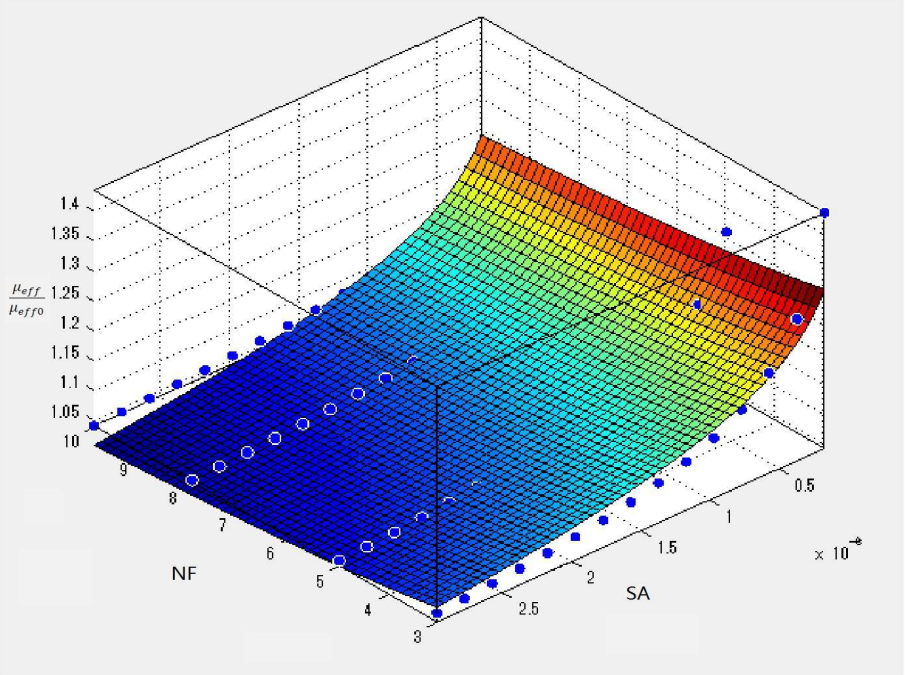

FIGURE 2.6: The 3D trade-off surface of μ_{eff}/μ_{effo} vs NF and SA for PMOS using fitting model $\mu_{eff}/\mu_{effo} = aNF^bSA^c$.

where SS_{tot} is the total sum of squares, SS_{res} is the sum of squares of residuals. SS_{tot} and SS_{res} can calculated as

$$SS_{tot} = \sum_i (y_i - \bar{y})^2 \tag{2.26}$$

$$SS_{res} = \sum_i (y_i - f_i)^2 \tag{2.27}$$

In the above equations of Eq. (2.26) and Eq. (2.27), y_i can be seen as the calculation value of μ_{eff}/μ_{effo} and f_i can be seen as the fitting value of μ_{eff}/μ_{effo}. $\bar{y}$ is the mean value of the total y_i,

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{2.28}$$

where n is the number of observations.

The R^2 which usually has a value between 0 and 1.0. An R^2 near 1.0 indicates that a regression line fits the data well, while an R^2 closer to 0 indicates a regression line does not fit the data very well. The two fitting surfaces are fitted well by having an R^2 of 0.943 and 0.935 respectively.

In the STI relative constraints, the variation of SA and NF will affect the mobility μ_{eff}. As seen in Fig. 2.5 and Fig. 2.6, the average variation of μ_{eff} is about 0.83% if SA

TABLE 2.2: Summary of fitting curves or surfaces for stress effect.

Trade-off type	Fitting Model	Coefficient Values	R^2
$\frac{\mu_{eff,n}}{\mu_{effo,n}}$ vs. NF and SA	$\frac{\mu_{eff,n}}{\mu_{effo,n}} = aNF^bSA^c$	a: 2.85 b: 0.50 c: 0.09	0.943
$\frac{\mu_{eff,p}}{\mu_{effo,p}}$ vs. NF and SA	$\frac{\mu_{eff,p}}{\mu_{effo,p}} = aNF^bSA^c$	a: 0.44 b: -0.05 c: -0.07	0.935

changed $0.1\mu m$. If NF is increased/decreased with one finger, the average variation of μ_{eff} is about 0.003%. Comparing to NF and SA, it is obviously that the ratio of the transistor S is the critical variable as shown in Eq. (2.11) and other constraints that we will discuss in Section 2.4.4, such as Eq. (2.37), (2.38), (2.39), and (2.40). Thus, the error caused by fitting curve of NF and SA is negligible.

2.4.3 WPE of BSIM model

Experimental analysis [24] shows that well proximity effect is strong function of distance of FET from mask edge, and electrical quantities influenced by it follow the same geometrical trend. A phenomenological model based on these findings has been developed by modifying some parameters in the BSIM model.

TABLE 2.3: Well-Proximity Effect Model Parameters.

Parameter	Description	Value
SCA	Integral of the first distribution function	\
SCB	Integral of the second distribution function	\
SCC	Integral of the third distribution function	\
SC	Distance to a single well edge	Variable
WEB	Coefficient for SCB	0
WEC	Coefficient for SCC	0
KU0WE	Mobility degradation factor for well proximity effect	0.1
SC_{ref}	Reference distance to calculate SCA, SCB and SCC	$1.0\mu m$

Due to the well proximity effect, the new mobility equation can be described as [21][22]:

$$\mu_{eff} = \mu_{eff,org} \cdot (1 + KU0WE \cdot (SCA + WEB \cdot SCB + WEC \cdot SCC)) \tag{2.29}$$

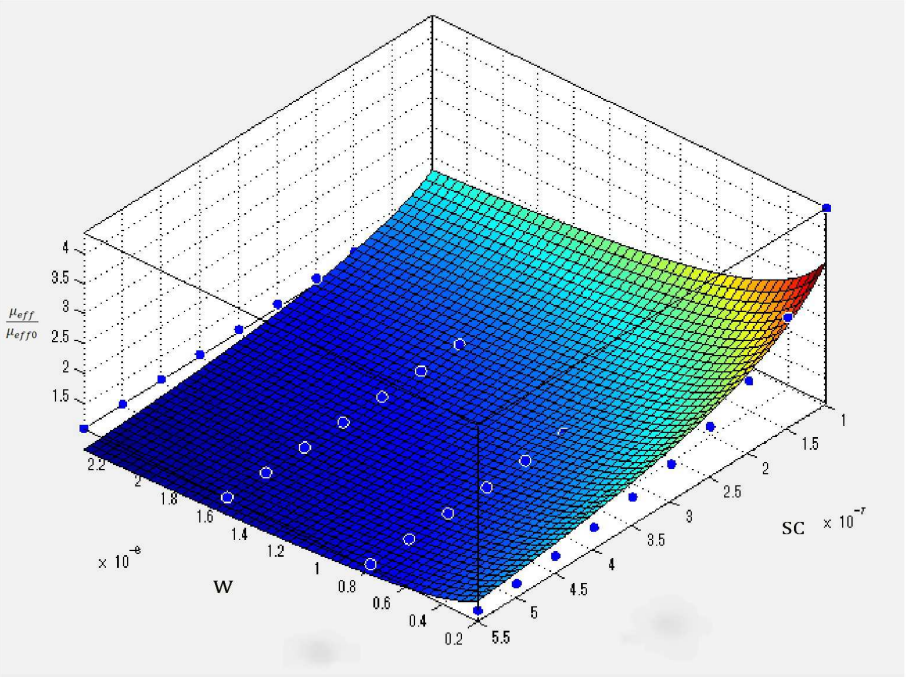

FIGURE 2.7: The fitting surface of WPE.

SCA, SCB, and SCC are the first, second, and third distribution function for scattered well dopant and the detailed description is shown in Table 2.3. KU0WE, WEB, WEC are fitting coefficients. For our model card, WEB and WEC are set as 0, so the above equation turns to:

$$\mu_{eff} = \mu_{eff,org} \cdot (1 + KU0WE \cdot SCA) \tag{2.30}$$

SC is defined as "The distance to a single well edge used in calculation of SCA when layout information is not available". If SCA is not given due to lack of detailed layout information, their estimation [25] can be made by simulators based on the assumption that for most of layouts, the devices are close to only one well edge(Fig. 2.3):

$$SCA = \frac{SC_{ref}^2}{W_{drawn}} \cdot (\frac{1}{SC} - \frac{1}{SC + W_{drawn}}) \tag{2.31}$$

For multi-finger case, we still assume that there is only one well edge close to the multi-finger devices as shown in Fig. 2.3. Thus, the calculation of SCA is the same as Eq. (2.31).

Using the same method of STI posynomial generation, this time we observe the trade-off of μ_{eff}/μ_0 versus W and SC. Take the value of $KU0WE = 0.1$, $WEB = WEC = 0$, $SC_{ref} = 1\mu m$. We constructed the fitting curve, as shown in Fig. 2.7. And the fitting

monomial equation is,

$$\mu_{eff}/\mu_o = aW_{drawn}^b SC^c \tag{2.32}$$

W_{drawn} in the above equation can be rewritten as

$$W_{drawn} = W/NF \tag{2.33}$$

and the final fitting equation is summed in Table 2.4 with an R^2 of 0.929 (The calculation of R^2 is the same as that we discussed in Section 2.4.2).

TABLE 2.4: Summary of fitting curves or surfaces for WPE.

Trade-off type	Fitting Model	Coefficient Values	R^2
$\frac{\mu_{eff}}{\mu_{effo}}$ vs. NF ,SC and W	$\frac{\mu_{eff}}{\mu_{effo}} = a(\frac{W}{NF})^b SC^c$	a: 1.04×10^{-5} b:-0.028 c:-0.527	0.929

In the WPE relative constraints, the W_{drawn} and SC will affect the mobility μ_{eff}. SC under $1.5\mu m$ is very sensitive parameter because the average variation of μ_{eff} will achieve over 200% per $0.1\mu m$. However, SC larger than $1.5\mu m$ has only about 1.0% average variation of μ_{eff} per $0.1\mu m$. According to mask layout rule, the SC has the minimum length of $1.5\mu m$. Thus, we can neglect the sensitive effect when SC under $1.5\mu m$. As for W_{drawn} , we find that it has 1.5% variation of μ_{eff} per $0.1\mu m$. Comparing to critical variable S in Eq. (2.11) and other constraints that we will discuss in Section 2.4.4, such as Eq. (2.37), (2.38), (2.39), and (2.40), the error caused by fitting the WPE relative constraint is negligible.

2.4.4 Posynomials for LDE-aware Circuit Performances

The two-stage op-amp circuit had been analyzed in many literatures, especially in [12] where a totally design of the circuit performance had been analyzed. In this section, we summarized the main constraints in which some parameters as mobility of $\mu 0$ were improved considering LDE. The original constant parameter $\mu 0$ is replaced as μ_{eff} which turns to be a monomial function as shown in Table 2.2 and Table 2.4.

- **Area**

The size of all the transistor areas can be approximately calculate as,

$$A = \sum_{i=1}^{8} L \cdot S_i(2SA + NF_i(L + SD)) \tag{2.34}$$

- **Transistors symmetry**

  For the input transistor pairs M_1 and M_2 and the current mirror M_3 and M_4, the layout of each pair should be identical,

$$\begin{aligned} &S_1 = S_2, NF_1 = NF_2, SA_1 = SA_2, SC_1 = SC_2. \\ &S_3 = S_4, NF_3 = NF_4, SA_3 = SA_4, SC_3 = SC_4. \end{aligned} \tag{2.35}$$

  These conditions are equality constraints between monomials, and are therefore readily handled by geometric programming.

- **Offset voltage**

  To reduce input offset voltage, the drain voltages of M_3 and M_4 must be equal, ensuring that the current I_5 is split equally between transistors M_1 and M_2.

$$S_3/S_6 = S_4/S_6 = S_5/(2S_7) \tag{2.36}$$

- **ICMR**

  $V_{cm,max}$ and $V_{cm,min}$ of the input common mode range (ICMR) need to be set to make transistors M_1, M_3 and M_5 stay in saturation,

$$\begin{aligned} (\frac{I_1}{\mu_{eff,n}C_{ox}/2S_1})^{1/2} + (\frac{I_5}{\mu_{eff,n}C_{ox}/2S_5})^{1/2} \\ \leq V_{DD} - V_{cm,max} + V_{TP} \end{aligned} \tag{2.37}$$

$$(\frac{I_1}{\mu_{eff,p}C_{ox}/2S_3})^{1/2} \leq V_{cm,min} - V_{SS} - V_{TP} - V_{TN} \tag{2.38}$$

- **Output Range**

  $V_{out,max}$ and $V_{out,min}$ of the output range need to be set to make transistors M_6 and M_7 stay in saturation,

$$(\frac{I_7}{\mu_{eff,p}C_{ox}/2S_6})^{1/2} \leq V_{out,min} - V_{SS} \tag{2.39}$$

$$(\frac{I_7}{\mu_{eff,n}C_{ox}/2S_7})^{1/2} \leq V_{dd} - V_{out,max} \tag{2.40}$$

- **Power**

  The quiescent power of the circuit is also need to be evaluated,

$$P = (V_{dd} - V_{ss})(I_{ref} + I_5 + I_7) \tag{2.41}$$

- **Gain**

  The improved gain function as state in Section 3.1,

$$A_v = \frac{2C_{ox}}{LM}(\mu_{eff,n}\mu_{eff,p}\frac{S_2S_6}{I_1I_7})^{1/2} \tag{2.42}$$

- **Gain bandwidth**

  The unity gain bandwidth can be approximated as

$$\omega_c = \frac{g_{m1}}{C_c} \tag{2.43}$$

- **Phase margin**

  The phase margin are calculated as

$$\sum_{i=2}^{4} arctan(\frac{\omega_c}{p_i}) \leq \frac{\pi}{2} - PM_{min} \tag{2.44}$$

  A simple approximation is given by $arctan(x) \approx x$, which is quite accurate for $arctan(x)$ less than 25°. Thus, we can approximate the phase margin constraint accurately as

$$\sum_{i=2}^{4} \frac{\omega_{c,approx}}{p_i} \leq \frac{\pi}{2} - PM_{min} \tag{2.45}$$

- **Slew rate**

  The slew rate can be expressed as

$$SR = min[2I_1/C_c, I_7/(C_c + C_{TL})] \tag{2.46}$$

This function can be rewritten as

$$\frac{C_c}{2I_1} \leq \frac{1}{SR_{min}}, \frac{C_c + C_{TL}}{I_7} \leq \frac{1}{SR_{min}}. \tag{2.47}$$

These two constraints are posynomials.

2.5 Design Case of Op-amp Circuit

In this section, we applied our GP optimization to a typical Op-amp circuit. In the first subsection we described the optimization framework with constraints, variables and other settings. In the second subsection, we executed HSPICE simulation using the optimal results. Meanwhile, we analysed the trade-off of LDE parameters.

2.5.1 Optimization framework

Analyzing the aforementioned specifications, we describe objectives and constraints according to the GP problem format, where the objective is to minimize the area, and the other circuit performances and LDEs are set as constraints, as shown in Table 2.5. Originally, a ratio of the width and length for each transistor is a variable. However, we assume each channel length to be a constant in the optimization process, so that each channel width is regarded as a variable. In order to determine the constant of channel length, we sweep the channel length from the minimum value to the maximum value according to the mask layout rule with 50nm intervals. We execute the GP again and again until its solution is feasible.

To clarify the validity of our concept, we developed design case of a typical two-stage op-amp on 90nm CMOS technology. The supply voltage is set to be 1.2V, and the load capacitance is 1pF and the specifications are shown in Table 2.7 and Table 2.8. Using GGPLAB [26] which is a Matlab-based toolbox for specifying and solving GP problems, we carried out two groups of optimizations as shown in Table 2.5 and the results are shown in Table 2.6. For both group of optimizations, we find that the GP solution is infeasible when channel length shorter than 0.5μm so as to achieve the high gain (larger than 60dB). In the first group (Group I) of GP optimizations, we consider only circuit performance constraints in which mobility μ_{eff} is set as constant and NFs are set as 1.

TABLE 2.5: Summary of GP form for 2-stage Op-amp Synthesis

	Group I	Group II	Description
Min.	Area	Area	Eq. (2.34)
Constraints	Circuit performance constraints.	Circuit performance constraints.	Eq. (2.35)∼(2.46)
		LDE constraints	Eq. (2.9), Table 2.2,2.4
Variables	S_i,C_c,I_{ref}	S_i,C_c,I_{ref}, SA_i, SC_i, NF_i	$i = 1, 2...8$
Parameter μ_{eff}	Constant	Intermediate variable	
Verification	Simlation IA, IB	Simlation II	Table 2.7,2.8

SA and SC are not seen as variables, so the results include only width and length. In the second group (Group II) of GP optimizations, we consider both circuit performance constraints and LDE constraints as shown in table 2.5. Besides width and length of each transistor, the optimal values of the second group include the number of finger NF and the length of SA and SC. Comparing the two groups of results, we can find that the widths of M6 and M7 are slightly changed if we consider LDE constraints.

The calculated results of circuit performances via GP are shown in Column "Program" Table 2.7 and Table 2.8. Observing Column "Program" of Group I and Group II, we can find that the output swing is the critical constraints because its value equals to the specification. In order to achieve high gain, we need large size of transistors of M2 and M6 but small current values of I_1 and I_7, as shown in Eq. (2.42). The size of M7 becomes large with M6 because of the offset voltage as shown in Eq. (2.36). Meanwhile, the right side of output range constrain is only 0.3V ($V_{out,min} - V_{SS}$ and $V_{dd} - V_{out,max}$), as shown in Eq. (2.39) and Eq. (2.40). Since our design objective is to minimum the size of the op-amp, the size of each transistor should be as small as possible. As a result, minimizing the large size of M6 and M7 will violate the constraint of output range very easily. Therefore, the constraints of the output range become critical because of the tight constraints of output range as shown in Eq. (2.39) and Eq. (2.40).

2.5.2 Simulation Verification using HSPICE

In this section, we did several simulations to verify the efficiency of our GP optimization results.

TABLE 2.6: Optimal results of the two-stage op-amp design problem.

	Group I	Group II			
Item	W_i/L_i	W_i/L_i	NF_i	SA_i	SC_i
M1	$3.0\mu m/0.5\mu m$	$3.0\mu m/0.5\mu m$	4	$0.18\mu m$	$1.5\mu m$
M2	$3.0\mu m/0.5\mu m$	$3.0\mu m/0.5\mu m$	4	$0.18\mu m$	$1.5\mu m$
M3	$3.0\mu m/0.5\mu m$	$3.0\mu m/0.5\mu m$	1	$0.18\mu m$	$1.5\mu m$
M4	$3.0\mu m/0.5\mu m$	$3.0\mu m/0.5\mu m$	1	$0.18\mu m$	$1.5\mu m$
M5	$5.0\mu m/0.5\mu m$	$5.0\mu m/0.5\mu m$	4	$0.18\mu m$	$1.5\mu m$
M6	$58.0\mu m/0.5\mu m$	$71\mu m/0.5\mu m$	3	$0.18\mu m$	$1.5\mu m$
M7	$56.0\mu m/0.5\mu m$	$55\mu m/0.5\mu m$	10	$0.18\ \mu m$	$1.5\mu m$
M8	$5.0\mu m/0.5\mu m$	$5.0\mu m/0.5\mu m$	4	$0.18\mu m$	$1.5\mu m$
C_c	$0.22pF$	$0.20pF$			
I_{ref}	$6.0\mu A$	$5.8\mu A$			
Time	3.22s	3.64s			

TABLE 2.7: Comparison of the Simulation with specifications(Group I).

		Group I		
	Specification	Program	Simulation IA	Simulation IB
STI and WPE options	-	none	off	on
NF,SA,SB,SC setting	-	none	none	default
Area	minimize	$1663\mu m^2$	-	-
ICMR	[0.4, 0.8]V	[0.3, 0.9]V	[0.2, 1.0]V	[0.3, 1.0]V
Output swing	[0.3, 0.9]V	[0.2, 0.9]V	[0.2, 1.0]V	[0.2, 1.0]V
Quiescent power	$\leq$2mW	0.19mW	0.09mW	0.07mW
Open-loop gain	$\geq$60dB	66dB	67dB	42dB
GB	$\geq$10MHz	12MHz	10MHz	8MHz
Phase margin	$[60°, 90°]$	78°	80°	100°
Slew rate	$\geq$10V/μs	26V/μs	20V/μs	20V/μs

First, we analyzed the trade-off curve between the channel length and the open-loop gain as shown in Fig. 2.8. Changing the channel length with the other constraints fixed, we checked the open-loop gain by the simulation. Since the simulation is performed in 90nm process, we varied the length from 100nm to 600nm. The star points in the figure corresponds to the simulation values, while the solid line does to the values of the GP formulation. It is observed that as the channel length increases, the open-loop gain increases. Op-amp with shorter channel length will have lower open loop gain. However, if we use Eq. (2.8), the results of gain would stay invariable obviously. Therefore, this result convinces us that fitting curves of Eq. (2.9) is reasonable.

Second, we verified an impact caused by the STI and the WPE using HSPICE. Using BSIM 4.5 or later model, the STI stress and the WPE can be simulated by setting

TABLE 2.8: Comparison of the Simulation with specifications(Group II).

		Group II	
	Specification	Program	Simulation II
STI and WPE options	-	LDE constraints	on
NF,SA,SB,SC setting	-	LDE variables	GP optimal results
Area	minimize	$1825\mu m^2$	-
ICMR	[0.4, 0.8]V	[0.3, 0.9]V	[0.3, 1.0]V
Output swing	[0.3, 0.9]V	[0.3 1.0]V	[0.2, 1.0]V
Quiescent power	$\leq$2mW	0.14mW	0.09 mW
Open-loop gain	$\geq$60dB	65dB	66dB
GB	$\geq$10MHz	12MHz	10MHz
Phase margin	$[60°, 90°]$	75°	80°
Slew rate	$\geq$10V/μs	25V/μs	20V/μs

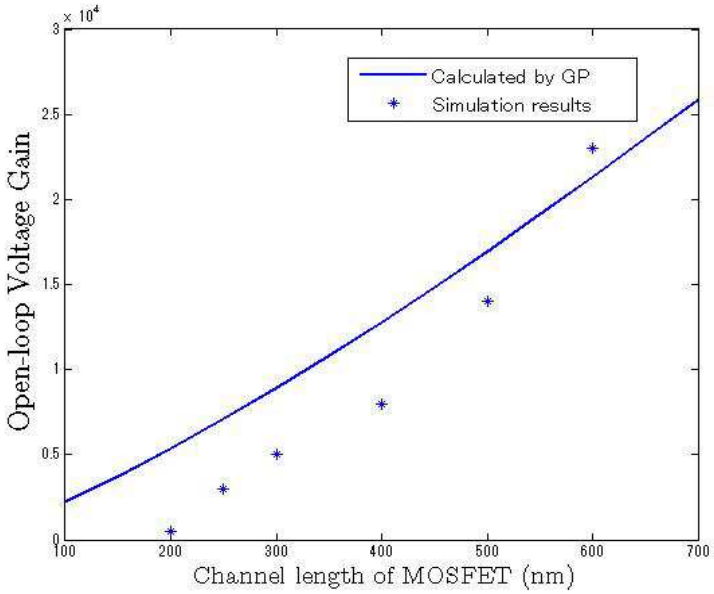

FIGURE 2.8: Trade-off curve between channel length and gain.

STIMOB=1 and WPEMOB=1 in the HSPICE model card. Thus, we execute for three kinds of simulations with the optimal results of the two groups. These simulations are named as Simulation IA, IB and II respectively, as shown in Table 2.7 and Table 2.8.

Simulation IA shows the result simulated with the optimal values of Group I. During this simulation, the options of the STI stress and the WPE inactivated. In Simulation IB, the results are also generated by simulating with the optimal values of Group I, but the options of the STI stress and WPE are activated with the NF, SA, SB and SC are set as the default values. In Simulation II, the results are simulated with the optimal

value of Group II with the options of the STI stress and WPE are activated, and NF, SA, SB and SC are set to be the results shown in Table 2.6.

As seen in Simulation IA, the results are satisfied with the specifications, but the GP optimization and the simulation both did not take the LDE parameters into account. However, in Simulation IB, several specifications such as the open-loop gain, the GB and the phase margin are not satisfied with the STI stress and WPE activated in the simulation. This means that default values used in Simulation IB are not proper if we do not consider the LDE constraints.

Next, as shown in Simulation II, when we consider the LDE parameters into GP posynomials, the specifications are re-satisfied. Comparing Simulation IB and Simulation II, we can find that the proposed LDE constraints can be used in GP forms to optimize the circuit.

Furthermore, the resultant area if we consider LDE constraints (Group II) is larger than the one we consider not (Group I). That is, we cannot estimate the accurate area by the circuit synthesis if we do not consider the STI stress and the WPE. It is known that the area estimation plays an important role in a hierarchical design strategy. These proved that the STI stress and WPE should be considered in the circuit synthesis. Therefore, we should notice that the resultant parameters of LDEs must be used for the layout constraints to satisfy the specifications.

Note that there are errors between GP program and simulations results. However, errors are inevitable because of the following reasons:

- The errors exist between the original data and fitting curves or surfaces which have to be posynomial form.
- Some constraints which are not posynomials originally need to be approximated to posynomials, for example, Eq. (2.45). Thus, error exists because of the approximation.
- Most of the constraints, especially circuit performance constraints, are created based on BSIM level 1 model which is easy for hand calculation. However, the simulation in this work is performed based on BSIM level 46 model. Therefore, errors exist because of the different BSIM systems.

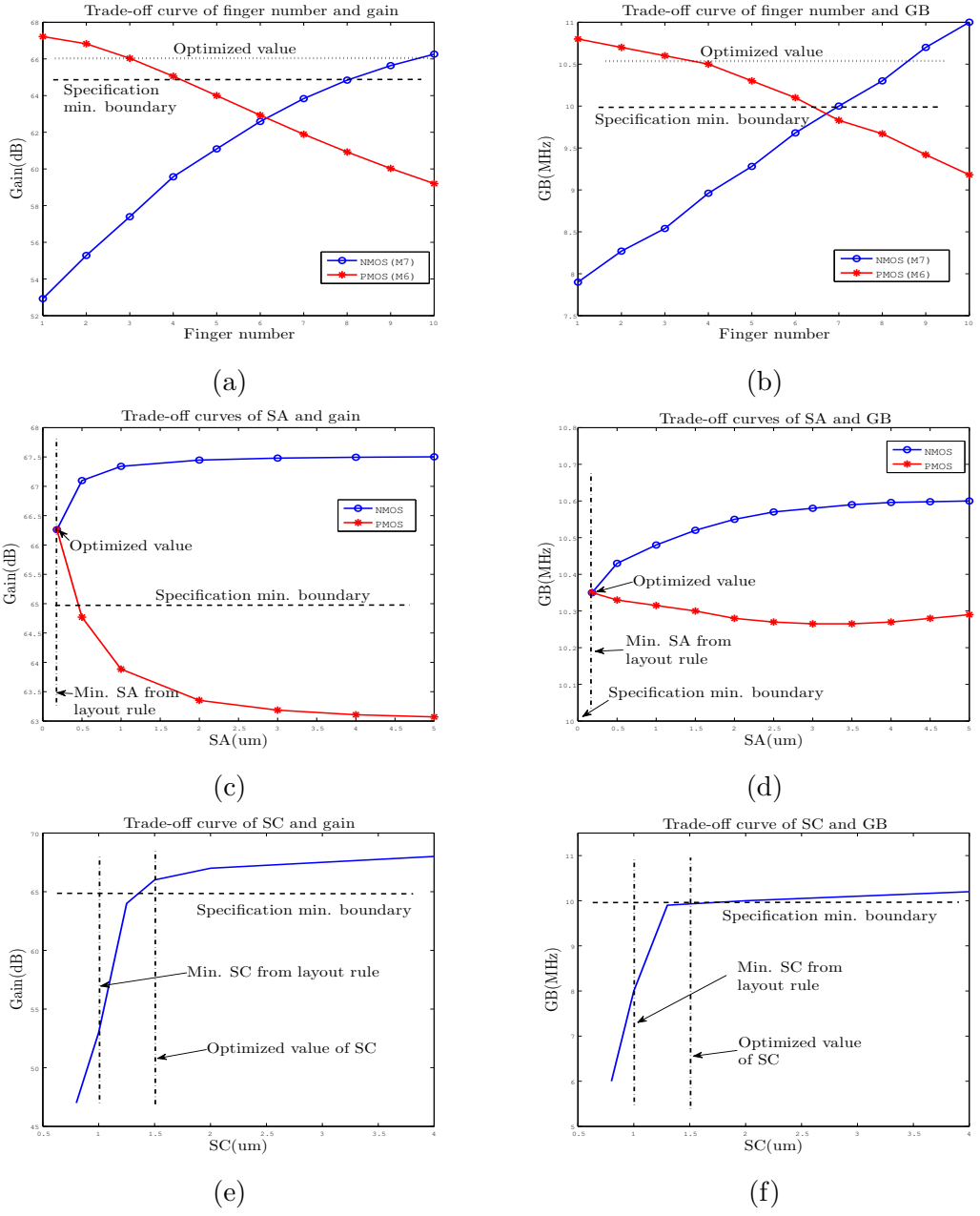

FIGURE 2.9: Trade-off analysis for LDE.

- Some simulation results can be directly generated from the HSPICE output files. However, some results need to be recalculated from the output waveform such as slew rate. As a result, there may be some errors.

Although errors exist, the results are satisfied with the specification. If more accurate results are required, fine tuning for some constraints with slightly tighting or relaxing their boundary is feasible.

To analyse the dependency of the LDE parameter on the circuit performance in details, we depicted the trade-off curves between one of NF, SA and SC and one of the gain and the GB. Note that the STI stress is related to the values of NF and SA, while the WPE is related to that of SC. The resultant curves are shown in Fig. 2.9.

As M6 and M7 are the larger transistors, we varied each finger number and fixed other parameters as constant. We found that as NF of NMOS increasing, the gain and GB increased. But for PMOS, as NF increasing, the gain and GB decreased. (See Fig. 2.9 (a) and (b).)

Next, as seen in Fig. 2.9 (c) and (d), as SA of NMOS increasing, the gain and GB increased. However, for PMOS, gain and GB decreased while SA increasing.

Besides, as shown in Fig. 2.9(e) and (f), the SC influences both of the gain and the GB. When SC is shorter than $1.5\mu m$, the gain and the GB increase dramatically. In the region of longer SC than $1.5\mu m$, the gain and the GB are not influenced.

Hence, in this design case, we need to consider both STI stress and WPE if we want get the minimum size with design specification satisfied. Note that for different process, the analysis results maybe different.

2.6 Summary

We presented LDE-aware analog circuit synthesis based on the GP formulation, where the circuit performances were formulated in terms of posynomials and monomials taking CLM parameter λ into account. As well, according to the BSIM model, we introduced the curve and surface fitting approaches to formulate the STI stress effect and the WPE in terms of posynomials. Applying our GP-base optimization to a two-stage op-amp, we clarified that the resultant circuit includes transistor sizes and LDE parameters generated by GP satisfied all the specifications under the verification of simulation based on the BSIM model with LDE options. Furthermore, we concluded that this GP-base circuit synthesis was available for the constraint generation of LDEs. The channel length of all transistors were assumed to be equal and constant in this work, so we attempt to formulate more accurate posynomial models for the STI stress and the WPE as well as

take channel length as variable with different values , and to provide a wide variety in design cases for analog circuits in future works.

Chapter 3

Digital Integrated Circuit Design

In this chapter we mainly discuss the design methodology of digital circuit design. We start from the very first Verilog/VHDL code design to the last P&R design. We also proposed some ideas to create the standard cells. By using new stardard cell of Programmable Dealy Element (PDE), we developed a DLL circuit. The methodology of it is proved efficiently and shows good results.

3.1 Introduction

High-speed synchronous interface circuits require that the controlling clock signals be accurately aligned. A dynamic de-skew circuit can be used to ensure good clock alignment across variations in process, voltage, and temperature variations (PVT). Variable delay element (DE) are often used in these circuits to manipulate the rising or falling edges of the clock or other pulses in integrated circuits (ICs). DE have many applications in VLSI circuits. The are used in digital delay-locked loops (DLLs) [27] and digital phase-locked loops (PLLs) [28], digitally controlled oscillators (DCOs) [29], and microprocessors and memory circuits[30].

The DE has many application in DLLs. The DLL is such a circuit, using a first-order closed-loop architecture that dynamically aligns its output clock signal with a reference clock signal. Two basic types of DLL architectures are currently used: analog [31] and digital [27]. The analog DLL uses a continuously variable delay line to remove the skew between the output clock and the reference clock. A digital delay line uses digital

elements, making the design more simple and portable, but with quantized steps in the delay time. The DLL has many similarities to a PLL [32]. One major difference is that rather than a Voltage-Controlled Oscillator (VCO), a delay line is used. A delay line usually comprises a chain of DE. Currently the delay elements can be categorized into three families [33]: transmission gate based, cascaded inverter based [28][30][34] and voltage controlled based [35][36].

DLL can be classified into DLL (ADLL) and digital DLL (DDLL). Compared to ADLL, all-digital (ADDLL) has the advantages of short lock time, easy system integration, insensitivity to power supply noise and process variation. To facilitate the DDLL for various clock generation circuits or phase-alignment circuits, the operating frequency range should be as large as possible to meet different production specifications, and the lock time should be the smaller the better. The highest operating frequency is limited by the bandwidth of a single delay unit while lowest operating frequency is restricted by the length of delay line. Thus, lock range is one of the most important issues of DLL.

A transmission gate based DE, as shown in Fig. 3.1(a), is a bidirectional switch consisting of a parallel connection of an NMOS and a PMOS that are controlled by completely signals. One of the disadvantages of using a transmission gate as DE is that the signal integrity of its output waveform is poor [33]. A cascaded inverter based DE, as shown in Fig. 3.1(b), is a pair of cascaded inverters. The propagation delay depends upon the time taken to (dis)charge the load capacitors. And a voltage-controlled delay element as shown in Fig. 3.1(c), consists of a cascaded inverter pair with an additional series-connected NMOS(PMOS) transistor in the pull-down(pull-up) of each inverter controlled by a global control voltage, varying which changes the delay of this element. The voltage controlled can be analog [36] or digital controlled [30], as shown in Fig. 3.2 and Fig. 3.3, respectively.

In this chapter, we proposed a DLL with multi-level L-decomposed PDE line. The transistor decomposition can be employed for direct fine grained adjustment of the channel length/width of the selected transistors. This fine grained adjustment can be used in the post-silicon tuning for the high accuracy analog applications [37]. Thus, we choose to use the L-decomposed transistors to realize the PDE. The PDE shows good linearity and high lock range with low power consumption. The rest of the DLL circuit are all digital parts which can be easily migrated to difference processes.

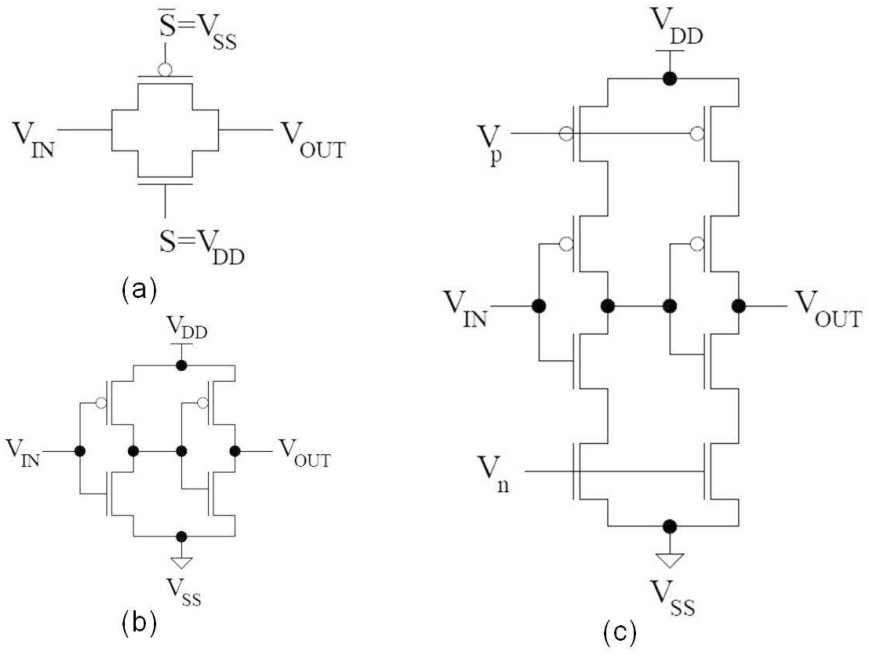

FIGURE 3.1: (a)Transmission gate based DE, (b)Cascade inverter based DE, (c)Voltage controlled based DE.

If the output of the delay were fed back to the input (forming an oscillator), the DLL could become a PLL. The other significant difference is that the DLL has a first-order response, where the PLL has a second-order response. This means that system stability is usually not an issue for DLL designs, making them easier to realize in silicon. In addition, because there is no VCO in the DLL, and therefore, no accumulation of phase error due to supply noise, the jitter performance of a DLL can be better than that of a PLL. In other words, the jitter transfer function of a DLL is equal to zero, because the reference clock both generates and synchronizes the output clock.

3.2 DLL Circuit Description

The architecture of the proposed DLL is shown in Fig. 3.4. It is composed of a phase detector(PD), a up/down counter, a decoder, a programming delay line and a PDE control circuit. The phase detector which generates UP and DN signals detects the difference between reference clock signal and feedback clock signal. UP means that the feedback signal leads reference signal while DN means feedback signal falls behind reference signal. The UP and DN signals control the direction of the counter. The PDE control circuit transform the output signals of counter into control signal of the PDE

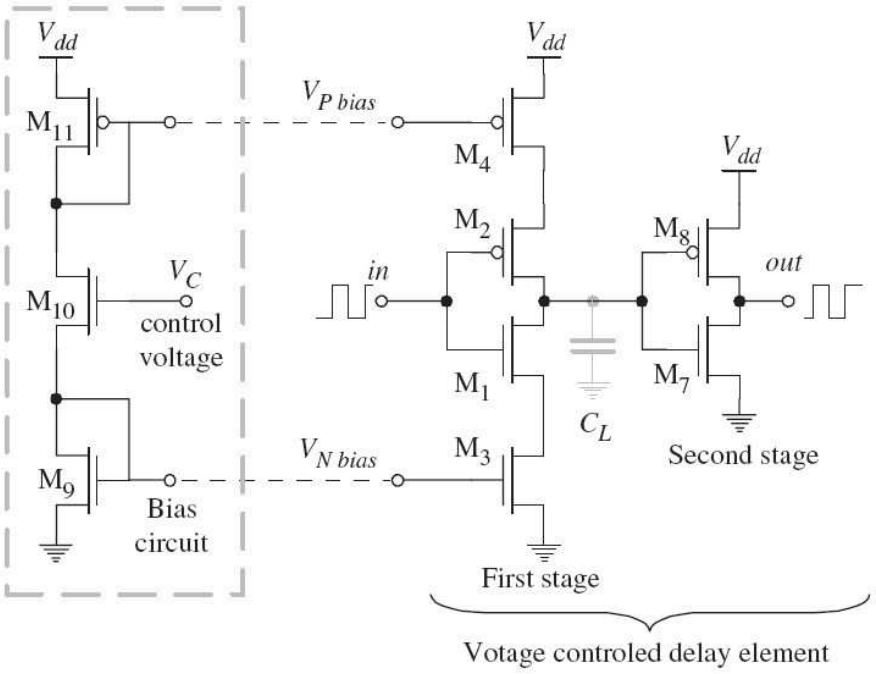

FIGURE 3.2: Analog voltage controlled DE.

line. The PDE line consists of multi-level L-decomposed PDEs. According to the output signals of PD, the PDE control circuit decides which level of the multi-level PDE is used.

3.2.1 Multi-level L-decomposed PDE

The delay line usually contain several delay units because of the narrow bandwidth of single delay unit. The proposed single level L-decomposed PDE, as shown in Fig. 3.5, is a mainly consists of a series-connected transistor with each gate connected by a switch. Each time only a switch can be activated. As the input code varies, the channel length varies. Thus the effective (dis)charging resistance and parasitic capacitances varies. The multi-level L-decomposed PDE unit shown in Fig. 3.6 is composed of M single level PDEs with different channel length. Note that the decomposed channel lengths increase in the multi-level PDE unit gradually.

To compute the propagation delay of the inverter is to integrate the capacitor (dis)charge current. This results in the expression of Eq. 3.1.

$$t_p = \int_{v_1}^{v_2} \frac{C_L(v)}{i(v)} dv \tag{3.1}$$

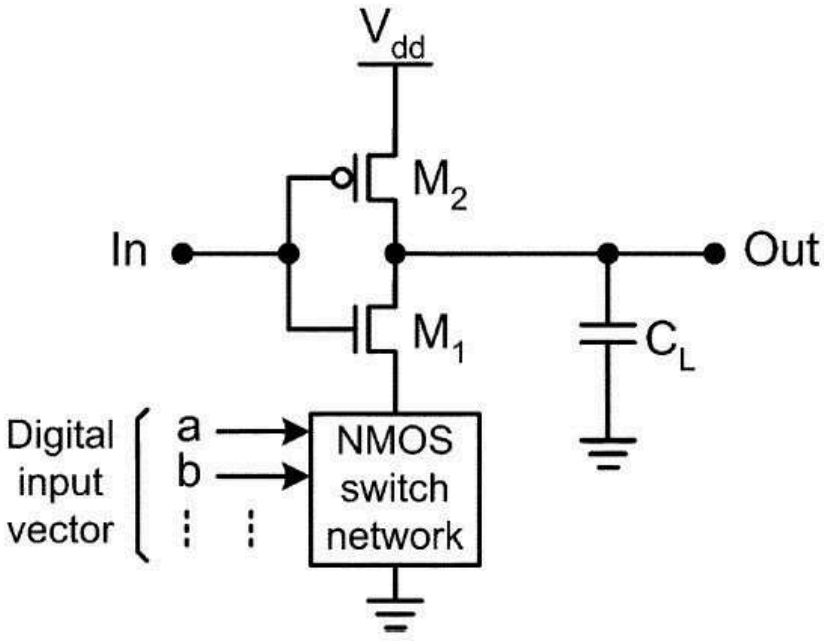

FIGURE 3.3: Digital voltage controlled delay elements.

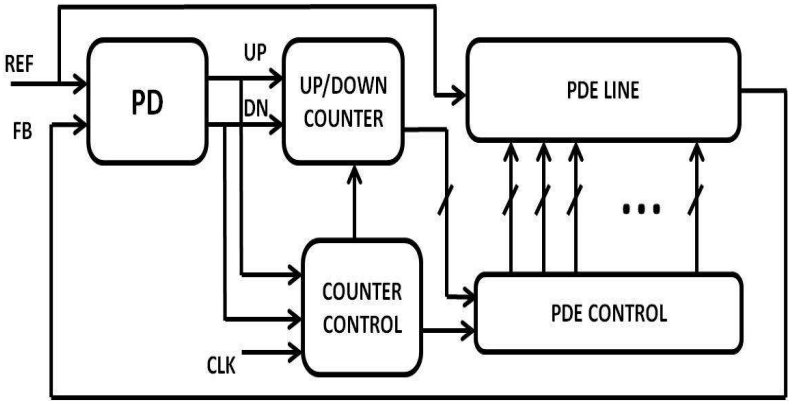

FIGURE 3.4: Proposed DLL architecture.

with i the (dis)charging current, v the voltage over the capacitor, and v_1 and v_2 the initial and final voltage. An exact computation of this equation is intractable, as both $C_L(v)$ and $i_(v)$ are nonlinear functions of v. Therefore we derive a reasonable approximation of the propagation delay adequate for manual analysis. The voltage-dependencies of the on-resistance and the load capacitor are addressed by replacing both by a constant linear element with a value averaged over the interval of interest. The preceding section derived precisely this value for the load capacitance. An expression for the average

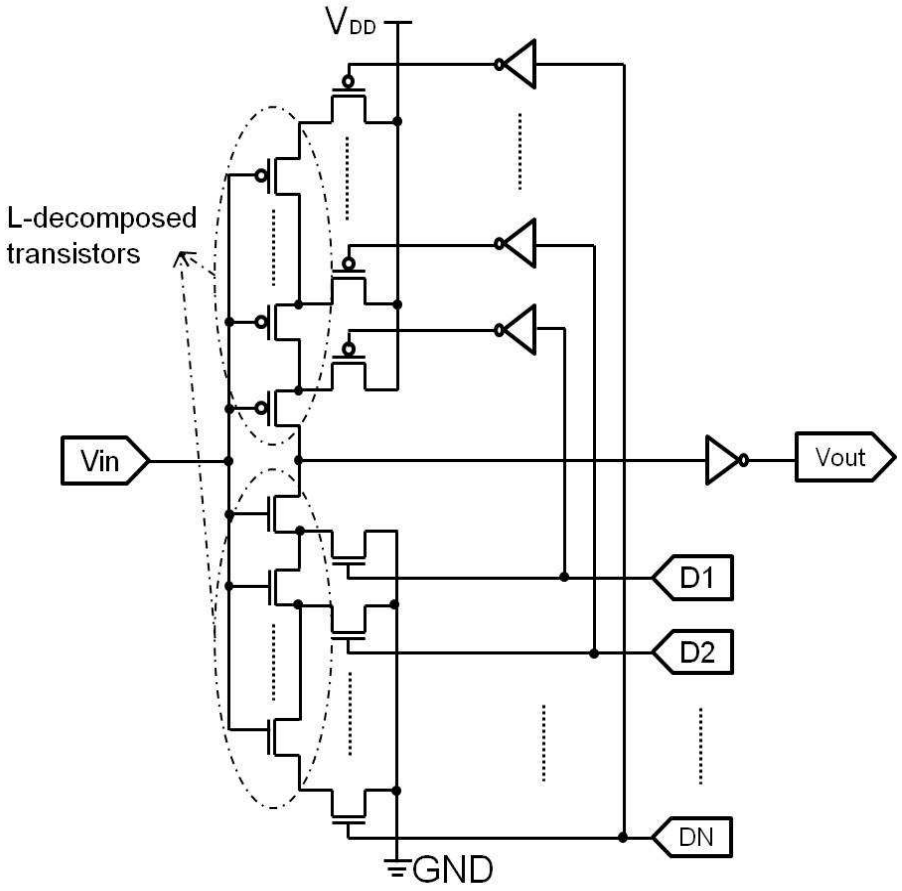

FIGURE 3.5: The proposed single level L-decomposed programming delay element.

on-resistance of the MOS transistor is

$$R_{eq} = \frac{1}{V_{DD}/2}\int_{V_{DD}/2}^{V_{DD}} \frac{V}{I_{DSAT}(1+\lambda V)}dV \approx \frac{3}{4}\frac{V_{DD}}{I_{DSAT}}\left(1-\frac{7}{9}\lambda V_{DD}\right) \tag{3.2}$$

where,

$$I_{DSAT} = k'\frac{W}{L}\left((V_{DD}-V_T)V_{DSAT} - \frac{V_{DSAT}^2}{2}\right) \tag{3.3}$$

Deriving the propagation delay of the resulting circuit is now straightforward, and is nothing more than the analysis of a first-order linear RC-network. The propagation delay of such a network for a voltage step at the input is proportional to the time-constant of the network, formed by pull-down resistor and load capacitance. Hence,

$$t_{pHL} = ln(2)R_{eqn}C_L = 0.69R_{eqn}C_L \tag{3.4}$$

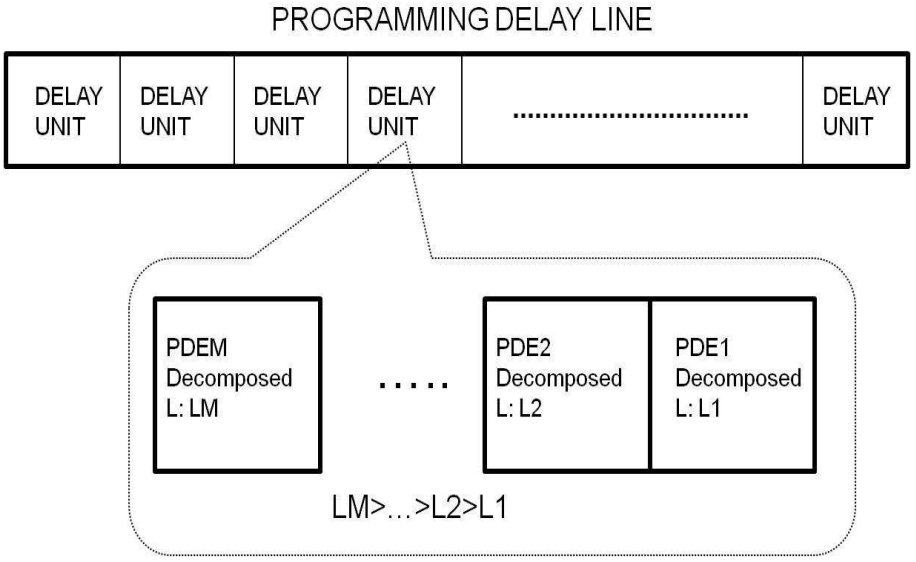

FIGURE 3.6: The proposed multi-level L-decomposed programming delay unit. The relationship among the decomposed channel length is LM>...>L2>L1.

Similarly, we can obtain the propagation delay for the low-to-high transition,

$$t_{pLH} = ln(2)R_{eqp}C_L = 0.69R_{eqp}C_L \tag{3.5}$$

The overall propagation delay of the inverter is defined as the average of the two values, or

$$t_p = \frac{t_{pHL} + t_{pLH}}{2} = 0.69C_L\left(\frac{R_{eqn} + R_{eqp}}{2}\right) \tag{3.6}$$

Contrast Eq. 3.2, Eq. 3.3 and Eq. 3.6, we can found the propagation delay of an inverter is inversely proportional to W/L, that is

$$t_p \propto \frac{1}{W/L} \tag{3.7}$$

For the PDE showed in Fig. 3.5, the length can be varied by input code which cause the propagation delay of PDE can be monotonically varied. As the simulation result showed in Fig. 3.7 under $65nm$ process, the delay is monotonically varied. Although the delay interval is slightly changed under different temperature, the linearity is unchanged. Fig. 3.8 shows the delay for short channel L-decomposed inverters. Compare the delay variation in Fig. 3.7 and Fig. 3.8, we can find that the delay resolution is different for different length. The shorter the channel length, the small the delay resolution.

Utilizing this characteristic we may create cascade PDEs to perform both coarse and fine adjusting of propagation delay. Using longer channel length in coarse tuning PDE and shorter channel length in fine tuning PDE, the delay resolution can achieve as low as $10ps$.

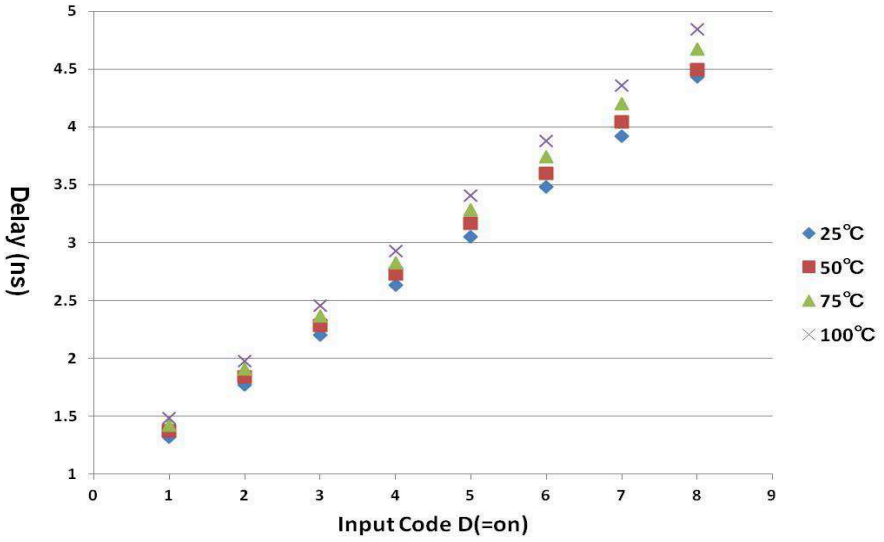

FIGURE 3.7: Temperature variations on the delay of the L-decomposed PDE with L=0.2μ. The width set as 0.12μ and 0.24mu for NMOS and PMOS respectively.

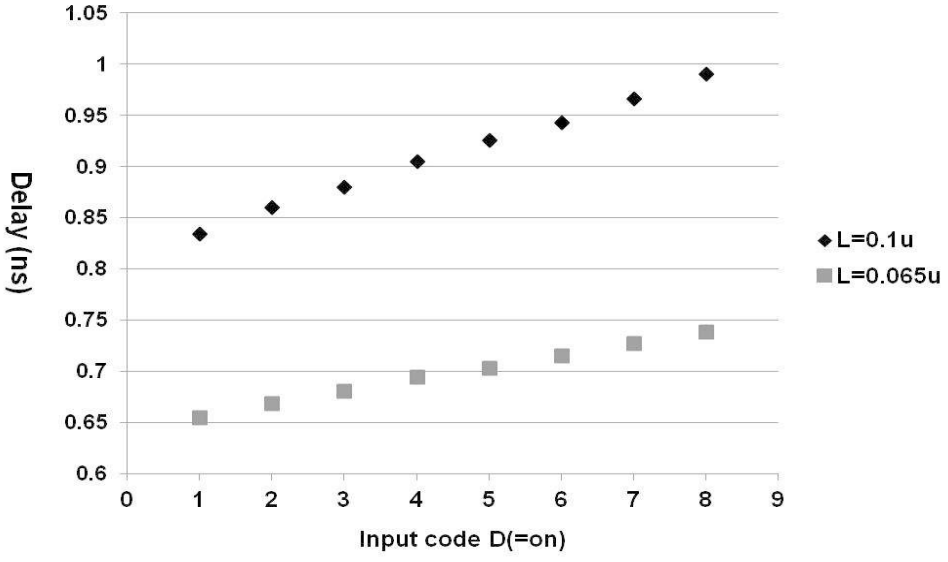

FIGURE 3.8: Variations on the delay of L-decomposed PDE for short channel length. The width set as 0.12μ and 0.24mu for NMOS and PMOS respectively.

3.2.2 UP/DOWN Counter

Since we need to decide the feedback wave is before or after the original wave, the UP/DOWN counter is necessary. However if we use the simple Verilog behavior level code, error might happen when we use Design Compiler matching to the standard cell library even if the waveform simulate by the behavior level is correct. Thus, we need to use other method to realize the UP/DOWN counter. We directly use the date flow formate to realize the UP/DN counter. For each bit of the counter circuit(called UP_DN_COUNT_CELL) is shown in Fig. 3.9. Q is the 1 bit output and TCU/TCD is the up and down wave for next bit.

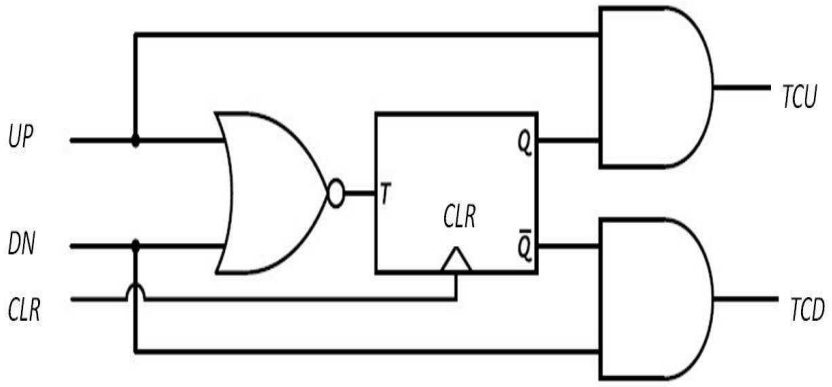

FIGURE 3.9: UP/DOWN counter cell.

3.2.3 Peripheral Control Circuit

Since the delay has been changed as multi-level delay cell. How to control these multi-level delay-cell becomes crucial. Basically, a SET signal comes out of COUNT_COUNTRL decide with level of PDE is selected as show in Fig. 3.10. The DEC signal is the output of decoder which decode the counter by specific mapping.

3.2.4 Other Parts of DLL Circuit

The delay resolution depends on the minimum composed channel length. If all the channel length set as minimum length, the searching time will increase. Thus, we use multi-level PDE to decrease the searching time. The counter control part shown in

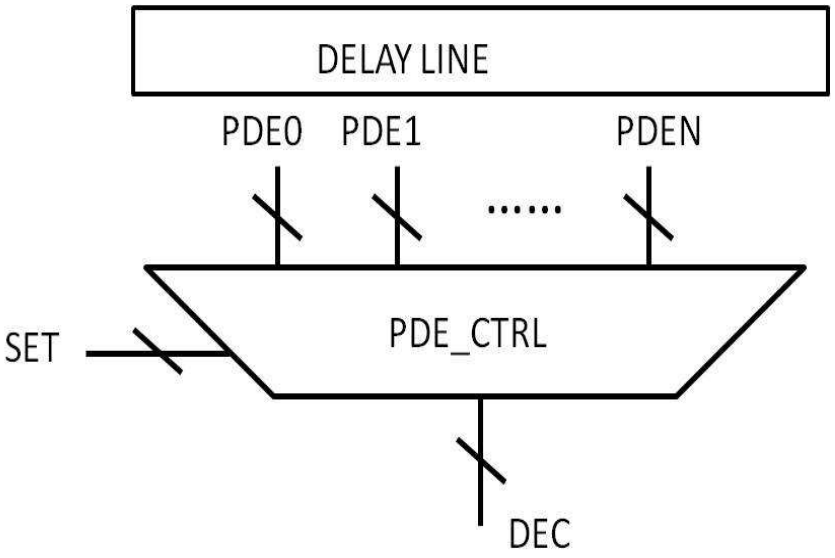

FIGURE 3.10: PDE control circuits.

Fig. 3.4 decides which level of the PDE should be used. We use a high frequency clock to count the output of PD. If the output difference is larger, the PDE control circuit as show in Fig. 3.4 directs to the high level of PDE line. Contrarily, if the output difference is small, the PDE control circuit directs to the lower level of the PDE line. Using this method, the searching cycle decreases as well as the bandwidth of each delay unit increases.

3.2.5 Searching Algorithm

DLL usually has the advantages of short lock time, easy system integration, insensitivity to power supply noise and process variation. The DLL searching algorithms can be roughly divided into three categories. The first one adopts sequential searching algorithms, the phase detector is a simple D-type flip-flop (DFF) but the lock time increase exponentially as the number of control bits increase. The second one adopts the flash architecture, i.e., the time-to-digital (TDC) scheme. Large chip area and power are required to achieve shortest lock time. The last one adopts the binary search algorithms, i.e., the successive approximation register-controlled (SAR) scheme. Even larger chip area are required. In this work, we use the sequential searching algorithm. If we use single level PDE. The searching time will absolutely increase as control bits increase. By using multi-level PDE, we can first search the best delay in the highest level of PDE with longest sub-channel. Then searching in the next higher level of PDE. So on and

so forth as we reached to the lowest level. Thus, the searching time will be lowered compare to the single level PDE cell.

3.3 Automatic Design Flow of for DLL

The design flow is shown in Fig. 3.11. Since the proposed DLL is all digital circuit except PDE. The PDE also can be created as standard cell because of it regular structure. A P&R tool, frequently used in digital circuit design flows for the purpose of automatic layout generation, uses standard cell libraries which are sets of layouts of often-used logic gates and it enables process portability of digital circuits. Our DLL structure is so simple with time-base architecture, that most of blocks can be designed using standard cells. Therefore, the DLL architecture is suitable for automatic layout using P&R tools.

Since the characteristic of the DLL is easy to modify just by editing several sections in the netlist, an automatic netlist generation is available by our script. It generates a HSPICE netlists and a Verilog gate-level netlists of the intended DLL form Verilog behavior-level codes. Then the layout can be generated uing P&R commercial tool.

3.3.1 Standard Cell Database

In this we introduce how to use Milkyway or ICC to make standard cell database.

Before you use IC Compiler, logical and physical libraries must be created that accurately reflect the characteristics of the available technology and cells that will be used to fabricate the chip. The physical libraries are prepared from layout data provided by an external source.

IC Compiler is a physical implementation tool that performs placement, clock tree synthesis, routing, and optimization of a chip design. To perform these tasks, it needs to read in a design netlist and both physical and logical libraries. These libraries contain information about the cells used in the design netlist. A physical library contains information about the geometry of the cells that are placed in the design and connected with power, ground, clock, and signal routes. This library information includes the cell dimensions, border, pin locations, and mask layers, as well as technology information such as wire tracks, antenna rules, and electromigration data. A logical library contains

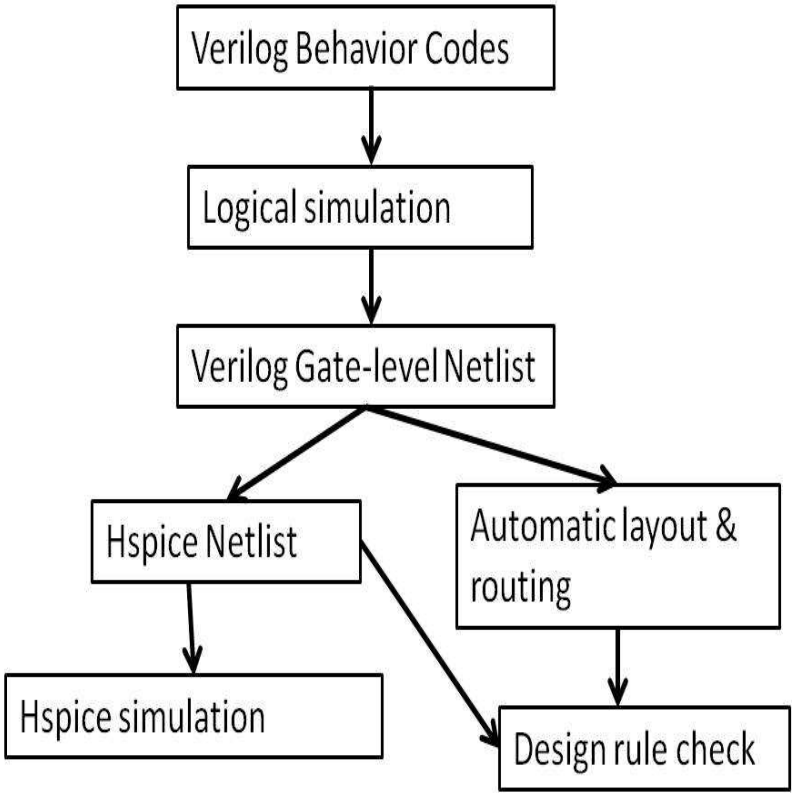

FIGURE 3.11: Automatic design flow.

functional information about these cells, including the logical function, timing characteristics, and power characteristics. IC Compiler needs this information to perform analysis and optimization of the design based on the physical characteristics and the timing, power, noise, and signal integrity requirements.

The physical libraries which is also called Milkyway Database include the following items

- CEL view: The full layout view of a physical structure such as a via, standard cell, macro, or whole chip; contains placement, routing, pin, and netlist information for the cell.
- FRAM view: An abstract representation of a cell used for placement and routing; contains only the metal blockages, allowed via areas, and pins of the cell.
- An interface logic model representation of a cell, which is created from a CEL view.
- A view of metal fill, which is used for chip finishing and has no logical function.

- A representation of the power and ground networks of a cell, created by PrimeRail or IC Compiler and used by PrimeRail for IR drop and electromigration analysis.
- A graphical view of physical design rule violations found by verification commands in IC Compiler.

The cell logic, timing, and power information is typically contained in a set of Synopsys database (.db) files produced by Library Compiler. The creator of the library determines the electrical behavior of the leaf-level cells by using a simulation-based characterization tool such as Liberty NCX in combination with a circuit simulator such as HSPICE. The characterization tool produces a set of files in Liberty (.lib) format.

The Liberty (.lib) files are ASCII-format files that fully describe the cell logic, timing, and power characteristics of the leaf-level logic cells. Library Compiler compiles the .lib files to produce .db files, which contain the same information as the .lib files, but in a compiled binary format that is more efficient for Galaxy tools to use.

Before you use IC Compiler, logical and physical libraries must be created that accurately reflect the characteristics of the available technology and contain the standard cells that will be used to fabricate the chip. IC Compiler needs these libraries to place, route, analyze, and optimize the chip design. The Synopsys tools for creating .db logical libraries are Liberty NCX, HSPICE, and Library Compiler. Liberty NCX and HSPICE characterize the standard cells and generate a set of .lib files that describe the cell behavior, and Library Compiler compiles the .lib files to produce the .db files. The use of these tools is beyond the scope of this user guide. For more information about creating .db files, see the documentation for the applicable products. The focus of this user guide is the preparation of physical library data in the Milkyway database format from the technology information provided by an outside source. This information is usually provided in one of the following standard data interchange formats:

- GDSII stream format, a well-established industry-standard data exchange format for integrated circuit layout information
- OASIS, a newer data stream format designed as an improved replacement for GDSII

- LEF/DEF, a set of standard data exchange formats for physical libraries (Library Exchange Format) and design data (Design Exchange Format)

To convert this information into the Milkyway database format, you need to read in the layout information in the provided format, add library information that is not provided in the exchange format, and write the data out to the Milkyway database. In addition, you need to prepare the standard cells for use in IC Compiler by identifying power and ground pins, flattening cells that are hierarchical, extracting FRAM views from CEL views, setting the place and route boundaries, and defining the wire tracks.

Fig. 3.12 shows the typical library preparation and usage flow for IC Compiler.

The most important task performed in the Milkyway Environment is Milkyway library preparation from the physical data provided by an outside source. Figure 3.13 summarizes the flow. We start by creating a new Milkyway library. Then we import the physical data in GDSII, OASIS, or LEF/DEF format and provide any additional information needed to create Milkyway CEL models for the cells in the library. Then we specify the power and ground ports, create the FRAM views of the cells, and specify other physical properties required in the FRAM model. The specific steps you need to perform depend on whether you are preparing macro cells or standard cells.

To prepare a Milkyway reference library from physical cell data, we import the data as a GDSII or OASIS data stream into the Milkyway Environment and translate that data into the Milkyway database format.

The GDSII stream format is an industry-standard data exchange format for integrated circuit layout information. A GDSII file contains information about layers, wire paths, boundaries, structures, arrays, and text labels in the cell or chip layout. When an outside source provides library cell information in GDSII format, we need to read the data file into the Milkyway Environment and convert the data stream into a cell library in the Milkyway database. In addition to the GDSII file, you typically provide a cell-type definition file and a layer mapping file to specify how to translate the data.

Each geometric object in a layout database has an associated layer number and data type number. In GDSII and OASIS, the layer number and data type are integers ranging from 0 to 32767. The Milkyway database supports layer numbers and data types ranging from 0 to 255. By default, numbers from 0 to 255 are left unchanged by a translation of

physical data from GDSII or OASIS to Milkyway format. Any GDSII and OASIS layer or data type numbers greater than 255 must be translated to lower numbers when they are read into the Milkyway database.

After you translate the physical design data into Milkyway library cells, you need to prepare the cells in the Milkyway Environment.

- Identify Power and Ground Ports

  After we import the cells to a Milkyway library from physical data that does not contain power and ground information, we must identify the power and ground ports in the cells. This must be done before blockage, pin, and via extraction to make the FRAM view for a cell.

- Flatten Hierarchical Cells

  To perform processing that involves geometries inside the hierarchy of a cell, we must first flatten the cell hierarchy to a level sufficient to provide the access required to perform the processing. For example, to remove unnecessary metal2 from metal1 pins at a lower level of hierarchy, we must flatten the cell to a level that makes the metal1 pins accessible. Flattening a cell removes a specified number of levels of hierarchy to make lower-level objects visible at the top level. This is often needed when importing a macro cell. When needed, flattening must be done before blockage, pin, and via extraction for creating a FRAM view. The Milkyway Environment tool might need access to geometries inside the hierarchy of the cell. In some cases, however, we might want to limit the access to lower levels. For example, we might want to limit the hierarchical access in blocks to avoid unnecessary processing and to achieve the blockage areas we want.

- Extract Blockage, Pin, and Via Information for FRAM View

  In a Milkyway reference library that is fully prepared for use in IC Compiler, each cell usually has both a CEL view and a FRAM view. The CEL view contains all of the physical mask layers of the cell, whereas the FRAM view is an abstraction of the cell containing only the information needed for placement and routing at the next higher level of the design hierarchy. When you place an instance of a cell, you use the FRAM view of the cell. The FRAM view includes the pin names and locations, the metal blockage areas where routes are not allowed, and the allowed

via areas. Other cell features are omitted in the FRAM view. The exclusion of unnecessary data can reduce the database size, increase the routing accuracy, and reduce the routing time. Fig. 3.14 shows two views of a standard cell, an inverter. The CEL view is on the left and the FRAM view is on the right. The CEL view contains all of the mask layers of the cell, including the polysilicon, diffusion, and internal contact layers. The FRAM view has only the power rails, input and output pins, and via landing areas (via regions) over the input and output pins. Note that the extraction process slices all metal polygons into rectangles. The FRAM view contains all the information necessary to place the inverter cell in a site row and route the connections to the input and output of the cell.

The buttons Extract Blockage, Extract Pin by Text, and Extract Via expand the dialog box to show the options related to the generation of blockages, pins, and vias, respectively. Use the Generate Boundary toggle buttons to specify which boundary edges are to be determined by the extraction process. By default, the existing boundary edges of the CEL view are used without change. If you generate a new boundary, the boundary depends on the locations of the extracted blockages, pins, and via landing areas. We can also extract the FRAM view from marco if we want.

- Creating Blockage Areas

  Milkyway creates blockage areas by merging any geometries on the same layer that are within merging distance of each other except for pins. By default, the merging distance is twice the minSpacing value plus the minWidth value in the technology file. This is the narrowest space that can allow a single minimum-width route to pass through.

- Set the Place-and-Route Boundary

  Synopsys tools can use the place-and-route boundary to align wire tracks and power and ground rails during cell placement. The place-and-route boundary is the bounding box of the unit tiles used by a standard cell, as shown in Fig. 3.15.

  If the cells are designed to overlap, you should make the place-and-route boundary smaller. For example, to make the power and ground pins of abutting rows overlap in a back-to-back floorplan, decrease the height of the place-and-route boundary

so the cells abut along the center of the power and ground pins, as shown in Fig. 3.16.

- Set the Multiple-Height Boundary

  We can adjust the place-and-route boundary for a double-height or triple-height cell in the Milkyway Environment. This command checks the power and ground height and width, suggests floorplan options, and checks the row spacing rule in the technology file

  In the Library Name text box, enter the name of the Milkyway library. In the Cell Name text box, enter the name of a single cell or a wildcard pattern; use the "pattern match" option with a wildcard pattern. In the Tile Name text box, enter the name of the tile used for adjusting the boundary. To adjust the place-and-route boundary as well as set the cell row orientation, select the "adjust PR boundary" option. Otherwise, you only set the cell row orientation.

- Define the Unit Tile Wire Tracks

  After we adjust the cell boundary and create the unit tiles, we need to create wire tracks in the unit tile. The unit tile cell must contain wire tracks, which assist the router in placing wires for each of the routing layers. Each layer has its own set of wire tracks, which run in the preferred direction for that layer. For example, wire tracks on metal1 run along the horizontal axis.

We follow the steps described above, and make the database of standcell. Fig. 3.17 is a stand cell layout of PDE.

3.3.2 Placement and Routing

We will transform a gate-level netlist into a placed and routed layout using Synopsys IC Compiler (ICC). ICC takes a synthesized gate-level netlist and a standard cell library as input, then produces layout as an output.

The first goal of a Place and Route (P&R) tool is to determine where each gate should be located on the physical chip (the placement portion of place and route). This process leverages heuristic algorithms to group related gates together in hopes of minimizing

routing congestion and wire delay. Typically P&R tools will focus their effort on minimizing the delay through the critical path, and to this end will resize gates, insert new buffers, and even perform local resynthesis. Additionally, P&R tools often have secondary algorithms to help reduce area for non-critical paths. After placement, ICC will attempt to route the design while minimizing wire delay. Along with routing, P&R tools often handle clock tree synthesis, power routing, and block level floorplanning.

Like Design Compiler, IC Compiler is an extremely complicated tool that requires many pieces to work correctly. Attempts at synthesis without providing the tools with properly formatted configuration scripts, constraint information, and numerous technology files for the target standard cells will only be met with more pain and sadness.

3.4 Experimental Results

To verify the linearity of the proposed multi-level PDE, we built two kinds of PDE lines using different processes. First, we built double level PDE using 65nm process, the channel is decompose into 8 for both level, and the decomposed channel length is 0.08μ and 0.25μ respectively. The delay curve is depicted as shown in Fig. 3.18. We can found that the linearity is good and the minimum solution is about 20ps. Second, we built same double (8x8) level PDE using 0.6μ process, the decomposed channel length is set as 0.6μ and 2.4 μ respectively. The delay curve is depicted as shown in Fig. 3.19. Overall the linearity is good for both process.

A layout of a single level PDE is shown in Fig. 3.20. The comparison of pre-layout and post layout simulation is shown in Fig. 3.21. We can found that the resolution of post-layout simulation is small than pre-layout one because of the layout parasitic capacitances. However, the well linearity of the delay versus input code remains.

TABLE 3.1: DLL performance comparison.

	Lock cycle	Power	resolution	Lock Range
65nm single	23	1.24mW	10ps	15∼500MHz
65nm double	7	1.37mW	10ps	5∼500MHz
0.6μm single	22	4.82mW	100ps	1.5∼50MHz
0.6μm double	8	4.94mW	100ps	0.5∼50MHz

We also compared the DLL performance of single-level and double-level PDE in both 65nm and 0.6μm. For the single-level PDE, the delay elements are decomposed into 32 sub-channels, the delay lines consists of 6 delay units. For double-level delay units, all the delay elements are decomposed into 8 sub-channels and all the deley lines consists of 3 delay units. The results are shown in Table 3.1. We can find that when we use multi-level PDE, the lock cycle decreased as well as the lock range increased. However, because of the relative control circuit the power increased. The layout of DLL is shown in Fig. 3.22.

3.5 Summary

In this chapter, we proposed a DLL with multi-level L-decomposed PDE. The multi-level PDE shows good linearity. Meanwhile, it can be easily designed with simple delay equations. The DLL with multi-level PDE lock faster and has wider lock ranges. However it costs more power than the DLL with single level PDE.

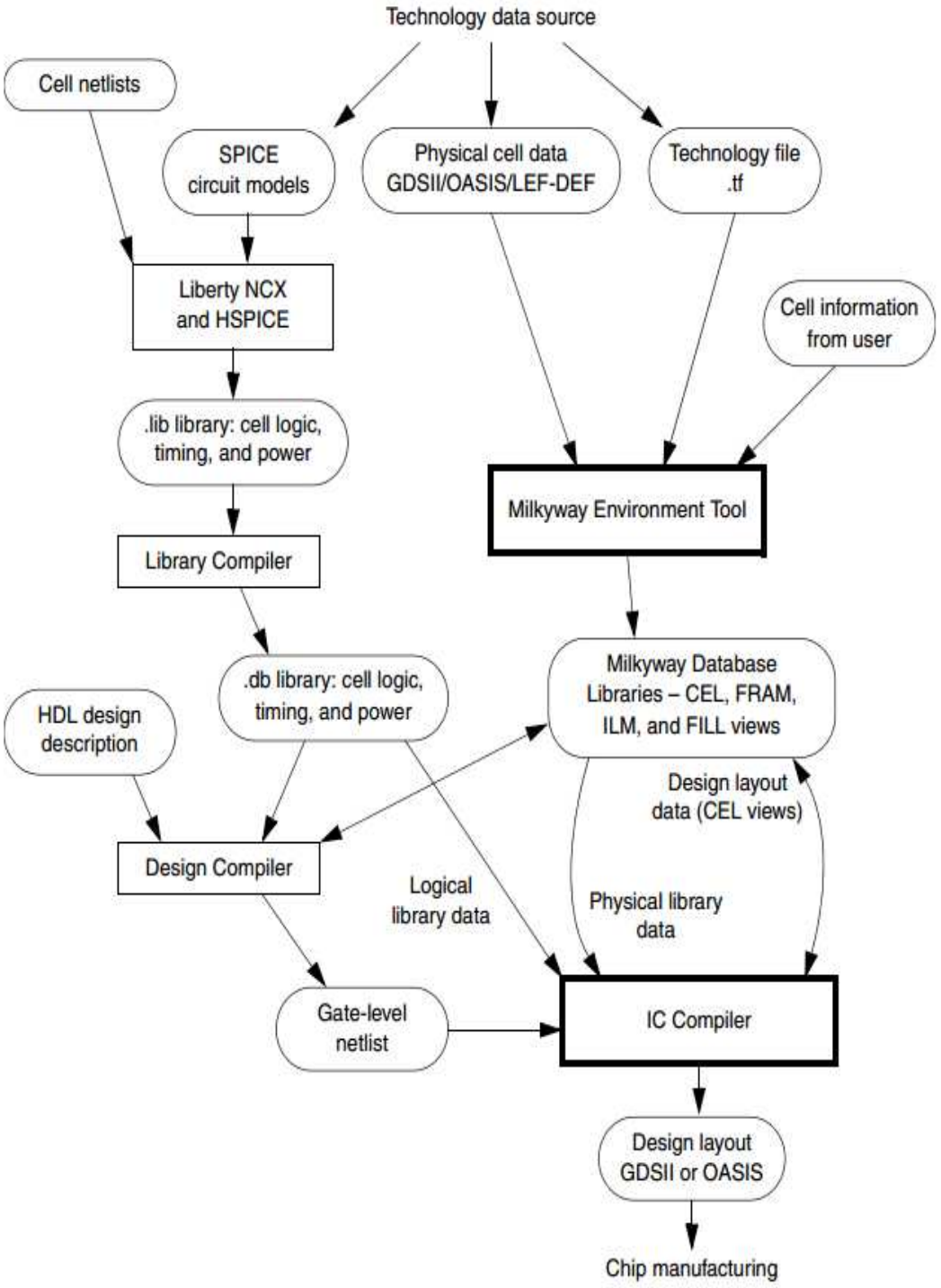

FIGURE 3.12: Typical physical and logical library preparation and usage.

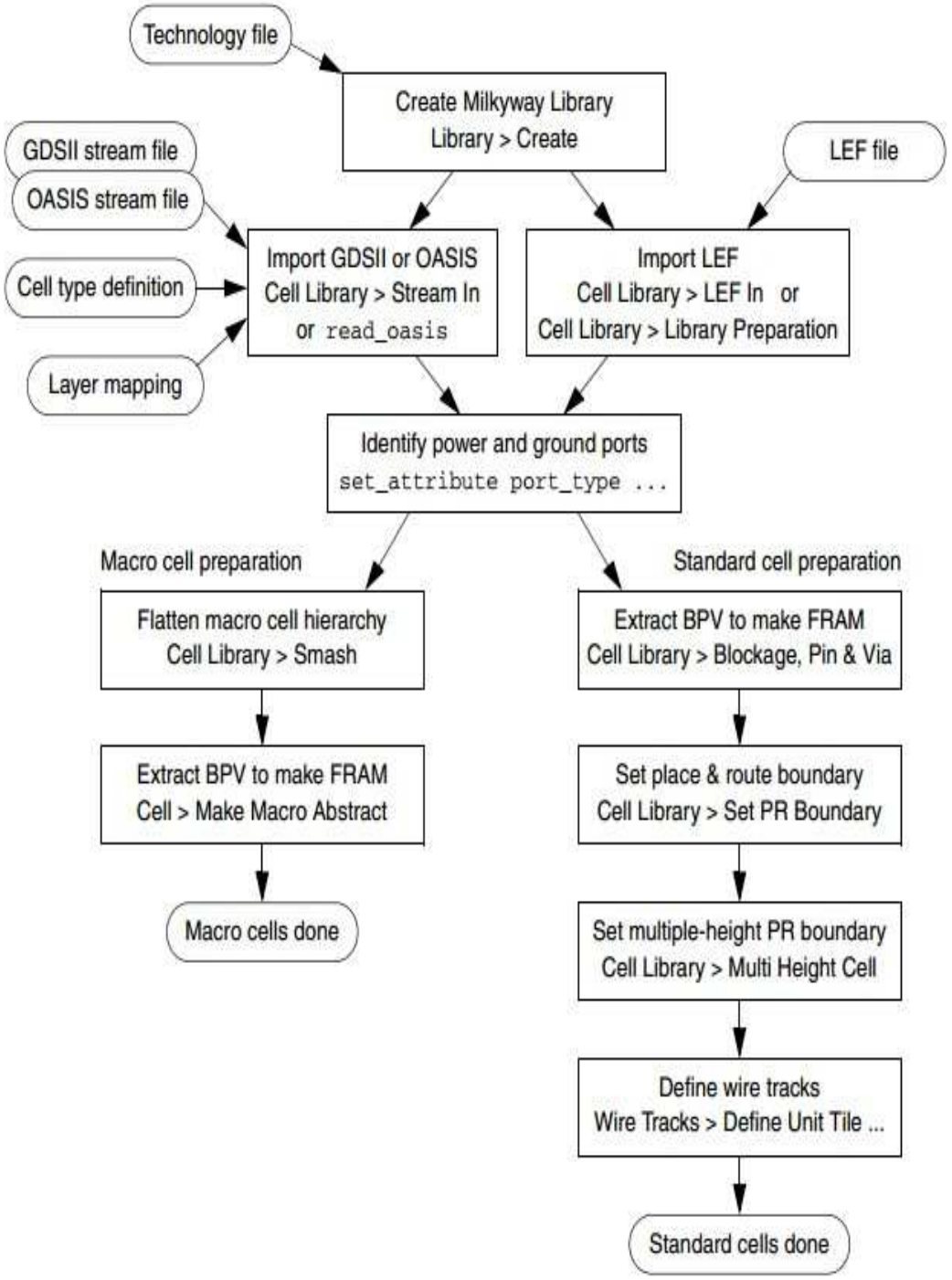

FIGURE 3.13: Library preparation flow in the Milkyway Environment.

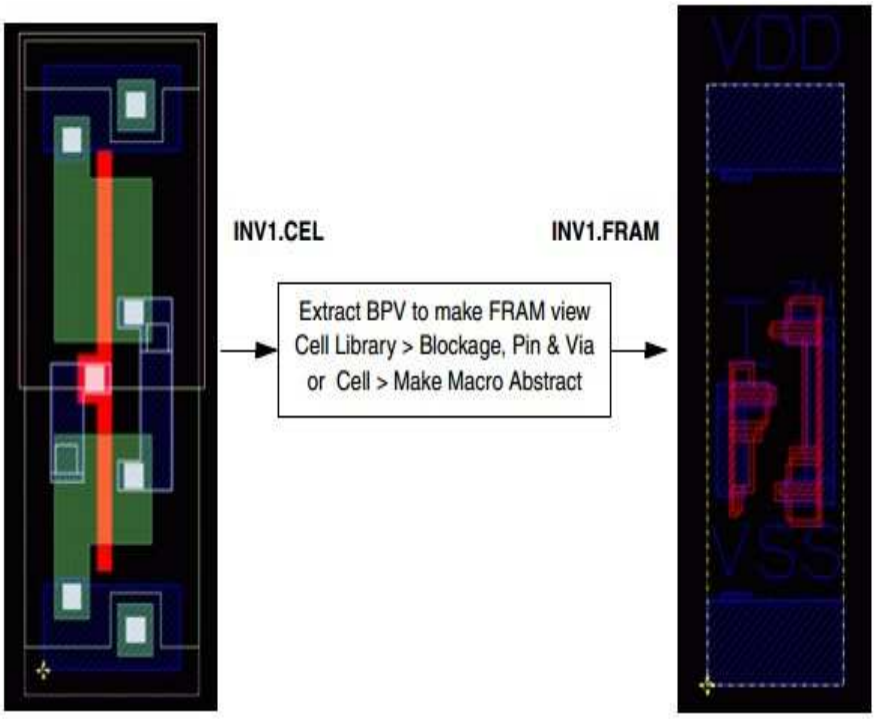

FIGURE 3.14: CEL and FRAM Views of a Standard Cell.

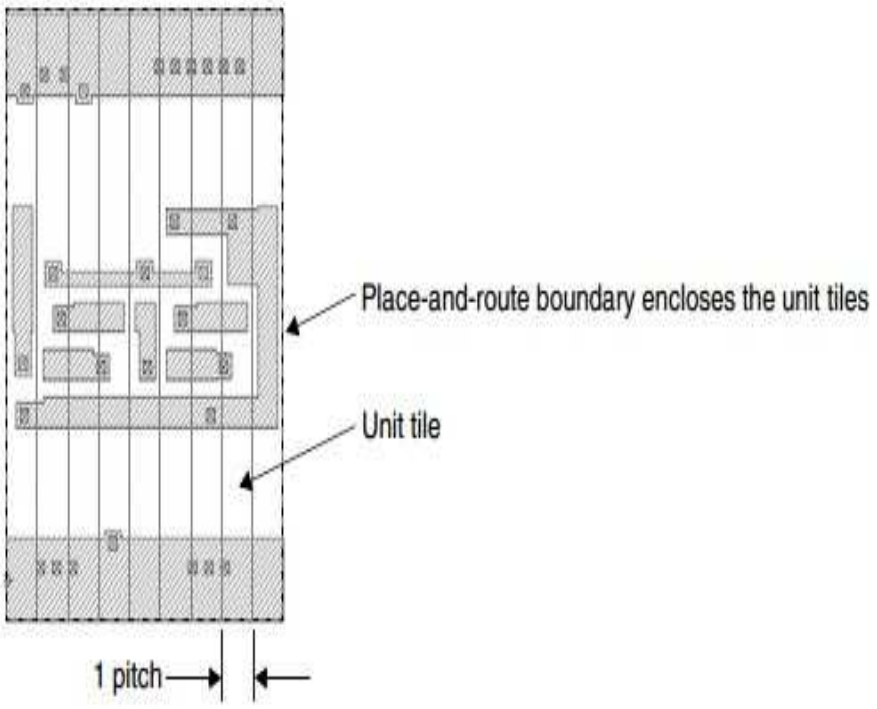

FIGURE 3.15: Place-and-Route Boundary Example.

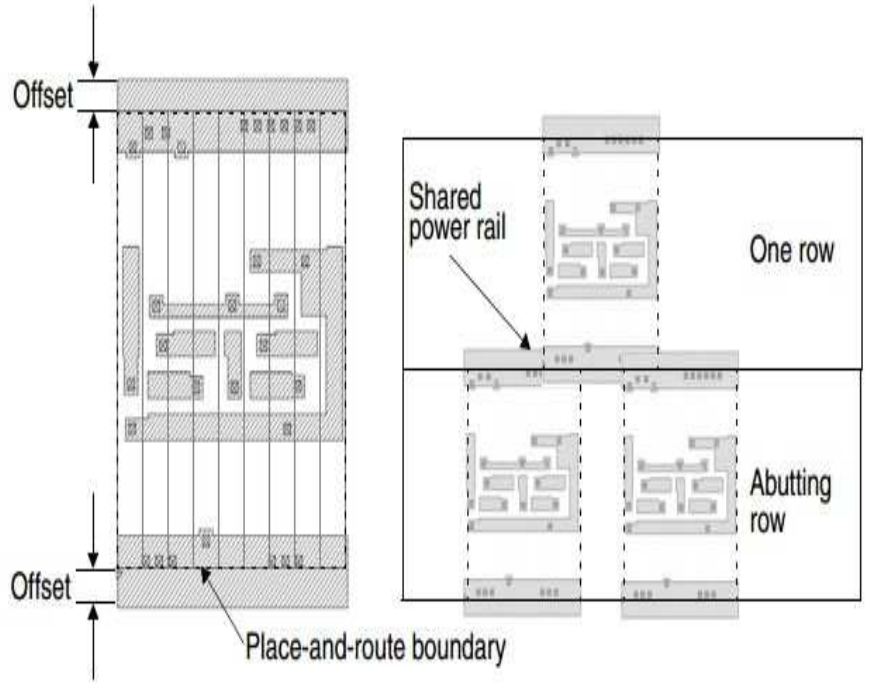

FIGURE 3.16: Overlapping Abutting Rows Example.

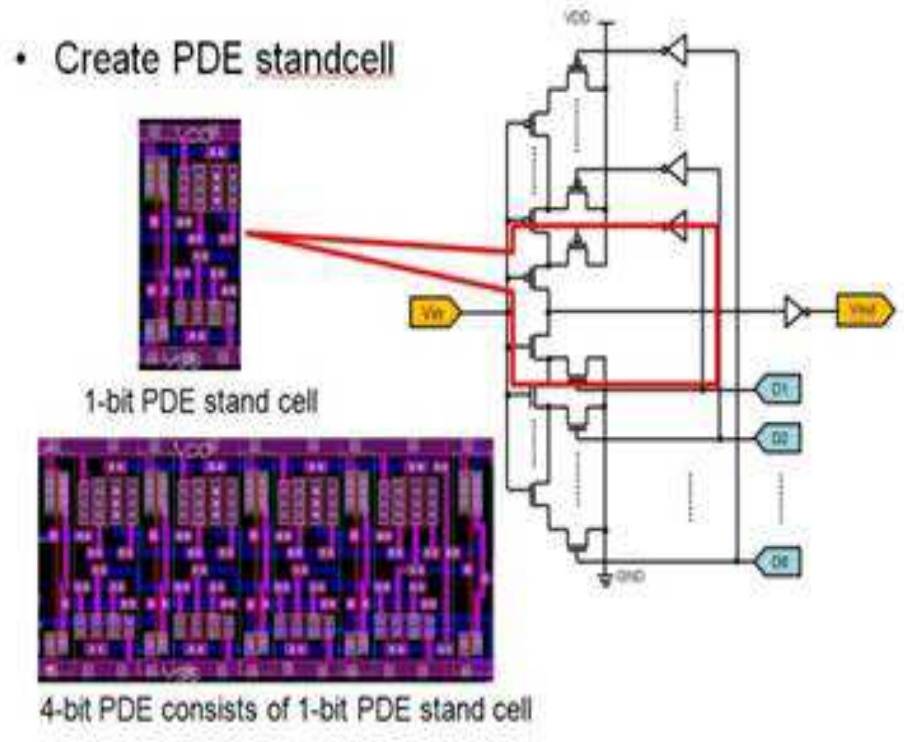

FIGURE 3.17: The standcell of PDE.

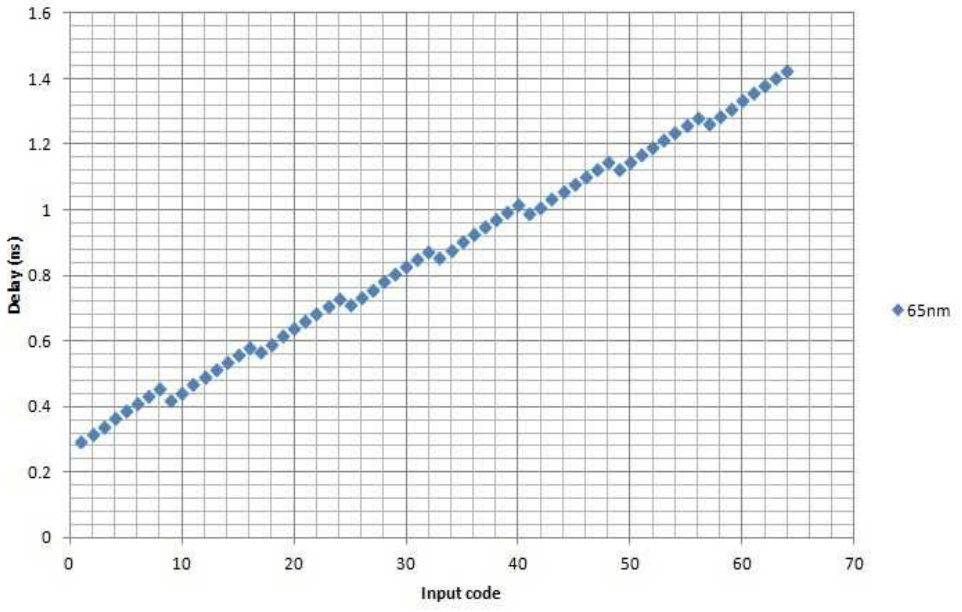

FIGURE 3.18: Variations on the delay of double level L-decomposed PDE using 65nm process.

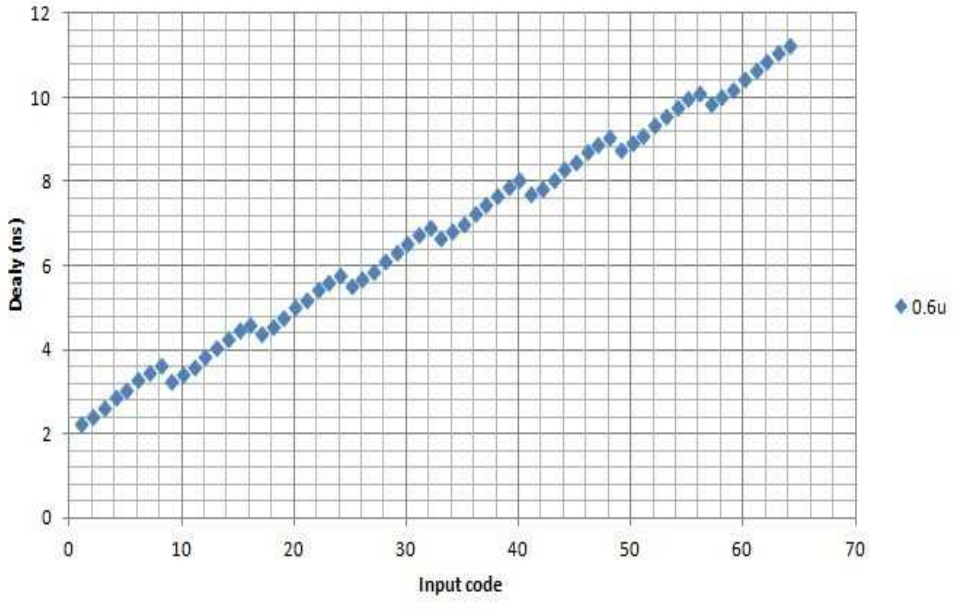

FIGURE 3.19: Variations on the delay of double level L-decomposed PDE using 0.6μm process.

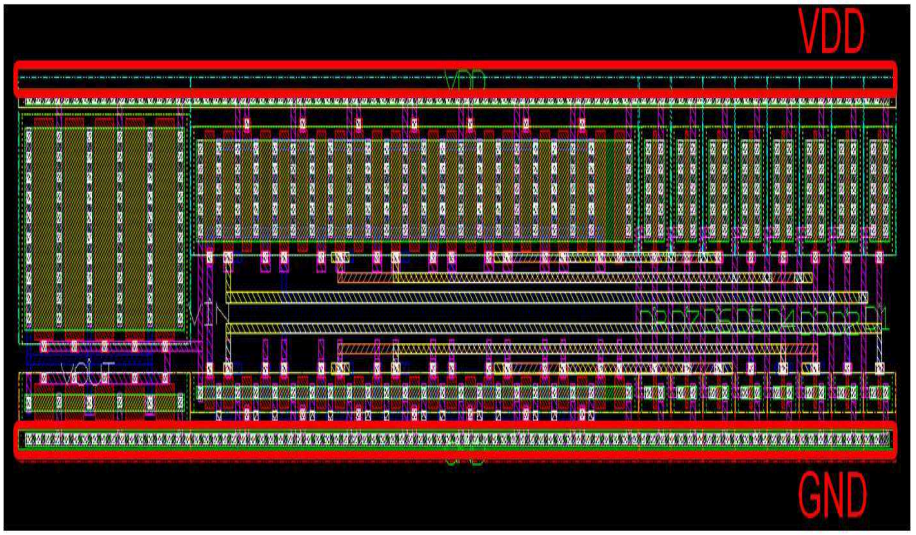

FIGURE 3.20: The layout of the proposed PDE.

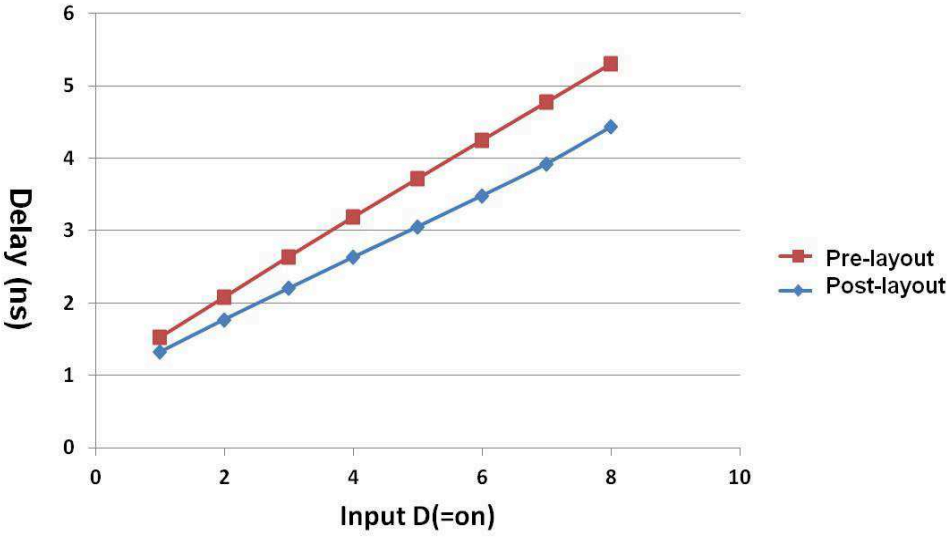

FIGURE 3.21: Pre-layout and post-layout simulation of the L-decomposed PDE with L=0.2μ.

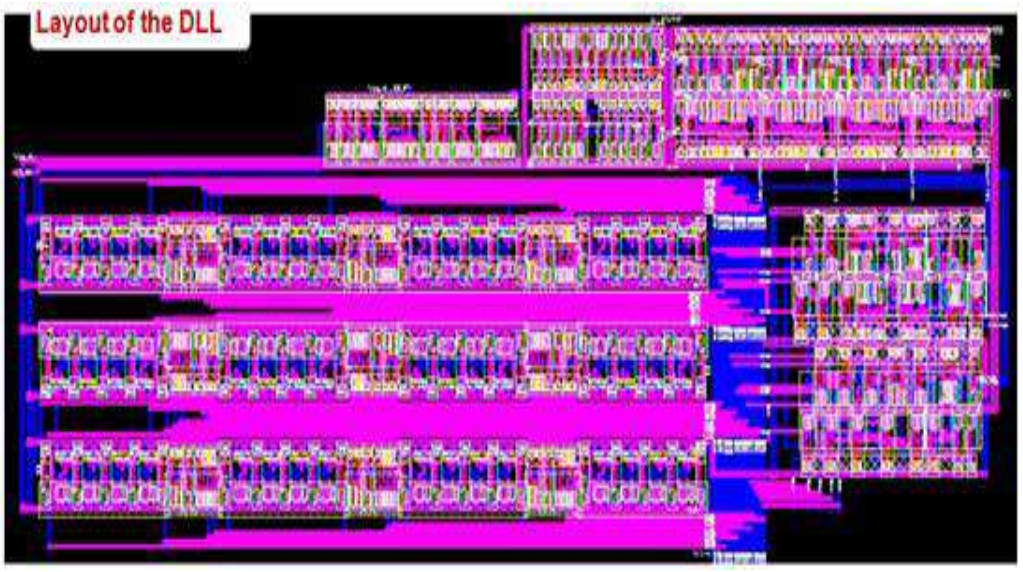

FIGURE 3.22: The layout of DLL.

Chapter 4

Memory Design

In Chapter 3, we mainly used the fixed standcell to create the DLL. In this chapter, however, we are going to use another kind of cell which is parameterized cell. The cell's size due to the layout-dependent effect can be changed. So that, we are trying to find out the trade-off between size, delay and some other performances.

4.1 Introduction

The advanced CMOS technology scaling is increasing the random and systematic variation. The SRAM is one of the most sensitive device to the variation because the cell size is very tiny to be integrated as much as possible. Meanwhile, SRAM design requires a balancing act between delay, area and power consumption. Thus, finding a good trade off among speed, power and area in memory design is very important. Usually the power consumption is proportional to active area of the circuit. In many process caches occupy 50% of the chip area [38]. Power dissipation in CMOS circuits can be categorized into two main components, dynamic and static power dissipation. Dynamic dissipation occurs due to switching transient current and charging and discharging of load capacitors. Static dissipation is due to leakage currents drawn continuously from the power supply. It is predicted that static power dissipation would contribute to 50% of the total power in the next generation process [39]. Many researchers addressed the power consumption problems by adjusting the threshold voltage V_{th}. In [40], the authors proposed a dynamic V_{th} technique by using body biasing to reduce leakage power

in SRAMs. However, the additional circuit for each SRAM cell would definitely increase the whole chip size. In [41], the authors directed that high V_{th} usually causes low leakage power consumption but causes high delay during SRAM operation. They used dual V_{th} transistor to decease the leakage power without increasing the delay. The delay during read and write operation determine the speed of SRAMs, and this is important in high speed application.

The threshold voltage can be changed by channel doping during chip manufacturing period and two or more threshold voltages will increase the fabrication cost. Besides, some layout-dependent effect such as STI stress or WPE will cause the variation of threshold voltage. Lots of paper have addressed this problem. In [16], the authors presented a model for the STI well contributing to the mobility, and proposes a diffusion filling to improve the timing performance. A piecewise linear approximation model was provided in [18] and this model was applied to various STI analysis. As for WPE, a MOSFET close to the edge of a well region must be characterized by the different threshold voltage. In [19], the authors demonstrated an impact of the WPE is crucial for analog designs. They showed that the threshold voltage increases as the transistor approaching near the well edge. In [42], both variations of STI and WPE effect are illustrated and a layout-dependent aware circuit synthesis had been performed. Therefore, these stress effect can not be neglect in nanometer design, especially in sensitive and high density memory design.

In this chapter, we proposed an SRAMs with parameterized cells and our research has the following contributions,

- We evaluated the stress effect of STI and WPE to the SRAM circuit.
- Utilizing the stress effect, we manage to construct the V_{th} variable SRAM array without increasing fabrication cost.
- We find a good trade-off among the leakage power dissipation, speed and area of SRAM circuitry.

The rest of the chapter is organized as follows; Section 4.2 describe the SRAM circuit and each block. Section 4.3 propose the variation caused by the stress effect of STI and WPE. In section 4.4, we construct the leakage power and delay model with the threshold

voltage parameter. The design case and verification are proposed in section 4.5. Section 4.6 concludes.

4.2 SRAM Circuit

As microprocessors and other electronics applications get faster and faster, the need for large quantities of data at very high speeds increases, while providing the data at such high speeds gets more difficult to accomplish. As microprocessor speeds increase from 25 MHz to 100 MHz, to 250 MHz and beyond, systems designers have become more creative in their use of cache memory, interleaving, burst mode and other high-speed methods for accessing memory. The old systems sporting just an on-chip instruction cache, a moderate amount of DRAM and a hard drive have given way to sophisticated designs using multilevel memory architectures. One of the primary building blocks of the multi-level memory architecture is the data cache.

A cache is memory used to temporarily store data. In many computer systems, whether a PC, a RISC workstation, or a mainframe, caches are used to:

- Store the instructions and data for the processor. Called the Level 1 or L1 cache, this small memory is found on the microprocessor chip and runs at processor speed.
- Optimize the amount of time required to provide data to the CPU when there is a miss on L1. This is called the Level 2 (L2) cache. It is many times larger than the L1 cache and is usually designed so that 90% of the time, the processor can find the data it needs there. In PC applications, the L2 cache runs at one half to one third of the processor speed. In workstations, the L2 cache may be required to operate at processor speed. The L2 cache resides on the board with the processor or on SIMMs (Single In-Line Memory Modules) that are located near the processor. Data stored by the processor in the L2 cache is periodically off-loaded to the DRAM (extended or Level 3 memory) or to one of the disk drives.

Increasingly, the designers of high performance systems, including PC and RISC based computers and high speed telecommunications applications, rely on synchronous SRAMs to design caches that provide data at the speeds they require.

There are many reasons to use an SRAM or a DRAM in a system design. Design tradeoffs include density, speed, volatility, cost, and features. All of these factors should be considered before you select a RAM for your system design.

- Speed. The primary advantage of an SRAM over a DRAM is its speed. The fastest DRAMs on the market still require five to ten processor clock cycles to access the first bit of data. Although features such as EDO and Fast Page Mode have improved the speed with which subsequent bits of data can be accessed, bus performance and other limitations mean the processor must wait for data coming from DRAM. Fast, synchronous SRAMs can operate at processor speeds of 250 MHz and beyond, with access and cycle times equal to the clock cycle used by the microprocessor. With a welldesigned cache using ultra-fast SRAMs, conditions in which the processor has to wait for a DRAM access become rare.

- Density. Because of the way DRAM and SRAM memory cells are designed, readily available DRAMs have significantly higher densities than the largest SRAMs. Thus, when 64 Mb DRAMs are rolling off the production lines, the largest SRAMs are expected to be only 16 Mb.

- Volatility. While SRAM memory cells require more space on the silicon chip, they have other advantages that translate directly into improved performance. Unlike DRAMs, SRAM cells do not need to be refreshed. This means they are available for reading and writing data 100time.

- Cost. If cost is the primary factor in a memory design, then DRAMs win hands down. If, on the other hand, performance is a critical factor, then a well-designed SRAM is an effective cost performance solution.

- Custom features. Most DRAMs come in only one or two flavors. This keeps the cost down, but doesn't help when you need a particular kind of addressing sequence, or some other custom feature. IBM's SRAMs are tailored, via metal and substrate, for the processor or application that will be using them. Features are connected or disconnected according to the requirements of the user. Likewise, interface levels are selected to match the processor levels. IBM provides processor specific solutions by producing a chip with a standard core design, plus metal mask options to define feature sets.

High performance SRAMs currently on the market are manufactured using one of three basic processes. A small number of offerings are made using bipolar circuits. Most are made using a BiCMOS technology in which most of the internal circuitry is CMOS while the interface circuits (including the drivers) are bipolar. A few, including all of IBM Microelectronics' high performance, synchronous SRAMs, are made entirely with CMOS circuits.

A typical SRAM block, shown in Fig. 4.1, consists of cell arrays, address decoders, column multiplexers, sense amplifiers, I/O, and a control circuitry. SRAMs usually support read and write operations. For these operations, the row decoder selects the appropriate word-line corresponding to the input address thereby activating a row in the memory array. The functionality and design of other components are briefly discussed in the following sections.

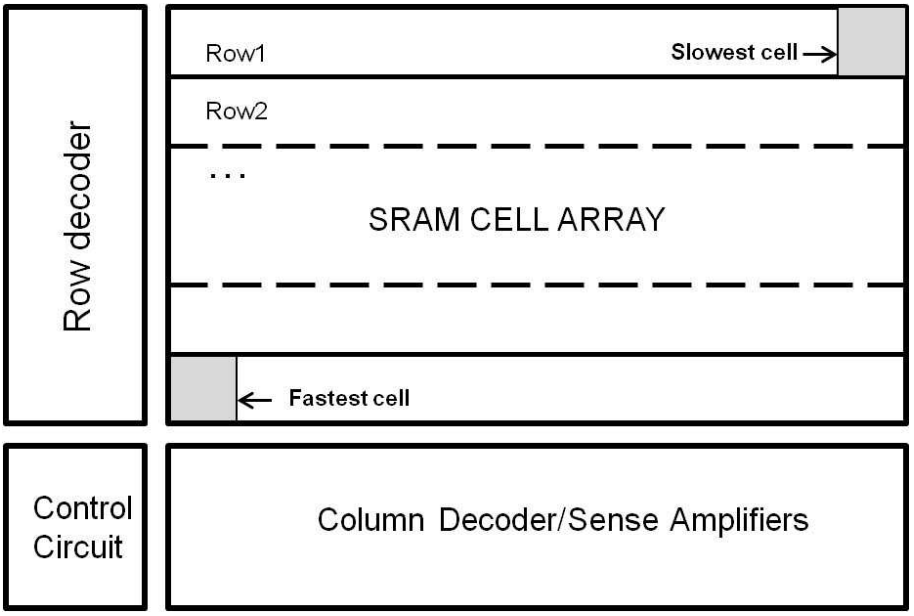

FIGURE 4.1: An SRAM block.

4.2.1 6T SRAM cell

Fig. 4.2 shows a 6-transistor(6T)SRAM cell. The bit value stored in the cell is preserved as long as the cell is connected to a supply voltage whose value is greater than data retention voltage. This feature, which is due to the presence of cross-coupled inverter inside the 6T SRAM, holds independent of the amount of leakage current. We usually

use the term static noise margin (SNM) to evaluate the capability of enduring noise of an SRAM cell. Traditionally all cell used in an SRAM block are identical which results in identical leakage characteristic for all cells. However, we considered the stress effect into the SRAM cell. The threshold voltage can be changed as long as the SRAM cell size varies.

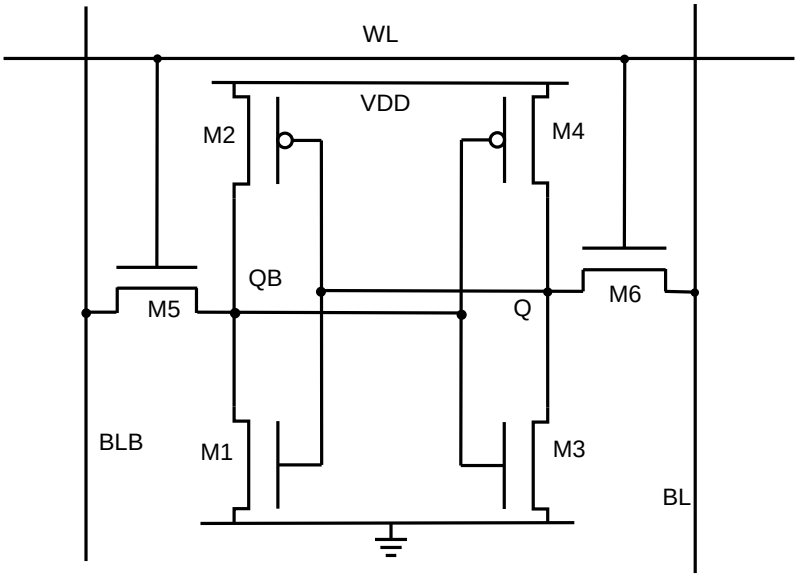

FIGURE 4.2: A 6T SRAM cell.

4.2.2 SRAM Array

Usually there is more than one cell array in an SRAM circuit. The size of the cell array depends on both performance and density requirements. Generally speaking, as technology shrinks, cell array are moving from tall to wide structures. However, since using wider arrays need more circuitry for column multiplexers and sense amplifiers, in case where a large area overhead is intolerable, the number of rows can be still high.

4.2.3 Row Decoder

Although the logical function of an row decoder is very simple, in practice designing it is complicated because the row decoder needs to interface with the core array cells and pitch matching with the core array can be difficult. The decoder usually consists of a single NAND gate and buffers for driving the word-line capacitance, is pitch-matched and placed next to each row.

4.2.4 Column Decoder and Sense Amplifiers

Column multiplexing is inevitable in most SRAM designs because it reduces the number of rows in the cell array and as a result increase the speed. Since the bit or bit-bar line is discharged about $200mV$ during a read operation, a sense amplifier is used to sense a small voltage difference and generate a digital value.

4.2.5 Monte Carlo Simulation

The Monte Carlo method was invented by scientists working on the atomic bomb in the 1940s, who named it for the city in Monaco famed for its casinos and games of chance. Its core idea is to use random samples of parameters or inputs to explore the behavior of a complex system or process. The scientists faced physics problems, such as models of neutron diffusion, that were too complex for an analytical solution, so they had to be evaluated numerically. They had access to one of the earliest computers but their models involved so many dimensions that exhaustive numerical evaluation was prohibitively slow. Monte Carlo simulation proved to be surprisingly effective at finding solutions to these problems. Since that time, Monte Carlo methods have been applied to an incredibly diverse range of problems in science, engineering, and business applications in virtually every industry.

Most business activities, plans and processes are too complex for an analytical solution, just like the physics problems of the 1940s. But we can build a spreadsheet model that lets you evaluate your plan numerically, you can change numbers, ask 'what if' and see the results. This is straightforward if we have just one or two parameters to explore. But many business situations involve uncertainty in many dimensions.for example, variable market demand, unknown plans of competitors, uncertainty in costs, and many others, just like the physics problems in the 1940s.

4.2.6 SRAM Timing Diagram

- Synchronous or Asynchronous

  SRAMs come in a variety of architectures and speeds, and in synchronous and asynchronous designs. Asynchronous SRAMs respond to changes at the device's

address pins by generating a clock signal that is used to time the SRAM's internal circuitry during a read or write operation. Although commonly used, this type of design runs into limitations at the high end of the performance range. For this reason, the fastest SRAMs are generally synchronous. Synchronous SRAMs use one or more external clocks to time the SRAM's operations. Because of the improved timing control possible with this method, access times and cycle times can be reduced to match the clock cycles of the fastest PC and RISC processors on the market today.

- Performance Specifications

  SRAM access and cycle times are routinely used to compare the offerings of various manufacturers. The quoted access time is the minimum amount of time required to read a bit of data from the memory, measured with respect to the initial rising clock edge in the SRAM read operation. This access time specification is measured under specific load, temperature, and power supply conditions, in which all critical timings meet the requirements set out in the product specification.

4.2.7 Existing Works

Due to the non-zero delay of the interconnects of the address decoder, word-lines, bit-lines, and the column multiplexer, read and write delay of cells in an SRAM block are different. Simulations show that for a typical SRAM block, the read time of the closest cell to the address decoder and the column multiplexer is 5-10less than that for the furthest cell. This gives an opportunity to reduce the leakage power consumption of the SRAM by increasing the threshold voltage of some of the transistors in SRAM cells.

On the other hand, due to the delay of sense amplifiers and output buffers in a read path, the write delay of a cell is lower than its read delay. Considering the fact that increasing the threshold voltage of the PMOS transistors in a 6T SRAM cell increases the write delay, without having much effect on its read delay, one may try to reduce the leakage power by increasing the threshold voltage of the PMOS transistors as long as the write time is below some target value.

It is known that each additional threshold voltage needs one more mask layer in the fabrication process, which results in increasing the fabrication cost. At the same time,

there are studies that show the benefit of having more than two threshold voltages is small. As a result, in many cases, only two threshold voltages are utilized. Therefore, we concentrate on the problem of low-leakage SRAM design when only two threshold voltages are available. However, it is possible to extend the results to handle more than two threshold voltages.

To reduce the leakage power consumption of a cell, the threshold voltage of all or some of the transistors of the cell can be increased. If the threshold voltage of all transistors within a cell is increased, the leakage reduction is the highest; however, since this scenario has the worst effect on the read delay of the cell, the number of cells that can be replaced is low. They use a symmetric cell configuration, which means the symmetric transistors within a cell have the same threshold voltages. Thus, there are eight different possibilities for assigning high and low threshold voltages to the transistors within a cell. These configurations are shown in Table 4.1. Here it is assumed that the threshold voltage of each transistor in the SRAM cell can be adjusted independent of other threshold voltages by changing the channel doping, which is deemed to be a safe assumption because in an SRAM cell the channels of the transistors are not too close to each other. However, in the Simulation Results Section it will be shown that even if only one threshold voltage is used inside the cell, power reduction can be considerable. The configuration ordering is such that, excluding C0 which is the original configuration, the leakage current saving of other configurations is monotonically decreasing as we move from C1 toward C7. Figure 4 shows the leakage current saving of different configurations versus the value of the high threshold voltage. The numbers in Figure 4 and the following ones are obtained using a 180nm CMOS technology with 1.8V for the supply voltage and 0.37V for the low threshold voltage at 100^0C.

Each of these configurations has different effects on read and write delays of cells. Figures 5 and 6 show the increase in read and write delays for each configuration for different values of the high threshold voltage. It can be seen that the increase in read delay for some configurations (e.g., C1 and C3) is very high, whereas it is very small for some other configurations (e.g., C6). It can also be seen from Figure 6 that not all configurations increase the write time; In fact, C4 and C7 decrease the write time.

TABLE 4.1: Possible configuration for high threshold voltage assignment.

Config.	High threshold transistors
C0	None
C1	M1,M2,M3,M4,M5,M6
C2	M3,M4,M5,M6
C3	M1,M2,M5,M6
C4	M1,M2,M3,M4
C5	M5,M6
C6	M3,M4
C7	M1,M2

4.3 Variation of SRAM Cell with Stress Effect

Stress effects discussed in this work include STI stress and WPE. The STI stress varies by the length of diffusion and WPE varies by the distance from transistors to the well edge. Fig. 4.3 shows a SRAM cell layout of 6T SRAM. Each adjacent cell is flipped cross the X or Y axis to minimize to cell array area. SA and SB represent for the variables of STI for each transistors. And SC stands for the distance form transistor to the well edge. STI stress and WPE are separately discussed in the following sections.

4.3.1 STI Stress

Many researchers had constructed their model for STI stress. The model we used here basically from BSIM4 model [22]. For STI stress,

$$VTH0_{STI} = VTH0_{org} + \frac{KVTH0}{K_{stress\_vth0}} \cdot (Inv\_sa + Inv\_sb - Inv\_sa_{ref} - Inv\_sb_{ref}) \tag{4.1}$$

where KVTH0, which is adjusting coefficient, is a constant for a certain process. K_{stress_vth0} is relate to the size of the transistor. Once the size of the transistor is fixed, K_{stress_vth0} can be seen as constant. sa_{ref} and sb_{ref} are reference values of SA and SB. Inv_sa and Inv_sb, which are refer to the reciprocal of SA and SB related function, can be calculated as,

$$Inv\_sa = \frac{1}{SA + 0.5 \cdot L} \tag{4.2}$$

$$Inv\_sb = \frac{1}{SB + 0.5 \cdot L} \tag{4.3}$$

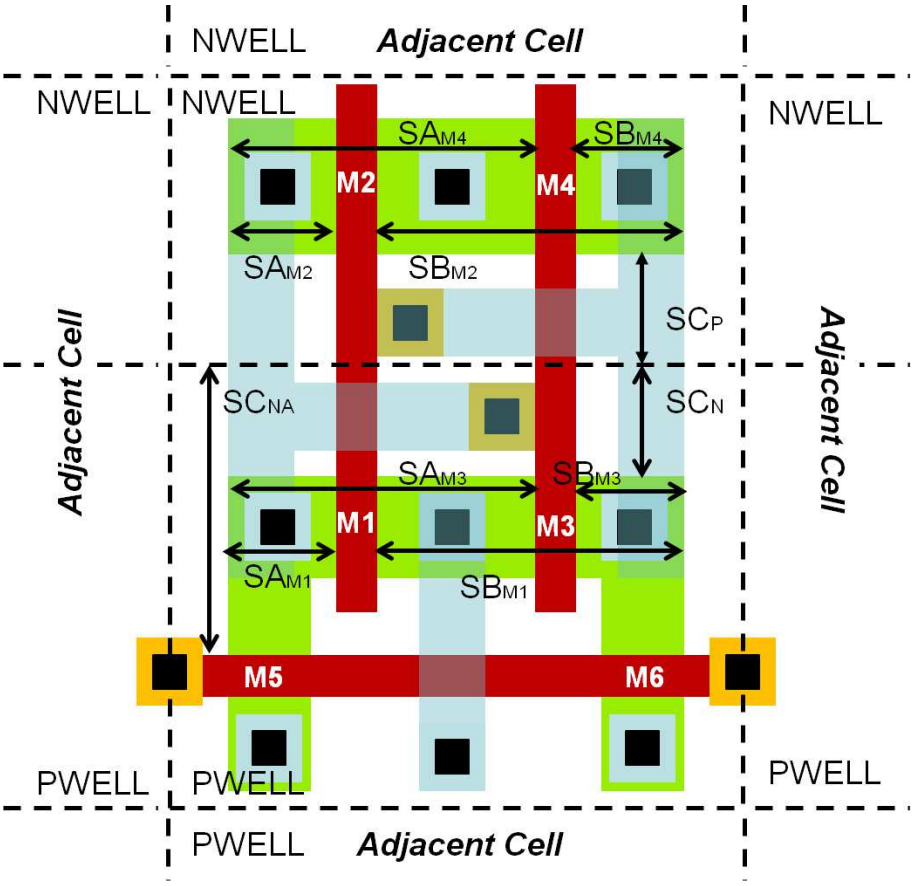

FIGURE 4.3: The layout of a 6T SRAM cell.

Here we observe the variation of MOSFET with STI stress effect by HSPICE simulation as shown in Fig. 4.4. We changed SA value and found that the largest variation of V_{th} is nearly 3.5% as SA approaching the minimal value.

As the minimum size transistors are typically used in SRAM cells to achieve a compact design, SRAM are very sensitive to process variation. Therefore, we used the Monte Carlo simulation technique to evaluate the SNM value. We set $SA = SA_{M1} = SB_{M3} = SA_{M2} = SB_{M4}$ and the distance between the two ploys of M1 and M3 as a constant. By adding Gaussian noise to the relative parameters in BSIM HSPICE model, we ran the simulation for 500 times. The results showed that the SNM can be approximated by a Gaussian distribution. The mean and standard deviation of SNM values are showed in Table 4.2. We can found that as SA turns longer, the σ value becomes smaller. Meanwhile, the $\mu - 3\sigma$, which in statistics mean the worst case of a distribution, turns greater as SA turns longer. Thus, we prefer longer SA to obtain less sensitive device. The longer SA can be realized using folded transistor or diffusion share as well.

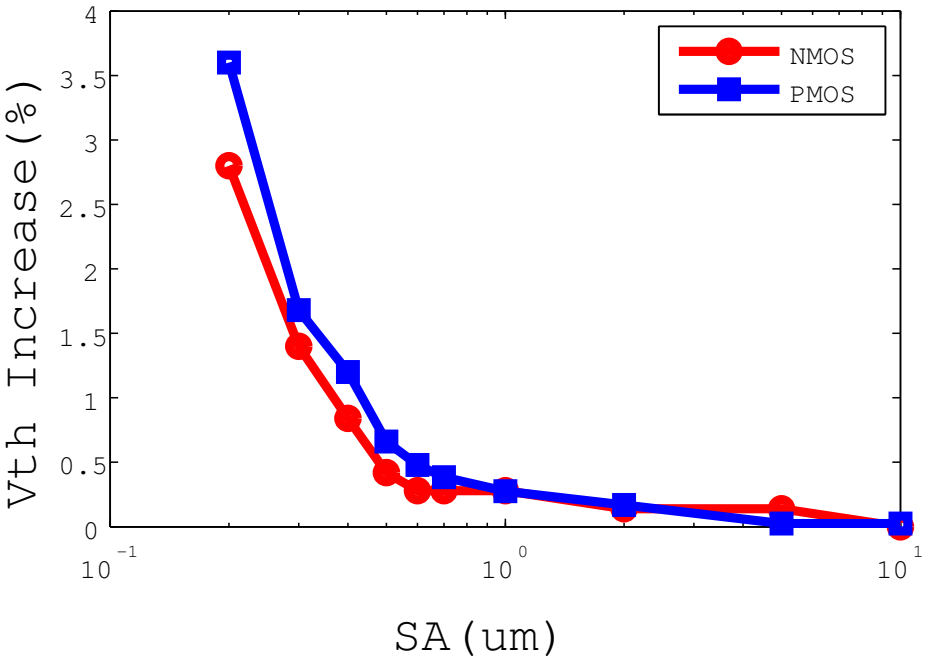

FIGURE 4.4: The variation of the threshold voltage V_{th} with STI stress.

TABLE 4.2: Mean and standard deviation of SNM with STI stress.

SA(μm)	μ	σ	$\mu - 3\sigma$
0.2	0.429	0.078	0.218
0.3	0.443	0.072	0.227
0.5	0.457	0.068	0.240
1.0	0.438	0.069	0.245
2.0	0.425	0.065	0.242

4.3.2 Well Proximity Effect

Since this layout-dependent phenomenon of WPE also increases the threshold voltage and thus decrease the drain current of the transistors, it is especially important for those circuits as SRAMs with high integration density. The corresponding MOSFET parameters of threshold voltage can be calculated as [19],

$$VTH0_{WPE} = VTH0_{org} + KVTH0WE \cdot \left(\frac{1}{SC_{eff}^{NWE} + SC0^{NWE}} - \frac{1}{SC_{ref}^{NWE} + SC0^{NWE}} \right) \tag{4.4}$$

where SC_{eff} is the effective distance from the well edge to the channel region. KVTH0WE, SC_{ref}, SC0 and NWE are the fitting parameters. SC_{eff} can be approximated as

$$SC_{eff} = \left\{ \frac{1}{W_{drawn} \cdot L_{drawn}} \cdot \left[\sum_{i=1}^{n} \left(W_i \cdot \int_{SC_i}^{SC_i + L_{drawn}} f(x)dx \right) + \sum_{i=n+1}^{m} \left(L_i \cdot \int_{SC_i}^{SC_i + W_{drawn}} f(y)dy \right) \right] \right\}^{0.5} \tag{4.5}$$

where f(x) is use to account for both lateral and vertical distribution variations of

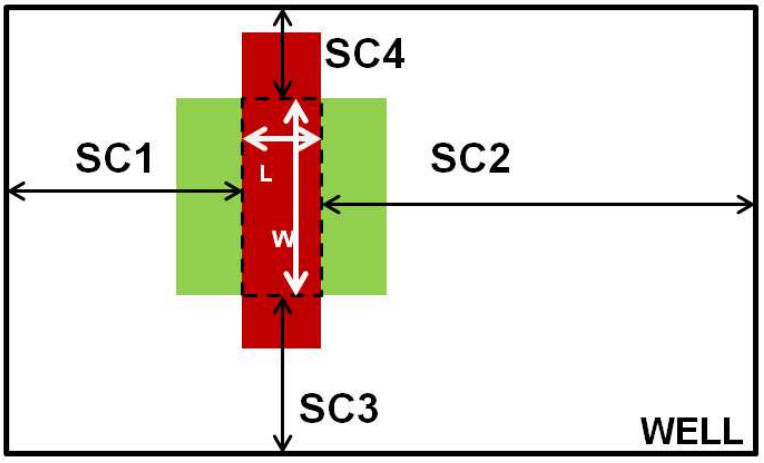

FIGURE 4.5: The description of SC with a transistor put in a well.

scattered well dopants. It is found that $1/x^2$ is a good approximation for f(x) to cover most conditions. In a regular rectangle well, there are two SCs in channel length direction and two SCs in channel width direction, as shown in Fig. 4.5. If the SC is too large, the integration on f(x) is too small that can be neglected. In the sram array, for PMOS M2 and M4 only SC_p is close to the well edge while other CSs are larger because of adjacent cells. SC_N is the closest SC for NMOS and SC_{NA} is the closest SC for access NMOS, as shown in Fig 4.3. Since M1 and M3, M2 and M4, M5 and M6 are symmetric

transistors, so there are only three kinds of SC_{eff},

$$SC_{P,eff} = \left[\frac{1}{W_P} \cdot \left(\frac{1}{SC_P} - \frac{1}{SC_P + W_P}\right)\right]^{0.5} \tag{4.6}$$

$$SC_{N,eff} = \left[\frac{1}{W_N} \cdot \left(\frac{1}{SC_N} - \frac{1}{SC_N + W_N}\right)\right]^{0.5} \tag{4.7}$$

$$SC_{NA,eff} = \left[\frac{1}{L_{NA}} \cdot \left(\frac{1}{SC_{NA}} - \frac{1}{SC_{NA} + L_{NA}}\right)\right]^{0.5} \tag{4.8}$$

For single MOSFET, we observed the variation of SC versus threshold V_{th} as shown in Fig. 4.6. We found that the V_{th} for both NMOS and PMOS increase dramatically when SC is decreasing below $1\mu m$. This phenomenon demonstrates in submicro process the WPE can not be ignored. As the same to STI stress, we performed Monte Carlo simulations to evaluate the SNM with WPE. We set $SC = SC_P = SC_N$, and thus $SC_{NA} = SB_{M5} + SC_N$. By adding Gaussian noise to the relative parameters in BSIM HSPICE model, we ran the simulation for 500 times. The mean and standard deviation of SNM values are showed in Table 4.3. We found that as SC turns longer, the SNM is more stable. Moreover, SRAM cell will have a worse value of SNM when SC under $0.2\mu m$.

Above all, we can found that shorter SA and SC value, that is, smaller size SRAM cell, has high threshold voltage but has worse SNM. Whereas, longer SA and SC value, that is, greater size SRAM cell, has low threshold voltage but has better SNM. Thus, utilizing these phenomenon, we maybe able to optimize the leakage current and delay of the SRAMs circuit as we will discuss in next section.

TABLE 4.3: Mean and standard deviation of SNM with WPE.

SC(μm)	μ	σ	$\mu - 3\sigma$
0.2	0.534	0.083	0.153
0.3	0.424	0.076	0.221
0.5	0.453	0.074	0.224
1.0	0.443	0.063	0.241
2.0	0.427	0.068	0.248

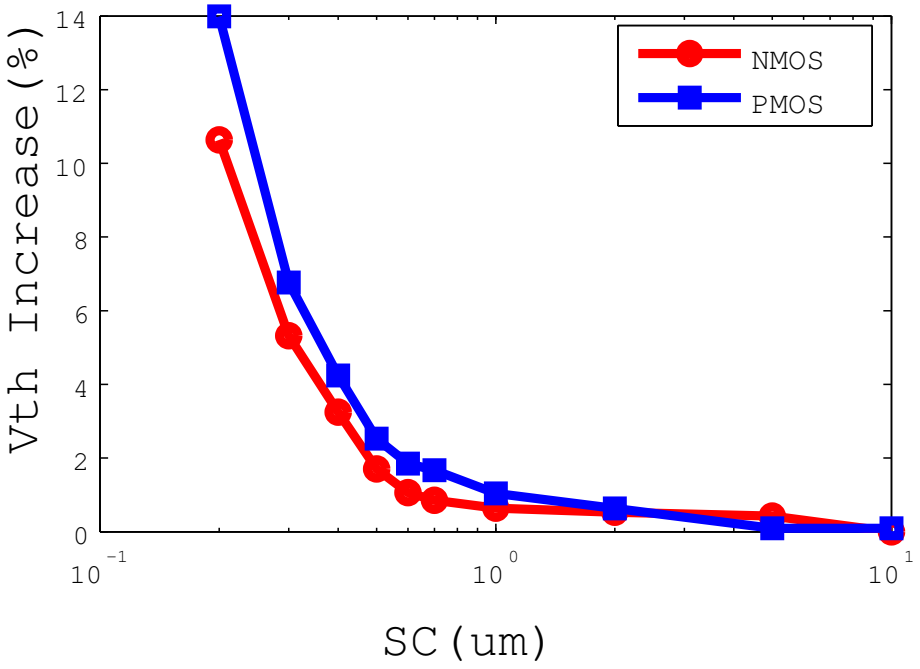

FIGURE 4.6: The variation of threshold voltage V_{th} with WPE.

4.4 Leakage Power and Delay of SRAMs

In SRAM design, power and delay are two very important specifications. Usually, SRAM with high threshold voltage transistors will have lower power dissipation but greater delay. Since stress effects changes threshold voltage of MOSFET dramatically under submicron process, we will discuss these characteristics in this section. However, the highest V_{th} increase with SC is up to 14% while its increase with SA is only 3.5%, we only evaluate the WPE effect to leakage power and delay.

4.4.1 Leakage Power of SRAM

Static noise margin is due to leakage currents drawn continuously from the power supply. However, the main contributor of leakage is the sub-threshold leakage current and is briefly discuss in this section.

In [43], the I_{Dsub} can be approximated as

$$I_{Dsub} = \left(\frac{W}{L}\right) \cdot K_{tech} \cdot 10^{\frac{-V_{th}}{S_t}} \tag{4.9}$$

where K_{tech} of a 6T SRAM can be approximated as 1.2. S_t is referred to as the sub-threshold swing parameter. Once the width and the length of a transistor is decided, the key parameter that influences I_{Dsub} in Eq. 4.9 is V_{th}.

In [44], it is proved that the leakage current of SRAM macro can be considered independent of SRAM operational phase. For a single SRAM the leakage current is,

$$I_{Dsub,SRAM} = I_{Dsub,N1} + I_{Dsub,N6} + I_{Dsub,P2} \tag{4.10}$$

The total leakage power of an SRAM cell is,

$$P(i,j) = V_{dd} \cdot I_{Dsub,SRAM}(i,j) \tag{4.11}$$

Thus the total power for a MxN SRAM macro is the summation of Eq. 4.11,

$$P_{macro} = \sum_{i=1}^{M} \sum_{j=1}^{N} P(i,j) \tag{4.12}$$

We varied $SC = SC_N = SC_P$, and observed the leakage current of a single SRAM cell. Form the result shown in Fig. 4.7, we can found that as SC increasing, the leakage power is increasing.

4.4.2 Delay

Due to the non-zero delay of the interconnects of the row decoder, word-lines, bit-lines and the column decoder, read and write delay of cells in an SRAM block are different. Simulations show that for a typical SRAM block, the read time of the closest cell to the row decoder and column decoder is 5-10% less than that for the furthest cell, as shown in Fig. 4.1. This gives opportunity to reduce the leakage power consumption of the SRAM by increasing the threshold voltage of some of the transistors in SRAM cells.

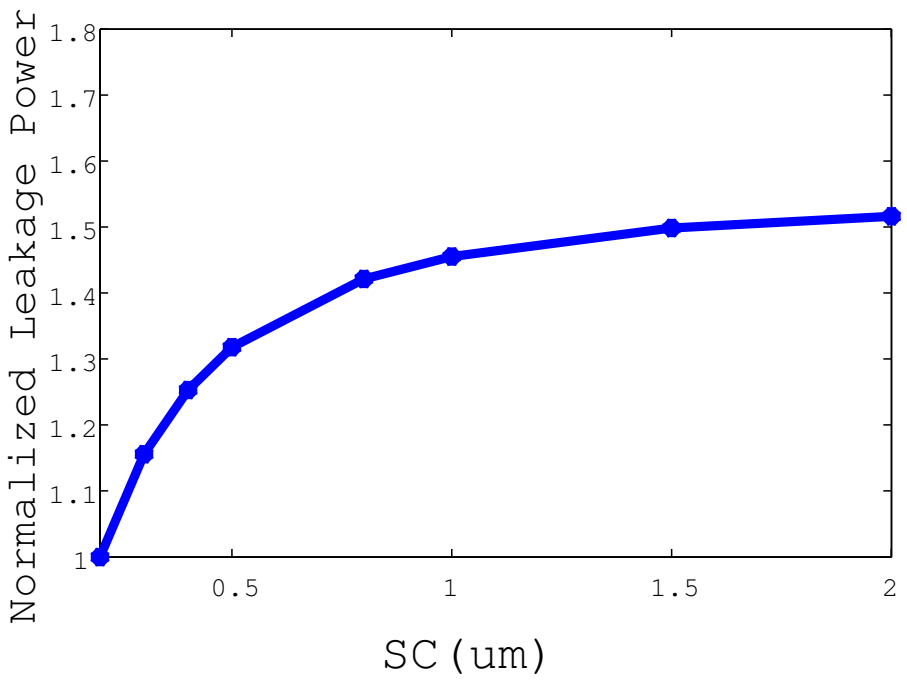

FIGURE 4.7: Leakage Power Variation of SRAM cell with WPE.

The read and write delay can be approximated [45] using,

$$T_{Read} = \tau_{PC} \cdot ln\frac{V_{DD}}{V_{DD} - V_{PC}} + \tau_{change\_read} \cdot ln\frac{V_{DD}}{V_{BL}} + T_{SA} \tag{4.13}$$

$$T_{Write} = \tau_{PC} \cdot ln\frac{V_{DD}}{V_{DD} - V_{PC}} + \tau_{change\_write} \cdot ln\frac{V_{DD}}{V_{write}} + T_{Flipping} \tag{4.14}$$

where, V_{PC} is the maximum voltage of bit-line and data-line at the end of pre-charge, V_{BL} is the voltage of bit-line at the end of read operation, T_{SA} is the delay of sense amplifier, V_{write} is voltage of bit-line that flipping process starts, $T_{Flipping}$ is the delay of flipping process and depended on transistor size of SRAM cell, and τ_{PC}, τ_{change_read},

τ_{change_write} obtain by following expression

$$\tau_{PC} = R_{eq\_PC} \cdot Max\{C_{DL}, C_{BL}\} \tag{4.15}$$

$$\begin{aligned}\tau_{change\_read} &= (R_{eq\_driver\_cell} + R_{eq\_access\_cell}) \cdot C_{BL} \\ &+(R_{eq\_driver\_cell} + R_{eq\_access\_cell} \\ &+R_{eq\_column\_access}) \cdot C_{DL}\end{aligned} \tag{4.16}$$

$$\begin{aligned}\tau_{change\_write} &= (R_{eq\_driver\_DL} \cdot C_{DL} \\ &+(R_{eq\_driver\_DL} + R_{eq\_column\_access}) \cdot C_{BL}\end{aligned} \tag{4.17}$$

where, R_{eq_PC} is equivalent resistance of pre-charging transistor, $R_{eq\_driver\_cell}$ is equivalent resistance of driver transistor of cell, $R_{eq\_access\_cell}$ is equivalent resistance of access transistor of cell, $R_{eq\_column\_access}$ is equivalent resistance of transmission gate, this transmission gate connect bit-line to data-line and $R_{eq\_driver\_DL}$ is equivalent resistance of data-line driver transistors. The equivalent resistance for each transistor during a transition obtained by following equation

$$R_{eq} = \frac{2.5}{\mu_{n,p}\frac{W}{L}C_{ox}(V_{DD} - |V_{th}|)} \tag{4.18}$$

The capacitor appeared in Eq. 4.15∼ Eq. 4.17 can be calculated as

$$C_{BL} = 1.1 \times C_{Junction\_AccessTransistor} \times 2^{Row} \tag{4.19}$$

$$C_{DL} = 2 \times C_{Junction\_ColumnAccessTransistor} \times 2^{Column} \tag{4.20}$$

where, $C_{Junction}$ is the drain junction capacitance of a transistor.

Replace Eq. 4.18 in Eq. 4.15∼ Eq. 4.17 and thus replace them in Eq. 4.13 and Eq. 4.14. We can found that the read delay and write delay are the V_{th} related function. Since the threshold voltage V_{th} is a function of SC. We observed the delay versus SC as shown in Fig. 4.8. We can found that as SC turns longer, the delay becomes small.

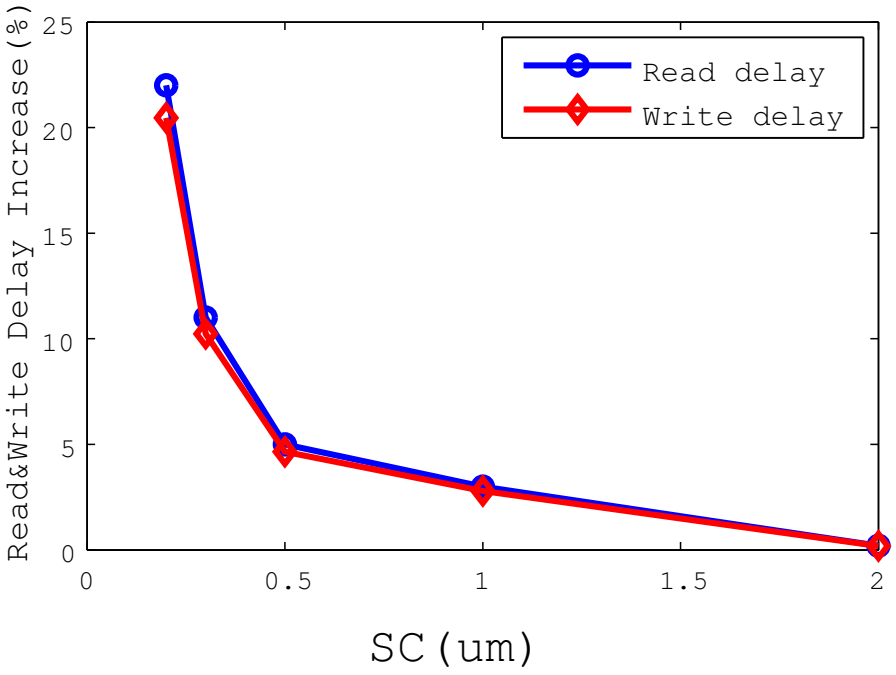

FIGURE 4.8: Read delay increase of a SRAM cell vs. SC.

4.5 SRAM Design Case and Verification

With these stress related model, we aim to design a SRAM array with non-uniform parameterized cells. In this section, we performed a specific SRAM marco design with specifications and then we verified our thoughts using HSPICE.

4.5.1 Design Case

As shown in Fig. 4.9. In order to make the area of the array as a rectangle and avoid the difficulty of routing with word-line and bit-line. Therefore we keep the SA of each column same and SC of each row same.

We designed a 32×64 SRAM with the specification of maximum read time $T_{read,max} < 230ps$ and write time $T_{write,max} < 230ps$. The SNM should not below $0.2V$ with $V_{DD} = 1.2V$. According to the model of leakage power and delay in section 4.4, we calculated the SA column by column and SC row by row to satisfy the specification. The results showed that cell(1,1) has the minimum size while cell(32,64) has the largest size. Each cell in the array also has different sizes as show in Fig. 4.10. SRAM with different cells

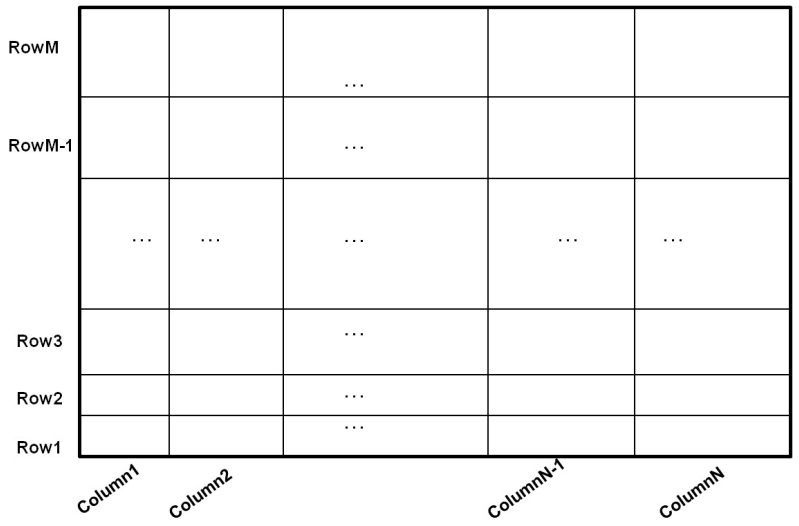

FIGURE 4.9: SRAM with non-uniform parameterized cells which have same SA value in same column and same SC value in same row. Cell(1,1) has the minimum size while Cell(32,64) has the largest size.

will definitely increase difficulty of layout. However, various of automatic layout tool has made such work easier.

4.5.2 Design Verification

In this section, we perform the HSPICE verification. We verified the 32×64 SRAM.

TABLE 4.4: HSPICE verification.

Item	Case 1	Case 2	Case 3
Area(μm^2)	2303	2578	3151
Read Time(ps)	261	219	215
Write time(ps)	252	209	207
Leakage Power(nW)	137.6	182.7	208.7
SNM(V)	0.195	0.223	0.241

We simulated three kinds of SRAMs and compared the area, worst case of read and write time, total leakage power of the array and worst case of SNM by Monte Carlo simulation. The results are shown in Table 4.4.

In case 1, we designed the SRAM cell with minimum SA and SC according to the layout rule. And we found that although it has less size and leakage power, the delay is larger and the worst case of SNM is very low.

In case 2, we designed the SRAM as the proposed methodology, while in case 3 we designed the SRAM with identical cell with all the performance satisfying the specifications. Comparing to case 3, even case 2 has the almost same delay with case 3, case 2 has a decrease of area with 18.2%. Also, case 2 has a improvement of leakage power with 12.5%.

4.6 Summary

In this chapter, we found that the stress effect of STI and WPE will affect the threshold voltage of MOSFET as well as affect the leakage power and delay of the SRAM. Utilizing these effect, we managed to design an SRAM with alterable size which has a good trade off among area leakage power and delay. Our method showed that the propose SRAM has an improvement of area and leakage power with 18.2 % and 12.5% respectively.

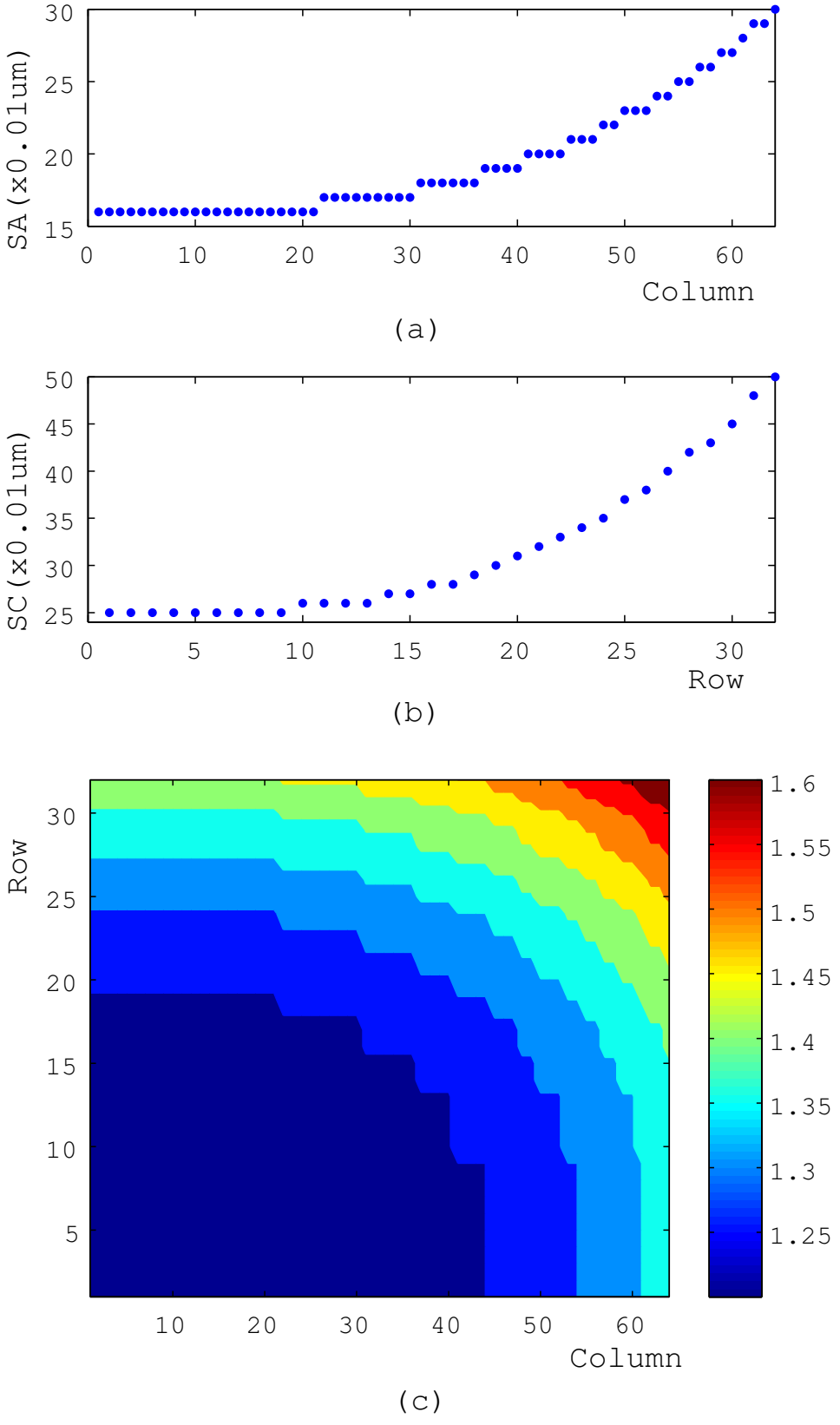

FIGURE 4.10: (a)SA value of each column. (b) SC value of each row. (c)Contour map of SRAM cell size(μm^2).

Chapter 5

Conclusion

We propose a mixed signal VLSI circuit design methodology in this thesis. We divide it into three parts, analog circuit, digital circuit and memory circuit.

In analog circuit design we mainly perform the synthesis via GP. From the specification, we can generate the best solution in a few seconds. The design flow can be summarized in the following:

- First, we decide the specification of circuit.
- Then, we analyze the circuit, and generate its constraints. Meanwhile, we check or calculate relative parameter from the semiconductor process.
- Once we get the constraints, we need to transform these constraints into monomial or posynomial forms. Also construct the layout-dependent constraints.
- We use the GP method to get the optimized the value (size,capacitance,etc.)
- After that we execute the HSPICE verification. If it is not satisfy the specification we tight or loosen the constraints and perform the GP solver again and again until it satisfy the specification.
- At last, we draw the layout and execute relative verifications.

We use op-amp circuit to demonstrate our ideas. Not only the GP solver can find the optimal length and width, but also it can find the finger number and relative LDE

values. And the simulation result shows that our method with the LDE constraints can be worked.

In digital circuit design, we mainly perform the synthesis through commercial software with our two kind of cells: fixed and parameterized cells. The design flow can be summarized as follows.

- First we design the circuit with its structure. Then we use VerilogHDL/VHDL to complete its behavior level.
- We verify its timing.
- Using Design Compiler to perform the logic synthesis.
- Create the standard cell date base including our proposed cells.
- Using IC Compile to perform the P&R.

We use DLL and SRAM to certificate our ideas. Using fixed cell, the DLL shows good performance and small chip size and easy to P&R. Using parameterized cell, we found that SRAM macro has a good trade-off between size and other circuit level performances.

Bibliography

[1] L.T. Wang, Y.W. Wang, and K.T.T. Chen. Electronic design automation: sythesis, verification, and test. 2009.

[2] D. Macmillen, R. Camposano, D. Hill, and T.W. Williams. An industrial view of electronic design automation. *IEEE*, 19:1428–1448, August 2000.

[3] A.B. Kahng, J. Lienig, I.L. Markov, and J. Hu. Vlsi physical design: from graph partitioning to timing closure. 2011.

[4] URL http://en.wikipedia.org/wiki/Electronic_design_automation.

[5] Achim G. Hoffmann. The dynamic locking heuristic - a new graph partitioning algorithm. *Proc. IEEE Int'l Symp. Circuits and Systems*, 1994.

[6] Kenneth M. Hall. An r-dimensional quadratic placement algorithm. 17(3):219 – 229, Nov 1970.

[7] C. Mead and L. Conway. Introduction to vlsi systems. 1980.

[8] E.S. Kuh and T. Ohtsuki. Recent advances in vlsi layout. *Proceedings of the IEEE*, 1990.

[9] R. J. Duffin, E. L. Peterson, and C. Zener. Geometric programming - theory and applications. 1967.

[10] S. Boyd, S. J. Kim, and S.Mohan. A tutorial on geometric programming. *Optimization and Engineering*, 8(1):67–127, January 2007.

[11] S. Boyd and S. J. Kim. Geometric programming for circuit optimization. *Proceedings of International Symposium on Physical Design (ISPD)*, pages 44–46, April 2005.

[12] Maria del Mar Hershenson, Stephen P. Boyd, and Thomas H. Lee. Optimal design of cmos op-amp via geometric programming. *IEEE Transactions on Computer-Aided Design*, 20(1):1–21, January 2001.

[13] Abhishek Debroy, Rahul Choudhury, and Tanmana Sadhu. Analysis on effective parameters influencing channel length modulation index in mos. *International Journal of Emerging Technology and Advanced Engineering*, Oct. 2012.

[14] Bo Liu, Bo Yang, and Shigetoshi Nakatake. Layout-aware variability characterization of cmos current sources. *IEICE Transactions on Electronics*, 2012(4):696–705, January 2012.

[15] Jiying Xue, Zuochang Ye, Yangdong Deng, Hongrui Wang, Liu Yang, and Zhiping Yu. Layout-dependent sti stress analysis and stress-aware rf/analog circuit design optimization. *Proceedings of the 2009 International Conference on Computer-Aided Design*, pages 521–528, 2009.

[16] Andrew B. Kahng, Puneet Sharma, and Rasit O. Tapaloglu. Exploiting sti stress for performance. *Proceedings of the 2007 IEEE/ACM international conference on Computer-aided design*, pages 83–90, Nov. 2007.

[17] R. O. Topaloglu. Standard cell and custom circuit optimization using dummy diffusions through sti width stress effect utilization. *Proc. CICC*, pages 619–622, 2007.

[18] Chi-Chao Wang, Wei Zhao, Frank Liu, Min Chen, and Yu Cao. Modeling of layout-dependent stress effect in cmos design. *ICCAD 2009*, pages 513–520, Nov. 2009.

[19] Yi-Ming Sheu, Ke-Wei Su, Shiyang Tian, Sheng-Jier Yang, and Chih-Chiang Wang. Modeling the well-edge proximity effect in highly scaled mosfets. *IEEE Transactions on Electron Devices*, 53(11):2792–2798, Nov. 2006.

[20] Drennan P.G., Kniffin M.L., and Locascio D.R. Implications of proximity effects for analog design. *Custom Integrated Circuits Conference*, (1):169–176, Spet 2006.

[21] Bsim4.5 compact model. . URL http://www-device.eecs.berkeley.edu.

[22] Bsim4.6 compact model. . URL http://www-device.eecs.berkeley.edu.

[23] Uc berkeley device group. URL http://www-device.eecs.berkeley.edu.

[24] Hook T.B., Brown J., Cottrell P., Adler E., Hoyniak D., Johnson J., and Mann R. Lateral ion implant straggle and mask proximity effect. *IEEE Trans. Electrion Devices*, 50(9):1946–1951, Spet 2003.

[25] Cmc website. URL http://www.eigroup.org/cmc.

[26] Ggplab: A simple matlab toolbox for geometric programming. URL http://www.stanford.edu/$\sim$boyd/ggplab.

[27] B. W. Garlepp, K. S. Donnelly, J. Kim, P. S. Chau, J. L. Zerbe, C. V. Tran C. Huang, C. L. Portmann, D. Stark, Y.-F. Chan, T. H. Lee, and M. A. Horowitz. A portable digital dll for high-speed cmos interface circuits. *IEEE J. Solid-State Circuits*, 34:632–644, May 1999.

[28] J. Dunning, J. Lundberg, and E. Nuckolls. An all digital phase locked loop with 50-cycle lock time suitable for high performance microprocessors. *IEEE J. Solid-State Circuits*, 30(4):412–422, Apr. 1995.

[29] M. Saint-Laurent and G. P. Muyshoondt. A digitally controlled oscillator constructed using adjustable resistors. *Proc. Southwest Symp. Mixed-Signal Design*, pages 80–82, 2001.

[30] M. Saint-Laurent and M. Swaminathan. A digitally adjustable resistor for path delay characterization in high frequency microprocessors. *Proc. Southwest Symp. Mixed-Signal Design*, pages 61–64, 2001.

[31] Y. Moon, J. Choi, K. Lee, D.-K. Jeong, and M.-K. Kim. An all-analog multiphase delay-locked loop using a replica delay line for wide-range operation and low-jitter performance. *IEEE J. Solid-State Circuits*, 35:377–384, Mar. 2000.

[32] J. G. Maneatis. Low-jitter process-independent dll and pll based on self-biased techniques. *IEEE J. Solid-State Circuits*, 31:1723–1732, Nov. 1996.

[33] Mahapatra, Nihar R., Alwin Tareen, and Sriram V. Garimella. Comparison and analysis of delay elements. *Circuits and Systems, MWSCAS-2002*, 2:II–473–II–476, Aug. 2002.

[34] Maymandi-Nejad, Mohammad, and Manoj Sachdev. A monotonic digitally controlled delay element. *IEEE J. Solid-State Circuits*, 40:2212–2219, Nov. 2005.

[35] Yuk-Wah Pang, Wing yun Sit, Chiu sing Choy, Cheong fat Chan, and Wai kuen Cham. An asynchronous cell library for self-timed system designs. *IEICE Transactions on Information and Systems*, E80-D(3):296–305, Mar. Mar.

[36] Jovanović, G. S., , and M. K. Stojčev. Current starved delay element with symmetric load. *International journal of electronics*, 93:167–175, Aug. 2006.

[37] Bo Yang, Qing Dong, Jing Li, and Shigetoshi Nakatake. Structured analog circuit design and mos transistor decomposition for high accuracy application. *Computer-Aided Design (ICCAD)*.

[38] C. Molina et al. Non redundant data cache. *Proc. ISLPED*, pages 274–277, 2003.

[39] International technology roadmap for semiconductors. URL http://public.itrs.net.

[40] C. Kim and K. Roy. Dynamic v_t sram: a leakage tolerant cache memory for low voltage microprocessor. *Proc. ISLPED*, pages 251–254, 2002.

[41] Behnam Amelifard, Farzan Fallah, and Massoud Pedram. Low-leakage sram design with dual v_t transistors. *Quality Electronic Design (ISQED)*, 2006.

[42] Yu Zhang, Bo Liu, Bo Yang, Jing Li, and Shigetoshi Nakatake. Cmos op-amp circuit synthesis with geometric programming models for layout-dependent effects. *ISQED*, pages 464–469, 2012.

[43] J. A. Butts and G. S. Sohi. A static power model for architects. *International Symposium on Microarchitecture*, pages 191–201, 2000.

[44] Mamidipaka, Khouri K. M., N. Dutt, and M. Abadir. Leakage power estimation in srams. *Univ. of Cal. Irvine CECS Techn. Report*, 2003.

[45] Azizi-Mazreah et al. Delay and energy consumption analysis of conventional sram. *Proc. Of World Academy of Science, Engineering and Technology*, 2008.