PATH DELAY TUNING FOR PERFORMANCE GAIN IN THE FACE OF RANDOM MANUFACTURING VARIATIONS

by

Kautalya Mishra

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
May 9, 2011

Keywords: Process variability, Path delay improvement, Performance tuning

Approved by

Adit Singh, Chair, James B. Davis Professor of Electrical and Computer Engineering
Vishwani Agrawal, James J. Danaher Professor of Electrical and Computer Engineering
Victor Nelson, Professor of Electrical and Computer Engineering

Abstract

Moore's laws predictions of transistor densities doubling every two years in an integrated circuit has held true for the past fifty years, and is predicted to hold true in the coming few years. But as technologies shrink to smaller dimensions and scaling becomes more aggressive, a number of factors are beginning to hinder the urge towards miniaturization.

Process variability, that introduces parametric variations in a device, is one such factor that is today seriously limiting clock rates in large synchronous designs. These variations are predicted to increase significantly with device scaling as silicon technology approaches the end of roadmap.

Prior research work on process variability shows that factors contributing to process variations affect the threshold voltage (Vth) of transistors. Varying Vth corresponds to varying path delays in a circuit, a few of which are significant outliers, with path delays much larger than the nominal delays the circuits were designed to have.

These variations make it hard to make any pre-fabrication predictions on the slowest paths in designs that would make room for a speeding methodology because of the random way in which they are distributed in a chip. With circuits today including several hundred million transistors, virtually every design will have a dozen of these outlier transistors that would limit the maximum clock frequency at which the chip can be correctly operated.

This thesis studies a new design architecture that allows for tuning and speedup of exceptionally slow paths in a chip to recover the lost performance and significantly increase the average clock speed attainable by the manufactured parts.

Acknowledgments

I'm very thankful and honored for having got the opportunity to work with Dr. Adit Singh, whose deep insight, energy and conviction was a source of inspiration and motivation. His words of advice and his work ethics have not just helped me work better academically, but also perform beyond it in other areas of life. I'm deeply indebted to him for having given me the opportunity to pursue research for him, and thankful for the lessons learned and the knowledge gained.

I'd like to thank Dr. Vishwani Agrawal for being on my thesis defense committee. Over the period of my Master's pursuit I've learned a number of lessons on dedication and humility while observing Dr. Agrawal both on and off classroom hours. His inquisitive nature and ease of approachability have been instrumental in me overcoming a number of brick walls in my understanding of various concepts.

I'd also like to thank Dr. Victor P. Nelson for also being on my thesis defense committee, and for the knowledge imparted through his courses, that have been of utmost importance as it satisfied a prerequisite requirement for entering the industry and starting my electrical engineering career.

I'd like to thank Auburn University for providing me with the necessary facilities and softwares that were required for my research work, and the people at Auburn who've been a part of my memorable stay at Auburn.

Last but not the least, I'd like to thank my family for their love, support and words of wisdom, and for letting me know that they are always there for me no matter what.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

SiO_2	Silicon-dioxide
μ	Mean
σ	Standard deviation
.pm	Primitive Model file
VHSIC	Very High Speed Integrated Circuits
CMOS	Complementary Metal-Oxide Semiconductor
GND	Ground
HfO_2	Hafnium-dioxide
ITRS	International Technology Roadmap for Semiconductors
LER	Line Edge Roughness
LWR	Line Width Roughness
MOSFET	Metal-Oxide Semiconductor Field Effect Transistor
MOS	Metal-Oxide Semiconductor
NMOS	N-type MOS transistor
OPC	Optical Proximity Correction
PMOS	P-type MOS transistor
PSG	Polysilicon granularity

PSM	Phase-Shifting Mask
RDF	Random Dopant Fluctuations
RET	Resolution Enhancement Techniques
SDE	Source-Drain Extension
Si	Silicon
TOX	Gate oxide thickness
Vdd	Supply voltage
VHDL	VHSIC Hardware Description Language
VOL	Difference between Vdd and Vth
Vth	Threshold voltage

Chapter 1

Introduction

The electronic industry has come a long way ever since the integrated circuit was invented in 1958. Gordon E. Moore's predictions, popularly known as Moore's Law, on transistor densities doubling every two years in an integrated circuit have held true so far, and is expected to hold true in the coming next decade at least. Increased complexity, faster clock rates, increased storage capacity and the ability to put in a lot more in to the same area have been the major driving forces favoring scaling.

But sustaining the scaling trend as forecast by Moore's Law is now facing serious challenges [1] in the fundamental limitations imposed by the inherent physical properties of the underlying materials, required tolerances down to a few atomic dimensions in the manufacturing processes, and statistical anomalies arising from the finite number of atoms constituting device structures in highly scaled technologies. Even though the scaling trend as predicted by the International Technology Roadmap for Semiconductors (ITRS) is expected to continue for another decade at least, performance gains measured in terms of processor speeds is expected to saturate or even fall back a little. Figure 1.1 represents the processor clock frequency variation with time.

A majority of the work done in this area, to compensate for the drop in processor speeds, has been in the field of multiprocessing, where microprocessors are being designed with multiple cores to improve performance through parallelism. Indeed impressive gains have been observed in cases of dual and quad core processors, which have even inspired thoughts of incorporating a dozen or even several hundred cores in future advanced designs.

But according to Amdahl's law, and backed by extensive research, it has been shown that additional processors only yield diminishing returns in most general purpose applications.

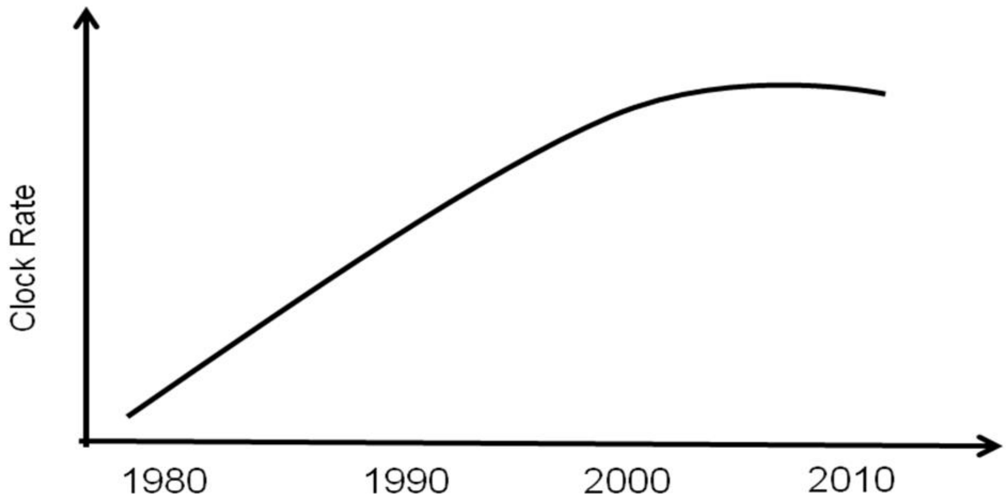

Figure 1.1: Processor clock frequency versus Time

Hence, in the long run, having multiple cores is not going to be a viable solution to improving performance. It is imperative that focus be given to addressing challenges of scaling that limit clock frequencies.

At the small dimensions devices are being fabricated today, manufacturing variations that produce changes in device parameters are found to occur randomly and in significant amounts. Because of their random nature, these variations are beyond the control of designers for them to make any prefabrication changes to their designs to overcome the effects of these variations. Such random process variations occur significantly in nanoscale devices, and are expected to be more significant in the 22nm technology and beyond.

Large complex designs today have several millions of transistors, and it is statistically likely that every manufactured part will have dozens of these slow transistors that are performance outliers, lying in the far ends of the distribution, beyond 4-6 standard deviations away from the nominal value. A presence of such an outlier along a path can significantly push the path delay up, and hence limit the frequency at which the device can be operated.

The focus of this thesis is to study a new tuning technique that would enable post-manufacture speedup of exceptionally slow paths, and hence provide the ability to push clock frequencies up to help get back much of the lost performance. The aim is to bring down the outlier path delays as close to the nominal average case delay, without changing the logic levels at which the gates switch, and ensuring the extra power dissipation is within acceptable limits.

Also, variations in processor speeds make it imperative that every chip or core be individually "speed binned" and operated at the fastest clock rate that it can reliably sustain, to take advantage of this tuning.

Tuning, again, is done post manufacture and not during the design stage. Process variability being random introduces parametric variations randomly, hence taking away any scope of predicting where the outliers might be to make any changes during the design stage.

It is also assumed in this work that the digital circuitry is designed in CMOS logic. This assumption is valid as CMOS logic continues to be the most power efficient logic, and is hence expected to be the dominant logic to design digital circuits with.

The aim of this work has been to enable tuning by adding a minimum amount of extra circuitry in each CMOS gate to speed up a particular transition at the output. Hence, once an exceptionally slow transistor is identified, cells along the path are programmed to speed up the slowest transition. This tuning, limited to only a few cells that are statistical outliers, can help push the clock frequencies up by a significant amount. This also happens with minimal impact on the total power dissipated in the chip, as will be discussed in the following chapters.

The following sections of this thesis are arranged as follows. Literature survey is presented in Chapter 2, effects of the presence of an outlier in Chapter 3, the CMOS tunable gate architecture that is to be included in the design is introduced in Chapter 4, followed by the steps taken to extract the SPICE netlist in Chapter 5, an outline of the method and the algorithm used to show the effectiveness of the design in Chapters 6, the simulation results in Chapter 7, power dissipation simulation and results in Chapter 8, and finally a conclusion in Chapter 9.

Chapter 2

Literature Survey

This chapter presents a brief study on the present and past research work done on process variability. The tuning architecture proposed in this work is novel, but the subject of process variability as such has been dealt with and studied in the past. A historical overview of process variability is first presented. That is followed by sections on the critical sources of variability. A final section on the impact of process variability is also presented.

2.1 Historical Overview of Process Variability

The semiconductor industry has distinguished itself by its rapid pace of improvement, which it owes largely to the industry's ability to exponentially decrease the minimum feature sizes used to fabricate integrated circuits [1]. But as the feature sizes get smaller, the variability associated with fabrication becomes more significant. It may seem like process variability is a new challenge, with the large amount of research literature today dedicated to it. But in fact process variability has always been a critical aspect of semiconductor fabrication [11].

Random variations in a semiconductor was first discussed in Shockley's 1961 paper titled "Problems related to p-n junctions in silicon," where the effect of statistical spatial fluctuations of donor and acceptor atoms with its resulting impact on the breakdown voltage in the junction is considered [12]. After that a number of other works on the effects of random ion implantations have been done. The most prominent ones include works by Schemmert and Zimmer on the sensitivity of threshold voltage (Vth) to implantation energy [13] , and works by Yokoyama et al. [14] on studying Vth sensitivity using Monte Carlo approach.

As can be seen, process variation has been a subject of interest for a long time. It has, though, become more significant in recent years as feature sizes get smaller and as we slowly approach the end of the roadmap, which poses certain fundamental limitations on further scaling.

2.2 Sources of Variability

Process variability may be categorized under two types in advanced CMOS technologies - local or intra-die variations and global or inter-die variations [8]. Local variability is over a short distance and normally within a die, while global variability is over larger distances and normally seen between die-die and wafer-wafer.

Global variability causes a shift in the mean value of sensitive design parameters such as channel length (L), channel width (W), layer thickness, resistivity, doping concentration, and body effect, while local variability introduces (1) systematic variability and (2) random variability.

Systematic variability includes variability caused by optical proximity corrections (OPC), phase-shifting mask (PSM), layout-induced strain, and well proximity effect, and can be addressed by more controlled Resolution Enhancement Techniques (RET) and layout designs.

Random variability includes several sources of which the critical ones are Random Dopant Fluctuations (RDF) [8, 11, 15, 16] , Line-edge and Line-width roughness (LER and LWR) [8, 11, 17, 18, 19, 20] , variations in gate oxide thickness (TOX) due to interface roughness [8, 11, 22] , polysilicon granularity [8, 11, 21, 22] , and high-K dielectrics with metal gates [8, 11, 21, 22].

Of the above variability types, random variability is the most critical because its impact on circuit performance is becoming increasingly significant for technology nodes below 45nm. The large outlier delays, observed in almost every device, that are almost four to five times above the average case worst case delays arise out of random variability. Also, addressing random variability requires innovative process and circuit design techniques and device modeling. The focus of this work, hence, has been to address the issue of random local variability.

A brief description of some important sources of random variability is presented below.

2.2.1 Random Dopant Fluctuation (RDF)

RDF has an increasingly significant effect on the MOS Vth in the sub-micron technologies. In MOSFET's, transistor channels are doped with dopant atoms to control the Vth. To keep short channel effects from degrading the Vth in technology nodes below 90nm, the doping concentrations are kept very high. For a MOSFET device with effective channel length, L_{eff}, effective width, W_{eff}, channel doping concentration, N_{CH}, and source-drain extension (SDE) junction depth, xj, the total number of dopant atoms in the channel, according to Saha, K. S. [8] , is given by:

$$Ntotal, chan\ = N_{CH}.(W_{eff}.L_{eff}).xj \tag{2.1}$$

According to Equation 2.1, as dimensions are scaled the number of doping atoms, given by $Ntotal, chan$, comes down drastically even though the doping concentration goes up, because the transistor dimensions L_{eff}, W_{eff} and xj reduce. Technology scaling involves reducing transistor area by a half with every generation. Hence, $Ntotal, chan$ would decrease exponentially over technology generations; 45nm technology node has only around a 100 dopant atoms in its channel [11]. Any small change in the number of atoms would hence imply a significant effective change in the doping concentration, and hence a significant difference between the Vth of one transistor and another. This is what is referred to as process variability. Its contribution towards device mismatch can be studied through the following equation,

$$\sigma V_{tran} = \left(\frac{\sqrt[4]{4.q^3.\epsilon_{Si}.\phi_B}}{2}\right).\frac{TOX}{\epsilon_{ox}}.\left(\frac{\sqrt[4]{N}}{\sqrt{W_{eff}.L_{eff}}}\right) = \frac{1}{\sqrt{2}}.\left(\frac{c_2}{\sqrt{W_{eff}.L_{eff}}}\right) \tag{2.2}$$

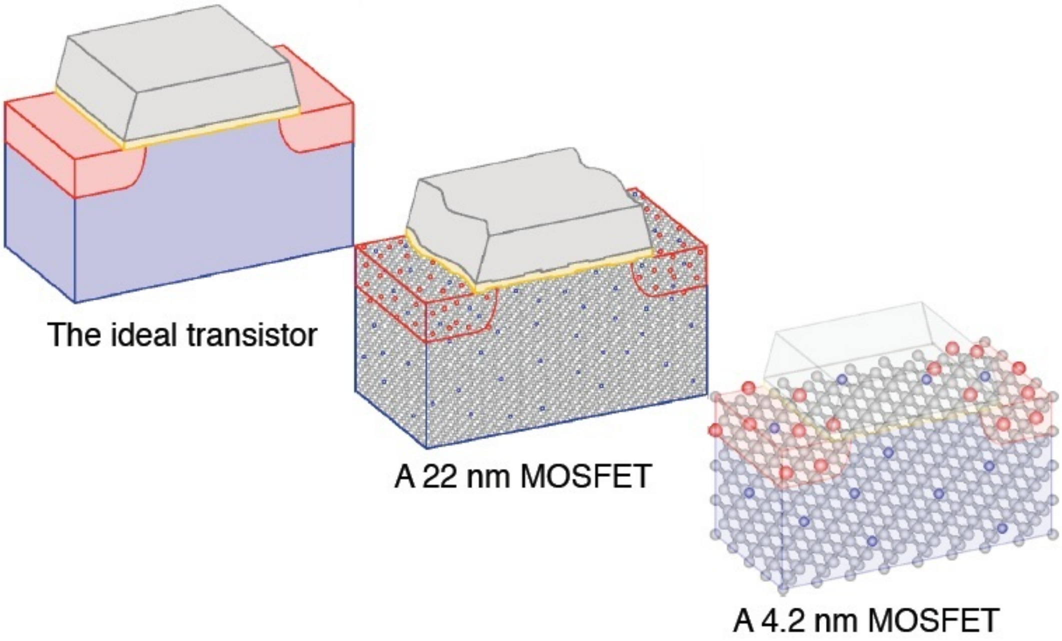

Figure 2.1: Random Dopant Fluctuations in sub-micron technologies

where q, ϵ_{Si}, ϵ_{ox}, and ϕ_B are the electron charge, permittivity of the silicon, permittivity of SiO_2, and the built-in potential of S/D-to-channel PN junction of MOFETs, respectively. C_2 is a constant that depends on the N_{CH} and TOX, and hence depends on $Ntotal, chan$. Even though $Ntotal, chan$ decreases with scaling, W_{eff} and L_{eff} reduce as well, and hence effectively the device mismatch becomes bigger with scaling [15].

A 3D numerical model with an adaptive local meshing scheme that allows prediction of Vth for arbitrary dopant profiles (see Figure 2.1) was developed and simulation results for 45nm and 65nm were compared to the observed variations [16]. RDF was found to be 65% of the total σVt in 65nm silicon, and 60% of the total σVt in 45nm silicon.

2.2.2 Line-Edge Roughness (LER) and Line-Width Roughness (LWR)

A second important factor responsible for process variability is one that arises out of variations in gate patterning, leading to non-ideal (rough) line edges. LER is a result of sub-wavelength lithography, which the semiconductor industry has been using for patterning

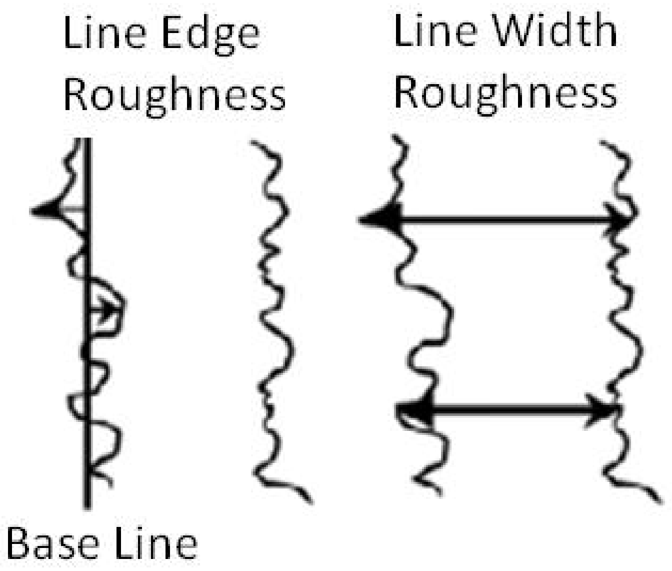

Figure 2.2: Definitions of LER and LWR

transistors since the 0.25μm technology node. For example, fabrication processes previously used the wavelength of light, λ=248nm, to pattern the minimum feature size, Critical Dimension CDmin=250nm and 180nm transistors. Wavelength λ decreased to 193nm for 130nm technology, and has remained the same since, even for 65nm transistors.

Until ultra-violet technology becomes available, sub-wavelength lithography will continue to be used for patterning, and will cause LER and LWR effects in scaled MOSFETs [8, 11].

Impacts of LER and LWR include increases in sub-threshold current and degradation in the Vth characteristics [17, 18, 19, 20]. Diaz et al [17] found transistor performance degradation to occur with an increase in the observed LER values. Experiments performed by Kim et al. [18] showed that LER effects began for gate lengths below 85nm. They also observed a four-order increase in the standard deviation of the sub-threshold current for the smallest gate lengths in their study. Fukutome et al. [19] in their experiments, observed that roughness of extension edges induced by gate LER depended on the implanted dose, halos (pockets), and various co-implantations. They showed an improvement of 4nm in

Vth roll-off with a decrease in the average LER, and confirmed that co-implants induced a degradation of 5mV in the standard deviation of Vth. Another important observation noted in experiments performed by Asenov et al. [20] was that LER and RDF effects act in a statistically independent manner, and that LER induced fluctuations have stronger channel length dependence and are hence expected to supplant RDF as the dominant source of variation.

The Vth-mismatch due to LER depends on the variability in W_{eff} of the MOSFETs and is given by [11] :

$$\sigma V_{TH,LER} \propto \frac{1}{\sqrt{W_{eff}}} < \sigma V_{TH,RDD} \tag{2.3}$$

2.2.3 Oxide thickness variation due to interface roughness

Conventional CMOS technologies with Silicon-dioxide (SiO_2) gate dielectrics are subjected to process variability introduced by the Si/SiO_2 and SiO_2/polysilicon-gate interface roughness through TOX variation [8]. In advanced MOSFETs with TOX of about 1nm, the interface roughness is found to be comparable to the TOX, and hence the variations associated with it can be about 50%. Such variations in the TOX introduce variations in the gate current Ig which produces a voltage drop in the polysilicon gate that changes the Vth significantly. Studies done by Asenov et al. [22] show that intrinsic Vth fluctuations induced by local TOX variations become comparable (∼30mV) to voltage fluctuations introduced by RDF for conventional MOS devices with dimensions 30nm and below. It is also found that process variations introduced by TOX variations due to interface roughness is found to be statistically independent of process variations introduced by RDF [8, 11]. The effective contribution towards Vth variation through the two effects is hence given by:

$$\sigma V_{TH,total} = \sqrt{\left(\sigma V_{TH}^{T_{OX}}\right)^2 + \left(\sigma V_{TH}^{RDD}\right)^2} \tag{2.4}$$

2.2.4 Polysilicon granularity (PSG)

PSG enhances the gate dopant diffusion along the grain boundaries which leads to non-uniform polysilicon gate doping and potential localized penetration of the dopants through the gate oxide into the channel region. The most significant source of fluctuation within the polysilicon gate though, is the Fermi level pinning at the boundaries between grains due to the high density of defect states. Local variations in the potential of up to 0.6V within the gate would reflect on fluctuations in the surface potential (ϕ_s) within the MOSFET channel leading to Vth mismatch between devices. In the sub 65nm MOSFETs these fluctuations are comparable to those introduced by RDF [8, 11, 21, 22].

2.2.5 High K dielectric morphology

Advanced CMOS technologies use high-K dielectrics like HfO_2 with metal gates to provide a thicker physical TOX to reduce the amount of gate leakage and ensure a thin electrical TOX required for the continuous scaling of MOSFETs. Its inclusion, however, introduces significant process variability due to the Si/high-K and high-K/MG interface roughness which causes mobility degradation and TOX variation, and the phase separation that is created between the crystallized grain and the amorphous SiO_x matrix which causes fluctuations in the channel potential under the gate. Also, the polycrystalline HfO_2 with random grain orientations causes dielectric constant to vary across the gate oxide. Hence, the surface potential ϕ_s varies with L and causes variability in Vth. This Vth variability due to the presence of high-K dielectrics increases with scaling, and is expected to be more than 30 mV for 10nm MOSFETs with TOX of about 4nm [8, 11, 21, 22].

2.2.6 Other sources

Other sources of process variability, besides the critical sources mentioned above are variations associated with patterning proximity effects, variations associated with polish such as shallow-trench isolation, gate and interconnect variation, variations associated with strain, high stress capping layers, and embedded silicon-germanium, and variations associated with implants and anneals.

2.3 Impact of Process Variability

Random variations introduced through RDF, LER/LWR, TOX variation, PSG, dielectric variations, and other sources has a critical impact on the yield, performance and reliability of the manufactured circuits. As a result parameters defined for transistors at the design stage vary significantly, post-manufacture, from their given nominal values. These variations are expected to get only worse with scaling.

All of the sources mentioned above primarily degrade Vth. RDF and LER contributions towards Vth variation are described in Equation 2.2 and Equation 2.3; of the two sources LER is expected to become more critical as scaling continues. TOX variation due to interface roughness has been shown to degrade threshold voltage as well, and for transistors with 1nm TOX the Vth variation introduced is almost comparable to that introduced by RDF. PSG and high-K dielectric effects degrade the threshold voltage Vth by affecting the surface potential (ϕ_s).

Hence the focus of this work has been on studying threshold voltage variability in circuits designed with CMOS logic. Only CMOS logic is assumed as it is the most power efficient logic among all logic types, and is expected to continue to be the dominant logic type to design with in future technologies as well.

Even though every process variability source affects Vth in a statistically independent manner, in this work values of Vth are drawn from a single normal distribution assuming the effects of all sources have been included. Hence for the purpose of simulation, every transistor in every circuit simulated has Vth drawn from a nominal distribution whose mean and standard deviation values are specified by the technology generation in which the transistors are being designed.

In conclusion, random process variability is an increasing concern as dimensions shrink and scaling continues. Different sources that contribute to variability have been studied and their effects analyzed. The next chapter studies the impact of variability on path delays.

Chapter 3

Effects of Outlier Presence on Path Delays

Extensive prior research suggests Deep Sub-Micron Technologies (DSM) are prone to high process variability [1-24]. Distributions representing parametric values of transistors, post fabrication, indicate a high standard deviation, implying a wide spread of the distribution, and a trend that the standard deviation would increase with scaling.

3.1 Normal distribution

Figure 3.1 compares two distributions of Vth for two different technology generations. As can be seen, the spread of the distribution gets wider while the height of the distribution gets shorter, as we scale from one technology generation to another.

Assuming a normal distribution for the spread, the results shown in Table 3.1 can be arrived at [25] :

Range(a,b) $(\mu + a * \sigma, \mu + b * \sigma)$	Population within (a,b)	Approximately 1 in
(0,1)	34.13%	3
(1,2)	13.59%	7
(2,3)	2.14%	47
(3,4)	0.13%	770
(4,5)	0.003%	30000
(5,6)	0.000028%	3.5 million
(6,∞)	0.0000001%	1 billion

Table 3.1: Normal distribution - Statistics

Table 3.1 represents the percentage of transistors that fall within the ranges specified on the right side of the distribution. A similar table can be constructed indicating the percentage of transistors that would fall in the left side of the distribution; being symmetrical

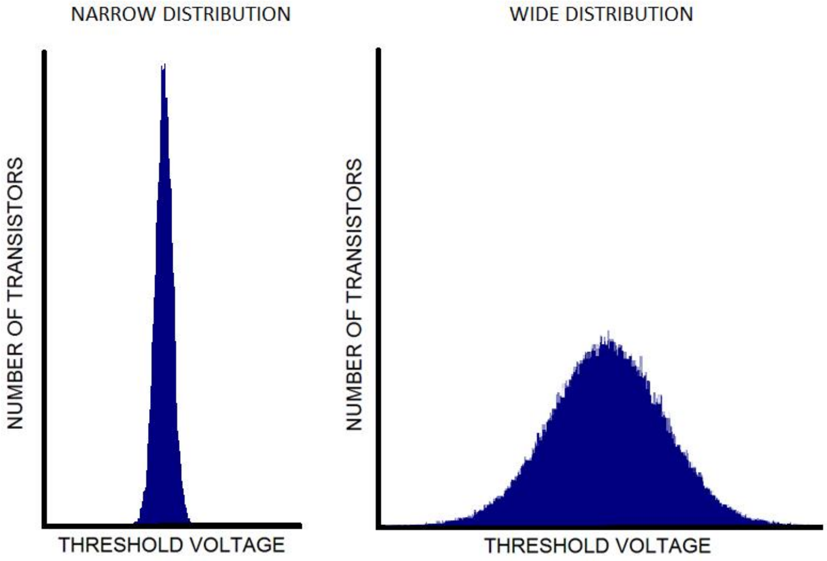

Figure 3.1: Threshold voltage variation with scaling

the percentages would be the same on both sides. Vth greater than nominal, i.e. falling on the right side of the distribution, would correspond to a slow transistor, while Vth less than nominal, i.e. falling in the left side of the distribution, would correspond to a fast transistor. It is the slow transistors that are of interest, as their presence on a path pushes path delays up by a factor which is greater than the speedup achieved by a symmetrically falling low Vth transistor, and is hence in effect responsible for the slow down.

Assume a nominal threshold voltage μ=0.5V, a standard deviation σ=50mV, and Vdd=1V, for a particular deep sub-micron technology [26]. Here $\mu + 5 * \sigma = 0.75$V. By Table 3.1, 1 in 3.5 million transistors lie beyond 5 standard deviations, i.e. approximately 300 in a design with a billion transistors.

Most designs today have billions of transistors in them, and hence, statistically there is every chance to find at least a few hundreds of such outlier transistors in every fabricated chip that lie far away in a distribution, at least 5 standard deviations away.

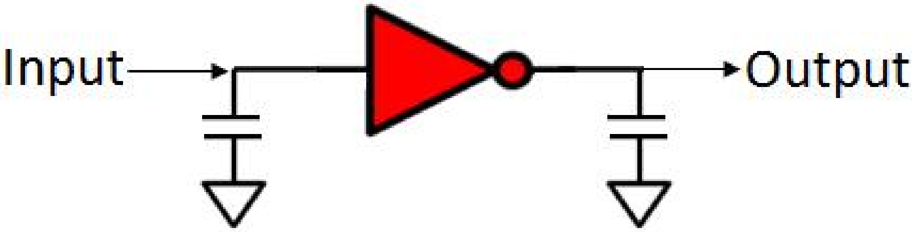

Figure 3.2: Standard inverter with an input and an output load

3.2 VOL

In the above example, Vth is greater than 0.75V beyond 5 standard normal deviations, i.e. the difference between Vdd and Vth, defined as VOL, is less than 0.25V. As Vth gets larger, it gets closer to Vdd, the supply voltage, and the delays increase exponentially.

To get a measure of this increase, simulations were done on a SPICE netlist of an inverter (Figure 3.2) with an input load and an output load, by varying Vth of the pull-up PMOS from 0.2V through the nominal value of 0.4V to 0.7V, for a supply voltage Vdd of 1V. The simulations were done using 180nm technology files, and the extraction of the SPICE netlist through the various Mentor Graphics tools. More on the technology and tools used is discussed later.

Vdd = 1V			
VOL=Vdd-Vth (V)	Vth (V)	Delay (ps)	Change from nominal (%)
0.3	0.7	377	508
0.4	0.6	167	170
0.5	0.5	95	53
0.6	0.4	62	0
0.7	0.3	44	-29
0.8	0.2	33	-46

Table 3.2: Delay versus VOL

Table 3.2 gives a sense of how delay varies with Vth. As Vth increases, inverter delays increase exponentially. For an increase of 0.2V above the nominal Vth, the increase in delay

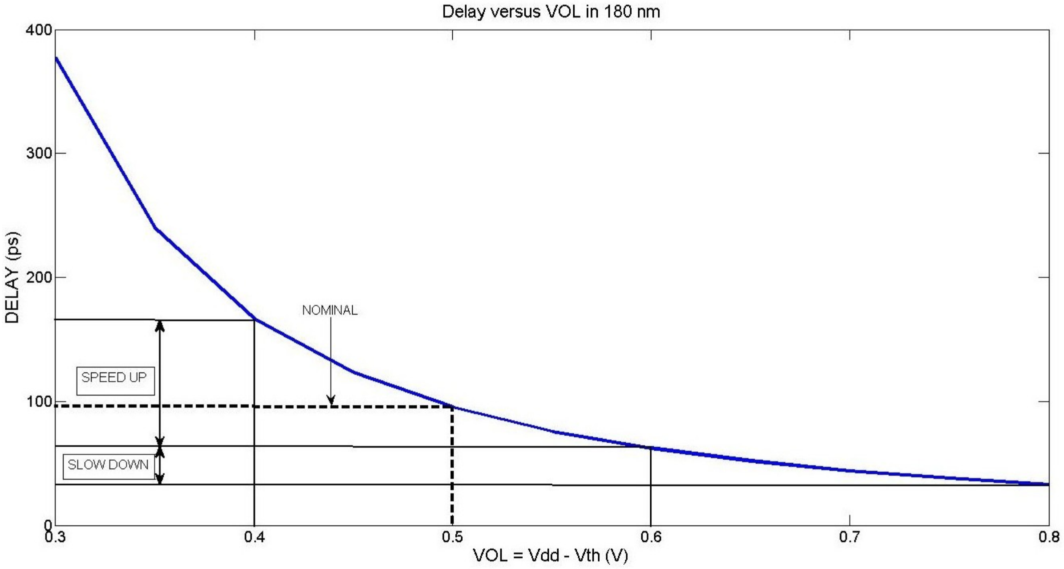

Figure 3.3: VOL versus Delay

is close to three times the nominal delay value. Hence, for transistors that lie beyond five standard normal deviations, the increase in delays will be more than four times the nominal.

The presence of such an outlier in a path can push up delays by a significant amount. To illustrate using an oversimplified example, assume an 8 length inverter chain, with all inverters having nominal delays ‘X’. The total delay of the inverter chain would be ‘8X’. Now assume a ‘4X’ increase in the delay of an outlier inverter that is on such a chain. The total delay is now pushed to ‘12X’, implying a 50% increase in path delays, and hence a 33% decrease in the clock frequency.

It is important to note that even though the distribution is symmetric, implying the presence of an equal number of fast transistors along a path in the circuit, the total delays on a path do not get averaged out. Figure 3.3, which is a plot between VOL and Delay, illustrates that for an equal change in Vth, the speed up is a lot smaller than slow down. Hence, there is a very small chance that a slow outlier node in a path would get compensated by a fast switching node.

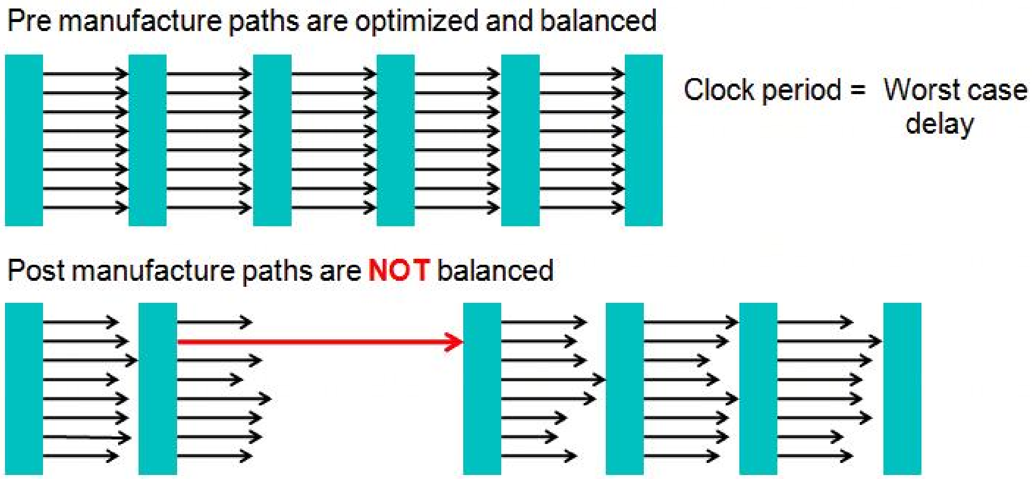
Pre manufacture paths are optimized and balanced
Clock period = Worst case delay
Post manufacture paths are NOT balanced

Figure 3.4: Process variability effects on pre-optimized designs

3.3 Effects of process variability on high performance applications

The presence of a slow outlier on a path doesn't necessarily imply that the path becomes critical. There are chances of the path itself being small in comparison to other paths. The large delays introduced by an outlier node in such a case might not make the path critical.

But most high performance applications today are speed optimized in the design stage to have close to approximately equal path delays so as to save on power. In such high performance designs where all paths are close to critical, the presence of an outlier node on any path can push the critical path delay up by a significant amount.

Figure 3.4 illustrates the case. It represents a pipelined architecture with the rectangles representing registers and arrow lengths representing path delays. The minimun clock period of a design equals the worst case delay. Hence, in the presence of an outlier, even though average case delays do not change by much, the clock period increases solely because of the exponential increase in the delay of a few outlier paths.

In conclusion, statistically every fabricated design with close to a billion transistors will have at least a few hundreds of outliers whose presence on any path can push clock delays up by a significant amount and hence restrict the frequency at which the circuit can be operated. The aim of post manufacture tuning is to bring down the clock period close to the average case delays.

Chapter 4

CMOS tunable gate architecture

This chapter introduces the tunable CMOS gate architecture, which forms the basis of this work. The tunable gate architecture design is first introduced, followed by the tuning strategy, a study on the sizing of the tuning transistors, the occurrence of an interesting case, and a final simulation result that shows the effectiveness of the tuning methodology on a single NAND gate.

4.1 CMOS tunable gates

Every CMOS gate has a pull-up network with PMOS transistors, and a pull-down network with NMOS transistors. Under static conditions either the pull-up network is 'ON' or the pull-down network is 'ON'. The output load capacitance of the gate charges to Vdd when the pull-up network is 'ON', while the capacitance discharges when the pull-down network is 'ON'. It is this charging or discharging time that determines the speed at which the gate switches.

The presence of a parallel path can speed up the charge time or the discharge time; it is this principle that lays down the foundation of the tuning strategy. A parallel path for charging and discharging the output load capacitance is provided to speed up an otherwise slow transistor. To introduce this parallel path in the design, a parallel PMOS transistor with tuning capability is introduced in to the pull-up network, and a parallel NMOS transistor with tuning capability is introduced in to the pull-down network. The resulting architecture resembles the design shown below in Figure 4.1.

To include the parallel path of charging or discharging, the corresponding tuning transistors have to be turned 'ON'. This is only done after diagnosis has been performed and

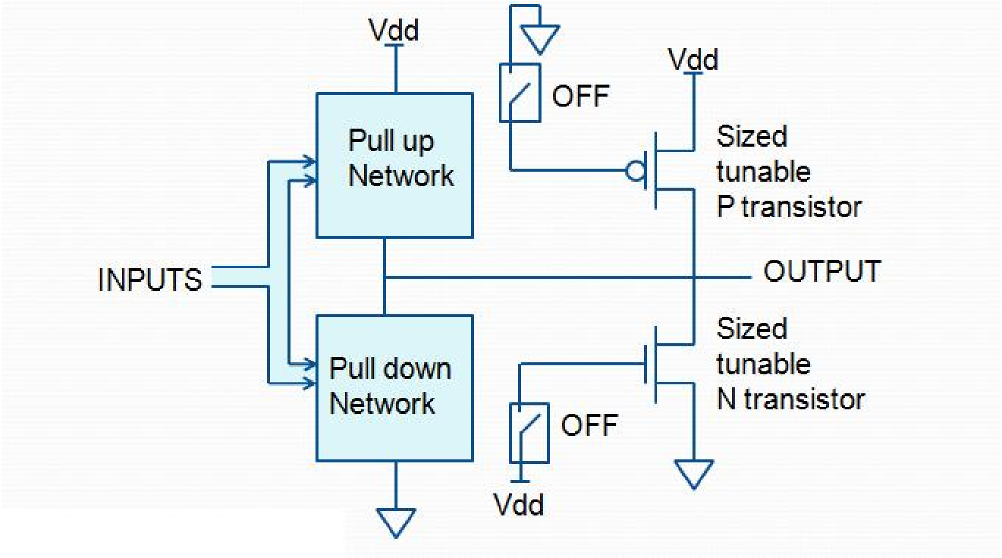

Figure 4.1: Tunable CMOS architecture

the slow outliers identified. The gate terminals of the tuning transistors are connected to a switch which can be turned ‘ON’ or ‘OFF’. The switches are ‘OFF’ when no tuning is required, and ‘ON’ while tuning is done. Once turned ‘ON’, logic of that particular gate switches from a CMOS type to a pseudo-NMOS type.

The tuning transistors have to be sized appropriately to provide good speed up without affecting the output voltage logic levels. More is discussed about the sizing of the tuning transistors in the subsequent sections of this chapter.

It is important to note that every node in the design comprising a pull-up and a pull-down network has to have this tuning capability in it (Figure 4.2). This is because of the inability to make any prefabrication predictions on the location of the outliers; outliers are created randomly and can hence fall anywhere on the chip. The chances of a gate having both a faulty PMOS and a faulty NMOS are very small; in its occurrence the fabricated chip would be discarded as yield loss.

There is also a possibility of the occurrence of a rare and an interesting case; it is possible that either the pull-up network or the pull-down network has a high resistance slow outlier and the corresponding tuning transistor, that would have to be turned ‘ON’ to bring down

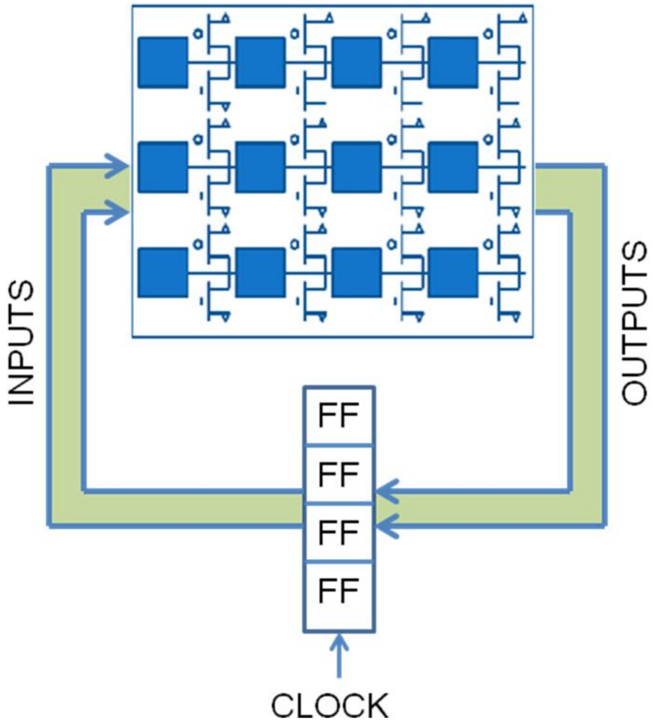

Figure 4.2: Hoffman model of Sequential circuits with tunable gates

the gate delay, be a low resistance low Vth outlier. In such a case tuning would not be possible because of the possible degradation of the voltage logic. This occurrence again is very rare and is considered a yield loss. More is discussed on this case later.

There are drawbacks to adding tuning transistors. One of the most significant being the additional parasitic capacitances that are added that push path delays up. But it will be seen in subsequent sections of this work that on tuning the slow outliers, the worst case path delays of the fabricated chip can be pulled well below the worst case path delays of a fabricated circuit that has no tuning capability in it.

Also, turning 'ON' a particular tuning transistor to speed up the slow transition would slow down the complementary transition. But it will be seen that the slowdown is much less than the speedup.

There are other important effects of including tuning transistors, a few being additional power dissipation and increase in the amount of area required. It will be seen that these effects are minimal and within affordable limits of adding tuning circuitry. More is discussed in the subsequent chapters.

Such a tuning architecture has been studied before by Ashouei et al. [5] but only as a defect tolerance methodology. In the presence of a fault, say in the pull-up network, the entire pull-up network is disconnected from the power rail and is replaced by a properly sized single always ‘ON’ pull-up transistor, hence converting the CMOS gate to a pseudo NMOS structure. A similar technique is adopted to replace a faulty pull-down network with an always ‘ON’ pull-down transistor. This method is similar to the one proposed in this work, with the difference that no additional circuitry is required to disconnect a network. Such a defect tolerant methodology, however, has not been applied for performance tuning. As will be seen, the potential exists to combine both defect tolerance and performance tuning in aggressive CMOS technologies.

4.2 Tuning strategy

As discussed in previous chapters, statistically there is every chance that a large design will have at least a few hundreds of outlier transistors that lie far away in the distribution, whose single presence on a path pushes path delays up by a significant amount. The slow transistor could either be in the pull-up network or the pull-down network; chances of both the pull-up and the pull-down network having a slow transistor are very small, in which case the chip would count as faulty and would fall under yield loss.

The presence of a slow transistor in the pull-up network would push up the rising delay as it would take longer for the output capacitance to charge through it. To counter the slow transistor, the tuning PMOS connected in parallel is turned ‘ON’ to provide a separate path for charging the output capacitance, as shown in Figure 4.3. The red dot in the pull-up network indicates the presence of a slow PMOS transistor.

In the same way, the presence of a slow transistor in the pull-down network would push up the falling delays as it would take longer for the output capacitance to discharge through

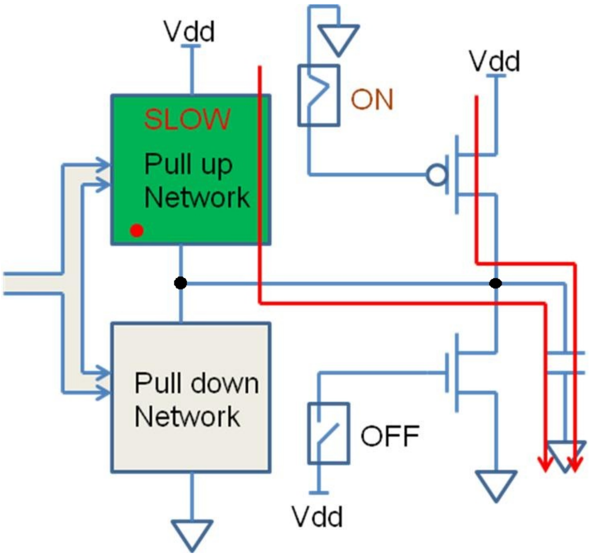

Figure 4.3: Rising transition speed-up

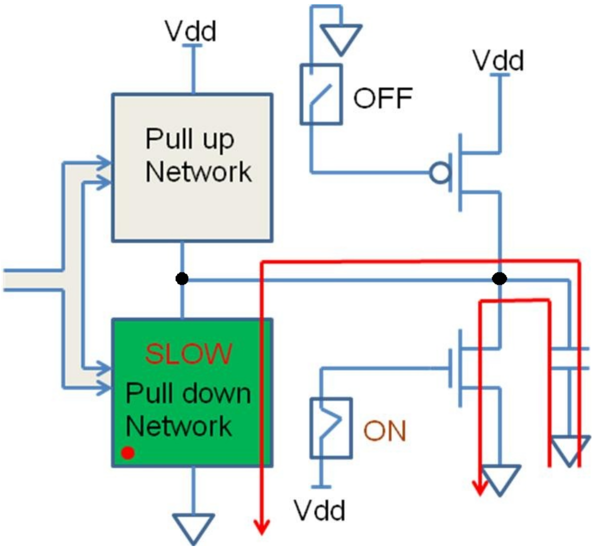

Figure 4.4: Falling transition speed-up

it. Their presence is countered by turning 'ON' the NMOS tuning transistor which would provide an additional path for discharge, as shown in Figure 4.4.

It is assumed here that an algorithm exists to diagnose the slow transistors in the design. Once diagnosed, the appropriate tuning transistors are turned 'ON' to bring down the gate delays.

4.3 Sizing of tuning transistors

Sizing of tuning transistors is an important factor determining the effectiveness of the tuning strategy. The tuning transistors should be sized large to provide good speed up, but at the same time be sized small enough to ensure that logic levels are maintained.

For instance assume the pull-up network to be slow. To speed up rise time the tunable PMOS transistor is turned 'ON'. Now, when the pull down network is activated, the circuit would behave like a voltage division network (Figure 4.5), with the output voltage being supply voltage times the fraction of pull down resistance to the total resistance. To ensure that this output voltage stays within bounds of logic '0', the resistance of the tuning PMOS should be high, or the W:L ratio of the transistor should be small.

A similar voltage division circuit is set up when an outlier transistor exists in the pull-down network and the NMOS tuning transistor is turned 'ON' while the pull-up network is activated.

Hence, a trade off between a large ratio and a small ratio is required for the strategy to work correctly. For the purpose of simulations in this work, a 4:1 resistance ratio between the tuning transistor and the corresponding complementary network is maintained. This ensures a LOW voltage of Vdd/5, and a HIGH voltage of 4Vdd/5, which are within acceptable ranges of logic '0' and logic '1', respectively.

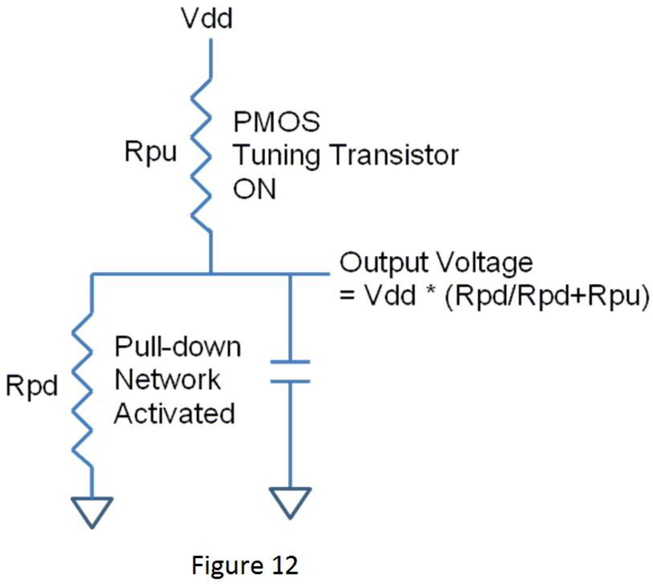

Figure 4.5: Voltage division on tuned nodes

4.4 Faulty tuning transistors

It is important to note that the tuning circuitry comprising the tuning transistors would also be exposed to process variability, as it is fabricated along with the other functional transistors of the chip. Hence, it could itself be a victim of process variability. A faulty tuning transistor would be of relevance only if it is required during the tuning process, otherwise it is of no importance.

There is a difference though in how a defective tuning transistor is perceived. As mentioned earlier, process variability affects Vth distribution. In other words the Vth of the transistor could be high, corresponding to a slow, high resistance, low leakage transistor, or the Vth could be low, corresponding to a fast, low resistance, high leakage transistor.

For a transistor in the pull-up network or the pull-down network, a high Vth would imply a faulty slow transistor as it takes longer to switch; a low Vth transistor would be fast to switch and hence not be an outlier candidate. But for a tuning transistor, a high Vth would correspond to a high resistance transistor, and even though slow, the effective

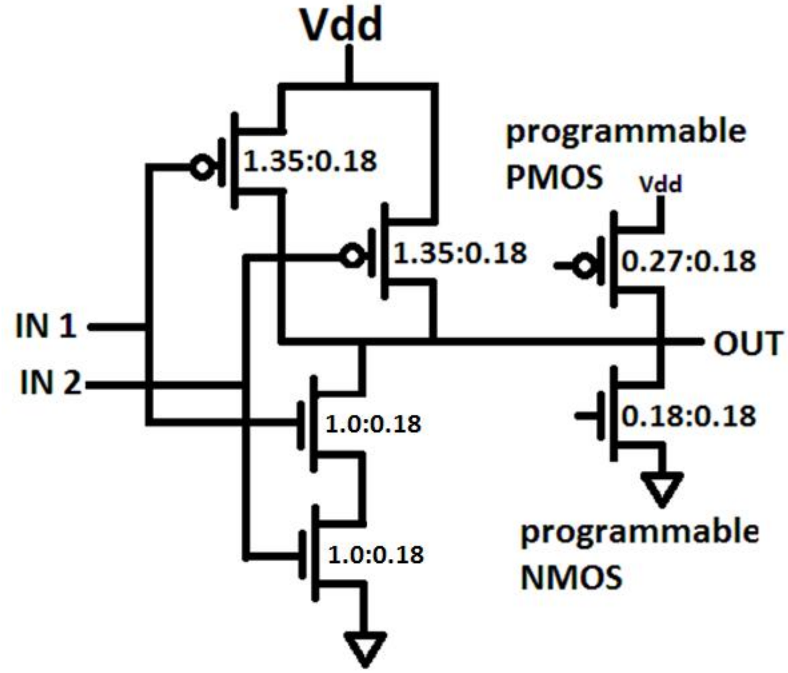

Figure 4.6: Tunable NAND gate

resistance of the slow outlier and the tuning transistor would be lower and hence some speed up will be achieved.

On the other hand, a low Vth tuning transistor, even though faster to switch, could degrade the output voltage logic so much that the output doesn't switch anymore. Such an occurrence would make it impossible to tune the outlier transistor to avoid voltage logic degradation. Hence, for the case of tuning transistors, it is the low Vth transistor that is considered defective and not the high Vth transistor. This occurrence is again not common, and in the chance that it is to occur, the fabricated chip would be discarded as a yield loss.

4.5 NAND gate simulations

To test the effectiveness of the tuning strategy, simulations were performed on a single NAND gate with input and output loads. Mentor Graphics tools were made use of to generate the NAND gate SPICE circuit with input and output loads. A gate level structural netlist was first created as a verilog file. This verilog file was fed in as an input to Design

Architect to create design view points. The design view points created in Design Architect are used in IC Station to generate the circuit layout. Once the layout was generated, it was verified for overflows and shorts, after which a layout versus schematic check was done using the LVS tool. Capacitances were finally extracted through Calibre PEX.

Once the final SPICE netlist was generated, tuning transistors connected to the NAND gate were added. This addition had to be done manually as the standard cell libraries do not have tunable gates in them. The width over length ratios for the tuning transistors were fixed so as to have a 4:1 resistance ratio between the tuning transistor and the complementary network of the gate. Additional capacitance was added to the gate outputs to include the effects of the tuning transistors. More is discussed on the SPICE netlist extraction in the following chapter.

The threshold voltages of the transistors were assumed to be nominal first, and the rise and fall time delays were noted. Another set of simulations were performed on the same NAND gate, but with one pull-up PMOS transistor having a threshold voltage 0.1V above nominal; rise and fall time delays were noted before and after tuning. The same simulations were repeated for Vth of PMOS being 0.2V above nominal, and 0.3V above nominal. Figure 4.6 shows the tunable NAND gate structure, and TABLE 4.1 provides the results of the above simulations.

180nm Technology					
Vth	Rising Delay		Falling Delay		Delay change
(PMOS)	Untuned	Tuned	Untuned	Tuned	(%)
0.4	113	89	130	142	-25.67
0.5	174	125	129	140	19.54
0.6	305	188	128	139	38.36
0.7	703	296	127	138	57.89

Table 4.1: Tunable NAND gate simulations

The transistor W:L ratios in the NAND gate are the standard ratios defined by the cell libraries. The W:L ratios of the tunable gates were fixed manually to ensure a 4:1 ratio between the tunable PMOS and pull-down, and tunable NMOS and pull-up resistances. As can be seen the performance gain for Vth = 0.4V is negative, as the complementary delays are pushed up, but sufficient gains are obtained as Vth of PMOS increases; for Vth = 0.7V the reduction in gate delay is close to 58%.

In conclusion, the methodology seems to be very effective in pulling back gate delays by a significant amount. Its effectiveness on a large circuit with long chains having multiple gates needs to be tested though.

The steps taken in generating the SPICE netlist for the purpose of simulations are discussed in the next chapter.

Chapter 5

SPICE Netlist Extraction

Extracting the SPICE netlist for the purpose of simulations is a very critical part of the work done here. A description of the tools used and the manual additions implemented are discussed in this chapter. Mentor Graphics tools consisting of ModelSim, Leonardo Spectrum, Design Architect, IC Station, LVS and PEX, have been used for extracting the transistor level SPICE netlist.

The circuits studied in this work include basic inverter chains, NAND gate chains and EXORtree circuits. Any larger circuit constructed, is constructed out of these simple circuits. The reason for studying these simple circuits is explained later.

The chapter starts with a brief description of the tools in the order in which they were used, followed by a description of the method used to include the tuning transistors.

5.1 Mentor Graphics Tools

5.1.1 ModelSim

The behavioral model of the circuit is first written either in VHDL or Verilog and then simulated in ModelSim. This VHDL/Verilog description forms the seed from which ultimately the transistor netilst is extracted.

Process variability, in the context of this work, was studied on basic inverter chains, NAND gate chains and EXORtree circuits. As the structural description of these circuits is known, a gate level structural description was written in Verilog. The behavioral description is not provided to ensure that no optimization is done on the circuit.

5.1.2 Leonardo Spectrum

Once the behavioral/structural description of the circuit is created in VHDL/Verilog, Leonardo Spectrum is used to generate the structural description of the circuit in Verilog format with the basic details of the gate delays specified through the .sdf file. It is to be ensured that no optimization is done while synthesizing an input structural netlist to generate a Verilog gate level structural description; optimization if done could change the gates used and a few inter-connections without of course changing the output functionality.

5.1.3 Design Architect

The next step in the extraction process is to load the Verilog file generated by Leonardo Spectrum into Design Architect to generate a schematic of the circuit. While generating the schematic it will be seen that standard cells with fixed width and length ratios, for a given drive, are used to construct the circuit. It is not possible to change the width or length information of the standard cells. This inability is the reason why tuning transistors are to be included manually at the end of the extraction process. More on this is discussed later. Design Architect also generates design view points which are used by IC Station in extracting the layout information.

5.1.4 IC Station

The next step is to open the schematic in IC Station to generate the layout. While generating the layout all overflows are to be connected, all shorts removed, and input output pins added. This would ensure that the design verifications in IC Station are carried out correctly.

5.1.5 LVS

The LVS tool is used next in performing the layout versus schematic check. This is done by comparing transistor dimensions and the interconnects of the transistors of the layout, with the transistor dimensions and interconnects of the transistors of the schematic.

5.1.6 PEX

The last step in the SPICE extraction of the functional transistors is carried out through PEX. Capacitances are extracted by the tools, that makes use of lookup tables in the technology file package. A final transistor level SPICE netlist with these capacitances extracted is generated. It is this SPICE netlist that is used in HSPICE for simulation.

5.2 Tuning transistors

So far, tuning transistors have not been included in the design. There are difficulties associated with including tuning transistors in the design. The most important being that standard cells do not come with the tuning capability. A number of different techniques were studied to implement the automated addition of tuning transistors.

As can be seen from Figure 5.1 above, tuning transistors together closely resemble the architecture of an inverter, with the difference that the gate terminals are connected to separate voltage sources instead of being connected together to an input voltage source, and input and output nodes are connected together. With this in mind the following architectures were implemented.

Figure 5.2 shows the close resemblance of an inverter connected to a gate output with the actual tuning architecture. It was proposed that every gate be connected to an additional inverter and the SPICE netlist extraction be done. Once the SPICE netlist was extracted, one would manually alter the connections between the inverter and the gate to transform the inverter architecture to a tuning architecture.

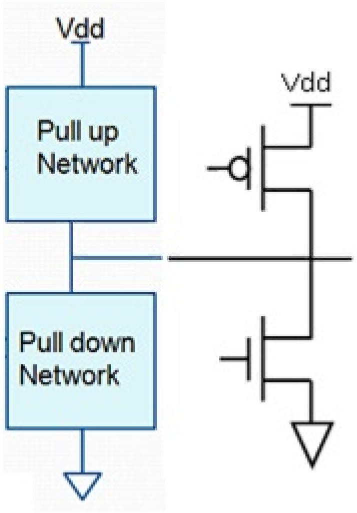

Figure 5.1: Tuning circuit

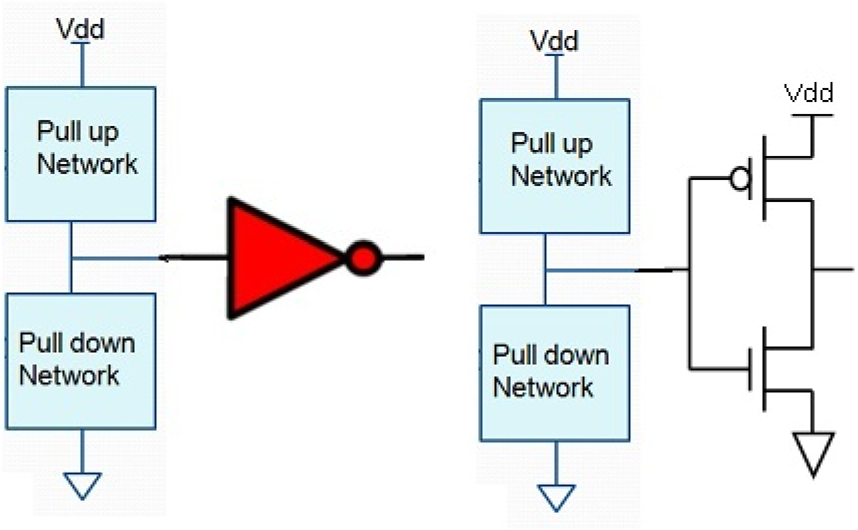

Figure 5.2: Inverter as a tuning circuit

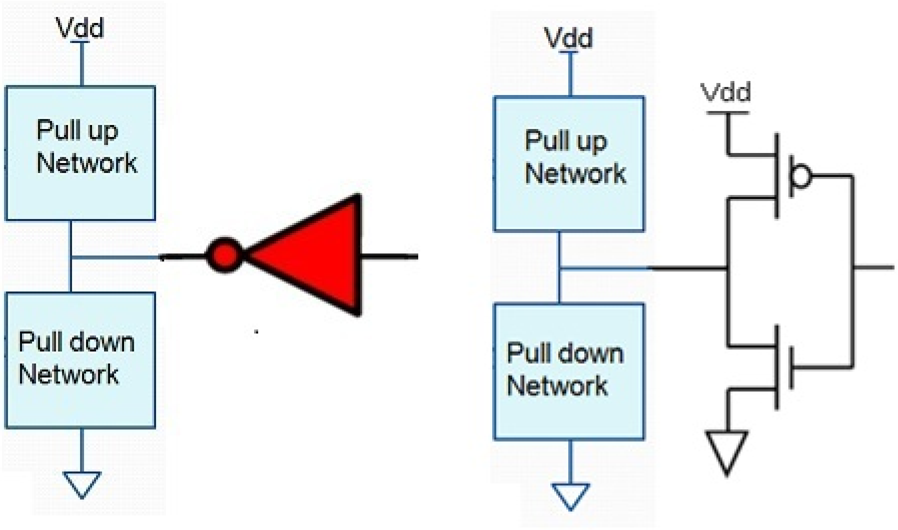

Figure 5.3: Inverted inverter as a tuning circuit

The problem associated with this architecture is that additional capacitance is added at the gate output nodes from the gate terminals of the tuning transistors. It is only the diffusion capacitance of the tuning transistors that is to be added. To overcome this problem the architecture shown in Figure 5.3 was proposed.

Figure 5.3 better resembles the tuning circuitry because of the direct addition of the diffusion capacitance of the transistors in the inverters with the node output capacitance. It was proposed that the inverter input node and its capacitance be removed once the SPICE netlist extraction was complete.

But this architecture could also not be implemented. As it was later found, while generating the schematic, inverter transistor width and length values could not be altered; only the standard cell values of width and lengths would be interpreted by IC Station in generating the transistor layout. It is important that the transistor dimensions of the tuning circuitry be altered to have a 4:1 resistance ratio between it and the corresponding complementary network in the node, for good speed up to be achieved without disturbing the voltage logic.

As a result it was decided that all tuning transistors be included manually in the SPICE netlist after the functional netlist was extracted from the various design tools described earlier. The transistor dimensions were also manually fixed to the required ratio. Additional parasitic capacitance that would have to be included to compensate for the tuning transistors, is also added manually to every node.

The value of this additional capacitance at every node is determined by extracting the diffusion capacitance of transistors in an inverter. The diffusion capacitance here happens to be the gate output capacitance of the inverter. But this capacitance extracted corresponds to the standard cell dimensions of the inverter whose L=180nm and W=450nm for the NMOS and W=990nm for the PMOS. To scale the capacitance, the ratio by which the widths of the transistors scale, is calculated. Width for the PMOS was scaled to 270nm and for NMOS was scaled to 180nm. Hence on an average the dimensions were scaled by a factor of 3. The inverter output capacitance, which happens to be the diffusion capacitance, is then divided by 3, and this value added to every node capacitance.

This strategy of adding tuning transistors was possible in the case of this work as simple circuits entirely constructed using a single gate, such as NAND gate or a NOT gate, were studied. For complex circuits with a mix of several gates, the above strategy would fail as it would be very time consuming to manually fix transistor dimensions of every tuning transistor connected to different types of gates; these dimensions would vary from one gate type to another. Future studies involves incorporating tunable gates into the standard cell libraries to make this automation possible.

Once the SPICE netlist extraction was complete, the SPICE file was simulated using HSPICE. The next chapter studies the algorithm implemented to study process variability on very large circuits.

Chapter 6

Tuning implemented on very large circuits

The preceding sections indicate the potential for improving performances of circuits through tuning. However, to truly identify the effectiveness of such a strategy, it is necessary to create an environment where the circuit has several million transistors. This chapter presents the work in creating such an environment through clever innovative techniques, and then simulating every circuit before and after tuning to calculate worst case delays and measuring performance improvement.

6.1 Building large circuits

Simulating large circuits is necessary because it is in such large circuits that transistors are exposed to process variability. In a small circuit, because of the small number of transistors involved, statistical outliers will not be found to occur in most cases. Chips fabricated today have a few hundred outlier transistors in them because of the close to billion transistors that are there in the chip.

6.1.1 Difficulties

There are a number of problems associated with simulating large circuits. First, every node in the circuit has to have the tuning capability, which means adding a PMOS and an NMOS transistor manually to every node. This has to be done manually because the library files that are used in generating circuit layouts and node capacitances, do not have gates with tuning capabilities in them. Hence doing the addition manually in a circuit that has several thousands of gates at least, would be very tedious.

Secondly, every circuit has to be simulated for different nominal threshold voltages and standard deviations, and different supply voltages, each be repeated at least a 1000 times to get an average case out.

It is also necessary to simulate a circuit with different sizes to get a sense of an approximate size beyond which there would be sufficient performance benefit to make this tuning technique applicable. A single HSPICE simulation for a large standard circuit takes several hours even on a fast computer. Such simulations are hence not very practical.

6.1.2 Basic algorithm

To get around this problem of being unable to simulate large circuits, an innovative algorithm is adopted. First a standard small circuit with less than a 100 transistors is simulated, with Vth for every transistor in the circuit dawn from a normal distribution. The same circuit is again simulated, but with different transistor threshold voltages, again drawn randomly from a normal distribution. This process of drawing Vth and simulating is done 20,000 times, and for each simulation the delays of the circuit are calculated.

Now, to construct a larger circuit from these smaller sub-circuits, a random ‘N’ number of sub-circuits are drawn from the collection of 20,000 circuits, and the largest delays of the ‘N’ sub-circuits selected determine the delay of the larger circuit. ‘N’ represents the number of small sub-circuits selected at a time to create the larger circuit. ‘N’ is allowed to vary from 10 to 10,000. The random pick of ‘N’ sub-circuits is done 1000 times each to get an average case for every size ‘N’. In this way large circuits can be constructed from smaller ones, and their delays calculated without having to simulate them.

6.1.3 Standard circuit used to create larger circuits

The standard small circuit picked in this work is an EXOR tree with 8 inputs and 1 output. The tree has 7 EXOR gates, each constructed with 4 NAND gates. Every NAND gate has the tuning capability added to it. There are 8 paths to the output, and each path has 6 active NAND gates on it.

The advantage of using an EXOR tree is that it is very testable, which would make the diagnosis easier, and a single bit change in the input is reflected at the output. This property is made use of in activating a single path individually, by sending out pulses through every input, one input at a time, and every pulse sent with enough spacing to allow the output to settle even in the presence of a slow outlier, before being pulsed at the next input to activate the next path.

Figure 6.1A is a figure of an EXOR gate constructed using NAND gates. The NAND gates here are all tunable. Figure 6.1B is a figure of an 8 input EXOR tree. Figure 6.1C represents a large circuit constructed out of 'N' EXOR trees, the number 'N' varying from 10 to 10,000.

6.2 Simulation

This section details the logic used while programming in MATLAB and PERL to create the environment described in the preceding section. The EXOR tree circuit was first constructed and synthesized as a SPICE file. This file was labeled 'without_tuning_circuitry'. Another EXOR tree SPICE file labeled 'with_tuning_circuitry' was created with tuning transistors connected to every NAND gate output. Two files were created so that a comparison could be drawn between the delays of the original circuit and the delays of the tuned circuit.

To start simulations on HSPICE, however, the primitive model ('.pm') files that would be needed to run with the SPICE netlists need to be created. The standard '.pm' file that is made available in the package cannot be used as it details the parameters of a single transistor. It is, however, used as a seed to construct the '.pm' files required for the EXOR tree simulations. The following subsection details the method used to generate primitive model files that incorporate variability.

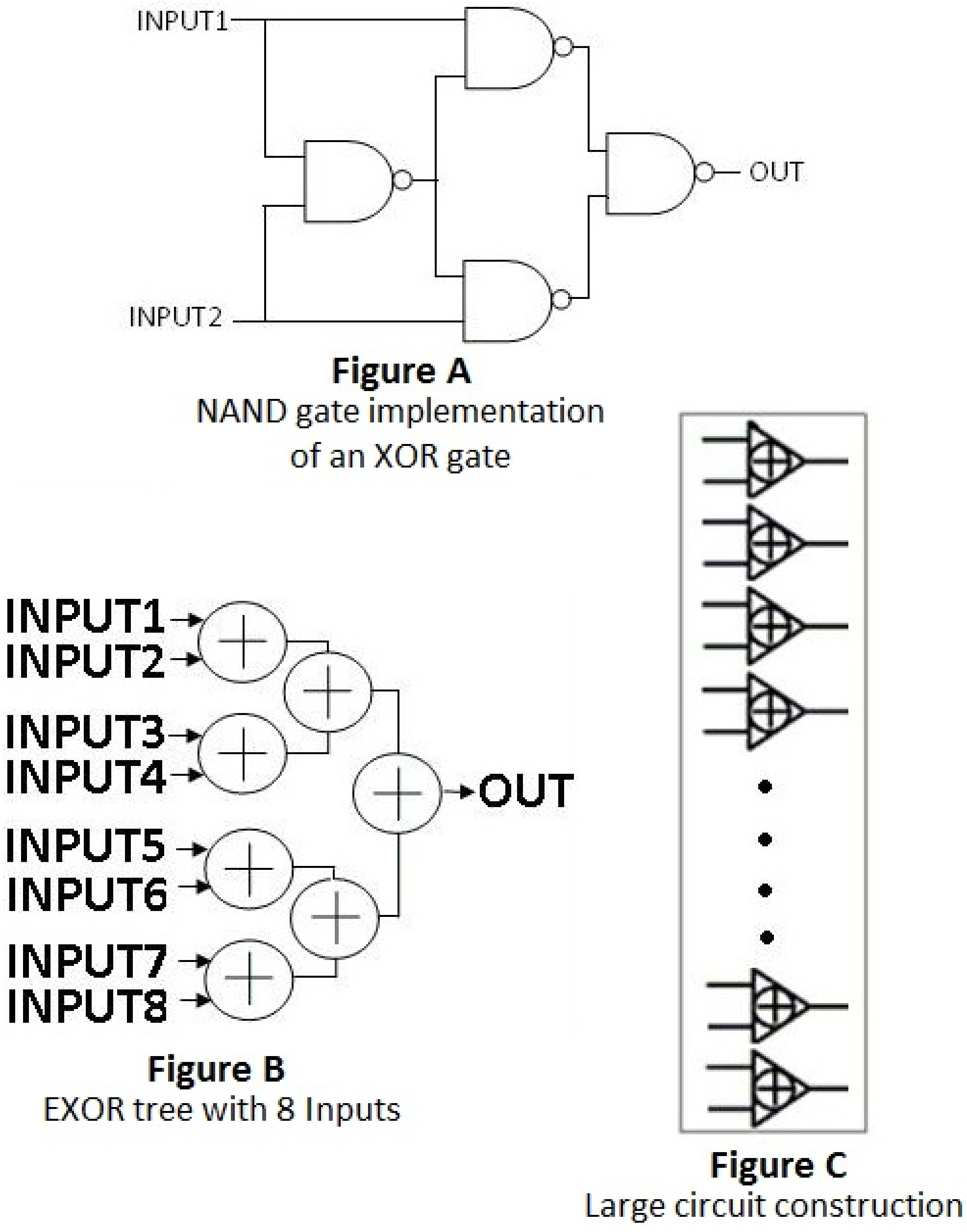

Figure A
NAND gate implementation of an XOR gate

Figure B
EXOR tree with 8 Inputs

Figure C
Large circuit construction

Figure 6.1: EXOR tree circuit

6.2.1 Primitive model files

As mentioned, the '.pm' files are included in the SPICE netlists before being simulated. To ensure that every transistor has Vth drawn from a normal distribution, and is hence different from every other transistor in the circuit, the '.pm' file for an EXOR tree circuit would have instances of the standard single transistor '.pm' file for every transistor in the circuit, with Vth of every instance different from the other.

In the 8 input EXOR tree circuit with tuning circuitry included, there are 84 PMOS transistors and 84 NMOS transistors. Hence, there will be 84 different descriptions of PMOS and 84 different descriptions of NMOS in a single '.pm' file to be included with the circuit SPICE netlist; these descriptions only vary in the value of Vth.

It was also mentioned earlier, in the section on the basic algorithm implemented to create large circuits, that 20,000 copies of the EXOR tree are to be simulated. All of these 20,000 copies will also have transistors that vary in Vth from every other transistor's Vth in any of the 20,000 copies. Hence, a MATLAB program to generate 20,000 * 84 (= 1,680,000) values of PMOS-Vth and 20,000 * 84 (= 1,680,000) values of NMOS-Vth was written that would extract the 1,680,000 Vth values from the standard normal distribution for a specific μ (mean) and σ (standard deviation).

These values are then assigned one by one to a '.pm' file. Every time 84 transistor instances have been written, a new '.pm' file is created to have the next 84 set of instances. In this way 20,000 different '.pm' files, with Vth for every transistor drawn from a nominal distribution was created, and stored in a specific folder, each file numbered sequentially from 1 to 20,000. The entire process of creating the 20,000 '.pm' files was repeated for different σ values.

This process of generating '.pm' files was also done for different values of Vdd. The reason being that, while drawing Vth from a distribution, there is a possibility of the value being greater than Vdd or less than zero. This would not be correct and needs to be avoided. In other words bounds are set for the Vth pick, depending on the value of Vdd.

The Vdd values chosen for simulation are 1.2V and 1.5V. For Vdd=1.2V, the lower bound was fixed at 0.1V and the upper bound at 1.0V. While for Vdd=1.5V, the lower bound was at 0.1V and the upper bound at 1.3V. Hence for every combination of Vdd and σ, '.pm' files were generated. All in all there were 4 sets of files, each with 20,000 '.pm' files in them, for the cases Vdd=1.2V σ=0.075V, Vdd=1.2V σ=0.150V, Vdd=1.5V σ=0.075V, and Vdd=1.5V σ=0.150V.

It is to be noted that the circuit labeled 'without_tuning_circuitry' would also use the same '.pm' files in spite of having fewer transistors; this circuit would have 56 PMOS transistors and 56 NMOS transistors. To ensure that transistors belonging to a gate in the two circuits labeled 'without_tuning_circuitry' and 'with_tuning_circuitry' are the same, the tuning transistors are only labeled from 57 to 84. This is important because if an outlier is to exist, it is to fall at the same location in the two circuits. Only then can a comparison be made between the delays of the two circuits.

6.2.2 Single EXOR tree simulations

The SPICE netlist files are now ready for simulation. It is to be noted that even though there are 20,000 different '.pm' files for a specific σ, there is only one copy of the SPICE netlist file; one for 'without_tuning_circuitry' and one for 'with_tuning_circuitry'. Only the '.pm' file that the SPICE file points to is modified every time.

A MATLAB program was written to open a SPICE netlist file, modify the '.pm' file to be pointed to, run the SPICE file on HSPICE, and extract the path delay information to be stored in a file; HSPICE was invoked from the MATLAB program. The path delay information was stored in a file labeled with the same number the '.pm' file was labeled with. This file had information on the delays of every path and also the worst case delay, found by calculating the largest delay in the simulation, along with the worst case path number and transition type. The worst case delays were also stored in a matrix in MATLAB to speed up its use by another MATLAB program.

The SPICE file had commands in it to extract every rising and falling transition delay information. It also has the capability to extract the power dissipated, but this command is made use of only later while simulating for power.

In this way 20,000 different delay files for the 20,000 copies were created and stored in a specific folder, along with the worst delays also being stored in a matrix in MATLAB (the matrix size was 20000 x 1). This process of simulating was repeated for different Vdd and σ values.

All in all there were 4 cases :

CASE 1 Vdd = 1.2V σ = 0.075V

CASE 2 Vdd = 1.2V σ = 0.150V

CASE 3 Vdd = 1.5V σ = 0.075V

CASE 4 Vdd = 1.5V σ = 0.150V

For all of the above cases, there were 2 basic simulations done on each of the 20,000 copies - one on ‘without_tuning_circuitry’ and the other on ‘with_tuning_circuitry’. In the process two different worst case delay matrices were also constructed. It is to be noted that no tuning has been performed yet.

6.2.3 Tuning

Once all simulations were done for both ‘without_tuning_circuitry’ and ‘with_tuning_circuitry’, and the worst case delays calculated and stored in a matrix, tuning was implemented. Only those circuits which have outlier transistors in them need to be tuned.

To pick outliers, a delay cut-off was established, and all worst case delays larger than the cut-off were declared outliers. This cut-off was selected so as to have between a 100 and 150 outlier cases out of the 20,000 circuits simulated for every case.

Once the cut-off was established, a search is performed through a MATLAB program to extract those circuit numbers that have their worst case delays greater than the cut-off.

The delay file for an outlier circuit is then opened and the value of the worst case delay, along with the path number of the largest delay and the transition type is extracted.

With this information in hand, the corresponding '.pm' file is opened, and a search is performed on the largest Vth values in the file. This search was done through a PERL script that would extract every line one-by-one from the '.pm' file and search for the patern 'Vth0' in the file. The PERL script is invoked from MATLAB, and hence forms a part of the same MATLAB code.

This search of Vth is restricted only to the transistors that fall on the critical path, and only those that are involved in the slowest transition. This is done to speed up the search. Details on which transistor lies on which path, for a particular transition, was predetermined and stored in the MATLAB code.

Since a comparison needs to be done on a Vth belonging to a PMOS and a Vth belonging to an NMOS, the average Vth of the NMOS transistors in a particular gate is compared with the Vth of a PMOS transistor. This is done because, in NAND gates, the NMOS transistors are connected in series, while the PMOS transistors are connected in parallel. Hence for a falling transition to occur, the discharge would happen through the series connection of the NMOS transistors; slowest charging would occur only through a single PMOS transistor.

Hence, Vth values of a PMOS transistor and average Vth values of the two NMOS transistors in a particular gate were calculated and extracted by the MATLAB program, for every gate in the critical path. Gates on this critical path are then sorted in descending order of Vth.

To tune the circuit, the tuning transistor connected to the gate with the highest Vth value, is to be turned 'ON'. If the the outlier transistor happens to be a PMOS transistor, the PMOS tuning transistor of that gate is turned 'ON'. On the other hand, if the outlier transistor happens to be an NMOS transistor, the NMOS tuning transistor of that gate is turned 'ON'.

The tuning transistors in the SPICE file are labeled with the gate number the tuning transistor belongs to. This makes the search for the tuning transistors connected to the gate under study easier.

Once the tuning transistor number is established, the voltage source connected to the gate terminal of the tuning transistor is determined. Here again, the voltage source is labeled conveniently to make its search possible. For an NMOS tuning transistor, the voltage source is initially connected to the GND (logic '0'). To enable tuning, the value of the voltage is switched to Vdd (logic '1'). In the case of a PMOS tuning transistor, the voltage source which was initially connected to Vdd (logic '1'), is switched to GND (logic '0') to enable tuning.

Once the tuning voltage source values are modified in the SPICE file, the MATLAB program invokes HSPICE to simulate the SPICE file. The delays again are calculated, and the worst case delay is noted. If the worst case delay falls below the cut-off, the circuit is said to be tuned, and the next outlier circuit is picked for tuning. If not, the second gate in the sorted list of gates on the critical path is tuned. Again, HSPICE is invoked to simulate the circuit, and path delays calculated and extracted by the program. If worst case delay stays above the cut-off again, the next gate in the sorted list is tuned. This process of picking a gate for tuning is carried on until tuning is achieved. If all single gate tuning cases are exhausted, the program would check for multiple gate tuning to see if tuning multiple gates on the critical path brings the worst case delays down.

In most cases, speedup was achieved by tuning the first gate in the sorted list of gates on the critical path. For the case Vdd = 1.2V $\sigma = 0.150$V, out of the 120 cases tuned, 112 cases were tuned by the tuning the first gate in the sorted list. For 7 circuits, tuning was achieved by tuning 2 gates in the critical path. 1 circuit could not be tuned; this is because of the occurrence of the special case. Table 6.1 gives the number of gates tuned for each of the cases studied.

CASE	Number of circuits tuned
Vdd = 1.2V $\sigma = 0.075V$	136
Vdd = 1.2V $\sigma = 0.150V$	120
Vdd = 1.5V $\sigma = 0.075V$	144
Vdd = 1.5V $\sigma = 0.150V$	113

Table 6.1: Number of circuits tuned

Once tuning is done, a final third matrix with tuned delay values is constructed and stored in MATLAB. Together the three matrices are used in measuring performance benefit. The motive behind storing values in a matrix is to provide faster accessibility of these data to other MATLAB programs.

It is to be noted that once an outlier path in a circuit is tuned, there is no possibility of another path becoming an outlier as a result of tuning. It is true that the complementary transition is slowed down, but this slow down would never be so large to become an outlier. Only outlier cases, with path delays significantly larger than average case delays, are being tuned.

6.2.4 EXOR tree simulation: example

Table 6.2 represents one EXOR tree simulation for the case Vdd = 1.2 and σ=0.150V. The table shows Rising Input 4 to have the largest delay (7.93ns) before tuning. The increase in the delays after adding tuning transistors is attributed to the parasitic capacitances introduced by the additional transistors. This critical path delay is brought down considerably on tuning to 1.786ns - a 77.5% reduction in the path delay, which is very significant. The example demonstrated here is that of an extreme outlier. On an average the benefits on single EXOR tree outlier circuits obtained were less. Table 6.3 gives the average single EXOR tree circuit outlier benefits.

It is to be noted that the entire process of finding outlier circuits, extracting details of the slowest path delays, sorting gates along a path in descending order of Vth, modifying

Input	Transition	Without any tuning circuitry (ns)	Untuned delays (ns)	Tuned delays (ns)
INPUT 1	RISING	1.301	1.373	1.377
	FALLING	1.222	1.295	1.295
INPUT 2	RISING	1.030	1.092	1.089
	FALLING	1.295	1.371	1.371
INPUT 3	RISING	1.499	1.161	1.079
	FALLING	0.9493	1.005	1.030
INPUT 4	RISING	7.930	8.323	1.716
	FALLING	0.9136	0.9644	0.9996
INPUT 5	RISING	0.8768	0.9277	0.9302
	FALLING	0.8978	0.9489	0.9482
INPUT 6	RISING	0.8381	0.878	0.8881
	FALLING	1.689	1.788	1.786
INPUT 7	RISING	0.9795	1.034	1.036
	FALLING	1.148	1.216	1.215
INPUT 8	RISING	1.010	1.068	1.067
	FALLING	1.310	1.384	1.382
		7.930	8.323	1.786

Table 6.2: Tunable EXOR tree simulation

CASE	PATH DELAY IMPROVEMENT
Vdd=1.2V $\sigma = 0.150V$	70%
Vdd=1.2V $\sigma = 0.075V$	15%
Vdd=1.5V $\sigma = 0.150V$	27%
Vdd=1.5V $\sigma = 0.075V$	10%

Table 6.3: Performance gain for a Single EXOR tree

the value of the voltage sources connected to tuning transistors, and invoking HSPICE to simulate the SPICE netlist, repeated over and over again until speedup is achieved, was done entirely by a single MATLAB program.

Another MATLAB program was written to pick a subset of 'N' EXOR trees to create a larger circuit. The 'N' EXOR trees constitute the sub-circuits put together to form the larger circuit. More is discussed on this in the following subsection.

6.2.5 Picking 'N' sub-circuits to create a larger circuits

Once 20,000 simulations were done for every case, on the two circuits labeled 'without_tuning_circuitry' and 'with_tuning_circuitry', and then tuning done on the outlier circuits to bring about speedup, larger circuits were constructed by randomly picking an 'N' number of sub-circuits, and then measuring the largest worst case delay of the 'N' worst case delays. This largest worst case delay would become the worst case delay of the larger circuit.

Picking up 'N' circuits here refers to drawing 'N' worst case delays from the three worst case delay matrices that were constructed earlier for every case. This is done by generating 'N' random numbers in MATLAB, and then extracting the matrix values of that index. Once the three values are extracted, benefit obtained on tuning is calculated.

This process of picking a specific 'N' sub-circuits is repeated a 1000 times to get an average case; doing only a few picks may include only special cases and not cover the average case.

The above process of picking 'N' sub-circuits 1000 times was repeated for different values of 'N' ranging from 10 to 10,000.

The next chapter presents the results of simulation done on large circuits, with plots and explanations on the observed trends of curves in the plots.

Chapter 7

Simulation Results

This chapter presents the results of circuit tuning on large circuits of varying sizes. For every size ‘N’ the pick and delay evaluation was done 1000 times and the average delay for the 1000 cases was calculated.

7.1 Plots

The following plots, Figure7.1, Figure7.2, Figure7.3, Figure7.4, represent variations of path delay versus circuit size for the cases Vdd=1.5V sigma=0.075, Vdd=1.2V sigma=0.075, Vdd=1.5V sigma=0.150, and Vdd=1.2V sigma=0.150 respectively. These path delays are the average path delays of the randomly generated 1000 circuits.

Curves in red correspond to delay versus circuit size variations for circuits without any tuning capability, while curves in blue represent the same for circuits with tuning capability in them but without any tuning done, and curves in green represent the same variations for tuned circuits.

7.2 Observation

It is observed that the blue curves are always above the red curves. This behavior is attributed to the addition of parasitic capacitance to the load capacitance of every gate as a result of adding tuning transistors. With increased capacitance it takes longer for every load capacitance to charge and discharge.

As circuit size gets larger, the chance of the presence of a larger extreme outlier in the circuit gets higher. This behavior explains the exponential increase in the delays of circuits with increasing circuit size, indicated here by the exponential rise of the red and blue curves.

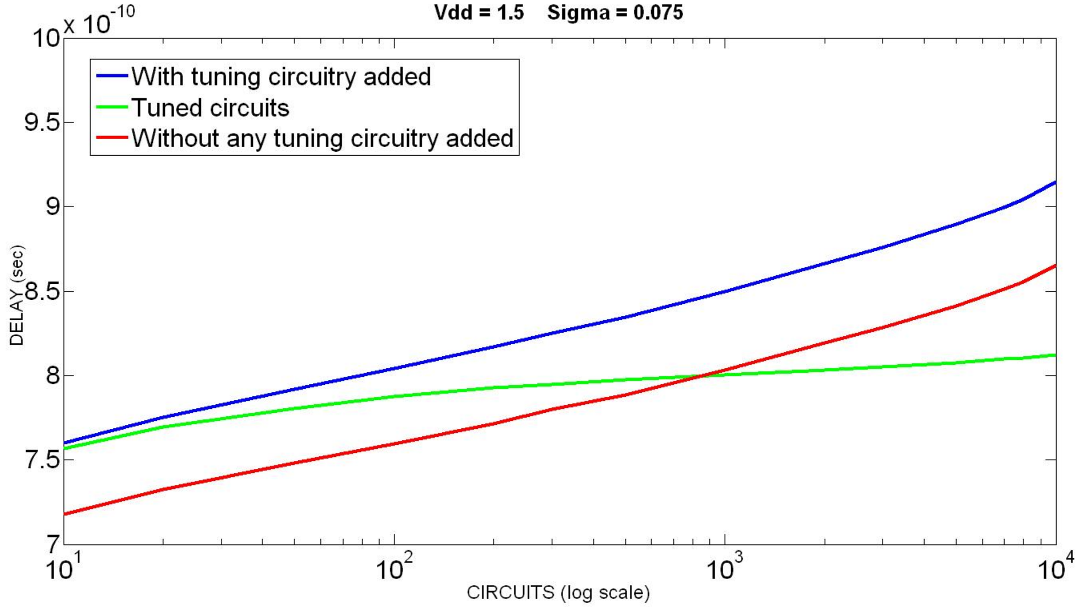

Figure 7.1: Simulation case : Vdd=1.5V sigma=0.075

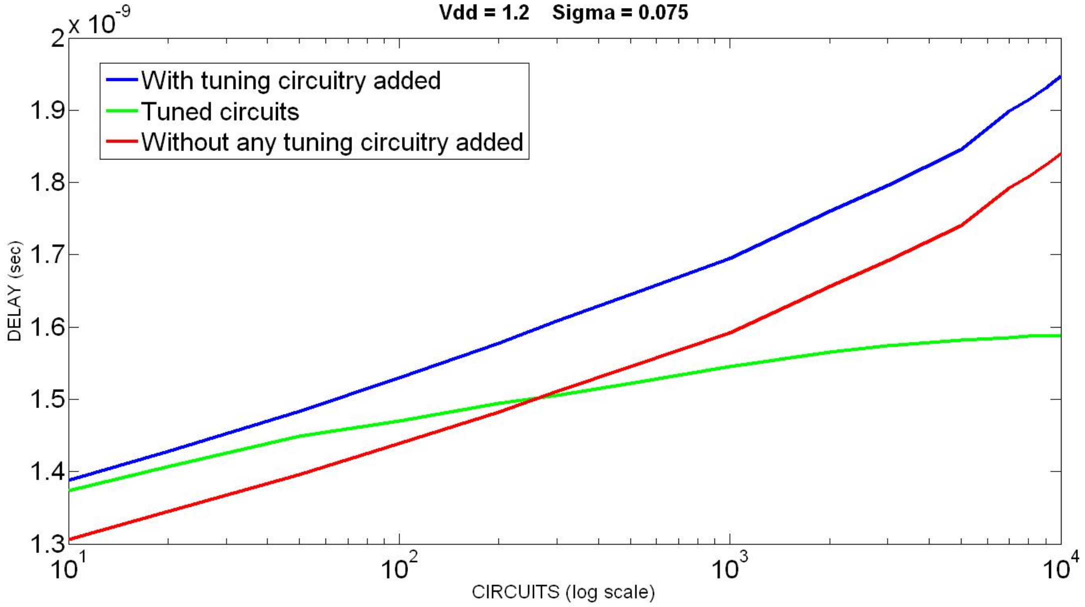

Figure 7.2: Simulation case : Vdd=1.2V sigma=0.075

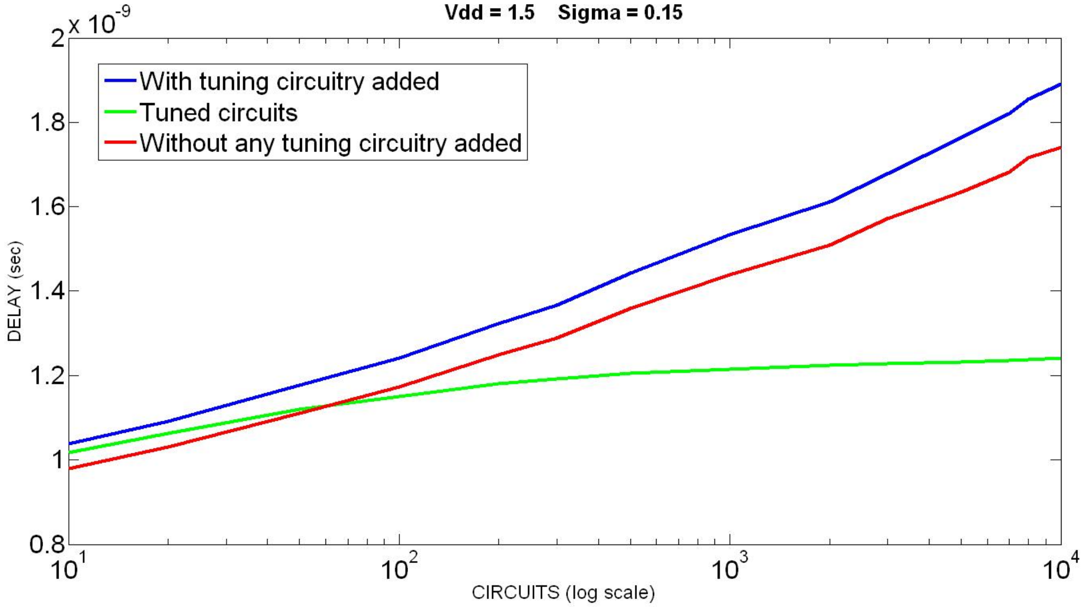

Figure 7.3: Simulation case : Vdd=1.5V sigma=0.150

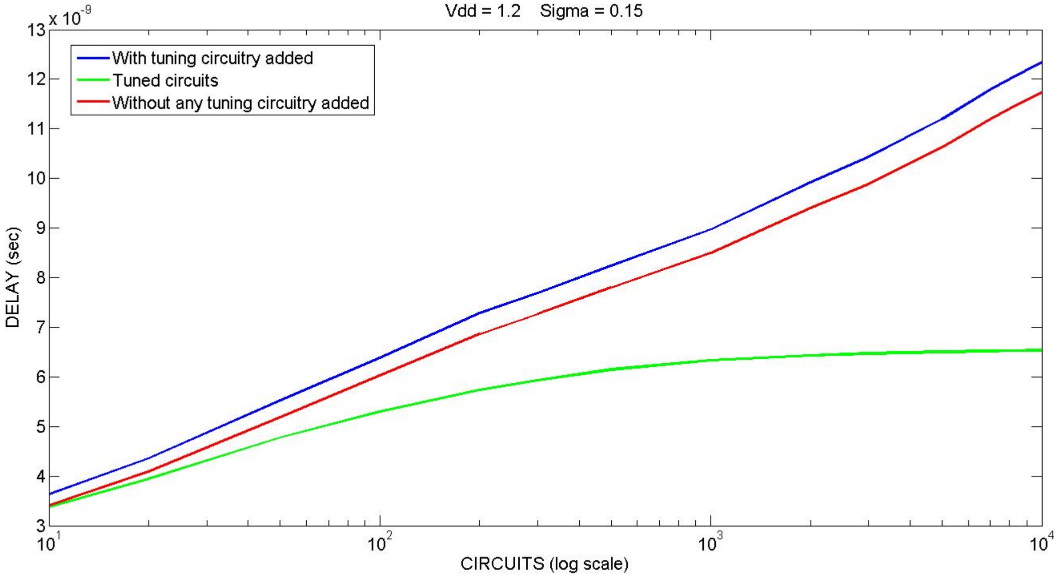

Figure 7.4: Simulation case : Vdd=1.2V sigma=0.150

The aim of tuning is to get the path delays to fall well below the outlier path delay value. As can be seen little or no benefit is obtained in small circuits, as the chance that an outlier will exist is less. This explains the initial behavior of the green curve when it tends to lie above the red curve. But beyond a certain circuit size we observe good returns, as the green curve seems to saturate while the red and the blue curves keep increasing.

Delays of circuits for a σ=0.150V distribution are greater than a σ=0.075V distribution. This is attributed to the fact that larger σ distributions have larger spreads, and as the spread gets wider the chances of picking a slow outlier is greater.

It is also observed that while the green curve appears to be saturating, a slight increase towards the end of the plot may be observed. This is attributed to the fact that a few EXOR trees out of the 20,000 random copies could not be tuned. The reason being the presence of a high resistance slow outlier in a network and a low resistance fast transistor in the tuning circuitry of the same network; an interesting case that was studied earlier. If tuning is done on such a node the voltage logic of the node would be disturbed, and the gate would very often not switch at all. However this special case is very rare and only 1 out of 20,000 cases were found to exhibit this behavior in the Vdd=1V and σ=0.150V case.

7.3 Benefits obtained

Table 7.1 shows the average case path delay improvements obtained for a circuit size of 10,000 sub-circuits under the different simulation environments.

CASE	PATH DELAY REDUCTION
Vdd = 1.2V $\sigma = 0.150V$	59.45%
Vdd = 1.2V $\sigma = 0.075V$	15.82%
Vdd = 1.5V $\sigma = 0.150V$	39.52%
Vdd = 1.5V $\sigma = 0.075V$	6.80%

Table 7.1: Performance gain for a circuit with 10,000 EXOR trees

7.4 Conclusion

Clearly the potential for performance improvement exists. Even though simulations were done in 180nm technology node, the distributions used for the purpose of simulation mimic current and future technology spreads. Hence similar or better benefits can be expected with smaller technology nodes.

The effects of the implemented design on power dissipation are studied in the following chapter.

Chapter 8

Power Dissipated with tuning

The tuning technique implemented provides performance benefit measured in terms of gain in clock frequency. Its impact on the extra power dissipated is studied in this chapter.

8.1 Static power dissipation

In a circuit, once an outlier is identified, the required tuning transistor of that node is turned ‘ON’ and remains ‘ON’ throughout. Hence when the complementary network in the same node is activated to an ‘ON’ state, there exists a direct path for current to flow from Vdd to GND; in other words, static power dissipation exists.

To measure the amount of power dissipated, simulations were done on a circuit size of 10,000 sub-circuits for the case Vdd=1.2V and σ=0.150V, as this case mimics the case impacted by process variability the most.

8.2 Simulation difficulty

While simulating for power it is important to simulate the circuit at a certain clock frequency, as dynamic power is a function of frequency. Most chips fabricated are run at a clock period that provides a 10% buffer to the critical delay in the chip, thus operating at a period 1.1 times the worst case delay.

With respect to the work done here, a circuit of size 10,000 would have to be selected randomly out of the 20,000 sub-circuits, and then simulated again at a clock period that is 1.1 times the worst case delay of that circuit to estimate power. This process of selecting 10,000 random sub-circuits out of 20,000, finding the worst case delay of the pick, then simulating it with a clock period that is 1.1 times the worst case delay, and repeating the process a 1000 times to get an average case out, would effectively require another 10 million (10000*1000) simulations of the EXOR tree sub-circuits, which is very tedious.

8.3 Algorithm

To get around the above problem, another innovative algorithm is adopted. First 1000 circuits of size 10,000 sub-circuits are drawn randomly from the 20,000 sub-circuits, and their worst case delays noted.

These 1000 circuits are then speed binned depending on the value of the worst case delay of the pick. For circuits without any tuning capability, 5 bins were created - those with worst case delays less than 9ns, those with worst case delays between 9ns and 10ns, those with worst case delays between 10ns and 11ns, those with worst case delays between 11ns and 12ns, and those with worst case delays between 12ns and 13ns. Every circuit in a bin can be operated with a clock period which is 1.1 times the largerst bin delay. Hence, the bin labeled less than 9ns would have circuits simulated with a clock period of 9.9ns. Similarly circuits in a bin labeled between 11ns and 12ns would be simulated with a clock period of 13.2ns. In the same way bins were created for circuits with tuning capability added but no tuning done.

Every one of the 20,000 circuits is now simulated again for the different bin clock frequencies. In other words, for circuits without any tuning capability added, the 20,000 EXOR trees would be simulated at clock periods 9.9ns, 11ns, 12.1ns, 13.2ns, and 14.3ns. Average power dissipated is evaluated for every simulation.

To create a circuit with 10,000 sub-circuits and evaluating the power dissipated, 10,000 random sub-circuits are drawn from the 20,000 sub-circuits. The worst case delay of this circuit would determine the bin that the circuit would fall in. Once the bin is identified, total power dissipated for every sub-circuit is obtained from the pre-simulated power dissipated values for that clock period, and then added up together to give the total power dissipated in the circuit.

Tuned circuits were also simulated for different cases. In the first case only 20 out of 20,000 circuits were tuned, and in another case 85 out of 20,000 circuits were tuned. These circuits were operated both at a clock period which was equal to the clock period at which they were operating at before any tuning was done, and at 1.1 times the worst case delay of the circuit after tuning was done.

In this way the number simulations has been cut down from 10 million to less than a hundred thousand.

8.4 Simulation

To carry out the above algorithm, the SPICE files generated to measure delays, are again simulated with references to the '.pm' files, this time, however, to also include power measurement. The same set of '.pm' files used to measure delays are again used here. Power measurement is done by including a SPICE command that measures the instantaneous power dissipated in a device over a time period. The total instantaneous power is then averaged over the clock period to get the average power dissipated. This power dissipation result includes both the dynamic and the leakage power.

8.4.1 Bins

As mentioned earlier, bins were created for simulating for power to reduce the number of simulations required. Table 8.1 shows the bins created for the two circuits. As can be seen bins for 'without_tuning_circuitry' start and end with a smaller clock period than bins for 'with_tuning_circuitry'. This is due to the fact that circuits with tuning circuitry in them would have larger gate delays, because of the presence of parasitic capacitances that are included through the addition of tuning transistors.

An important point to note is that no clock is actually applied to the circuit input. It is the width of the input pulse that represents the clock period. This is due to the fact that a

without_tuning_circuitry	with_tuning_circuitry
8-9ns	9-10ns
9-10ns	10-11ns
10-11ns	11-12ns
11-12ns	12-13ns
12-13ns	13-14ns

Table 8.1: Bins

clock if applied must have a period which is greater than the worst case pulse width required by the circuit to function correctly.

8.4.2 SPICE

Once the bin periods are established, a MATLAB program is written to simulate every one of the 20,000 sub-circuits at 1.1 times the bin frequency. The values of the power dissipated in every circuit are extracted and stored in a matrix.

The next set of simulations are done on tuned circuits. The challenge behind tuning is that, an outlier circuit first needs to be tuned to bring about speedup, and then a search needs to be done to decide which bin the tuned circuit would fall in. Once the bin is chosen, the tuned circuit is again simulated, but this time at a clock period which is 1.1 times the bin period.

In the beginning, simulation is done for delay measurement. Hence, the period at which simulation is done, is kept large enough to accommodate the outlier delays. When simulating finally for power, the values of the pulse widths in the SPICE netlist had to be modified to the bin period, before being simulated again to measure power. This modification of pulse widths was done though a PERL script, invoked from the MATLAB program of power simulations.

While simulating with a circuit size of 10,000, it was found that all circuits once tuned had a worst delay in the range of 6-7ns. Hence, for tuned circuits, only 1 bin, with a period of 7.7ns, was created.

Once tuned circuits are simulated, the power dissipation values are stored in a third matrix. These matrices, created for power, are then made use of, along with the three delay matrices created earlier, in generating larger circuits.

Power dissipated with tuning, but no modification done to the pulse widths, was also calculated and stored for the purpose of comparisons, which will be studied in the proceeding sections.

Tuning was also done for two cases differing in the number of circuits tuned. In one case only 20 out of the 20,000 circuits were tuned, and in another 85 out of the 20,000 circuits were tuned.

8.4.3 Picking 10,000 sub-circuits to create a larger circuit

To generate a larger circuit of size 10,000 sub-circuits, a random set of 10,000 numbers was generated by a MATLAB program. With this as an index, the delay of the circuits were found from the three delay matrices, and the bins were then chosen, to decide which bin to pick the power dissipation values from. Once the power dissipation values for the 10,000 circuits were picked, they were added to give the total power dissipated in the large circuit with 10,000 EXOR tree circuits. This process of picking a random set of 10,000 numbers and calculating power is repeated 1000 times to get an average case.

8.5 Simulation results

CASE	Without any tuning circuitry	Untuned	Tuned	Change
20 gates Tuned	0.0656 W	0.0652 W	0.0654 W	0.24%
85 gates Tuned	0.0657 W	0.0654 W	0.0657 W	0.41%
20 gates Tuned and Speeded	0.0655 W	0.0653 W	0.0902 W	38.2%
85 gates Tuned and Speeded	0.0656 W	0.0652 W	0.107 W	64.1%

Table 8.2: Power dissipated with tuning

Table 8.2 shows that the percentage increase in the power dissipated with tuning alone and no speedup done is less than 1%. In the cases above more gates are being tuned than prescribed earlier - 1 in 3.5 million that lie beyond 5 sigma should be tuned. But in our case we are forced to tune a little more because of the smaller circuit size involved. In spite of tuning more, the excess power dissipation observed is still only less than 1%, which would be a lot lower in the 1 in 3.5 million case. On tuning 85 gates instead of 20, the increase in power was also very substantial.

It is also observed that the power dissipation increases to 38% and 64% on operating the tuned circuits at their maximum clock frequencies. This is expected because the circuits are now being operated at approximately 38% and 64% higher clock frequency.

Also, the power dissipated in circuits without any tuning capability included is greater than in circuits with tuning capability included, in spite of the larger leakage power produced, because these circuits are operated at a higher clock frequency. Circuits with tuning capability included but no tuning done would have larger circuit delays than circuits without any tuning circuitry added, because of the additional capacitance added by the tuning transistors. Hence operating them at a lower clock frequency would generate less power.

The above results are in tune with the expected results. On an average, keeping the frequency and Vdd supply voltage the same, switching from NMOS logic to CMOS logic reduced the power dissipation by approximately 2 orders of magnitude. Hence, tuning gates should also, on average, consume power that would go up by 2 orders of magnitude, because effectively the logic changes from CMOS to a pseudo NMOS type. Hence, assuming an increase in per gate consumption of power by 2 orders of magnitude, the increase in power consumed in a circuit with 280,000 gates, and having 25 gates that are tuned would cause an increase in the power consumed by 0.89%, which is very close to the obtained results.

The next chapter presents the final conclusions and observations on the work done.

Chapter 9

Conclusion

In this work a new design methodology that targets slow down brought about by process variability, and allows for post fabrication speedup is studied.

Random process variability is bound to occur and affect every chip fabricated as scaling goes beyond 22nm. Statistically every fabricated chip will have dozens of outlier transistors that would limit the maximum clock frequency at which the chip can be operated. Often considered as a defect, there exists a need to address this problem so as to reap the benefits of scaling.

The design methodology incorporates the idea of adding redundant transistors to the circuit which can be turned 'ON' in case an outlier is detected. Tuning is limited to outlier gates, which in a circuit size of a 100 million would equal a few hundred gates. This tuning would be beneficial in pulling back worst case path delays close to the average case delay, at the expense of a very small increase in the power dissipated. Simulation results indicate that worst case path delays were brought down by 50%, while power dissipation only increased by a meager 0.44%.

While reviewing the proposed tuning architecture, there were a few concerns about the area overhead, diagnosis difficulty, and implementing tuning itself.

It is true that there is an area overhead, but this area overhead is compensated by the savings obtained on Silicon itself, through the fault tolerance capability of this architecture. The implemented architecture is fault tolerant as turning 'ON' a tuning transistor could provide a path for charging or discharging the gate output capacitance of an otherwise faulty node. Hence, even though additional area is used up by the tuning circuitry, tuning when implemented could increase the yield, and buy back a portion of the used up Silicon.

Diagnosis of outliers is difficult indeed, but a scheme that enables it is being developed and studied. The following is a brief description of the work being done. The scheme first involves running path delay tests on the circuit to detect the slow paths. Vectors pairs that would detect slow paths are determined by running ATPG. When tested on a chip, an incorrect output logic measured within a particular time frame for the vector pair applied would implicate a set of gates that could be outlier candidates. Different sets of outlier candidates are determined for different vector pair inputs and the corresponding outputs where the fault was detected. Starting with this set of implicated gates, an algorithm is implemented to filter out the correct outliers from the set. This diagnosis scheme is being worked on, and when ready would make it possible to exercise the tuning strategy.

Tuning implementation is perceived to be implemented by overlaying a programmable memory like mesh on the chip, with the gate terminals of the tuning transistor connected to the memory array. The memory array is programmable to imply that a tuning transistor can be turned 'ON' by programming the memory appropriately. To make this implementation affordable, the memory mesh could be fabricated in thin-film technology, where amorphous Si is used in the fabrication process. Such a technology would have a very low drive, but for the sake of tuning, would suffice. Hence, it is valid to assume that such a tuning architecture has scopes of applicability in future technology nodes.

The work described in the thesis is only the beginning. The cases studied were naive in the sense that a simple circuit with the same gates and equal path delays is assumed, and adding tuning transistors and the additional loading capacitance had to be done manually and with small approximations. Also, simulations were done in 180nm technology, which might seem to indicate irrelevance of the obtained results, but the distributions worked with are pertinent to the advanced technologies, and hence close to applicable.

Future work involves using 45nm technology for the purpose of simulations, adding tunable gates to libraries so that its extraction can be automated, and working with standard circuits to make the case more authentic and applicable. Also a definite diagnosis strategy that would enable the detection of outlier gates in a chip, and a tuning methodology that would enable implementing the tuning scheme, needs to be studied.

Bibliography

[1] http://www.itrs.net/links/2009ITRS/Home2009.html, The International Techonology Roadmap for Semiconductors, 2009.

[2] S. Nassif, K. Bowman, "Design for Variability in DSM Technologies," ISQED 2000.

[3] S. Borkar, "Design Challenges of Technology Scaling," IEEE Micro, Vol. 19, Issue 4, pp. 23-29, August 1999.

[4] M. Ashouei, M. Nisar, A. Chatterjee, A. D. Singh, A. Diril, "Probabilistic Self-Adaptation of Nanoscale CMOS Circuits: Yield Maximization under Increased Intra-Die Variations," International Conference on VLSI Design, Bangalore, India, pp. 711-716, January 2007.

[5] M. Ashouei, A. D. Singh, A. Chatterjee, "A Defect-Tolerant Architecture for End-of-Roadmap CMOS," European Test Symposium, Freiburg, Germany, May 2007.

[6] A. D. Singh, "Scan Based Testing of Dual/Multi Core Processors for Small Delay Defects", in Proc. International Test Conference, 2008.

[7] A. D. Singh, "A Self-Timed Structural Test Methodology for Timing Anomalies due to Defects and Process Variations", in Proc. International Test Conference, 2005.

[8] K. S. Saha, "Modeling Process Variability in Scaled CMOS Technology," IEEE Design and Test of Computers, Vol. 27, Issue 2, pp. 8-16, March/April 2010.

[9] B. Cheng, D. Dideban, N. Moezi, C. Millar, G. Roy, X. Wang, S. Roy, A. Asenov, "Statistical-Variability Compact-Modeling Strategies for BSIM4 and PSP," IEEE Design and Test of Computers, Vol. 27, Issue 2, pp. 26-35, March/ April 2010.

[10] V. Wang, K. Agarwal, S. R. Nassif, K. J. Nowka, D. Markovic , "A Design Model for Random Process Variability" in proc. ISQED pp. 734-737, 2008.

[11] C. Kenyon, A. Kornfeld, K. Kuhn, M. Liu, A. Maheshwari, W. Shih, S. Sivakumar, G. Taylor, P. VanDerVoorn, K. Zawadzki, "Managing Process Variation in Intel's 45nm CMOS Technology." Intel Technology Journal , Vol. 12, Issue 2, pp. 92-110, June 2008.

[12] W. Shockley, "Problems related to P-N Junctions in Silicon." Solid-State Electronics, Vol. 2, pp. 35-67, January 1961.

[13] W. Schemmert, G. Zimmer, “Threshold-Voltage Sensitivity of Ion-Implanted MOS Transistors due to Process Variations.” Electronics letters, Vol. 10, Issue 9, pp. 151-152, May 1974.

[14] K. Yokoyama, A. Yoshii, S. Horiguchi, “Threshold-sensitivity Minimization of Short-Channel MOSFET’s by Computer Simulation.” IEEE Journal of Solid-State Circuits, Vol. 15, Issue 4, pp. 574-579, August 1980.

[15] P. A. Stolk, F. P. Widdershoven, D. B. M. Klaassen, “Modeling Statistical Dopant Fluctuations in MOS Transistors,” IEEE Transactions on Electron Devices, Vol. 45, Issue 9, pp. 1960-1971, September 1998.

[16] K. Kuhn, J. Kelin , “Reducing Variation in Advanced Logic Technologies: Approaches to Process and Design for Manufacturability of Nanoscale CMOS.” IEEE International Electron Devices Meeting, IEDM Technical Digest, pp. 471-474, December 2007.

[17] C. H. Diaz, H. J. Tao, Y. C. Ku, A. Yen, K. Young, “An Experimentally Validated Analytical Model for Gate Line Edge Roughness (LER) Effects on Technology Scaling.” IEEE Electron Device Letters, Volume 22, Issue 6, pp. 287-289, June 2001.

[18] H. W. Kim, J. Y. Lee, J. Shin, S. G. Woo, H. K. Cho, J. T. Moon, “Experimental Investigation of the Effect of LWR on Sub-100-nm Device Performance.” IEEE Transactions on Electron Devices, Vol. 51, Issue 12, pp. 1984-1988, December 2004.

[19] H. Fukutome, Y. Momiyama, T. Kubo, Y. Tagawa, T. Aoyama, H. Arimoto, “Direct Evaluation of Gate Line Edge Roughness Impact on Extension Profiles.” IEEE Transactions on Electron Devices, Vol. 53, Issue 11, pp. 2755-2763, November 2006.

[20] A. Asenov, A. R. Brown, J. H. Davies, S. Kaya, G. Slavcheva, “Intrinsic Parameter Fluctuations in Decananometer MOSFETs introduced by Gate Line Edge Roughness.” IEEE Transactions on Electron Devices, Vol. 50, Issue 5, pp. 1254-1260, May 2003.

[21] K. Bernstein, D. J. Frank, A. E. Gattiker, W. Haensch, B. L. Ji, S. R. Nassif, E. J. Nowak, D. J. Pearson, N. J. Rohrer, “High-performance CMOS Variability in the 65-nm Regime and Beyond,” IBM Journal of Research and Development, Vol. 50, Issue 4/5, pp. 433-449, July 2006.

[22] A. Asenov, S. Kaya, J. H. Davies, “Intrinsic Vth Fluctuations in Decananometer MOSFETs Due to Local TOX Variations,” IEEE Transactions on Electron Devices, Vol. 49, Issue 1, pp. 112-119, January 2002.

[23] S. Changhwan, S. Xin, K. L. Tsu-Jae, “Study of Random Dopant Fluctuation (RDF) Effects for the Trigate Bulk MOSFET,” IEEE Transactions on Electron Devices, Vol. 56, Issue 7, July 2009.

[24] A. T. Putra, A. Nishida, S. Kamohara, T. Hiramoto, “Random Threshold Voltage Variability Induced by Gate-Edge Fluctuations in Nanoscale Metal-Oxide-Semiconductor Field-Effect-Transistors,” Applied Physics Express 2, Vol. 024501, January 2009.

[25] http://en.wikipedia.org/wiki/Normal_distribution, Normal distribution.

[26] http://ptm.asu.edu/modelcard/HP/32nm_HP.pm, 32nm High Performance PTM models.

[27] A. D. Singh, K. Mishra, A. Faraz, A. Chatterjee, "Path Delay Tuning for Performance Gain in the face of Random Manufacturing Variations," in proc. International Conference on VLSI Design, Bangalore, India, January 2011.