Algorithm Based Robust Transistor Sizing for Optimal Power - Delay Designs in CMOS Digital VLSI Circuits

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Electronics and Communication Engineering by Research

by

Prateek Gupta
20162066
prateek.g@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
July, 2019

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Algorithm Based Robust Transistor Sizing for Optimal Power - Delay Designs in CMOS Digital VLSI Circuits" by Prateek Gupta, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Dr. Zia Abbas

To

Papa and Mummy for everything

Acknowledgments

The last three years I spent in IIIT-H is the best phase of my life which made me a different person not only as a good researcher but also a good human being. I would like to thank all those people who helped me in successfully completing this beautiful phase of my life.

First and foremost, I owe a deepest sense of gratitude to my thesis advisor Dr. Zia Abbas for his timely advice and support through out my MS journey. Without his guidance I can't even imagine to complete this research on time. It was his motivation only that I got interest into the research and published research papers in one of the best conferences of VLSI which was just an unimaginable dream for me, three years ago. I remember many times he used to sit beside me to motivate me to the research and gave life lessons. He has played a very important role in shaping my personality and moulding my thoughts in the right direction. He is really very kind, simple and soft-hearted human being. It was his trust and belief that I got a chance to teach the students during the TAship period under him.

I am also extremely grateful to Dr. Abhishek Srivastava for giving me an opportunity to do TAship under him. Working with him was totally a different experience. I am thankful to him that he made me do even little things with so much perfection.

I am thankful to my seniors cum friends Rhishi, Pratik, Mahesh, Nachiket, Prakash, Upender, Deepa, Shivakrishna, Hari for their help and motivation in the course work as well as in my research. I am very inspired from Rhishi bhaiya as I learnt many good things from him; from problem solving techniques to tree plantation etc. Thanks to Boshak (Suman) and Jitender for always being there whenever and wherever I needed you. Thanks to Anish, ZakirSukesh, Pulkit, , Jayant, Josyula, Durga, Bhargav, Kishore, Krishna, Kali, Priyam, Deepak, Muskan, Ruchi and Rakesh for being an integral part of my life at IIIT-H. Special thanks to Arpan and Ashfakh for fruitful discussions and valuable suggestions throughout this journey. Thanks to the co-authors of my research papers Shubham, Harshini and Shirisha for doing such a quality work and helping me. I thank all my CVEST lab mates for making the group meetings memorable.

Thanks to Yuktahar mess staff for always providing the healthy food which helped me in focussing on my research. I thank Srikanth sir, mess admin who helped me while working as Yuktahar Mess Co-ordinator for which I got the Commendation Award as well.

Thanks to Dr. Semparithi Aravindan for helping me a lot in managing the CVEST servers and all kind of tools installation on them. Thanks to Karthik sir, Sailaja ma'am and Pushpalatha ma'am for

their help and support in the administrative works. Last but not the least, I thank all my friends in IIIT Hyderabad.

Abstract

The need for the more and more number of devices, in turn higher numbers of functions in a single Integrated Circuit (IC) and greater operating speed are the prime reason for the continuous downsizing of CMOS technology. However, such progressive miniaturization of device dimensions in CMOS circuits has also imported an enormous increase in process variability and therefore, performances figures like propagation delays and power dissipation notably the leakage power are severely degraded.

Downscaling the technology nodes in CMOS integrated circuits at or below 45nm have fastened the performance of the circuits but at the cost of the power consumption. CMOS Integrated Circuits have expelled the barrier to achieve minimum power consumption undoubtedly, though it is getting quite difficult to achieve desired performances in the circuits in ultra-deep sub-micron regime. This expeditious scaling in IC causes the dominating behaviour of static power over dynamic power.

Enormous increase in process variations (due to progressive CMOS technology scaling) along-with the temperature and supply voltage variations are severely degrading the fabrication outcome of digital circuits i.e. circuits are not accomplishing the specification bounds of the required performances. Therefore, process and operating variations aware optimization has become a very essential task in VLSI design. Moreover, many specifications in a circuit have challenging trade-offs, hence demand effective optimization skills.

With this vision, this thesis presents various optimization algorithms based robust transistor sizing calculation for various nanoscale CMOS digital circuits. The algorithms used to optimize the fitness function along with constraints set are meta heuristics such as evolutionary based Genetic Algorithm, thermo based Simulated Annealing algorithm, swarm based Artificial Bee Colony and Particle Swarm Optimization algorithm. The objective is to minimize the static i.e. leakage power without degrading the operating frequency (i.e. keeping the propagation delays in bound) and area.

Statistical variation aware robust transistor sizing (width/length) is calculated for the various 2 input and 3 input basic logic gates such as AND, OR, NOT, NAND, NOR, XOR, Full-adder etc. and the obtained sizing is used to size the various other complex cells such 32-bit ripple carry adder, carry select adder, carry save adder and multiplier and few ISCAS benchmark circuits. Notably, our methodology targets semi-custom digital design: once the cell library layout are designed, they can be placed wherever needed in a semi-custom design by technology mapping tools. Monte Carlo simulations have been performed for all the logic cells with 50,000 random statistical samples to check the circuit's yield,

which is found to be around 99.8%. The leakage power is reduced up to 88% without giving any penalty to the propagation delay.

Also, the author has proposed two novel techniques to minimize the stand by mode leakage power of 1 bit full adder cell and compared their results with the nominal one. The optimization is done for 45nm, 32nm and 22nm metal gate high-k productive technology model. The obtained PVT variations aware transistor sizing (width/length) are used to optimize other full adder based cells.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The advent of the CMOS technologies has no doubt removed the obstacle of achieving low power consumption in the mobile and battery portable devices. But this era came up with an ultra deep sub-micron regime where, leakage power occupies most of the part of the total power dissipation in the circuits. According to a report of ITRS (International Technology Roadmap for Semiconductors) [3] [6], the static power dissipation in the CMOS transistors would dominates the dynamic power over the years, as shown in Fig. 1.1. Thus there is a need of the techniques to improve the leakage power dissipation without compromising the other performances such as propagation delay which is in trade off. With this down scaling of the technology and demand of the high performance circuits, the short channel effects come into the picture, thus the robust transistor sizing with desired performance constraints is the ultimate need in the future VLSI technology and the optimization algorithms will accomplish such tasks perfectly.

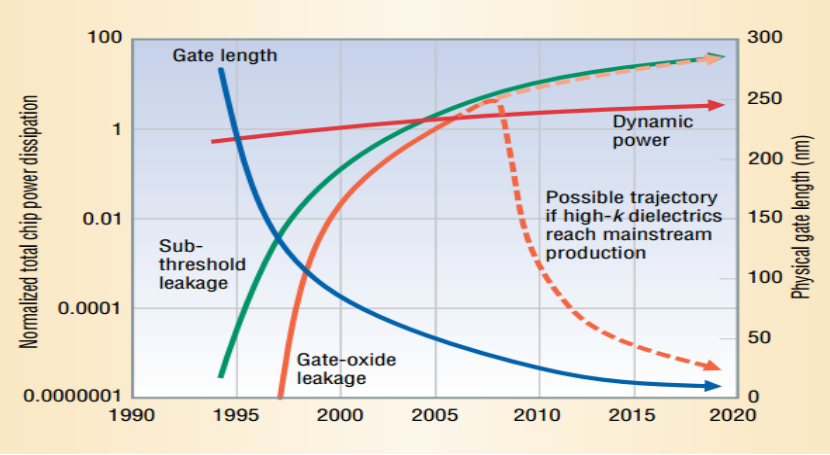


Figure 1.1: Trend of leakage power

To maintain the reliability of the device and constant power dissipation per unit area, the supply voltage has also been reduced thus there is a steady reduction in threshold voltage of the transistor as well, it leads to the increment in the leakage power. In the earlier CMOS technology nodes, 350nm or higher, the process variations effects on the device performance were meager but as the technology scaled, the effects have become crucial and decisive for the proper functioning of the device after the fabrication. There can be both die to die and within-die variations in the processes such as threshold voltage, gate oxide thickness, junction depth, channel doping concentration, physical gate equivalent oxide thickness, effective channel length etc [10],[34]. Generally, corner analysis is widely used to make sure the unaffected working of the chip from the process. Fast (F), Typical (T) and Slow (S) corners are provided by the foundry in their respective process development kits (PDKs) but these corners might not be potent at some of the transitional values [28]. Another method of validating the robustness of the circuit is to apply Monte Carlo simulation check using statistical Gaussian (normal) distributed samples of the various process parameters but this method is computation and time expensive.

In addition to the process variation, the variation in the operating conditions such as temperature and supply voltage can also cause the device performances to vary from their nominal values. For example, a circuit designed for 0.8 V, may work at 0.75 V due to fluctuation in the supply voltage which ultimately operates at slower speed than the nominal. Similarly, the higher temperature may be good for speed but not for static power which increase exponentially with temperature [14]. Such fluctuations need to be avoided to increase the parametric yield of the integrated circuits.

1.2 Thesis Contribution

The contribution of this thesis are as follows:

- We Proposed a novel idea to minimize the stand by mode leakage power of PVT invariant 28-Transistor Full-adder cell and it's applications by calculating robust transistor sizing using swarm based Artificial Bee Colony (ABC) algorithm.

- We have performed the leakage power and propagation delay optimization of all the basic digital circuit such as 2 input/ 3 input NAND, NOR, AND, OR, XOR, Full-adder and few benchmark circuit by varying the width (W) and length (L) of each and every transistor in the circuit using different algorithms considering the PVT variations.

- we optimize the the leakage power, delay using thermo based Simulated Annealing (SA) algorithm while calculating the robust transistor sizing of the transistors of the circuit and further we improved the local minima search of SA algorithm using gradient descent applied over the sizing obtained using SA.

- Also, we improved the convergence speed and efficiency of the existing variants of swarm based Particle Swarm Optimization (PSO) algorithm and compared the results for various mathematical benchmark functions as well as for the digital circuits.

1.3 Thesis Outline

The thesis is organized as follows:

In chapter 2, we discuss an overview of performance optimization, power dissipation in ICs, different types of leakage mechanism, leakage dependent factors and the methods to control the leakage are explained briefly.

In chapter 3, the effect of operating and process variations on the leakage power and propagation delay of the circuit are observed and thus the problem statement is defined as to achieve the robust transistor sizing using various algorithms.

Chapter 4 discusses the various optimization algorithms implemented and the improvements done on it along with their implementation on various logic cells have been shown.

Chapter 5 proposes two novel techniques to minimize the stand by mode leakage power of full adder cell by calculating the PVT invariant transistor sizing using swarm based algorithm.

Finally, this thesis has been concluded with some future work.

Chapter 2

Various Aspects of Performance Optimization in Digital Circuits: An Overview

2.1 Technology Scaling

When Intel made their first CPU in 1971, it contained 2300 transistors with clock speed of 0.7MHz. While, compared to today's where a CPU contains billions of transistors and can reach the speed of over 4 GHz. This has become possible due to technology scaling. The direct advantage of technology scaling is to fabricate higher number of transistors on each silicon wafer which leads to cheaper circuits. This led to the widespread adoption of mobile phone, tablets, computers and the information technology revolution.

The effect of technology scaling was first observed by Gordon Moore in the 1960's that number of transistors in an integrated circuit doubles after every 18 months or so. Figure 2.1 shows the observation of Gordon Moore. In 1974, Robert Dennard showed how to realize Moores law with metal oxide

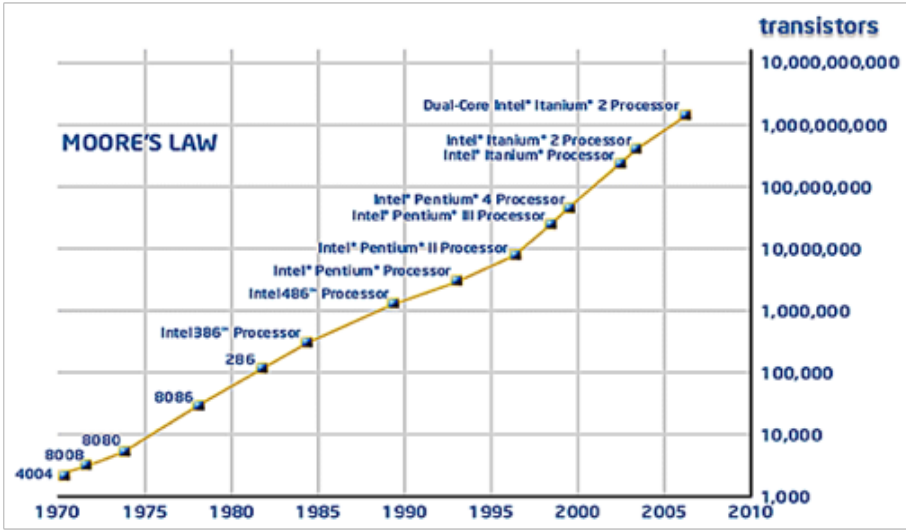


Figure 2.1: Moore's law observations (source: Intel)

semiconductor field effect transistors (MOSFETs). Moore's law worked perfectly for about 50 years as anticipated but later it slowed down due to the various short channel effects and trade off aroused in the circuits [37]. The processor fabricated using 45nm technology contains twice the number of transistors as compared to 65nm technology node. Currently the technology node has been down scaled below 10nm and is expected to reduce more although Intel has suggested silicon transistors can only keep shrinking for another five years. Scaling of the MOSFET device requires proportional reduction in channel length, channel width, oxide thickness and threshold voltage etc. which leads to adequate increment in the leakage currents with each scaled technology node[5].

2.2 Transistor Sizing

In today's nanoscaled technology, the low power dissipation is the basic need of any circuit to perform well while keeping delay and other performances with in the specification constraint. This is a crucial task which can be achieved if all the transistors in the circuit are properly sized. The transistor sizing is said to be robust if the performances are invariant of the temperature, supply voltage and various process parameters. In my work, the nominal transistor sizes are chosen as minimum sized transistor (width/length) for equal rise and fall time for the particular input to the output.

In a static CMOS, the propagation delays can be modelled as dependent on RC, where R is the effective resistance and C is the capacitive load. From the drain current equation of the Mosfet given in Eq. 2.1 for linear mode, we know that the drain current is proportional to width (W) of that transistor, whereas resistance (R) is inversely proportional to it while capacitance C is proportional to size of the transistor in subsequent stage.

$$I = \mu C_{ox} \frac{W}{L} \Big((V_{gs} - V_{th})^2 - \frac{V_{ds}^2}{2} \Big) \tag{2.1}$$

where, μ is charge mobility, W is gate width, L is gate length and C_{ox} is gate oxide capacitance per unit area, V_{gs} is the gate to source voltage, V_{th} is threshold voltage of the transistor and V_{ds} is the drain to source voltage. The W and L are treated as variable. Various algorithms and programming method are then applied to achieve optimal size for each transistor in the circuit. In [33], mathematical programming based concept of generalized gradient is proposed to size the transistors. Non linear programming (NLP) and General Geometric Programming (GGP) are used in [32] for the transistor sizing. Transistor sizing of custom high-performance digital circuits with parametric yield considerations is done in [8].Many optimization algorithms such as Genetic algorithm, Particle swarm optimization algorithm, Artificial bee colony algorithm etc have been proposed in past which can solve complex non parametrical problems, multi-dimensional, non continuous and non differential problems [15], [19], [23].

2.2.1 Design Parameter Range

The design parameters here are the widths and lengths of all the transistors present in circuit. The width and length are varied in a range so as to achieve the robust transistor sizing with performance

Table 2.1: Design Parameter Range

Technology node	Width Range (W)	Length range (L)
45nm	$180\text{nm} - 1500\text{nm}$	$45\text{nm} - 55\text{nm}$
32nm	$128\text{nm} - 1200\text{nm}$	$32\text{nm} - 42\text{nm}$
22nm	$88\text{nm} - 800\text{nm}$	$22\text{nm} - 32\text{nm}$

constraints. These parameters are altered in steps of 1nm which is the step size available in the 45nm layout geometries for alteration. The nominal transistor sizing taken in this paper are from [4] for 45nm technology and obtained proportionally for other scaled technology nodes also such as 32nm 22nm. These nominal design parameters are calculated in cited paper by keeping the equal rise and fall time of the delay for corresponding technology nodes. The range of transistor's width and length taken in this research are given in Table 2.1.

2.3 Power Dissipation in ICs

Power dissipation is the most important concern in circuit design and performance optimization. In battery operated portable devices, high power consumption reduces the battery life. Thus to prolong the battery life and battery efficiency, the power consumption needs to be reduced. The power dissipation in circuits can be classified into two parts: dynamic power (power dissipated during Transient state condition.) and static power (power dissipated during Transient state condition) [29]. The power dissipation also depends on which mode the circuit is operating, it can be active mode or standby mode. With the progressive downscaling of VLSI technology for performance improvement in computing systems led to an enormous increase in leakage power (static power) and even it dominates the dynamic power in the ultra deep submicron regime.

2.3.1 Dynamic Power Dissipation

The Dynamic power dissipation in circuit is due to the logic transition (i.e. charging and discharging of load capacitor) and due to short circuit current that flows between supply and ground when both pull up and pull down network of CMOS are conducting simultaneously ie.

$$P_{dynamic} = P_{switching} + P_{short} \tag{2.2}$$

$$P_{switching} = \alpha C_{load} V_{dd}^2 f \tag{2.3}$$

Where α is switching activity, C_{load} is the load capacitance, V_{dd} is the supply voltage and f is the clock frequency.

$$P_{short} = \frac{\beta}{12}(V_{dd} - 2V_{th})^3 \frac{t_r}{t_p} \tag{2.4}$$

Where β is the DC gain of the device, t_r is the rise time of the input signal and t_p is its time period.

2.3.2 Static Power Dissipation

Current flowing in the circuit without even any transition in the input signal is called leakage current. That is, when there is no change in the input the static power dissipation in the circuit occurs. Leakage current increases by the factor of 10 with reduction of the threshold voltage by 100mv [14].

2.4 Sources of Leakage Current in Nanoscaled CMOS

There are different leakage mechanisms contribute to total static power dissipation out of which the sub-threshold leakage contributes most. Various sources of leakage current are mentioned below:

2.4.1 Gate Oxide Tunneling Leakage

With channel length scaling, gate oxide thickness, junction depth are also scaled to maintain good transistor aspect. Thus, due to thin oxide layer in MOSFET and under strong electric field across oxide layer the tunneling of electrons through the gate into substrate causes leakage current to flow through the gate terminal which increases exponentially with gate oxide thickness (T_{ox}) and supply voltage (V_{DD}) [11]. Gate oxide leakage currents have been partially limited by the introduction of high-K (permitivity) dielectrics in CMOS technologies [26].

2.4.2 Sub-threshold leakage

Small threshold voltage is desirable for high I_{on} current but if we set V_t very small value then there will be leakage current exist. A minimum amount of gate to source voltage $(V_{gs} > V_t)$ is required to turn the device in on state.
At $V_{gs} < V_t$, a MOSFET device is in the off state, therefore an undesirable leakage current due to minority carriers will exist between drain and the source terminal in the device operating in weak inversion region [21], [38], [39]. This current is called the sub-threshold current, which is primary contributor to the MOSFET off-state current, I_{off} and i.e. in what is called the "sub-threshold" region. The sub-threshold leakage is increases with the reduction in the transistor threshold voltage, which is scaled down with the supply voltage for maintaining high speed and supply voltages are scaled to withstand the electric field. This exponential increase in sub-threshold leakage current can be observed from equation 2.5.

$$I_{ds} = K(1 - e^{(-V_{ds}/V_T)})e^{(V_{gs}-V_t+\eta V_{ds})/nV_T} \tag{2.5}$$

where η is drain induced barrier lowering coefficient, K and n are technology functions, V_{ds} and V_{gs} are drain to source and gate to source voltages, V_T is the voltage equivalent to temperature and V_t is the transistor threshold voltage.

2.4.3 Gate Induced Drain Leakage

When the electric field is very strong near the drain junction in the MOSFET with grounded gate and drain potential at V_{DD}, causes electron-hole pair generation through avalanche multiplication in the drain gate overlap region while majority carriers are recombined near drain region which results in gate induced drain leakage (GIDL). The holes are swept away fron the substrate and electrons are collected by the drain results in GIDL. From BSIM model, the expression for the GIDL current can be expressed as in below equation 2.6 [1]:

$$I_{GIDL} = AGIDL.W_{effCJ}.N_f.\frac{(V_{DS}-V_{GS}-EGIDL)}{3T_{ox}}exp(-\frac{3.BGIDL.T_{ox}}{V_{DS}-VGS-EGIDL})\frac{V_{DB}^3}{CGIDL.V_{DB}^3} \quad (2.6)$$

Where AGIDL, BGIDL, CGIDL and EGIDL are model parameters. N_f is the number of fingers in the device. W_{effCJ} is the effective width of drain diffusion.

2.4.4 Drain Induced Barrier Leakage

This is the most common factor of increasing leakage current in the circuit due to **short channel effect**. A MOS device is said to be have short channel when the channel length is comparable to the depletion region thickness of the source and drain junctions.

DIBL effect occurs in the MOSFET when the drain is at high potential and there is reduction in the threshold voltage of the transistor. In weak inversion regime, the height of the potential barrier present between source and channel can decrease due to high drain voltage, which leads to increase in the drain current. Thus the drain current here is not only due to gate voltage but also due to drain voltage, leads to DIBL.

2.5 Leakage Dependent Factors

There are various factors which can modify the leakage current in the circuit such as input vector dependency, stacking effect and loading effect.

2.5.1 Input Vector Dependency

In standard digital cells, leakage power not only depends upon the various transistor's parameters such as channel length, junction depth, oxide thickness, threshold drop but also on the input vector applied at it's input terminals [31]. From Figure 2.2, it can be seen that how leakage power varies with the input vector for different technology nodes in a 28T full adder cell. It can be observed from Figure 2.1 that the leakage at input vector (111) which is the minimum leakage vector (MLV) of the cell is significantly lesser (about 1.4 times) than leakage at input vector (001) which is the worst leakage state (WLS) of the cell. The leakage values shown are calculated using HSPICE tool using Berkeley Short-channel IGFET Model (BSIM4) for metal gate high K (MGK) predictive technology models (PTMs).

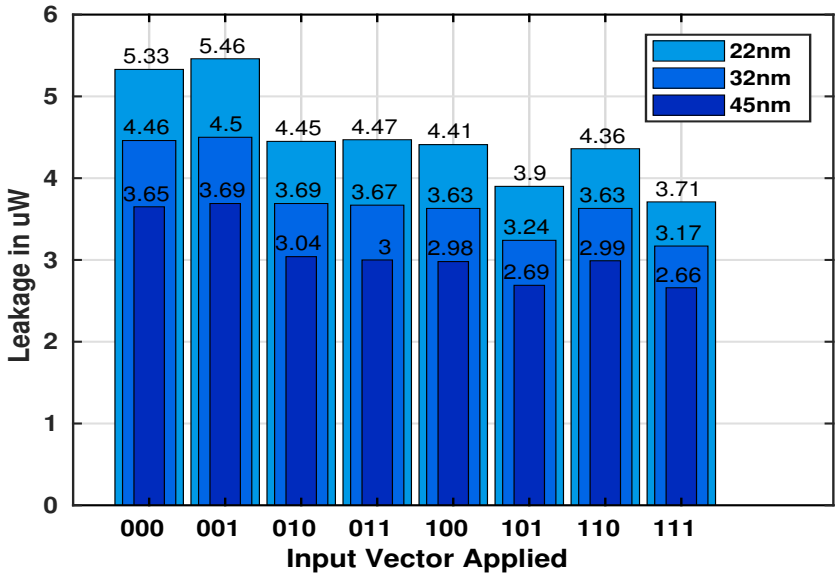


Figure 2.2: Variation in leakage with input vector at different technology nodes in full adder circuit.

2.5.2 Stacking Effect

Stacking effect leads to the stacking of transistors in series, so as to reduce the sub-threshold leakage in significant amount. It has been observed that the leakage current in series connected two off devices is smaller than that in the one off device. Figure 2.3 shows the single off transistor and a stack of two off transistors [30].

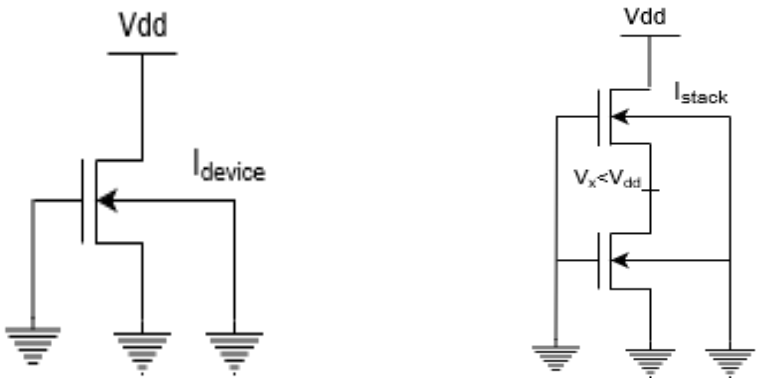


Figure 2.3: Single off transistor and a stack of two off transistors

2.6 Conventional Techniques to control the leakage in ICs

2.6.1 Dual Threshold

In a dual-threshold designs, the low-V_t transistors are used in circuit's critical paths and the high threshold transistors in the rest paths to reduce the sub-threshold leakage which is the most dominating leakage current. In Dual-V_t CMOS, high V_t transistors help in reducing the leakage power and single low V_t transistors has same critical delay [5].

2.6.2 Power Gating

To reduce the maximum amount of leakage power dissipation in the stand by mode is to turn off the power supply, the technique is called power gating.

Generally, the power gating implementation requires a multi-threshold CMOS technique. The main logic blocks are implemented using low threshold, high performance transistors whereas the sleep transistors connected between pull up network (PUN) of CMOS and supply & pull down network (PDN) and ground are provided with the high threshold voltage as shown in figure 2.4. Thus, In the **active** mode, the sleep transistor is on and therefore circuit function as usual. In **stand by** mode, the sleep transistor is turned off and the circuit is disconnected from the supply and ground and connected with virtual nodes. The important thing is to size the sleep transistor properly so as the voltage drop across them should be minimum while it is on. It involves few flaws such as increased area, performance penalty and several noise effects etc.

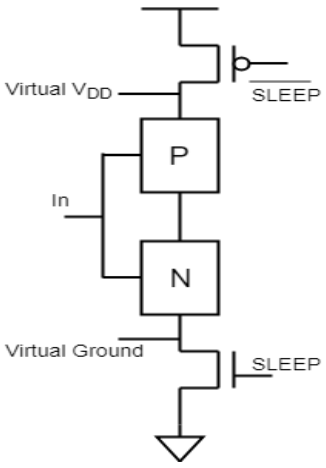


Figure 2.4: Power gating circuit

Table 2.2: Results obtained for CMOS NOR2 cell

Inputs (A B)	Output (O)	Leakage (in nW)
0 0	1	517
0 1	0	501
1 0	0	223
1 1	0	169

2.6.3 Input Vector Control (Minimum Leakage Vector Method)

From Figure 2.2, we have seen that the leakage power is a strong function of the input vector applied to the circuit due to the reason of effect of input value on the number of off transistors in the cell.

Table 2.2 shows the leakage power obtained for a two-input NOR gate for 45nm MGK PTM model card with 1.0V supply voltage and 25^oC temperature. In this case the logic gate dissipates minimum leakage power when both the inputs are one i.e. when both the PMOS transistors are off while both the NMOS transistors are on. The effective resistance between the supply and ground is the sum of resistances of two OFF PMOS transistors in the series connection. Thus to drive the circuit while in STANDBY mode, this input vector '11' if applied through pass transistors to the inputs of the gate can reduce the leakage power a lot.

2.6.4 Body Biasing

Generally in MOSFETs, the body and source terminals are supplied with the same potential ie. V_{sb}=0 for both NMOS and PMOS.

But if we connect source potential at higher point than the body ie. $V_{sb} > 0$, then the threshold voltage of the MOSFET increases which reduces the leakage power dissipation but at the cost of the speed. This method of reducing the leakage power in the circuits is called as reverse body biasing (RBB).

In other case, if $V_{sb} < 0$, it decreases the threshold voltage of the MOSFET which increase leakage current through the device which is called the forward body biasing technique (FBB). The only advantage of this technique is that the device becomes faster.

The RBB and FBB are limited by the breakdown voltage and latch up problem in the CMOS respectively.

2.7 Propagation Delay

In general terms, the propagation delay (τ_d) of any logic gate defines how fast it response to the change at the input. For a CMOS inverter, it is measured as the average response time for low-to-high

output transition (τ_{lh}) i.e. time taken by the capacitive load of inverter to charge up to supply voltage (V_{DD}) and for high-to-low output transition (τ_{hl}) i.e. time taken by the same load to discharge through ground (Gnd), given by the below equation and shown in the figure 2.5.

$$\tau_d = \frac{(\tau_{lh} + \tau_{hl})}{2} \tag{2.7}$$

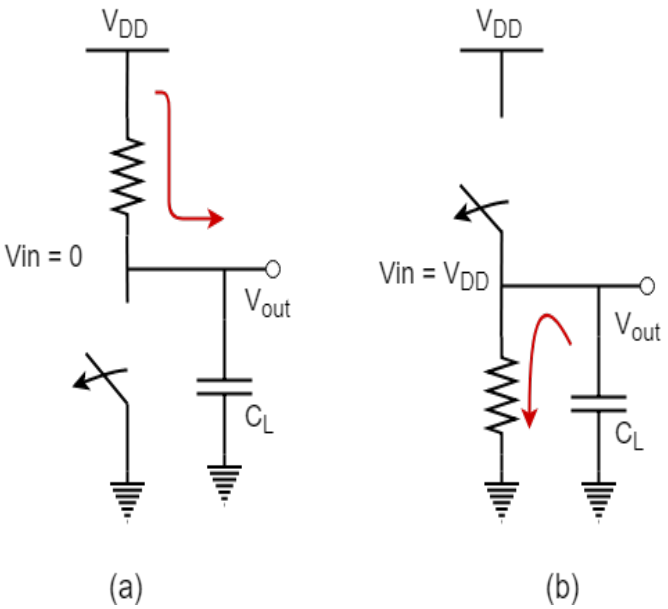


Figure 2.5: Propagation delay a) τ_{lh} and b)τ_{hl}

2.7.1 The Fanout-of-four (FO4) Delay:

As most of the designers know the FO4 delay in their process hence it is suitable to define the times in terms of FO4 delays [36].

The RC delay model can be easily understand by an example of an inverter pair. The propagation delay of an inverter pair connecting one back to back with different effective resistance is given by equation (1) [36].

$$T_{pd,pair} = t_{dr} + t_{df} = (R_{eff,P} + R_{eff,N})(C_D + C_G) \tag{2.8}$$

$$T_{pd,pair} = t_{dr} + t_{df} = C_G(R_{eff,P} + R_{eff,N})\left(\frac{C_D}{C_G} + 1\right) \tag{2.9}$$

where, t_{dr} and t_{df} is the rise and fall delay, $R_{eff,P}$ and $R_{eff,N}$ are the effective resistance of PMOS and NMOS respectively, C_D is the parasitic output capacitance and C_G is the input capacitance. Generally, the term $\left(\frac{C_D}{C_G}\right)$ is considered as constant and is usually equal to 1 ie. m=1 and the second term 1 in

the equation (3) indicates that each inverter is loaded by one other identical inverter. Thus the fanout 4 (FO4) delay can be calculated similarly. The average propagation delay is given as follows:

$$T_{pd,avg} = \frac{t_{dr} + t_{df}}{2} = C_G(R_{eff,P} + R_{eff,N})(m + 4) \tag{2.10}$$

The fanout of 4 nominal sized inverter of their corresponding technology node have been taken as a load at the output of 28T full adder circuit. It is shown in Fig. 1 for the carry output (C_{out}) loading FO4; similarly it has been taken in the sum (S) output also. Thus the capacitive load calculated for the mentioned FO4 delay for different technology nodes are 0.56fF, 0.35fF and 0.21fF for 45nm, 32nm and 22nm PTM models respectively.

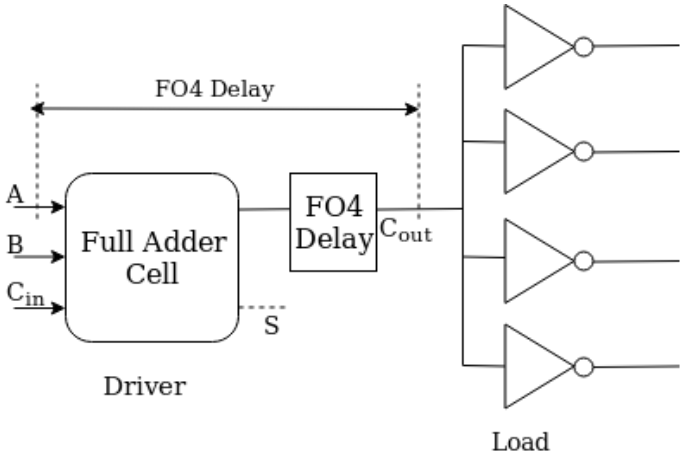


Figure 2.6: The fanout-of-4 (FO4) delay

Chapter 3

Effect of Operating and Process Variations on the Circuits Performances: Problem Statement

The operating variations include the variation in supply voltage and temperature and the process variations include the variations in various process parameters such as oxide thickness, junction depth etc. Thus the objective is to find the robust transistor sizing which is invariant of such variations for minimum leakage power dissipation without compromising with the propagation delay which is in trade-off with it.

3.0.1 Corner Analysis

The most common technique used in the industries to check the worst case operation of the device is to check the functioning of the device at all the process corners such as SS (slow-slow) i.e. both NMOS and PMOS are slow, SF (slow-fast) i.e. slow NMOS and fast PMOS, FS (fast-slow) means NMOS is fast but PMOS is slow, FF (fast-fast) i.e. both NMOS and PMOS are fast and TT (typical-typical) which is the normal operation of NMOS and PMOS. These corners are displayed in figure 3.1.

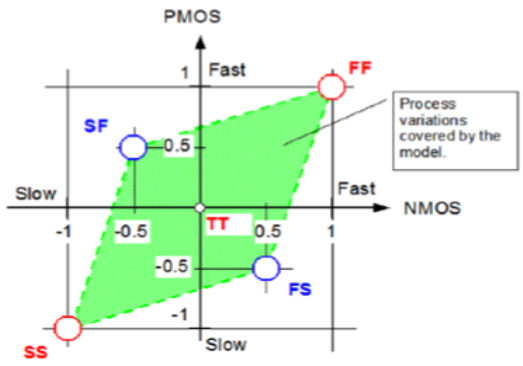


Figure 3.1: Corner Model

It is assumed that if the final design passes the test on all these corners then certain yield is guaranteed. However, this method is now facing few crucial challenges [27] as the device may not be functional

at some combinations of the intermediate value of these corners. The digital corners do not include local variation effect failing which affect the analog device functioning.

3.0.2 Operating Parameters Range

From equation (2.5) and Fig. 3.3, it is clear that sub-threshold leakage increases exponentially with temperature and proportionally with supply voltage also delay increases with reduction in supply voltage (Fig. 4) thus the temperature and voltage for each performance corners are determined. For different technology node used in this paper, the normal operating condition (NOC) and worst operating conditions (WOC) for leakage and delay are displayed in Table 3.1. For worst leakage computation, 125^oC temperature and $+10\%$ variation with nominal value of supply voltage has been taken. Similarly, the performance (delay) corner in this paper has 125^oC temperature and -10% variation with nominal value of supply voltage. A nominal temperature of 25^oC and supply voltage of 1.0V is assumed for predictive technology models (PTM) of 45nm and 32nm while 25^oC temperature and 0.8V supply voltage is assumed for 22nm technology node [2].

Table 3.1: Operating Variations Constraints

Technology node	NOC (temp, Vdd)	WOC (temp, Vdd)
45nm 32nm	(25^oC, $1.0V$)	(125^oC, $1.10V$) for leakage computation. (125^oC, $0.90V$) for delay computation.
22nm	(25^oC, $0.8V$)	(125^oC, $0.88V$) for leakage computation. (125^oC, $0.72V$) for delay computation.

3.0.3 Statistical Parameters

The impact of statistical variations on circuit performance was insignificant in previous CMOS technologies (0.5μm or higher) [9] and therefore the performance was predictable over the device life time. Statistical (process) variation can be broadly classified into inter-die or global variations (eg. t_{ox}) and intra-die or local variations (eg. threshold voltage, V_{th}). Global variations are wafer-to-wafer or chip-to-chip variations , thus their impact sweeps for long-range while the local variations have short-range impact [4]. The Berkeley Short-channel IGFET Model (BSIM4) [1] with metal gate high-k PTMs for reported nanoscaled technology nodes are used in the netlists, simulated using HSPICE tool [18]. At 3σ value of $\pm 10\%$ variation in 10 process parameters in NMOS and PMOS are considered and for 45nm they as reported in Table 3.2, similarly done for other technology nodes as well. The abbreviations of various process parameters mentioned in Table 3.2 are given as follows: channel-length offset parameter $(lint)$, channel-width offset parameter $(wint)$, physical gate equivalent oxide thickness $(toxp)$, electrical gate equivalent oxide thickness for both NMOS $(toxe_n)$ and PMOS $(toxe_p)$, nominal gate oxide thickness for gate dielectric tunneling current model only for both NMOS $(toxref_n)$ and PMOS $(toxref_p)$, junction depth (xj) and channel doping concentration at depletion edge for zero body bias

Table 3.2: Process variation considered ($45\mathrm{nm}$ tech.)

Sr. No.	Process parameter	Lower deviation	Nominal	Higher deviation
1	$lint$	2.35e-09	2.7e-09	3.05e-09
2	$wint$	4.34e-09	5e-09	5.59e-09
3	$toxp$	5.74e-09	6.5e-09	7.26e-09
4	$toxe_n$	7.96e-10	9e-10	10.28e-10
5	$toxe_p$	7.97e-10	9.2e-10	10.37e-10
6	$toxref_n$	7.96e-10	9e-10	10.28e-10
7	$toxref_p$	7.97e-10	9.2e-10	10.37e-10
8	xj	1.22e-08	1.4e-08	1.57e-08
9	$ndep_n$	5.77e+18	6.5e+18	7.26e+18
10	$ndep_p$	2.45e+18	2.8e+18	3.12e+18

for NMOS ($ndep_n$) and PMOS ($ndep_p$). Table II also depicts the nominal and deviated value in each process parameter. Of course, lower deviation in some parameters leads to more amount of leakage such as in ($toxp$), ($toxe_n$), ($toxe_p$), ($toxref_p$), ($ndep_p$) and vice versa for propagation delay.

3.0.4 Effect of variations on leakage and delays

To observe the impact of temperature and supply voltage on the leakage power and propagation delays respectively, simulations have been done on HSPICE tool and displayed in Figure 3.3 and Figure 3.4 for CMOS 28-Transistor full adder cell whose circuit diagram is shown in Figure 3.2. It can be clearly seen the leakage power for all the input vectors of full adder is increasing with respect to temperature and the supply voltage reduction causing delay to increase. Also, the impact of the corresponding worst values of all the process parameters variations together on leakage and delay is been displayed in Figure 3.5 and Figure 3.6 respectively. From Figure 3.5, it is clear that alone process parameters variation can cause drastic increment in the leakage which is upto 3.5 times the nominal leakage though in case of delay the highest value of delay increased to 16.1ps from 15.3ps ($delayHL\_B\_Co$; delay is estimated when input node B takes the transition from high to low and the output node is C). Finally, after considering all process, voltage and temperature variations, their impact on leakage and delay has been observed and depicted in Figure 3.7 and Figure 3.8.

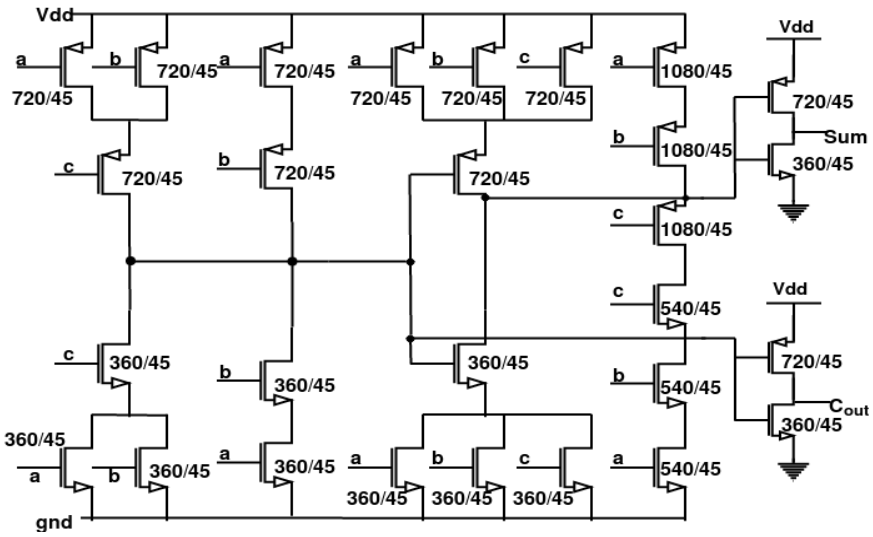


Figure 3.2: Nominal Full Adder cell (45nm). [13]

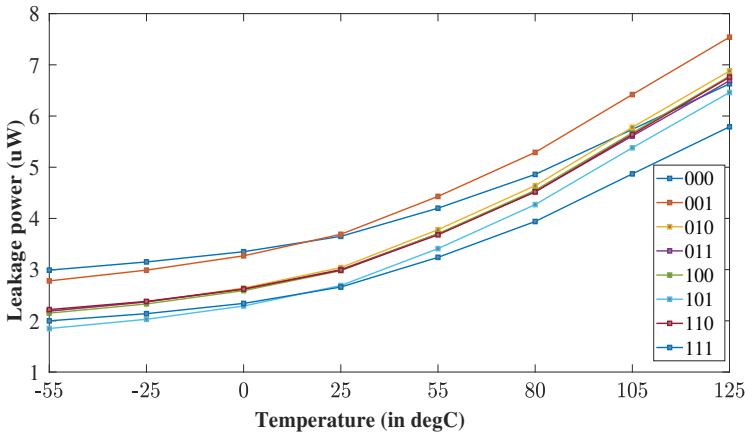


Figure 3.3: Impact of temperature on leakage power of full adder cell

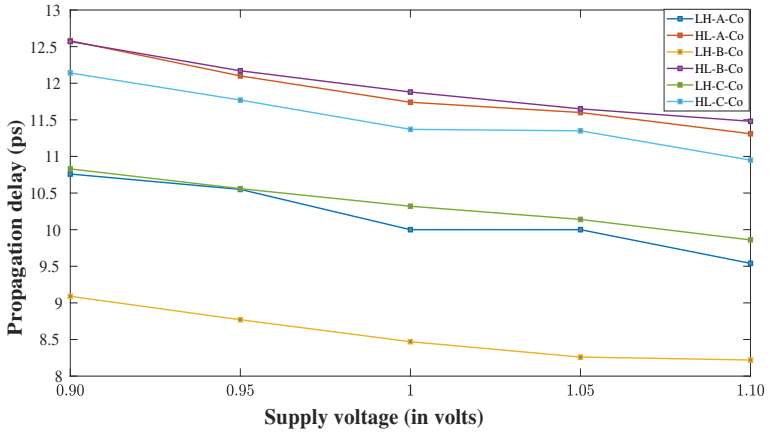


Figure 3.4: Impact of supply voltage on delays of full adder cell

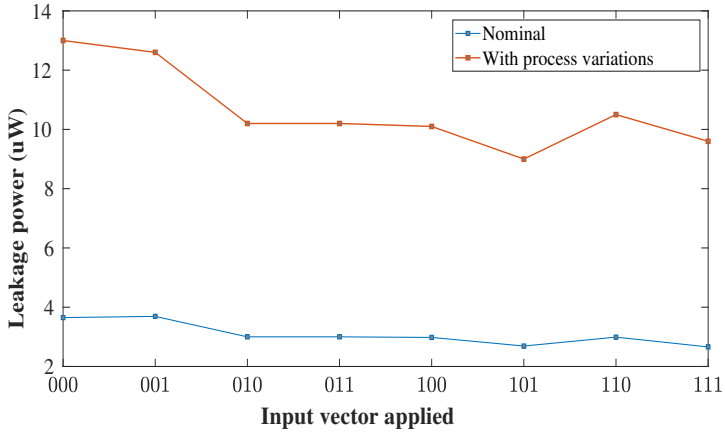


Figure 3.5: Impact of process parameters variations alone on leakage power of full adder cell (NOC)

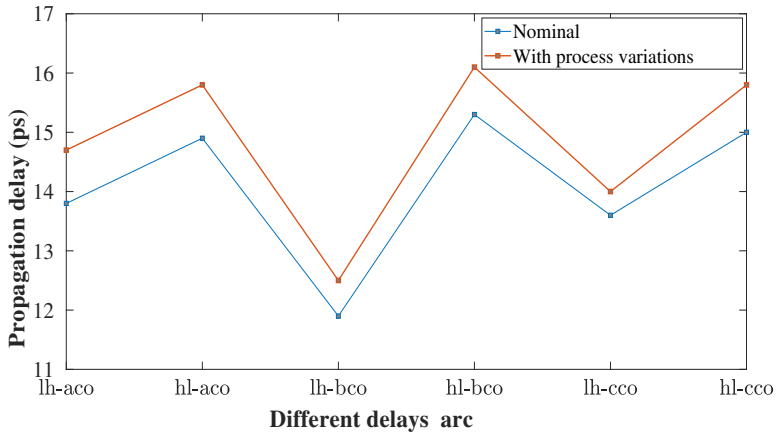


Figure 3.6: Impact of process parameters variations alone on delays of full adder cell (NOC)

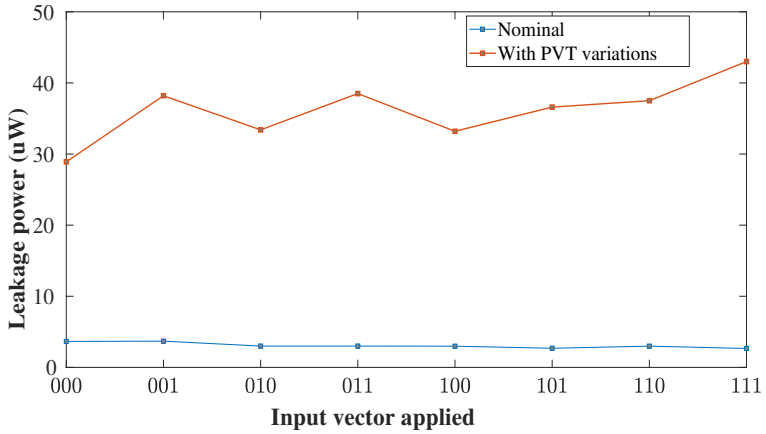


Figure 3.7: Impact of PVT variations on leakage power of full adder cell (WOC)

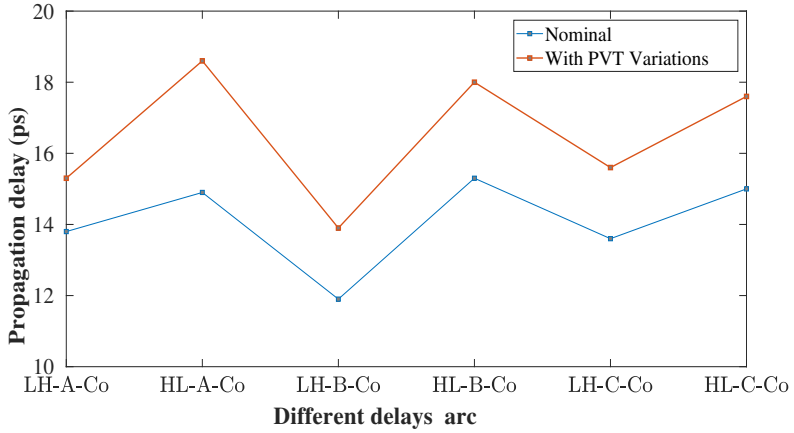


Figure 3.8: Impact of PVT variations on delays of full adder cell (WOC)

3.0.5 Yield

Generally, the yield of integrated circuits can be thought of as the ratio of the number of properly working chips to the total number of chips manufactured. To check the yield, the performance corresponding to those varied process parameters are noticed, the aspect of yield is how much percentage of devices meet the specification. Here, for each set of parameters, circuit simulation is used to determine whether it passes or it fails to meet the specifications. Thus the objective is to increase the yield as much as possible. In this research the yield obtained is between 99% to 100% that is shown clearly in the upcoming section.

Monte Carlo (MC) simulation for generated 50,000 random samples within the range of $\pm 10\%$ of each statistical parameter are performed to observe the effects of variations in device characteristics on circuit performances. The design metrics- leakage and delay exhibit Gaussian distribution with such

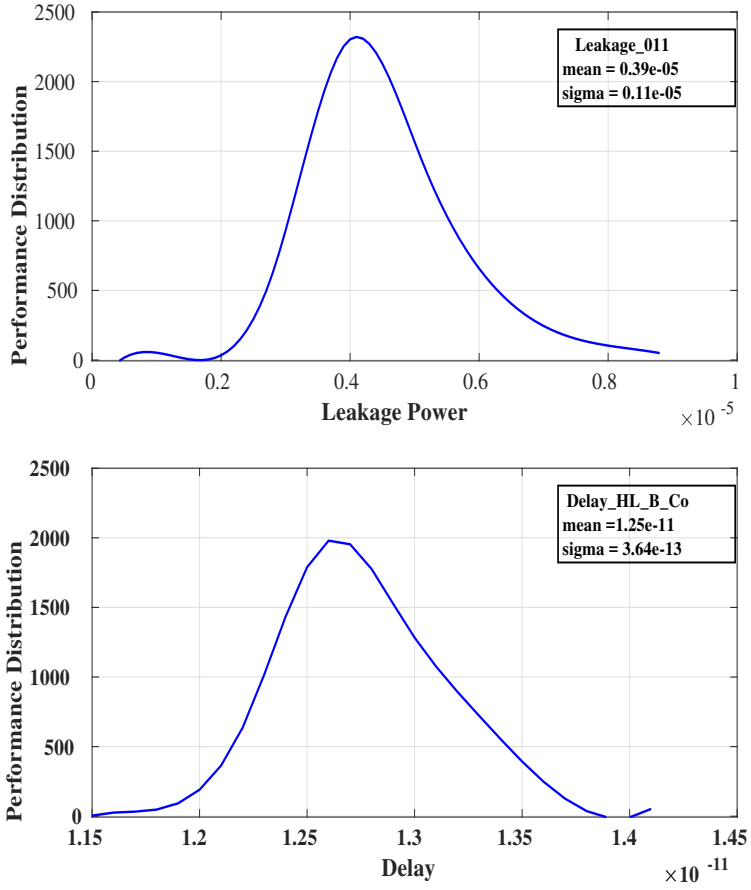


Figure 3.9: Performance distribution plots

variations are displayed in Fig. 3.9 for 45nm technology node at nominal transistor sizing of full adder cell.

Chapter 4

Optimized Standard Digital Cells using Various Algorithms and their Proposed Version

All the conventional techniques discussed earlier for power reduction such as dual threshold, multi threshold, power gating etc each have there own pros and cons in terms of the trad off between power and delay. While, with the help of algorithms we can keep the multiple performance specifications in a bound while optimizing the firness fucntion. Many evolutionary algorithms (EAs), swarm based algorithms have been proposed in past for optimization purpose which are population based heuristics algorithms. To obtain the PVT invariant robust transistor sizing of standard digital cells for optimal leakage, delay and area, we have implemented some of those algorithms. Their working is explained in the below sections.

4.1 Biologically Inspired non-dominated sorting genetic algorithm II (NSGA II)

John Holland in 1960s introduced the Genetic algorithm which is nature inspired based on principle of natural genetics and natural selection. In order to optimize the multiple objective simultaneously, a fast non dominated sorting approach with m number of objective and N population to be sorted was employed named as NSGA [12]. It leads to the complexity of $O(mN^2)$. It's an Elitism evolutionary algorithm which confirms that at least one set of the best individual of the current generation is passed on to the next generation. The pseudo code of NSGA II algorithm is given in Algorithm I. It starts with generating the set of random initial population followed by selection of two parents and applying the process of crossover and mutation on their offspring. The population is classified into various non dominated sorting fronts and then the density of the solutions surrounding a particular solution in the population is estimated using crowding distance measure [25]. Based on the crowding distance metric, the individuals are selected.

Algorithm 1: Pseudo code for NSGA II algorithm

```
1:  procedure NSGA II
2:      Generate random solution for initial population set P;
3:      while stop criterion not satisfied do
4:          for s ←0; s < P; s ← s+2 do
5:              Select two parents P1 and P2;
6:              Execute crossover of P1, P2 and generate offspring O1, O2;
7:              Mutate O1 and O2;
8:              Evaluate O1, O2 and put them in offspring set
9:          end for-
10:         Create a union set from population and offspring and clear population;
11:         Build Pareto fronts pf from union based on the dominance rule;
12:         Set the current front as pf[0];
13:         while Population length < P do
14:             Sort solutions from current front using Crowding distance;
15:             for Each solution in the ordered front do
16:                 if population length < P then
17:                     Include the solution in population;
18:                 end if
19:             end for
20:             Go to the next Pareto front;
21:         end while
22:     end while
23: end procedure
```

4.2 Swarm based Particle Swarm Optimization Algorithm (PSO)

PSO algorithm [15], is basically inspired from nature such as bird flocks or fish schools. The ability of PSO can be judged by two properties, Exploration and Exploitation [10]. Both of them are important to ensure the performance of the algorithm. PSO starts with a group of particles with randomized positions and velocity. Let $x_i(t)$ be the position of i^{th} particle in the search space at time instance t. During every iteration the position of the particle can be changed by adding velocity $v_i(t)$ to it [11].

$$x_i(t) = x_i(t) + v_i(t) \tag{4.1}$$

$$v_i(t) = wv_i(t-1) + c_1r_1(p_i(t) - x_i(t)) + c_2r_2(p_g(t) - x_i(t)) \tag{4.2}$$

In (2), $p_i(t), p_g(t)$ represents i^{th} particle personal best and swarm global best positions at instance t; c_1 and c_2 represents random forces in direction of $p_i(t)$ and $p_g(t)$; r_1 and r_2 are random numbers between [0,1]; w is the inertia weight parameter. High value of w leads to better exploration and low value of w leads to better exploitation.

The algorithm has to run for a certain number of iterations and the p_g position at the end of the process gives the optimum position. PSO is generally preferred over GA because it allows greater diversity and

exploration over a single population. PSO also offers faster convergence over GA. The pseudo code of the PSO algorithm is given in Algorithm 2 below.

Algorithm 2: Pseudo code for PSO algorithm

```
1:  begin
2:    for Each particle do
3:      initialize particle
4:    end for
5:    for Each Particle do
6:      Calculate fitness value
7:      if fitness value is better than the best fitness value (pbest) in history then
8:        Set current value as the new pbest
9:      end if
10:   end for
11:   Choose the particle with the best fitness value of all the particles as the gbest
12:   for Each Particle do
13:     Calculate particle velocity according to equation 4.2
14:     Calculate particle position according to equation 4.1
15:   end for
16:   While maximum iterations or minimum error criterion is not attained
17: end
```

4.3 Swarm based Artificial Bee Colony Algorithm (ABC)

ABC is a swarm intelligence based optimization algorithm proposed in [22] based on the concept of how artificial bees search for their food. The pseudo code of the algorithm is given in Algorithm 3. Here, the bees are of three types; employed bees, onlookers and scouts. The initialization of food sources is given in Eq.4.3.

$$x_{ij} = x_j^{min} + rand(0,1)\big(x_j^{max} - x_j^{min}\big) \tag{4.3}$$

where i is number of food sources and j is the number of design parameters. x_j^{min} and x_j^{max} is the minimum and maximum values of parameter j. The employed bees travel to the food sources in the memory and determine the neighbourhood food source given by Eq. 4.4.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{4.4}$$

where j and k are randomly chosen parameters and neighbourhoods respectively and ϕ_{ij} is a random number within [-1, 1]. In the algorithm, the one with the highest fitness is selected. After reaching the food source they come back to their hive and dance on this area, the onlooker bee waits on the dance area and makes a decision to choose the food source by evaluating the probability function given by Eq. 4.5.

$$p_i = fitness_i(x_i) / \sum_{j=1}^{n} fitness_j(x_j) \tag{4.5}$$

This process will continue until the maximum trial is reached. Scout is a bee that searches food randomly [24] using Eq.4.3. The working of the ABC algorithm and its performance comparison with other evolutionary and the population based algorithm can be referred from the survey work in [23]. Here, the food can be considered as the minimum value of the solution (leakage power) of the fitness function in the optimization process and the bees as the different transistor sizings, which provide those solutions.

Algorithm 3: Pseudo code for ABC algorithm

begin
  Initialize the food sources = 20, trials = 100, population of solutions be x_ij and also lower and upper bound of parameters;
  Evaluate the population;
  Cycle ← 1, MaxIter ← 600;
  for cycle in range(MaxIter) **do**
    for each employed bee i **do**
      Choose a food source x_k in the neighborhood of x_i and select jth parameter out of total design parameters (D);
      Generate a food source v_i in the neighborhood of x_i and x_k using Eq. 4.4;
      Evaluate fitness and select the best between x_i and v_i and increment trails;
    end for
    for each onlooker bee i **do**
      Select a food source x_i based on the probability p_i using Eq. 4.5;
      Choose a food source x_k in the neighborhood of x_i;
      Generate a food source v_i in the neighborhood of x_i and x_k by Eq. 4.4;
      Evaluate fitness and select the best between x_i and v_i and increment trails;
    end for
    if there exists an abandoned food source or limit has been reached; **then**
      Scout bee determine a new food source by Eq. 4.3;
    end if
    Memorize the best solution achieved so far.
  end for
end

4.4 Thermo Based Simulated Annealing Algorithm (SA)

In the Annealing process, a metal is first heated until it melts and then it is cooled down slowly so that atoms attain the equilibrium state, as later atoms move less freely with less energy than earlier. The pseudo code of the SA algorithm is displayed in Algorithm 4. It begins by generating the random initial solutions. Then random changes are done to the current solution x_c that is proportional to current temperature. Thus the fitness function value obtained at a current solution is compared with that of the new solution x_n obtained. The current solution gets replaced by a new solution if the new solution is better or if the probability function given in equation (2) is higher than a randomly generated number

between 0 and 1.

$$p = \frac{1}{1 + e^{\frac{-(E(c,T)-E(n,T)}{T}}} \tag{4.6}$$

where E(c,T) and E(n,T) are the corresponding energy values of current state (c) and new state (n) respectively.

To avoid premature convergence, the rate of temperature reduction should be slow, as given in equation (4.7).

$$T_{i+1} = \alpha * T_i \tag{4.7}$$

where i+1 is the number of iterations and α is the cooling rate ($0 < \alpha < 1$) taken as 0.7 in this paper. In this way, some worse solutions may be accepted at the beginning of the SA, but in the end, only improved ones are allowed. Further information on Multi-Objective SA (MOSA) can be easily found in the survey work in [35].

Algorithm 4: Pseudo code for SA algorithm

```
1:  begin
2:    Initialize(temperature, starting point)
3:    while cool iteration <= max iterations do
4:      temp iteration = 0
5:      while temp iteration <= nrep do
6:        temp iteration = temp iteration +1
7:        Select a new point in the neighborhood
8:        Compute the current cost (for new point)
9:        δ = current cost - previous cost
10:       if δ¡0 then
11:         accept neighbor
12:       else
13:         accept with the probability given in Eq.4.6
14:       end if
15:     end while
16:     Reduce temperature using Eq.4.7
17:   end while
18: end
```

4.4.1 Proposed Multiobjective Simulated Annealing using Gradient Descent (MO-SAGRAD) for Improved Performances

We have implemented Archived multi-objective simulated annealing (AMOSA) [7] algorithm and the obtained optimal transistor sizing is applied as the initial point for the numerical gradient descent algorithm (as explained in Algorithm 5) to further converge the objective function globally and in order to avoid the stuck at local minima as a novel concept in SA. The multi-objective optimization generates the Pareto optimal (PO) solutions that is a solution where no exist other solution in the search space

that dominates it [13]. Here we are using it to optimize leakage and delay which generally are in trade-off thus the generated PO solutions are stored in an archive hence named as AMOSA and it is been explained through the pseudo codes in the Algorithm 6.

The initial/nominal sizing of the transistors for all the digital cells in this paper have been taken from [4] which has been calculated for equal rise and fall times.

$$\triangle dom_{avg} = \frac{(\sum_{i=1}^{k} \triangle dom_{i,new-pt}) + \triangle dom_{curr-pt,new-pt}}{(k+1)} \tag{4.8}$$

$$prob = \frac{1}{1 + e^{(\triangle dom_{avg} * temp)}} \tag{4.9}$$

$$\triangle dom_{avg} = \frac{(\sum_{i=1}^{k} \triangle dom_{i,new-pt})}{k} \tag{4.10}$$

$$prob = \frac{1}{1 + e^{(-\triangle dom_{min})}} \tag{4.11}$$

where k, is the total number of points in an archive which dominates new-pt and $\triangle dom_{min}$ is minimum of the difference of domination amounts between new point and k points.
new-pt = new point
curr-pt = current point

4.5 Implementation of algorithms on digital circuits

The above mentioned algorithms have been implemented for 2 and 3 input CMOS NAND, NOR, AND, OR, XOR and full adder cell to obtain the robust transistor sizing by varying the width and length of each transistor. The optimal transistor sizing obtained for basic digital circuits can be used to implement different standard benchmark circuits and that we have displayed for few cells such as C17 cell, Parity checker, C432 (Interrupt Controller), 4-bit Multiplier, 32-bit ripple carry adder, 32-bit carry select adder, 32-bit carry save adder etc. The fitness function considered to be minimized is the average leakage power for all the possible input combination for corresponding cell keeping the propagation delay in the initial highest delay. Notably, our methodology targets semi-custom digital design: once the cell library layout are designed, they can be placed wherever needed in a semi-custom design by technology mapping tools.

The procedure can be easily understood by an example of a CMOS 1 bit full adder cell at $32\mathrm{nm}$ technology shown in Figure 3.2 which consist of 28 transistors connected in pull up and pull down network. Using the algorithms the leakage power for the average of all possible eight input combinations 000, ..., 111 have been optimized by varying the design parameters ie. width (W) and length (L) of the transistors in the range given in Table 2.1, with predefined step size in the respective technology nodes,

Algorithm 5: Numerical Gradient Descent

begin
  $f \rightarrow$ fitness function to be minimized; $X \rightarrow$ search space
  $p_{new} \rightarrow$ initial point; $h \rightarrow$ small value; $\alpha \rightarrow$ step-size
  while True **do**
    for j $\leftarrow$ 1 to problem dimension (d) **do**
      $y \leftarrow$ [$p_{new}(1)$,..,$(p_{new}(j) + h)$,..,$p_{new}(d)$]
      $g(j) \leftarrow \frac{f(y) - f(p_{new})}{h}$
      if $g(j) > 0$ **then**
        $temp(j) \leftarrow p_{new}(j)$ - α
      else
        $temp(j) \leftarrow p_{new}(j)$ + α
      end if
    end for
    if $temp \approx p_{new}$ **then**
      $p_l \leftarrow p_{new}$
    end if
  end while
  $p_{new} \leftarrow temp$
  return p_l
end

keeping all delays in the bound (critical path delay at initial transistor sizing of respective SDCs). The netlist of full adder for 32nm technology with initial (nominal) sizing is given in Appendix B and the 32nm Metal-gate/High K PTM model used is given in Appendix A.

At $32\mathrm{nm}$ technology node, the nominal operating conditions (NOC) and worst operating conditions (WOC) are defined in Table 3.1 and process variations considered are displayed in Table 4.1.

For the CMOS 1 bit full-adder cell, the WOC conditions for leakage and delays are observed at (125^oC, 1.10V) and (125^oC, 0.90V) respectively. It is evident from the Table 4.2 that the average leakage power has been reduced up to 14% and 27% at NOC and WOC with the application of PSO and up to 33% and 38% with SA respectively. The reduction percentage with ABC NSGA algorithms compared to the initial for NOC and WOC are 55% and 80% 56% and 81% respectively.The same observation is depicted in Fig. 4.1. In FA the initial delay of the critical path for NOC and WOC is $10.7\mathrm{ps}$ and $11.7\mathrm{ps}$ respectively, which has been taken as a bound for maximum delay during the complete optimization procedure. The capacitive load considered is FO4 minimum sized inverter cells.

Figure 4.2 shows leakage power reduction for basic logic cells using vaious reported algorithm for 32nm MGK technology for both NOC and WOC condition. The average leakage power reduction achieved is up to 88% keeping the propagation delays at the initial state displayed in Figure 4.3.

Algorithm 6: The Archived Multi-objective Simulated Annealing (AMOSA) Algorithm

begin
  Set T_{max}, T_{min}, α, iters
  Initialize the archive
  current-pt = random(archive)
  while (temp $> T_{min}$) **do**
    for i=0; i<iter; i++ **do**
      new-pt = perturb(current-pt)
      if current-pt. dominates new point **then**
        obtain $\triangle dom_{avg}$ and prob from eq. (3), (4)
        new-pt $\leftarrow$ current-pt with probability = prob
      end if
      if current-pt and new-pt are non dominating to each other **then**
        if new-pt is dominated by k (k$\geq$ 1) points in archive **then**
          obtain probability and $\triangle dom_{avg}$ from eq. (4), (5)
          Set new-pt as current-pt with probability=prob
        end if
        if new-pt is non dominant w.r.t. all the points in the archive **then**
          new-pt $\leftarrow$ current-pt and add new-pt to archive
        end if
        if new-pt dominates K (K$\geq$ 1) points of the archive **then**
          new-pt $\leftarrow$ current-pt and add it to archive
          Remove all the k dominated points from the archive
        end if
      end if
      if new-pt dominates current-pt **then**
        if new point is dominated by k (k$\geq$ 1) point in archive **then**
          Calculate $\triangle dom_{min}$ and prob from eq. (6)
          Set point of archive which correspond to $\triangle dom_{min}$ as current-pt with probability=prob
        else
          set new-pt as current-pt
        end if
        if new-pt is non dominating w.r.t. points in archive **then**
          new-pt $\leftarrow$ current-pt and add it to archive
          if current-pt is in archive, remove it from archive **then**
          end if
        end if
        if new-pt dominates k other points from archive **then**
          new-pt $\leftarrow$ current-pt and add it to archive
          Remove all k dominated points from archive
        end if
      end if
    end for
    temp = α * temp.
  end while
end

Table 4.1: Process variation considered ($32\mathrm{nm}$ tech.)

Sr. No.	Process parameter	Lower deviation	Nominal	Higher deviation
1	$lint$	2.35e-09	2.7e-09	3.05e-09
2	$wint$	4.34e-09	5e-09	5.59e-09
3	$toxp$	5.74e-09	6.5e-09	7.26e-09
4	$toxe_n$	7.96e-10	9e-10	10.28e-10
5	$toxe_p$	7.97e-10	9.2e-10	10.37e-10
6	$toxref_n$	7.96e-10	9e-10	10.28e-10
7	$toxref_p$	7.97e-10	9.2e-10	10.37e-10
8	xj	1.22e-08	1.4e-08	1.57e-08
9	$ndep_n$	5.77e+18	6.5e+18	7.26e+18
10	$ndep_p$	2.45e+18	2.8e+18	3.12e+18

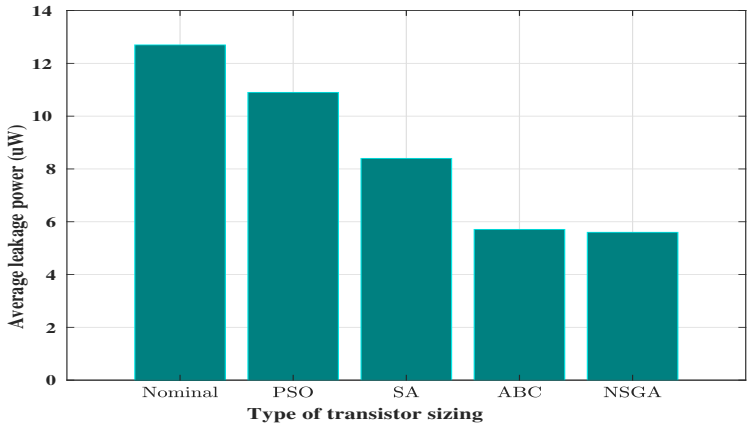
The obtained optimal sizing for basic cells are used to optimize the various complex cells such as C17 cell, Parity Checker, C432, 4-bit multiplier, 32 bit ripple carry adder etc. and their results are reported in Table 4.3.

Table 4.2: Results obtained for CMOS 28T 1-bit Full Adder cell at 32nm technology

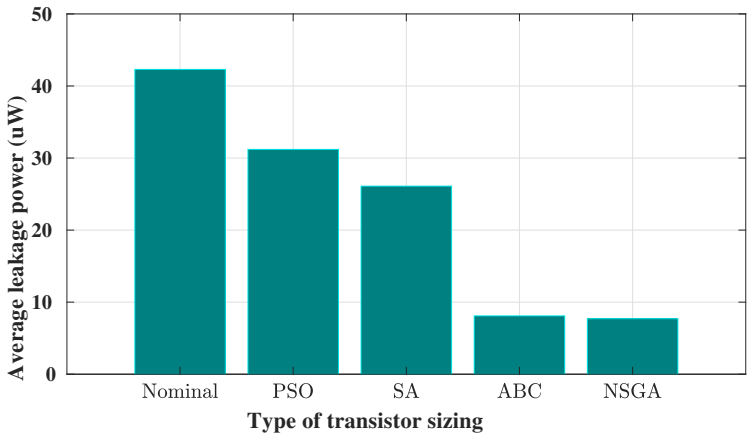
Performances	Nominal		PSO		SA		ABC		NSGA	
	NOC	WOC	NOC	WOC	NOC	WOC	NOC	WOC	NOC	WOC
$Leak\_000$ (μW)	15.8	38.6	11.5	30.0	9.4	26.9	6.72	8.9	6.6	8.7
$Leak\_001$ (μW)	15.2	46.5	12.0	28.1	9.1	30.2	6.07	8.4	5.9	8.2
$Leak\_010$ (μW)	12.2	40.1	10.5	25.5	7.7	23.3	5.36	7.6	5.1	7.3
$Leak\_011$ (μW)	12.2	45.4	10.2	35.6	7.9	26.1	5.69	8.5	6.0	7.4
$Leak\_100$ (μW)	12.0	39.4	10.7	26.3	8.5	21.09	6.07	8.4	6.2	8.3
$Leak\_101$ (μW)	10.5	42.3	9.21	34.3	7.5	28.4	5.09	7.6	4.7	6.8
$Leak\_110$ (μW)	12.3	42.2	11.3	35.5	9.0	29.9	5.24	7.6	5.1	7.5
$Leak\_111$ (μW)	11.0	44.1	11.9	33.9	8.4	23.1	5.41	7.3	5.2	7.1
$Average$ (μW)	12.7	42.3	10.9	31.2	8.4	26.1	5.71	8.1	5.6	7.7
$T_{plh\_a\_Cout}$ (ps)	8.9	9.4	7.5	8.4	7.8	8.9	10.6	11.3	10.5	11.4
$T_{phl\_a\_Cout}$ (ps)	10.5	11.7	9.2	10.3	8.9	10.1	10.5	11.7	10.5	11.6
$T_{plh\_b\_Cout}$ (ps)	7.3	8.3	7.8	9.1	8.3	9.5	9.8	10.3	9.8	10.7
$T_{phl\_b\_Cout}$ (ps)	**10.7**	**11.7**	6.5	7.1	6.7	7.4	10.6	11.7	10.7	11.6
$T_{plh\_c\_Cout}$ (ps)	8.9	10.1	10.6	11.7	10.5	11.7	10.0	10.4	10.2	10.9
$T_{phl\_c\_Cout}$ (ps)	10.2	10.8	8.8	9.8	9.1	10.1	10.4	11.5	10.6	11.6

Table 4.3: Average leakage power obtained for benchmark circuits at $32\mathrm{nm}$ technology

Sr. No.	Digital Circuit Type	Average Leakage Power (μW)									
		Nominal		PSO		SA		ABC		NSGA	
		NOC	WOC	NOC	WOC	NOC	WOC	NOC	WOC	NOC	WOC
1	$C17$	10.4	33.8	6.82	23.9	5.1	18.2	3.26	11.8	2.5	9.5
2	$Parity\ Checker$	5.44	5.65	4.91	4.98	4.9	5.0	3.30	3.51	3.1	3.3
3	$C432\ (Interrupt\ Controller)$	173	242	168	240	170	240	163	247	150	230
4	$4-Bit\ Multiplier$	12.0	12.2	9.3	10.5	8.9	9.9	6.27	7.63	5.9	7.1
5	$32-Bit\ Ripple\ Carry\ Adder$	406	1246	282	845	290	860	182	476	160	450
6	$32-Bit\ Carry\ Skip\ Adder$	750	2330	549	1608	530	1400	374	986	290	700
7	$32-Bit\ Carry\ Select\ Adder$	810	2540	593	1705	450	1400	382	1006	350	805

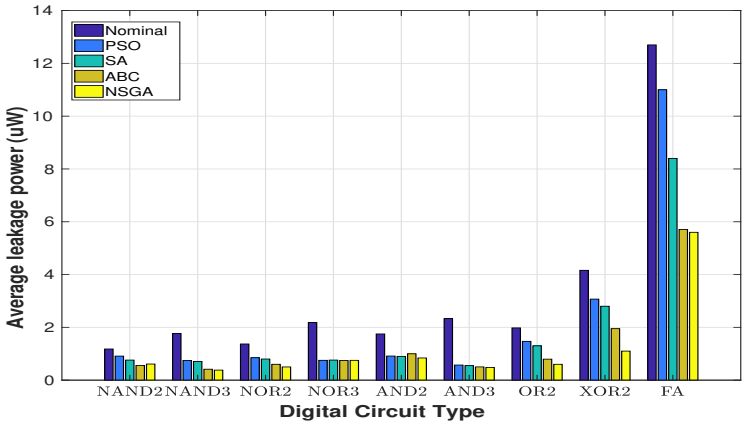


(a)

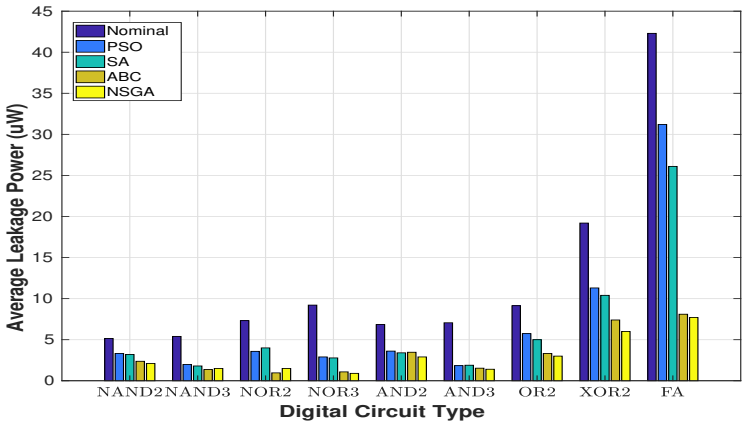


(b)

Figure 4.1: Leakage power reduction for 32nm full adder cell using various algorithms a) for NOC b) for WOC



(a)



(b)

Figure 4.2: Average leakage power reduction for various logic cells using various algorithms for 32nm technology a) for NOC b) for WOC

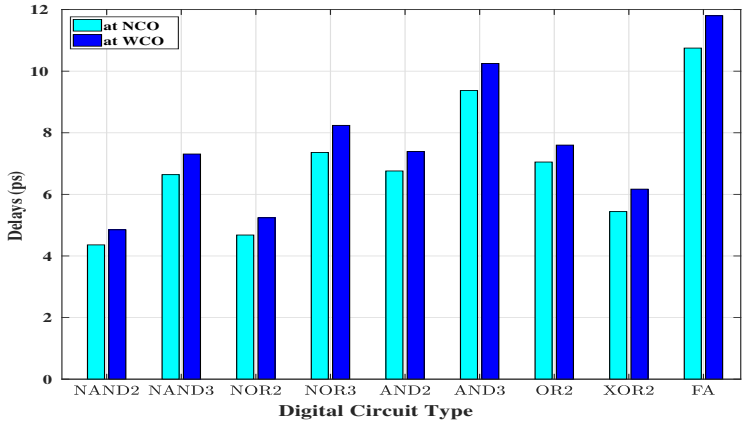


Figure 4.3: Initial highest delays taken as bound in basic CMOS digital circuits at $32\mathrm{nm}$ technology

Chapter 5

Proposed Technique of Stand by Mode Leakage Power Reduction for PVT Invariant CMOS 1-bit Full Adder cell

A robust transistor sizing with good yield is required for high performance designs. The transistor sizing is constrained by the specifications for leakage, delay and area. All these performance figures should be satisfied for a range of operating variation (temperature and supply voltage), the design parameters (transistor width and length) and process parameter fluctuations. Here, it is achieved with the help of efficient swarm based optimization algorithms. The swarm based algorithms such as Ant colony optimization, Particle swarm optimization (PSO) are generic population based metaheuristic algorithms. The proposed design approaches involve two different ways to reduce the standby mode leakage power and to find optimal transistor sizing; Input Vector Control (IVC) and Power Gating (PG) [17], [16]. Both the reported techniques involve ABC algorithm [23] to find its associated performance parameters.

5.1 Proposed Approaches

5.1.1 Approach I -

For IVC approach, the design flow for estimating the optimal sizing of each transistor in circuit under test is displayed in Fig. 5.1. It involves two steps; MLV computation and MLV implementation. MLV computation initializes with defining the performance constraints i.e. the leakages and delays constraints for normal operating condition (NOC) and worst operating conditions (WOC) considering the process variations. The effect of individual process parameter with $\pm 3\sigma$ variation on leakage and delay is observed and the worst process values for leakage and worst process values for delays are considered in SPICE netlist. The initial sizing of used 28-T nominal full adder mirror type circuit has been taken from [4] is shown in figure 3.2. The minimum leakage value for out of all 8 input vectors (000 to 111) in the full adder cell is assigned as L_{min}. Then using reported algorithm, leakage at one input vector is minimized at a time while keeping leakage across other inputs under constraints and checked for leakage lesser than L_{min} to find MLV and thus L_{min} gets replaced by the leakage correspond to the MLV. Thus MLV computation step gives MLV and final sizing of all 28 transistors of the full adder cell.

Coming to next step of IVC implementation, The obtained MLV is applied to the circuit by applying Input Vector Control technique that is implemented using transmission gate technology so as to drive the circuit in standby mode with minimum power consumption. Since a single NMOS or PMOS pass transistor suffers from a threshold drop while transmission gate fixes this problem at the cost of two transistors in parallel [40]. Again, using algorithm minimized the stand by mode leakage (ie. leakage at MLV) keeping active mode leakage in constraint and delays under the bound of maximum initial delays and obtained the sizing of externally used sleep transistors. It needs to be noted that, even after adding the multiple sleep transistors in 1 bit full adder cell, finally the leakage is minimized while keeping the delays and other parameters under the bound of nominal ones. By comparing the leakage obtained at MLV at final obtained transistor sizing with that of at the initial sizing, the reduction in stand by mode leakage can be obtained.

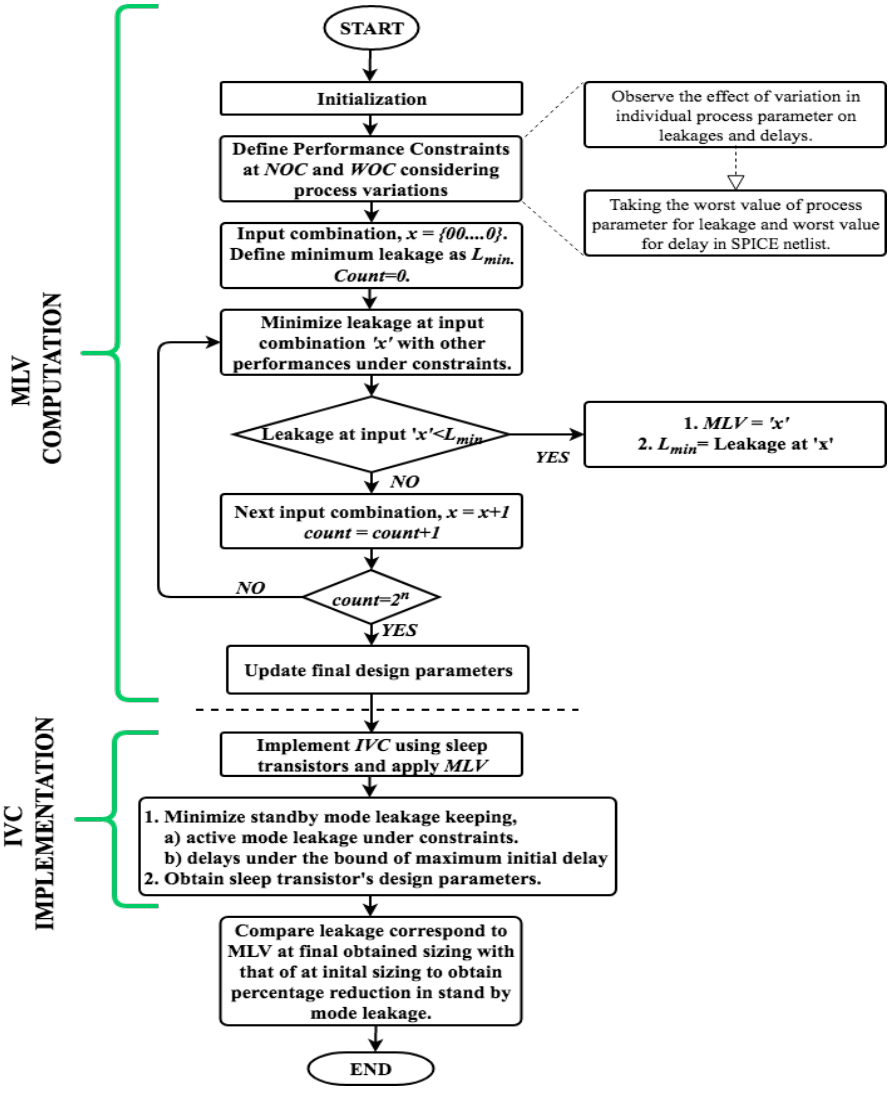


Figure 5.1: Design Flow

5.1.2 Approach II -

The second design approach involves stand-by mode leakage reduction using power gating technique by shutting off the idle blocks. This is very esteemed technique that is efficient in reducing the leakage power dissipation but with the condition, if sleep transistors are sized carefully, otherwise it may lead to some negative effects also.

Generally, the power gating implementation requires a multi-threshold CMOS technique. The main logic blocks are implemented using low threshold, high performance transistors whereas the sleep transistors connected between PUN and supply and PDN and ground are provided with the high threshold voltage. It involves few flaws such as increased area, performance penalty and several noise effects etc [20]. However, In this paper the authors have tried to overcome the existed problems of power gating with the help of proper transistor sizing achieved by optimization algorithms. The authors have tried both the particle based optimization and artificial bee colony based algorithm separately to size the main logic blocks transistors as well as the sleep transistors in the very next step. Unlike the conventional power gating based technique, the threshold voltage of all the transistors including sleep transistors are taken as same value so as to avoid the difficulty of using multiple threshold voltage levels in the single chip. It facilitates the reduction of standby mode leakage by turning off the sleep transistors and creates a virtual supply (vdd) and ground (gnd) while keeping the other performances under constraint. In Active mode, the sleep transistors are turned on thus the circuit functions as usual. Again, the optimal sizing of all transistors in the full adder including sleep transistors calculated. Here also, the initialization steps are same as mentioned in approach I as defining the performance constraints for NOC and WOC with process parameter variations.

Both the approaches guarantees for the high yield which is reported in result section; as the performance constraints taken in the first step are for the worst value of all the individual process parameter variation taken together. Thus the design parameters calculated using the algorithms are chosen somewhere in the feasible region which satisfies for the nominal conditions also.

5.2 Results and Discussions

We have applied both the proposed approaches on 28-T full adder cell shown in Fig. 3.2, their nominal sizing for $45\mathrm{nm}$ technology have been referred from [4] which have been calculated by keeping the equal rise and fall times. Similarly the nominal sizing for other $32\mathrm{nm}$ and $22\mathrm{nm}$ technology nodes have been calculated proportionally. The performances obtained after optimization are compared with those at nominal sizing. The active mode leakage and delay components are depicted in Table 5.1 - 5.3 for different technology nodes and their respective transistor sizing have been displayed in Figure. 5.2 - 5.7 respectively. In order to obtain the robust transistor sizing, the leakage has been minimized keeping all the delay components within the bound of maximum initial delay for respective operating condition which are highlighted in the tables.

From Table 5.1. for 45nm technology, the maximum propagation delay at Nominal sizing observed is 13.3ps and 15.4ps for NOC and WOC conditions respectively which are taken as limit for all the delay components in the both the approaches while performing the optimization. The initial standby mode leakage at nominal sizing is given as $9.59\mu\text{W}$ and $43.0\mu\text{W}$ for NOC and WOC respectively. The MLV obtained is 111 which leads to the stand by mode leakage power of $4.52\mu\text{W}$ using approach I and $3.82\mu\text{W}$ using approach II at NOC and for the WOC it is obtained as $12.4\mu\text{W}$ using approach I and $14.7\mu\text{W}$ using approach II. Thus the overall standby mode leakage reduction using approach I is around 53% and 70% for NOC and WOC respectively while using approach II, the reduction is approx 60% and 66% for NOC and WOC respectively. The average active mode leakage power as depicted in Table 5.1 is reduced for NOC and WOC upto 57% and 58% using approach I and upto 51% and 60% using approach II respectively. The author has compared the percentage reduction in average active leakage power (45nm) using the proposed approaches with the average leakage obtained using worst case distance method in [4]. In [4], the percentage reduction in average leakage obtained at NOC and WOC was 52% and 74% respectively but at the cost of 16% increment in the highest delay value. While, with the proposed approaches the reduction obtained is 57% and 60% without any penalty in the propagation delay.

Similarly, for 32nm and 22nm also the results have been observed as depicted from Table 5.2 and Table 5.3 respectively. The amount of leakage reductions are plotted using matlab plots and can be seen in the Fig. 5.8.

Power Consumption in Identification in the Optimization Process:
The reported work is based on the simulation (iterations), thus the power consumed in the identification of power consuming circuit can be given as the product of power dissipated by circuit under test (CUT) and number of iterations ie.

$$Power\,Consumed = (Power\,dissipated\,by\,CUT)\;X\;(Number\,of\;Iterations) \tag{5.1}$$

where Number of iterations depend on circuit's topology, number of performance figures, number of inputs in the circuit etc.
For example, In a 1 -bit full adder circuit (32nm) the power dissipated by the CUT is $7.8\mu\text{W}$ and if 1000 iterations are used then the power dissipation in identification would be around 7.8e-3 watt. Though it seems a big value but the circuit with optimal sizing is ready, it can be used in millions of the chips, for example in ABC algo the number of iterations used are 8 Lakh. Power dissipated by CUT is the power dissipated by the circuit under test at nominal sizing before optimization as displayed in table present in Appendix 3.

The area at nominal sizing observed is 753sq.fm which has been reduced up to 360sq.fm and 378sq.fm using approach I and approach II respectively. Thus, even after addition of external sleep transistors, the area has got reduced with out giving any penalty to the propagation delay. The additional area due to sleep transistors used in approach I is 97sq.fm.

To make sure the robustness of the obtained transistor sizing using proposed approaches for different technology nodes, the Monte Carlo simulations for randomly generated 50,000 samples have been applied. The Yield achieved is between 99.6% to 99.9% using both the approaches for all reported technology nodes.

Table 5.1: Results obtained at $45\mathrm{nm}$ technology using proposed approaches.

Performances (Leakage in Watts, Delay in seconds)	Nominal (NOC)	Optimized-Approach I (NOC)	Optimized-Approach II (NOC)	Nominal (WOC)	Optimized-Approach I (WOC)	Optimized-Approach II (WOC)
Leakage_000	1.30e-05	4.74e-06	5.67e-06	2.89e-05	1.69e-05	1.22e-05
Leakage_001	1.26e-05	4.84e-06	5.41e-06	3.82e-05	1.54e-05	1.26e-05
Leakage_010	1.02e-05	4.32e-06	4.98e-06	3.34e-05	1.45e-05	1.22e-05
Leakage_011	1.02e-05	4.80e-06	5.12e-06	3.85e-05	1.58e-05	1.72e-05
Leakage_100	1.01e-05	4.05e-06	5.27e-06	3.32e-05	1.38e-05	1.25e-05
Leakage_101	9.00e-06	4.16e-06	4.87e-06	3.66e-05	1.54e-05	1.71e-05
Leakage_110	1.05e-05	4.53e-06	5.12e-06	3.75e-05	1.69e-05	1.79e-05
Leakage_111	9.59e-06	4.52e-06	5.03e-06	4.30e-05	1.24e-05	1.58e-05
Avg. Leakage	**1.06e-05**	**4.49e-06**	**5.18e-06**	**3.62e-05**	**1.51e-05**	**1.47e-05**
Delay_LH_a	1.14e-11	1.29e-11	1.32e-11	1.26e-11	1.42e-11	1.43e-11
Delay_HL_a	**1.33e-11**	1.33e-11	1.26e-11	**1.54e-11**	1.49e-11	1.45e-11
Delay_LH_b	9.60e-12	1.10e-11	1.10e-11	1.05e-11	1.23e-11	1.27e-11
Delay_HL_b	1.33e-11	1.30e-11	1.28e-11	1.48e-11	1.50e-11	1.44e-11
Delay_LH_c	1.12e-11	1.22e-11	1.33e-11	1.21e-11	1.33e-11	1.38e-11
Delay_HL_c	1.26e-11	1.24e-11	1.05e-11	1.40e-11	1.39e-11	1.17e-11

Table 5.2: Results obtained at 32nm technology using proposed approaches.

Performances (Leakage in Watts, Delay in seconds)	Nominal (NOC)	Optimized-Approach I (NOC)	Optimized-Approach II (NOC)	Nominal (WOC)	Optimized-Approach I (WOC)	Optimized-Approach II (WOC)
Leakage_000	1.59e-05	5.85e-06	7.60e-06	3.86e-05	2.29e-05	1.78e-05
Leakage_001	1.53e-05	5.92e-06	7.33e-06	4.66e-05	1.94e-05	1.78e-05
Leakage_010	1.23e-05	5.10e-06	6.49e-06	4.01e-05	1.80e-05	1.61e-05
Leakage_011	1.22e-05	5.72e-06	6.76e-06	4.54e-05	2.04e-05	2.10e-05
Leakage_100	1.21e-05	5.24e-06	6.73e-06	3.94e-05	1.79e-05	1.63e-05
Leakage_101	1.06e-05	5.21e-06	6.22e-06	4.24e-05	1.85e-05	2.03e-05
Leakage_110	1.23e-05	5.47e-06	6.51e-06	4.23e-05	1.80e-05	2.23e-05
Leakage_111	1.10e-05	5.45e-06	6.31e-06	4.42e-05	1.23e-05	1.79e-05
Avg. Leakage	**1.27e-05**	**5.50e-06**	**6.74e-06**	**4.24e-05**	**1.84e-05**	**1.87e-05**
Delay_LH_a	8.91e-12	9.69e-12	1.05e-11	9.44e-12	1.05e-11	1.19e-11
Delay_HL_a	1.06e-11	1.07e-11	9.32e-12	1.18e-11	1.19e-11	1.04e-11
Delay_LH_b	7.38e-12	8.87e-12	9.27e-12	8.34e-12	9.93e-12	1.06e-11
Delay_HL_b	**1.08e-11**	1.08e-11	1.01e-11	**1.19e-11**	1.19e-11	1.11e-11
Delay_LH_c	8.96e-12	9.92e-12	1.07e-11	1.02e-11	1.07e-11	1.14e-11
Delay_HL_c	1.02e-11	1.02e-11	9.51e-12	1.09e-11	1.12e-11	1.04e-11

Table 5.3: Results obtained at 22nm technology using proposed approaches.

Performances (Leakage in Watts, Delay in seconds)	Nominal (NOC)	Optimized-Approach I (NOC)	Optimized-Approach II (NOC)	Nominal (WOC)	Optimized-Approach I (WOC)	Optimized-Approach II (WOC)
Leakage_000	6.69e-06	2.35e-06	2.80e-06	1.37e-05	4.99e-06	5.38e-06
Leakage_001	7.43e-06	3.05e-06	3.22e-06	1.45e-05	5.74e-06	5.95e-06
Leakage_010	6.05e-06	2.84e-06	2.95e-06	1.27e-05	4.76e-06	5.62e-06
Leakage_011	7.02e-06	3.95e-06	3.53e-06	1.25e-05	6.07e-06	5.71e-06
Leakage_100	6.83e-06	3.30e-06	3.35e-06	1.32e-05	6.02e-06	5.98e-06
Leakage_101	6.98e-06	3.91e-06	3.57e-06	1.19e-05	6.66e-06	5.63e-06
Leakage_110	8.71e-06	4.63e-06	4.09e-06	1.41e-05	6.82e-06	6.21e-06
Leakage_111	9.72e-06	5.49e-06	4.66e-06	1.27e-05	7.17e-06	6.46e-06
Avg. Leakage	**7.43e-06**	**3.69e-06**	**3.52e-06**	**1.32e-05**	**6.03e-06**	**5.87e-06**
Delay_LH_a	9.02e-12	1.04e-11	9.26e-12	1.03e-11	1.22e-11	1.06e-11
Delay_HL_a	**1.08e-11**	1.02e-11	1.10e-11	**1.20e-11**	1.14e-11	1.19e-11
Delay_LH_b	7.63e-12	9.28e-12	8.73e-12	8.47e-12	1.08e-11	9.54e-12
Delay_HL_b	1.06e-11	1.08e-11	1.03e-11	1.15e-11	1.18e-11	1.13e-11
Delay_LH_c	8.78e-12	9.94e-12	9.10e-12	9.50e-12	1.13e-11	9.85e-12
Delay_HL_c	1.02e-11	1.05e-11	9.49e-12	1.12e-11	1.15e-11	1.05e-11

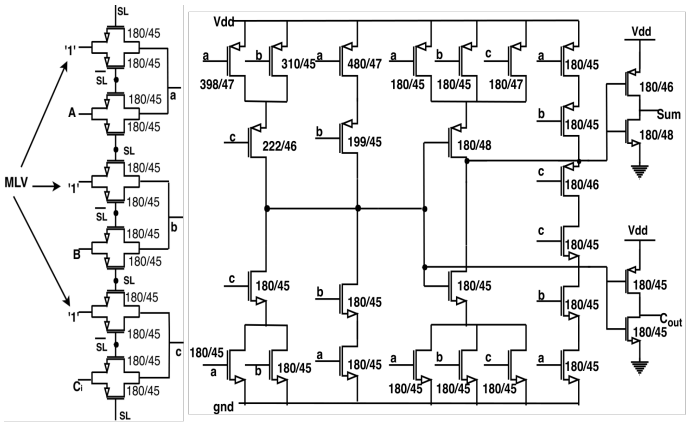


Figure 5.2: Robust transistor sizing obtained in 28-T full adder cell using approach I for 45nm tech.

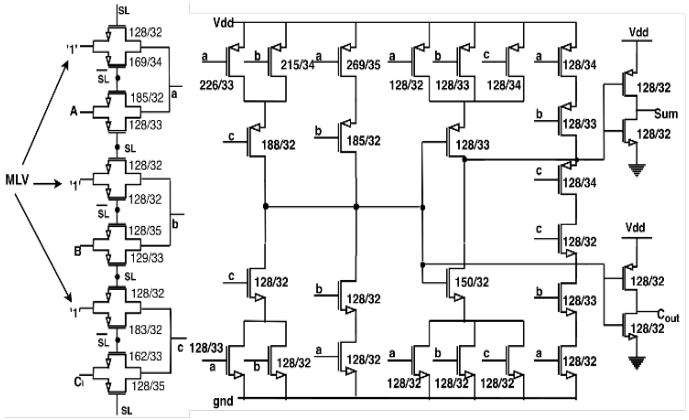


Figure 5.3: Robust transistor sizing obtained in 28-T full adder cell using approach I for 32nm tech.

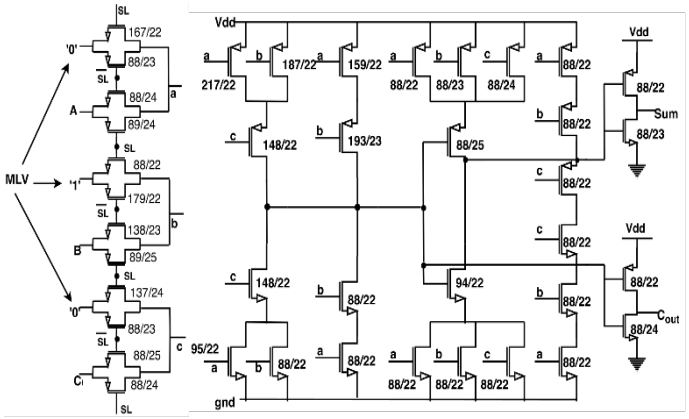


Figure 5.4: Robust transistor sizing obtained in 28-T full adder cell using approach 1 for 22nm tech.

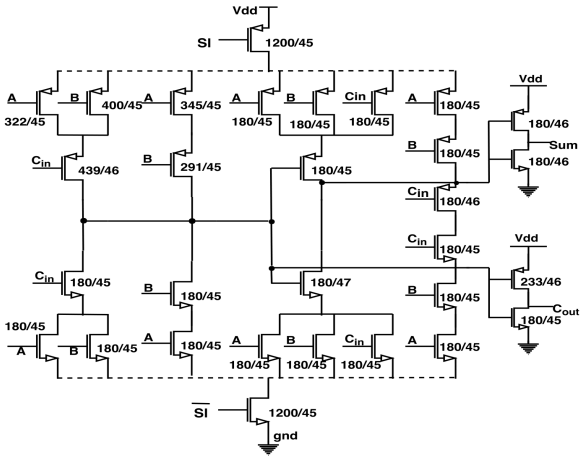


Figure 5.5: Robust transistor sizing obtained in 28-T full adder cell using approach II for 45nm tech.

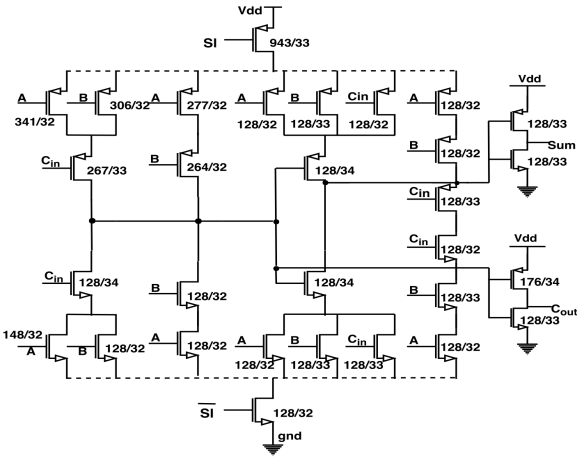


Figure 5.6: Robust transistor sizing obtained in 28-T full adder cell using approach II for 32nm tech.

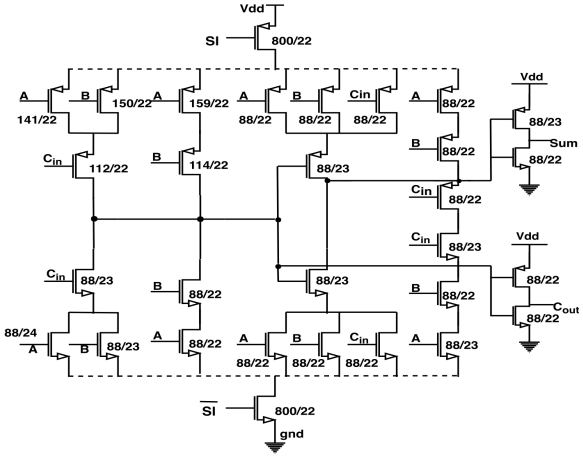
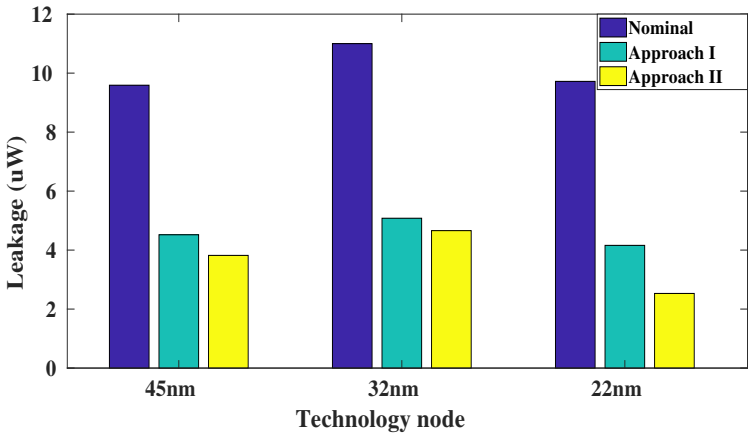
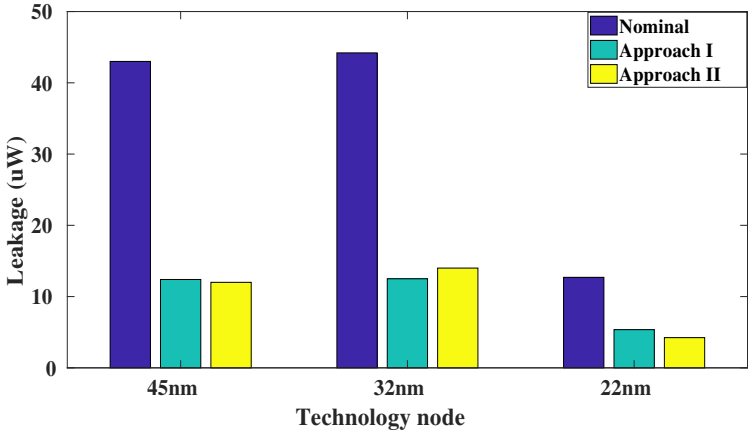


Figure 5.7: Robust transistor sizing obtained in 28-T full adder cell using approach II for 22nm tech.



(a)



(b)

Figure 5.8: Stand by mode leakage power obtained with different approaches a) for NOC. b) for WOC.

Chapter 6

Conclusions

Various heuristics algorithms have been implemented in this thesis to optimize the logic cells for minimum leakage power and propagation delay. The implemented optimization algorithms include evolutionary based Genetic algorithm, thermo based Simulated Annealing algorithm, swarm based Particle Swarm Optimization algorithm and Artificial Bee Colony algorithm. The obtained PVT variations aware robust transistor sizing of basic cells have been used to size the benchmark circuits such as 32 bit adders, ISCAS cells, multiplier etc. The overall leakage power reduction achieved is up to 88% using reported algorithms. As there is trade off between power and delay thus the leakage optimization have been performed while keeping propagation delay arc at the initial states.

In second part, two novel techniques have been proposed which minimizes the stand by mode leakage power in the CMOS 1-bit 28T full adder cell. The performance of optimized full adder cell has been compared with the initial one and the percentage leakage reduction is achieved up to 75%. Around 50K statistical samples have been taken to check the yield of the cells which is obtained up to 99.8%. The simulations have been performed for 45nm. 32nm 22nm metal gate high-k productive technology model cards using HSPICE tool. The computation time to fully optimize one circuit is approximately 2 hours for PSO, 0.5 hour for SA algorithm, 1.5 hours for ABC algorithm and 2 hours for NSGA based approach considering 400 iterations on 32 GB RAM, 12 core machine.

The future work may include implementation of the algorithms on the advanced VLSI technologies, like SOI, SOS and FINFETs. Also, sizing the sequential logic cells is an inevitable part of such kind of optimization in VLSI.

Appendix A

PTM Model

```
    * PTM 32nm Metal Gate / High-K

.model nmos nmos level = 54

+version = 4.0 binunit = 1 paramchk= 1 mobmod = 0
+capmod = 2 igcmod = 1 igbmod = 1 geomod = 1
+diomod = 1 rdsmod = 0 rbodymod= 1 rgatemod= 1
+permod = 1 acnqsmod= 0 trnqsmod= 0
+tnom = 27 toxe = 7.5e-010 toxp = 5e-010 toxm = 7.5e-010
+dtox = 2.5e-010 epsrox = 3.9 wint = 5e-009 lint = 1.95e-009
+ll = 0 wl = 0 lln = 1 wln = 1
+lw = 0 ww = 0 lwn = 1 wwn = 1
+lwl = 0 wwl = 0 xpart = 0 toxref = 7.5e-010 xl = -14e-9
+dlcig = 1.95e-009
+vth0 = 0.3558 k1 = 0.2 k2 = 0 k3 = 0
+k3b = 0 w0 = 2.5e-006 dvt0 = 1 dvt1 = 2
+dvt2 = 0 dvt0w = 0 dvt1w = 0 dvt2w = 0
+dsub = 0.078 minv = 0.05 voffl = 0 dvtp0 = 1e-011
+dvtp1 = 0.1 lpe0 = 0 lpeb = 0 xj = 1e-008
+ngate = 1e+023 ndep = 8.7e+018 nsd = 2e+020 phin = 0
+cdsc = 0 cdscb = 0 cdscd = 0 cit = 0
+voff = -0.13 nfactor = 2.1 eta0 = 0.005 etab = 0
+vfb = -1.058 u0 = 0.0238 ua = -5e-010 ub = 1.7e-018
+uc = 0 vsat = 182130 a0 = 1 ags = 0
+a1 = 0 a2 = 1 b0 = 0 b1 = 0
+keta = 0.04 dwg = 0 dwb = 0 pclm = 0.06
+pdiblc1 = 0.001 pdiblc2 = 0.001 pdiblcb = -0.005 drout = 0.5
```

+pvag = 1e-020 delta = 0.01 pscbe1 = 2.0e+009 pscbe2 = 1e-007
+fprout = 0.2 pdits = 0.01 pditsd = 0.23 pditsl = 2300000
+rsh = 5 rdsw = 80 rsw = 40 rdw = 40
+rdswmin = 0 rdwmin = 0 rswmin = 0 prwg = 0
+prwb = 0 wr = 1 alpha0 = 0.074 alpha1 = 0.005
+beta0 = 30 agidl = 0.0002 bgidl = 2.1e+009 cgidl = 0.0002
+egidl = 0.8 aigbacc = 0.012 bigbacc = 0.0028 cigbacc = 0.002
+nigbacc = 1 aigbinv = 0.014 bigbinv = 0.004 cigbinv = 0.004
+eigbinv = 1.1 nigbinv = 3 aigc = 0.020014 bigc = 0.0027432
+cigc = 0.002 aigsd = 0.020014 bigsd = 0.0027432 cigsd = 0.002
+nigc = 1 poxedge = 1 pigcd = 1 ntox = 1 +xrcrg1 = 12 xrcrg2 = 5
+cgso = 9e-011 cgdo = 9e-011 cgbo = 0 cgdl = 7.5e-013
+cgsl = 7.5e-013 clc = 1e-007 cle = 0.6 cf = 1.1e-010
+ckappas = 0.6 ckappad = 0.6 vfbcv = -1 acde = 1
+moin = 15 noff = 1 voffcv = 0
+kt1 = -0.154 kt1l = 0 kt2 = 0.022 ute = -1.1
+ua1 = 1e-009 ub1 = -1e-018 uc1 = -5.6e-011 prt = 0
+at = 33000
+fnoimod = 1 tnoimod = 0 noia = 6.25e+041 noib = 3.125e+026
+noic = 8.75e+009 em = 41000000 af = 1 ef = 1
+kf = 0 tnoia = 1.5 tnoib = 3.5 ntnoi = 1
+jss = 1.2e-006 jsws = 2.4e-013 jswgs = 2.4e-013 njs = 1
+ijthsfwd= 0.1 ijthsrev= 0.1 bvs = 10 xjbvs = 1
+jsd = 1.2e-006 jswd = 2.4e-013 jswgd = 2.4e-013 xjbvd = 1
+pbs = 1 cjs = 0.0018 mjs = 0.5 pbsws = 1
+cjsws = 1.2e-010 mjsws = 0.33 cjswgs = 2.1e-010 cjd = 0.0018
+cjswd = 1.2e-010 mjswd = 0.33 pbswgd = 1 cjswgd = 2.1e-010
+mjswgd = 0.33 tpb = 0 tcj = 0 tpbsw = 0
+tcjsw = 0 tpbswg = 0 tcjswg = 0 xtis = 3
+dmcg = 0 dmci = 0 dmdg = 0 dmcgt = 0
+dwj = 0 xgw = 0 xgl = 0
+rshg = 0.4 gbmin = 1e-010 rbpb = 5 rbpd = 15
+rbps = 15 rbdb = 15 rbsb = 15 ngcon = 1

.model pmos pmos level = 54
+version = 4.0 binunit = 1 paramchk= 1 mobmod = 0
+capmod = 2 igcmod = 1 igbmod = 1 geomod = 1
+diomod = 1 rdsmod = 0 rbodymod= 1 rgatemod= 1

+permod = 1 acnqsmod= 0 trnqsmod= 0
+tnom = 27 toxe = 7.7e-010 toxp = 5e-010 toxm = 7.7e-010
+dtox = 2.7e-010 epsrox = 3.9 wint = 5e-009 lint = 1.95e-009
+ll = 0 wl = 0 lln = 1 wln = 1
+lw = 0 ww = 0 lwn = 1 wwn = 1
+lwl = 0 wwl = 0 xpart = 0 toxref = 7.7e-010 xl = -14e-9 +dlcig = 1.95e-009
+vth0 = -0.24123 k1 = 0.2 k2 = -0.01 k3 = 0
+k3b = 0 w0 = 2.5e-006 dvt0 = 1 dvt1 = 2
+dvt2 = -0.032 dvt0w = 0 dvt1w = 0 dvt2w = 0
+dsub = 0.1 minv = 0.05 voffl = 0 dvtp0 = 1e-011
+dvtp1 = 0.05 lpe0 = 0 lpeb = 0 xj = 1.008e-008
+ngate = 1e+023 ndep = 3.5e+018 nsd = 2e+020 phin = 0
+cdsc = 0 cdscb = 0 cdscd = 0 cit = 0
+voff = -0.13 nfactor = 2.1 eta0 = 0.0042 etab = 0
+vfb = -1.058 u0 = 0.00306 ua = -5e-010 ub = 1.6e-018
+uc = 0 vsat = 78000 a0 = 1 ags = 1e-020
+a1 = 0 a2 = 1 b0 = 0 b1 = 0
+keta = -0.047 dwg = 0 dwb = 0 pclm = 0.1
+pdiblc1 = 0.001 pdiblc2 = 0.001 pdiblcb = 3.4e-008 drout = 0.6
+pvag = 1e-020 delta = 0.01 pscbe1 = 2e+009 pscbe2 = 9.58e-007
+fprout = 0.2 pdits = 0.08 pditsd = 0.23 pditsl = 2300000
+rsh = 5 rdsw = 80 rsw = 40 rdw = 40
+rdswmin = 0 rdwmin = 0 rswmin = 0 prwg = 0
+prwb = 0 wr = 1 alpha0 = 0.074 alpha1 = 0.005
+beta0 = 30 agidl = 0.0002 bgidl = 2.1e+009 cgidl = 0.0002
+egidl = 0.8 aigbacc = 0.012 bigbacc = 0.0028 cigbacc = 0.002
+nigbacc = 1 aigbinv = 0.014 bigbinv = 0.004 cigbinv = 0.004
+eigbinv = 1.1 nigbinv = 3 aigc = 0.011942 bigc = 0.0012217
+cigc = 0.0008 aigsd = 0.011942 bigsd = 0.0012217 cigsd = 0.0008
+nigc = 1 poxedge = 1 pigcd = 1 ntox = 1 +xrcrg1 = 12 xrcrg2 = 5
+cgso = 9e-011 cgdo = 9e-011 cgbo = 0 cgdl = 3e-011
+cgsl = 3e-011 clc = 1e-007 cle = 0.6 cf = 1.1e-010
+ckappas = 0.6 ckappad = 0.6 vfbcv = -1 acde = 1
+moin = 15 noff = 1 voffcv = 0 +kt1 = -0.14 kt1l = 0 kt2 = 0.022 ute = -1.1
+ua1 = 1e-009 ub1 = -1e-018 uc1 = -5.6e-011 prt = 0 +at = 33000
+fnoimod = 1 tnoimod = 0 noia = 6.25e+041 noib = 3.125e+026
+noic = 8.75e+009 em = 41000000 af = 1 ef = 1
+kf = 0 tnoia = 1.5 tnoib = 3.5 ntnoi = 1

+jss = 2e-007 jsws = 4e-013 jswgs = 4e-013 njs = 1 +ijthsfwd= 0.1 ijthsrev= 0.1 bvs = 10 xjbvs = 1
+jsd = 2e-007 jswd = 4e-013 jswgd = 4e-013 xjbvd = 1
+pbs = 1 cjs = 0.0015 mjs = 0.5 pbsws = 1
+cjsws = 9.4e-011 mjsws = 0.33 cjswgs = 2e-010 cjd = 0.0015
+cjswd = 9.4e-011 mjswd = 0.33 pbswgd = 1 cjswgd = 2e-010
+mjswgd = 0.33 tpb = 0 tcj = 0 tpbsw = 0
+tcjsw = 0 tpbswg = 0 tcjswg = 0 xtis = 3
+dmcg = 0 dmdg = 0 dmcgt = 0 xgw = 0 +xgl = 0
+rshg = 0.1 gbmin = 1e-012 rbpb = 50 rbpd = 50
+rbps = 50 rbdb = 50 rbsb = 50 ngcon = 1

Appendix B

Spice Netlist

```
**********Netlist for Full-adder cell**********
.param gflag__totalflag_mos__mos__cmos040lp=0
.param gflag__globalflag_mos__mos__cmos040lp=1
.param svtlp_dev=1
.include '../32nm_MGK.pm'
.options GMIN=1e-30
.options POST
.options AUTOSTOP
.options INGOLD=2 DCON=1
.options GSHUNT=1e-15
.options RMIN=1e-15
.options ABSTOL=1e-25 ABSVDC=1e-25
.options RELTOL=1e-20
.options RELVDC=1e-2
.options NUMDGT=4 PIVOT=1e-13
.option MEASDGT=6
.temp t
.PARAM pvdd=0
.PARAM Vin_A=0
.PARAM Vin_B=0
.PARAM Vin_C=0

vvdd vdd 0 pvdd
vgnd gnd 0 dc=0

Vina nodea 0 Vin_A
Vinb nodeb 0 Vin_B
```

```
Vinc nodec 0 Vin_C

**netlist
Mp1 1 nodea vdd vdd pmos l=32n w=512.0n
Mp2 1 nodeb vdd vdd pmos l=32n w=512.0n
Mp3 nodecon nodec 1 vdd pmos l=32n w=512.0n
Mn1 5 nodea gnd gnd nmos l=32n w=256.0n
Mn2 5 nodeb gnd gnd nmos l=32n w=256.0n
Mn3 nodecon nodec 5 gnd nmos l=32n w=256.0n
Mp4 4 nodea vdd vdd pmos l=32n w=512.0n
Mp5 nodecon nodeb 4 vdd pmos l=32n w=512.0n
Mn4 nodecon nodeb node4 gnd nmos l=32n w=256.0n
Mn5 node4 nodea gnd gnd nmos l=32n w=256.0n
Mp6 2 nodea vdd vdd pmos l=32n w=512.0n
Mp7 2 nodeb vdd vdd pmos l=32n w=512.0n
Mp8 2 nodec vdd vdd pmos l=32n w=512.0n
Mp9 nodes0n nodecon 2 vdd pmos l=32n w=512n
Mn6 3 nodea gnd gnd nmos l=32n w=256.0n
Mn7 3 nodeb gnd gnd nmos l=32.0n w=256n
Mn8 3 nodec gnd gnd nmos l=32n w=256.0n
Mn9 nodes0n nodecon 3 gnd nmos l=32n w=256n
Mp10 9 nodea vdd vdd pmos l=32n w=768.0n
Mp11 8 nodeb 9 vdd pmos l=32n w=768.0n
Mp12 nodes0n nodec 8 vdd pmos l=32n w=768.0n
Mn10 7 nodea gnd gnd nmos l=32n w=384.0n
Mn11 6 nodeb 7 gnd nmos l=32n w=384.0n
Mn12 nodes0n nodec 6 gnd nmos l=32n w=384.0n
Mp13 nodeco nodecon vdd vdd pmos l=32n w=512.0n
Mn13 nodeco nodecon gnd gnd nmos l=32n w=256.0n
Mp14 nodes0 nodes0n vdd vdd pmos l=32n w=512.0n
Mn14 nodes0 nodes0n gnd gnd nmos l=32n w=256.0n

*******Worst process parameters for leakage*******
.param wintPAR = 4.5e-009
.param lintPAR = 2.145e-009
.param lmin=32n
.param wPAR=32n
.param toxe_parN =6.75e-010
```

```
.param toxp_par =4.5e-010
.param toxrefPARn =8.25e-010
.param xjPAR=1.1e-008
.param ndepPARn=7.83e+018
.param toxe_parP=6.93e-010
.param toxrefPARp=8.47e-010
.param ndepPARp=3.15e+018

.DATA In_Var
Vin_A        Vin_B        Vin_C        t        pvdd
0            0            0            25       1
0            0            1            25       1
0            1            0            25       1
0            1            1            25       1
1            0            0            25       1
1            0            1            25       1
1            1            0            25       1
1            1            1            25       1
0            0            0            125      1.1
0            0            1            125      1.1
0            1            0            125      1.1
0            1            1            125      1.1
1            0            0            125      1.1
1            0            1            125      1.1
1            1            0            125      1.1
1            1            1            125      1.1
.ENDDATA

.DC dummy 1 1 1 SWEEP DATA=In_Var

.MEAS DC Vsup AVG v(vdd)
.MEAS DC Ivdd AVG i(vvdd)
.MEAS DC VIS param='-Vsup*Ivdd'

.MEAS DC Va AVG v(nodea)
.MEAS DC Ia AVG i(Vina)
.MEAS DC VIA param='-Va*Ia'
```

```
.MEAS DC Vb AVG v(nodeb)
.MEAS DC Ib AVG i(Vinb)
.MEAS DC VIB param='-Vb*Ib'

.MEAS DC Vc AVG v(nodec)
.MEAS DC Ic AVG i(Vinc)
.MEAS DC VIC param='-Vc*Ic'

.MEAS DC leakage param='VIS+VIA+VIB+VIC'
.meas dc sum avg v(nodes0)
.meas dc carry avg v(nodeco)
.END
```

Appendix C

Power Consumed in Identification of Power Dissipation Circuit

Table C.1: Power Consumed in Identification of Power dissipation circuit (Pident)

Circuit type	Power dissipation by CUT (μW)
Full adder	7.8
NAND2	0.59
NAND3	0.95
NOR2	0.25
NOR3	0.95
OR2	0.63
XOR2	0.82

Related Publications

1. Prateek Gupta, Shirisha Gourishetty, Harshini Mandadapu, Zia Abbas, *"PVT Variations Aware Robust Transistor Sizing for Power-Delay Optimal CMOS Digital Circuit Design"*, in IEEE International Symposium on Circuits and Systems (ISCAS) 2019, Sapporo, Japan.

2. Prateek Gupta, Harshini Mandadapu, Shirisha Gourishetty, Zia Abbas, *"Robust Transistor Sizing for Improved Performances in Digital Circuits using Optimization Algorithms"*, in 20th International Symposium on Quality Electronic Design (ISQED) 2019, California, USA.

3. Prateek Gupta, Zia Abbas, *"Statistical Variation Aware Algorithm Based Optimal Transistor Sized Digital Cells with Reduced Stand-by Mode Leakage Power"*, in Microelectronics Journal (MEJ) Elsevier (to be submitted).

4. Prateek Gupta, Shubham Kumar, Zia Abbas, *"Optimal Transistor Sizing of Full-Adder Block to Reduce Standby Leakage Power"*, in 22nd International Symposium on VLSI Design and Test (VDAT) 2018, India.

5. Prateek Gupta, Zia Abbas, *"Multi-Objective Optimization Algorithm Based Transistor Sizing for Improved Power-Delay-Area in Digital Circuits"*, in 15th India Council International Conference INDICON 2018, India.

Bibliography

[1] Bsim 4.6.1 mosfet model, users manual, tech. rep., EECS Department, University of California, Berkeley, 2007, available: http://www.srware.com/xictools/docs/model_docs/bsim4.6.1/bsim461_manual.pdf.

[2] Predictive technology model (ptm) [ONLINE]. available: http://ptm.asu.edu/.

[3] Z. Abbas and M. Olivieri. Impact of technology scaling on leakage power in nano-scale bulk cmos digital standard cells. *Microelectronics Journal*, 45(2):179–195, 2014.

[4] Z. Abbas and M. Olivieri. Optimal transistor sizing for maximum yield in variation-aware standard cell design. *International Journal of Circuit Theory and Applications*, 44(7):1400–1424, 2016.

[5] A. Agarwal, S. Mukhopadhyay, A. Raychowdhury, K. Roy, and C. H. Kim. Leakage power analysis and reduction for nanoscale circuits. *IEeE Micro*, 26(2):68–80, 2006.

[6] S. I. Association et al. International technology roadmap for semiconductors 2005 edition. *http://www. itrs. net/*, 2006.

[7] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb. A simulated annealing-based multiobjective optimization algorithm: Amosa. *IEEE transactions on evolutionary computation*, 12(3):269–283, 2008.

[8] D. K. Beece, C. Visweswariah, J. Xiong, and V. Zolotov. Transistor sizing of custom high-performance digital circuits with parametric yield considerations. *Optimization and Engineering*, 15(1):217–241, 2014.

[9] S. Bhunia and S. Mukhopadhyay. *Low-power variation-tolerant design in nanometer silicon*. Springer, 2010.

[10] L. Cheng. *Statistical analysis and optimization for timing and power of VLSI circuits*. University of California, Los Angeles, 2010.

[11] C.-H. Choi, K.-Y. Nam, Z. Yu, and R. W. Dutton. Impact of gate direct tunneling current on circuit performance: A simulation study. *IEEE Transactions on Electron Devices*, 48(12):2823–2829, 2001.

[12] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In *International conference on parallel problem solving from nature*, pages 849–858. Springer, 2000.

[13] K. Deb and D. K. Saxena. On finding pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. *Kangal report*, 2005011, 2005.

[14] B. S. Deepaksubramanyan and A. Nunez. Analysis of subthreshold leakage reduction in cmos digital circuits. In *Circuits and Systems, 2007. MWSCAS 2007. 50th Midwest Symposium on*, pages 1400–1404. IEEE, 2007.

[15] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*, pages 39–43. IEEE, 1995.

[16] P. Gupta and Z. Abbas. "statistical variation aware algorithm based optimal transistor sizeddigital cells with reduced stand-by mode leakage power". In *to be submitted in Microelectronics Journal (MEJ) Elsevier.*

[17] P. Gupta, S. Kumar, and Z. Abbas. Optimal transistor sizing of full-adder block to reduce standby leakage power. In *International Symposium on VLSI Design and Test*, pages 88–99. Springer, 2018.

[18] D. M. Harris, N. Wei, Z. Wang, A. Fikes, and A. Thaker. Evaluation of predictive technology models. *Microelectronics journal*, 80:7–17, 2018.

[19] R. Hassan, B. Cohanim, O. de Weck, and G. Venter. A copmarison of particle swarm optimization and the genetic algorithm. *American Institute of Aeronautics and Astronautics*, 2004.

[20] H. Jiang, M. Marek-Sadowska, and S. R. Nassif. Benefits and costs of power-gating technique. In *Computer Design: VLSI in Computers and Processors, 2005. ICCD 2005. Proceedings. 2005 IEEE International Conference on*, pages 559–566. IEEE, 2005.

[21] J. Kao, S. Narendra, and A. Chandrakasan. Subthreshold leakage modeling and reduction techniques. In *Proceedings of the 2002 IEEE/ACM international conference on Computer-aided design*, pages 141–148. ACM, 2002.

[22] D. Karaboga. An idea based on honey bee swarm for numerical optimization. Technical report, Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, 2005.

[23] D. Karaboga and B. Akay. A comparative study of artificial bee colony algorithm. *Applied mathematics and computation*, 214(1):108–132, 2009.

[24] D. Karaboga and B. Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of global optimization*, 39(3):459–471, 2007.

[25] S. Kukkonen and K. Deb. Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems. In *2006 IEEE International Conference on Evolutionary Computation*, pages 1179–1186. IEEE, 2006.

[26] B. H. Lee, S. C. Song, R. Choi, and P. Kirsch. Metal electrode/high-k dielectric gate-stack technology for power management. *IEEE Transactions on Electron Devices*, 55(1):8–20, 2008.

[27] X. Li, P. Gopalakrishnan, Y. Xu, and L. T. Pileggi. Robust analog/rf circuit design with projection-based performance modeling. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26(1):2–15, 2007.

[28] T. McConaghy, K. Breen, J. Dyck, and A. Gupta. *Variation-aware design of custom integrated circuits: a hands-on field guide*. Springer Science & Business Media, 2012.

[29] S. P. Mohanty, N. Ranganathan, E. Kougianos, and P. Patra. *Low-power high-level synthesis for nanoscale CMOS circuits*. Springer Science & Business Media, 2008.

[30] S. Narendra, V. De, D. Antoniadis, A. Chandrakasan, and S. Borkar. Scaling of stack effect and its application for leakage reduction. In *Proceedings of the 2001 international symposium on Low power electronics and design*, pages 195–200. ACM, 2001.

[31] S. Narendra, Y. Ye, S. Borkar, V. De, and A. Chandrakasan. Leakage dependence on input vector. In *Leakage in Nanometer CMOS Technologies*, pages 21–39. Springer, 2006.

[32] M. Radfar, S. P. Mozafari, K. Shah, and J. Singh. Analysis of geometric and non-linear programming as optimization algorithms for low power vlsi circuits. In *Electrical and Computer Engineering (CCECE), 2011 24th Canadian Conference on*, pages 000617–000620. IEEE, 2011.

[33] J.-M. Shyu, A. Sangiovanni-Vincentelli, J. P. Fishburn, and A. E. Dunlop. Optimization-based transistor sizing. *IEEE Journal of Solid-State Circuits*, 23(2):400–409, 1988.

[34] A. Srivastava, D. Sylvester, and D. Blaauw. *Statistical analysis and optimization for VLSI: timing and power*. Springer Science & Business Media, 2006.

[35] B. Suman and P. Kumar. A survey of simulated annealing as a tool for single and multiobjective optimization. *Journal of the operational research society*, 57(10):1143–1160, 2006.

[36] I. Sutherland, R. F. Sproull, B. Sproull, D. Harris, and E. D. Harris. *Logical effort: designing fast CMOS circuits*. Morgan Kaufmann, 1999.

[37] T. N. Theis and H.-S. P. Wong. The end of moore's law: A new beginning for information technology. *Computing in Science & Engineering*, 19(2):41–50, 2017.

[38] A. Wang, B. H. Calhoun, and A. P. Chandrakasan. *Sub-threshold design for ultra low-power systems*, volume 95. Springer, 2006.

[39] A. Wang, A. P. Chandrakasan, and S. V. Kosonocky. Optimal supply and threshold scaling for subthreshold cmos circuits. In *VLSI, 2002. Proceedings. IEEE Computer Society Annual Symposium on*, pages 7–11. IEEE, 2002.

[40] N. H. Weste and D. Harris. *CMOS VLSI design: a circuits and systems perspective*. Pearson Education India, 2015.