LOW POWER SYSTEM-ON-CHIP DESIGN USING VOLTAGE ISLANDS: FROM APPLICATION TO FLOORPLAN

by

Dipanjan Sengupta

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

The Faculty of Graduate Studies

(Electrical and Computer Engineering)

The University of British Columbia

(Vancouver)

July 2010

Abstract

With the continued trend in device scaling and the ever increasing popularity of hand held mobile devices, power has become a major bottleneck for the development of future generation System-on-Chip devices. As the number of transistors on the SoC and the associated leakage current increases with every technology generation, methods for reducing both active and static power have been aggressively pursued.

Starting with the application for which the SoC is to be designed, the proposed design flow considers numerous design constraints at different steps of design process and produces a final floorplanned solution of the cores. *Voltage Island Design* is a popular method for implementing multiple supply voltages in a SoC. Use of multiple threshold voltages with power gating of cores is an attractive method for leakage power reduction. This thesis addresses the design challenges of implementing multiple supply and threshold voltage on the same chip holistically with the ultimate goal for maximum power reduction.

Specifically, given the power-state machine (PSM) of an application, the high power and low power cores are identified first. Based on the activity of the cores, threshold voltage is assigned to each core. The next step is to identify the suitable range of supply voltage for each core followed by voltage island generation. A methodology of reducing the large number of available choices to a useful set using the application PSM is developed. The cores are partitioned into islands using a cost function that gradually shifts from a power-based assignment to a connectivity-based one.

Additional design constraints such as power supply noise and floorplan constraints can offset the possible power savings and thus are considered early in the design phase. Experimental results on benchmark circuits prove the effectiveness of the proposed methodology. On average, the use of multiple V_T and power gating can reduce almost 20% of power compared to single V_T. A proper choice of supply voltages leads to another 4% reduction in power. Compared to previous methods, the proposed floorplanning technique on average offers an additional 10% power savings, 9% area improvement and 2.4X reduction in runtime.

Table of Contents

Abstract ..... **ii**
Table of Contents ..... **iii**
List of Tables ..... **vi**
List of Figures ..... **vii**
List of Abbreviations ..... **ix**
Acknowledgments ..... **xi**
Chapter 1: Introduction **1**
1.1 Motivation 1
1.2 Research Directions 4
1.3 Research Objectives 10
1.4 Thesis Organization 11
Chapter 2: Background **13**
2.1 Overview of Chapter 13
2.2 Sources of Power Consumption 13
2.2.1 Active Power 14
2.2.2 Static Power 17
2.3 Low Power Design Techniques 20
2.3.1 Dynamic Power Reduction Techniques 21
2.3.1.1 Clock Gating 21
2.3.1.2 Operand Isolation 22
2.3.1.3 Gate-level Transformation 22
2.3.1.4 Path Balancing 23
2.3.1.5 Gate Sizing 24
2.3.1.6 Transistor Sizing 25
2.3.1.7 Voltage Scaling 25
2.3.1.8 Dynamic Voltage Scaling 26
2.3.1.9 Multi-VDD Approach 27
2.3.2 Static Power Reduction Techniques 31
2.3.2.1 Transistor Stacking 32
2.3.2.2 Standby Input Vector Selection 33

2.3.2.3 Dual-VT Technique 33
2.3.2.4 Variable Threshold CMOS (VTCMOS) 35
2.3.2.5 Power Gating 37
2.4 Floorplanning 40
2.4.1 Slicing Floorplans 40
2.4.2 Non-Slicing Floorplans 41
2.4.3 Floorplan Algorithms 42
2.4.3.1 Simulated Annealing 42
2.4.3.2 Integer Linear Programming 43
2.4.3.3 Genetic Algorithm 43
2.4.3.4 Knowledge-based Floorplanning 44
2.4.4 Modern Floorplan Constraints 44
2.4.4.1 Fixed-outline constraint 44
2.4.4.2 Hierarchical Floorplanning 45
2.4.4.3 Power Supply Aware Floorplanning 45
2.5 Summary 46
Chapter 3: Supply and Threshold Voltage Selection 48
3.1 Overview 48
3.2 Preliminaries 49
3.2.1 Task Graph and Task Mapping 49
3.2.2 Core-Level Power Modeling 50
3.2.3 System-level power Modeling 51
3.3 Power Optimization Framework 55
3.3.1 Block-level multi-VT design 56
3.4 VT selection 57
3.5 Challenges in Multiple Supply Voltage Design 60
3.6 VDD Selection 61
3.6.1 Voltage Assignment Table Generation 65
3.7 Summary 71
Chapter 4: Supply Noise and Voltage Selection Issues in Floorplanning 73
4.1 Overview 73

4.2 Decoupling Capacitance Allocation 75
4.2.1 Power Grid Noise 75
4.2.2 Voltage Islands and 1-D Power Grid Representation 76
4.2.3 Current Model and Noise Correlation 78
4.2.4 Decap Selection in Voltage Island SoCs 83
4.2.4.1 Linear Increments 83
4.2.4.2 Binary Search 84
4.2.4.3 Exponential Increase 85
4.2.5 Effect of Core Reassignment on Decap Value 87
4.2.6 Fixed Decap Allocation Problem 88
4.3 Floorplan Constraints 90
4.3.1 Connectivity-aware Supply voltage Selection 92
4.4 Summary 97
Chapter 5: Voltage Island Floorplanning 99
5.1 Overview 99
5.2 Design Exploration 100
5.3 Voltage Assignment for Floorplanning 104
5.3.1 Proposed Voltage Assignment Methodology 106
5.3.2 Cost Function 108
5.4 Floorplanning 110
5.5 Overall Design Flow 112
5.6 Connectivity-Aware Voltage Island Design 121
5.7 Summary 123
Chapter 6: Conclusions and Future Work 125
6.1 Research Summary and Contributions 126
6.2 Research Limitations 130
6.3 Future Work 132
References 134

List of Tables

Table 3.1: Optimal Voltages Table 65
Table 3.2: Voltage-Energy Table 69
Table 3.3: Voltage Assignment Table 71
Table 4.1: Voltage Choices For Different Cores 76
Table 4.2: Comparison Of Search Techniques 86
Table 4.3: Comparison of Noise and Power 89
Table 5.1: Power vs. Level Shifters 109
Table 5.2: Power saving due to Multi-V_T Design 113
Table 5.3: Comparison of Voltage Selection Algorithms 114
Table 5.4: Power and Area Comparison 116
Table 5.5: Connectivity Aware vs. Energy-Aware 122

List of Figures

Figure 1.1: Power density trend for future technology generations [3] 2
Figure 1.2: Trends in battery max power and chip power [6] 3
Figure 1.3: Power reduction design space [8] 4
Figure 1.4: Floorplan time vs. number of IP blocks vs. number of islands 9
Figure 1.5: Voltage Island Design comparison 11
Figure 2.1: Active Power (a) CMOS Logic (b) Simple Inverter 14
Figure 2.2: Leakage current components of an NMOS transistor [34] 18
Figure 2.3: Subthreshold and Gate leakage in CMOS circuit 19
Figure 2.4: Implementation of DVS in the IEM926 [68] 27
Figure 2.5: CVS implementation 28
Figure 2.6: SoC with (a) Single Voltage Island and (b) Three Voltage Islands 29
Figure 2.7: Transistor Stacking 32
Figure 2.8: Dual V_T implementation [91] 34
Figure 2.9: VTCMOS (a) Circuit implementation (b) System implementation 36
Figure 2.10: Body Biasing by changing source voltage with respect to body voltage 37
Figure 2.11: Power Gating 38
Figure 2.12: Activity Profile without Power Gating [7] 39
Figure 2.13: Activity Profile with Power Gating [7] 39
Figure 2.14: (a) Slicing Floorplan (b) Slicing Tree and (c) Normalized Postfix expression 41
Figure 2.15: (a) Non-Slicing Floorplan (b) HCG and (c) VCG [119] 42
Figure 3.1: Task Graph 49
Figure 3.2: Generic System-level Design 50
Figure 3.3: Different power modes of cores in a SoC 52
Figure 3.4: Power State Machine 53
Figure 3.5: Variable Activity Circuit 57
Figure 3.6: Delay vs. V_{DD} 58
Figure 3.7: Power consumption for 1% activity 59
Figure 3.8: Power consumption for 100% activity 59
Figure 3.9: Power, Delay and EDP vs. V_{DD} for a core 63
Figure 3.10: Energy vs V_{DD} for variable activity and delay 64

Figure 3.11: Outline for Supply Voltage Selection Algorithm 68
Figure 3.12: Energy-Based voltage choice 69
Figure 4.1: 1-D analysis of Voltage Island 77
Figure 4.2: HSPICE simulation of current waveform 78
Figure 4.3 : Voltage Island current model 79
Figure 4.4: Power Grid Noise vs. Decap vs. I_{avg} 82
Figure 4.5: Results for Linear Incremental increase in decap value 84
Figure 4.6 : Results for Binary Search method for decap selection 85
Figure 4.7: Results for Exponential Search method for decap selection 86
Figure 4.8: Effect of moving IP block from one island to the other 88
Figure 4.9: Comparison of three metrics 90
Figure 4.10: A SoC with placement constraint 91
Figure 4.11: Connections among the cores 92
Figure 4.12: Connectivity-based voltage choice 94
Figure 4.13: Function calculation 94
Figure 4.14: Voltage Island Floorplan with placement constraints 95
Figure 4.15: Comparison of three voltage assignments 97
Figure 5.1: Power vs. number of Voltage Islands 102
Figure 5.2: Comparison of design exploration 103
Figure 5.3: Comparison of design exploration 105
Figure 5.4: Voltage Island Floorplanning 111
Figure 5.5: Overall design flow 112
Figure 5.6: Floorplan of *n50* using (a) technique in [21] (b) proposed approach 117
Figure 5.7: Average and Peak temperature comparison 118
Figure 5.8: IR drop consideration 119
Figure 5.9: Floorplan solutions from (a) without and (b) with decap consideration 120
Figure 5.10: Design parameters vs. Power vs. Overhead 121
Figure 5.11: (a) Energy-aware (b) Connectivity-aware floorplan 123

List of Abbreviations

3G	Third Generation
ABB	Adaptive Body Biasing
ACG	Adjacent Constraint Graph
APC	Adaptive Power Controller
ASIC	Application-Specific Integrated Circuit
BSG	Baseline-Sliceline Grid
BSIM	Berkeley Short-Channel IGFET Model
CAD	Computer Aided Design
CMOS	Complimentary Metal-Oxide-Semiconductor
CPF	Common Power Format
CS	Corner Sequence
CVS	Clustered Voltage Scaling
DAG	Directed Acyclic Graph
DVFS	Dynamic Voltage and Frequency Scaling
DVS	Dynamic Voltage Scaling
DVTS	Dynamic Threshold Voltage Scaling
ECVS	Extended Clustered Voltage Scaling
EDP	Energy-Delay Product
e-VIP	Early Voltage Island Partitioning
GA	Genetic Algorithm
HCG	Horizontal Constraint Graph
HVT	High-V_T
IC	Integrated Circuits
IEM	Intelligent Energy Manager
ILP	Integer-Linear Programming
IP	Intellectual Property
LVT	Low-V_T
MPEG	Moving Picture Experts Group
MPSoC	Multi-Processor System-on-Chip
MSMV	Multi-Supply Multi-Voltage

NP	Nondeterministic Polynomial Time
PDA	Personal Data Assistant
PDP	Power-Delay Product
PEP	Power-Energy Product
PSM	Power State Machine
PSO	Power Shut Off
RTL	Register Transfer Level
SA	Simulated Annealing
SoC	System-On-Chip
SP	Sequence Pair
SVT	Standard-V_T
TCG	Transitive Closure Graph
TCG-S	Transitive Closure Graph with a Sequence
UPF	Unified Power Format
VAT	Voltage Assignment Table
VCG	Vertical Constraint Graph
VLSI	Very Large Scale Integration
VTCMOS	Variable Threshold Complimentary Metal-Oxide-Semiconductor

Acknowledgments

First, I would like to thank my research supervisors Prof. Res Saleh and Prof. Steve Wilton. Dr. Saleh's never ending support, infinite amount of energy and all the knowledge conveyed in the past has been instrumental during my graduate studies. Dr. Wilton's positive outlook, patience, and guidance during the difficult times in my PhD years, have been invaluable to me.

I would also like to thank the numerous professors at UBC who served on my supervisory committees, provided useful feedback on my research and have assisted me at different times during my UBC graduate years. They include Dr. Andre Ivanov, Dr. Shahriar Mirabbasi, Dr. Guy Lemieux, Dr. Mark Greenstreet and Dr. Alan Hu.

Special thanks to the SoC support staff, Dr. Roberto Rosales and Roozbeh Mehrabadi, who were always willing to lend a hand when needed. My academic life at UBC was enriched through friendships with my peers and friends. Many thanks to Cristian, Partha, Victor, Neda, Shohaib, Usman, Karim, Melody, Shirley, Xiongfei and Jeff.

I am deeply indebted to my parents for their constant support and enthusiasm. Their sacrifice and encouragement made my dreams come true. And finally I would like to thank my wife Madhuja for being on my side during the ups and downs of graduate life. She has provided constant support and made my graduate life an enjoyable one.

CHAPTER 1 : INTRODUCTION

1.1. Motivation

With the advent of deep submicron technology, it is possible to integrate hundreds of millions of transistors on the same chip. This enables applications to be implemented using System-on-Chip (SoC) design methodologies that utilize predesigned IP (Intellectual Property) blocks or cores [1]. The increasing demand of wireless, third generation (3G), cellular mobility with a wide variety of integrated applications (such as text messaging, email access, still and video photography, digital music downloads and players, internet surfing, streaming video and video games) and the increasing need for both wireline and wireless internet appliances in home and small offices have further enhanced the demand for SoCs.

For over two decades, technology scaling has been following Moore's law of doubling the number of transistors on a chip every two years. With every technology generation, reducing gate delay and having more transistors on the same chip allowed designers and consumers to enjoy the benefit of faster, cheaper and yet more complex SoCs. In the past, the principle goal of chip design was to deliver the maximum possible performance, often at the expense of area and power. However, with continued technology scaling, the levels of integration enabled by SoC design have inadvertently pushed the power profiles of such systems beyond acceptable power density limits [2]. In fact, the combination of greater functional integration coupled with higher clock speeds enabled by technology scaling have conspired to increase the overall power density of the chips. As an indication of the potential severity of the problem, Figure 1.1 [3] shows a plot of the power density of microprocessor chips against successive technology generations.

This trend depicted in the figure is extremely worrisome and has forced power to become an important challenge for higher levels of integration due to further device scaling. Higher power density has a negative impact on the performance as well as reliability of the chip [4]. Moreover, dissipating more heat has a large impact on the packaging and cooling technology and the associated cost [5]. Thus, for almost all applications, reducing overall power consumption and localized power density of SoC's is essential for the foreseeable future.

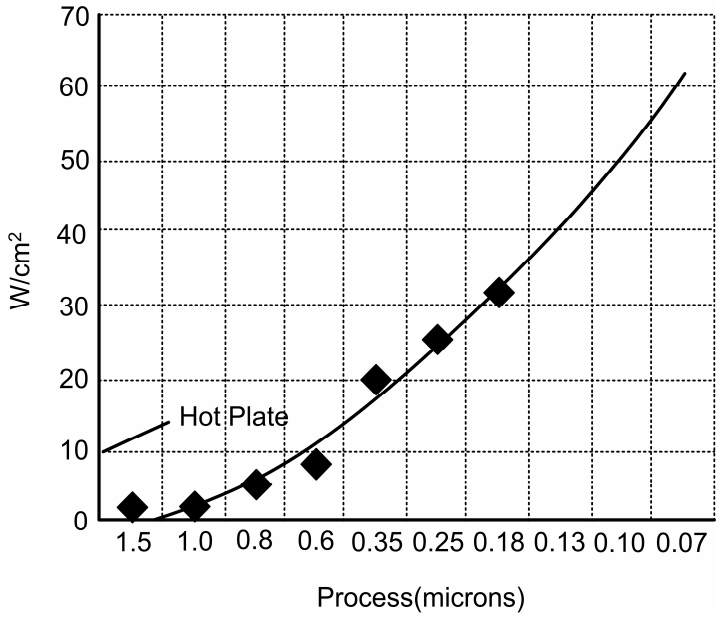


Figure 1.1: Power density trend for future technology generations [3]

In addition, battery powered electronics, such as cellular phones, PDAs, and other hand-held devices, account for a large and rapidly growing revenue market for semiconductor industry [3]. One of the characteristics for such systems is the limited battery life that impacts the systems' utility as well as the duration and mobility. Unfortunately, battery technology is unable to keep pace with what is needed to support the increasing complexity, functionality and performance of the systems they power [6]. Figure 1.2 illustrates the widening gap between the trends of processor power consumption and the improvement in battery power

capacity. As these systems have fixed throughput requirements, a careful tradeoff between power and performance is the key to extending battery life.

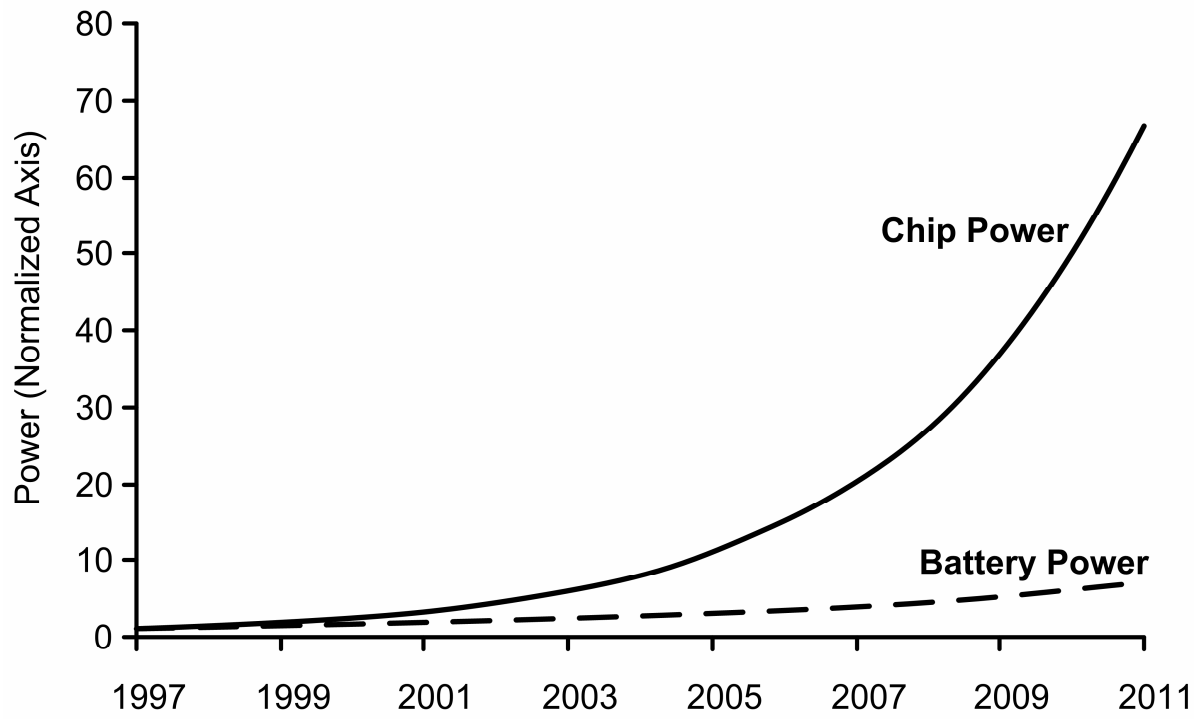


Figure 1.2: Trends in battery maximum power and chip power [6]

The pursuit of efficient methods for reduction in power consumption has moved to the forefront of the SoC design challenge. Today, satisfying the power budget is one of the most important design goals in SoC design. If not considered early in the design process, it becomes a significant problem very late in the design cycle when evaluating first silicon results. At this stage, it is almost impossible to redesign the part to fit into the market window due to stringent design cycle time requirements. One could just ship the design even though it does not meet the power specifications; however, exceeding the power budget can be extremely detrimental. It might require the chip packaging to be redesigned or have unacceptable reliability issues or insufficient battery power to run the device [7]. Furthermore, since energy issues have entered

into the mainstream public debate, it is important to minimize power as much as possible. So, from large desktop units to mobile devices, the drive today is towards “green” SoCs.

1.2. Research Directions

The aim of this research is reduction in power consumption of SoCs. For high performance SoCs, this would lead to cooler chips, while hand-held devices would have a longer battery life. Low-power design methodologies can be realized at different abstraction levels such as physical, logical, architectural, algorithmic and system levels [8]. Figure 1.3 depicts the different design levels that may be optimized for power savings. Each level offers a different set of tradeoffs that must be understood before they can be used effectively.

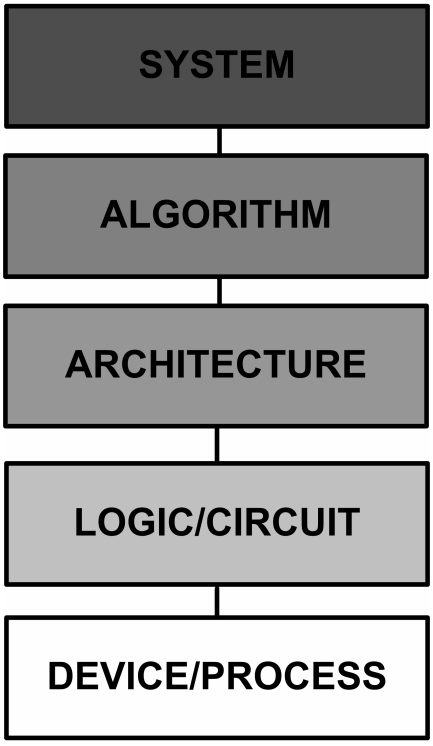


Figure 1.3: Power reduction design space [8]

Power reduction through process and technology changes involves reducing capacitances and leakage currents by scaling device geometry, improving interconnect materials, the use of

low-k dielectrics, the availability of multiple threshold voltages [9], etc. To minimize power at the circuit level, designers focus on the reduction of circuit activity by performing logic and clock gating [10], substrate biasing [11], bus loading and coding, and re-encoding of sequential circuits. Moving up another level, the use of parallelism, pipelining, and power shut off (PSO) for unused IP blocks [12] and multi-supply multi-voltage (MSMV) [13] are common approaches for architectural level power reduction. Power savings at the algorithm level involves data coding, and minimizing the number of operations in the context of software or firmware running on the system. Specifically the use of dynamic voltage and frequency scaling (DVFS) [14] is one of the popular approaches. Finally, power optimization at the system level is achieved through hardware/software co-design and ultimately holds the most promise. However, this also presents the highest level of complexity in minimization of the overall power or energy profile.

The most effective approach to producing an energy-efficient design is to consider power optimization throughout the top-down design process. A power-driven methodology must begin from the system/algorithmic phase, continue through the architectural phase, be refined through logic and circuit design and finalized in the physical implementation with power-aware synthesis, placement, and routing [15]. On the other hand, the size and complexity of SoCs today has increased enormously. This leads to a larger number of constraints to be considered in the design of these chips. If these constraints are considered only in the final phase of the design, large opportunities for power savings available in the earlier phases would be missed.

Among all the power reduction techniques discussed, the use of multiple voltages is one of the most attractive solutions in core-based SoC design [16] and forms the centerpiece of the research described in this dissertation. This technique necessitates delivering multiple supply voltages to the same chip and requires the design of several power grids to deliver current to the IP blocks. Theoretically, it is possible to have any number of supply voltages on the same chip but only a few supply voltages are practical [17]. The selection of a specified number of supplies is a key research topic pursued in this work.

With every new technology generation, device geometries are scaled by roughly 0.7X. According to past history, the threshold voltage (V_T) should also be scaled down by a commensurate amount. However, due to the exponential relation between subthreshold leakage current and threshold voltage, leakage power contributes significantly to the total power consumption of the chip. It has been projected that the transistor off-state current per micrometer of transistor width increases by ~5X per generation [18]. Thus, leakage power has also become a critical design concern for any VLSI system today. One of the popular techniques for reducing leakage power is the use of multiple threshold voltages (multi-V_T). This allows the designer to selectively apply low, high and standard threshold voltages to different transistors depending on whether the goal is high speed or low leakage, or perhaps a compromise between the two. The selection of the appropriate threshold voltages for different blocks is also a key topic in this work.

Based on the above discussion, two different methods of power reduction of SoC designs are pursued in this research, namely, chip-level dynamic power optimization using MSMV (or voltage islands), and block-level leakage power optimization using multi-V_T. While both the

techniques help in the reduction of power consumption, they also raise new challenges in the design optimization process. Chip-level dynamic power optimization focuses on optimal assignment of a supply voltage to each core and the generation of a voltage island-based chip floorplan. Once the Intellectual Property (IP) blocks are placed, the next step is to identify the leaky blocks and use a higher V_T for further power reduction. The proper choice of V_T is dependent on the application being executed by the SoC. Although each of these techniques provides some power reduction, implementing both of them concurrently provides additional opportunities for design improvement. For example, if the MSMV technique is implemented exclusively, then the choice of minimum supply voltage for each block is only based on delay constraints. However, this will eventually make leakage power optimization a more difficult task. Additionally, if V_T selection for the individual blocks is done independently, then information regarding the activity of block while executing the application is not exploited. Depending upon the application, if the choice of V_T and V_{DD} can be made together with the other design factors, then there exists a potential for maximizing the power savings.

A new challenge in voltage island design is the management of supply noise in multiple power grids. With scaling of the supply voltage, there is a commensurate increase in noise problems due to *IR* and *Ldi/dt* effects [19]. Typically, excessive voltage drops are reduced by the insertion of decoupling capacitance (decap) to ensure that the supply voltage stays within the noise budget. Solutions that meet power grid noise constraints require proper allocation of these decaps within the voltage islands. The amount of decoupling capacitance needed in a voltage island depends on the blocks assigned to the island, which is determined as part of an iterative loop and therefore not known *a priori*. The additional space required by the decaps can violate the overall area constraint of the chip. While this problem has not been addressed

directly, the importance of proper decap allocation with multiple supply voltages has been demonstrated in [20]. Thus, such constraints have an impact on the potential power savings that can be achieved. This work also addresses power grid noise constraints on the proposed low power design methodology.

Classical floorplanners consider a single V_{DD} in the entire SoC. In voltage island design, cores with same V_{DD} must be placed contiguous to one another. Thus to implement voltage island design, the floorplanning methodology needs to be modified. As in [1][21], placement requirement for island creation is implemented in a two-stage floorplanning technique: island-level floorplanning followed by chip-level floorplanning. This floorplanning technique is heavily dependent on the voltage assignment of the cores and can consume significant portion of the design time. Moreover, the number of IP blocks for future technologies is expected to grow exponentially [22]. In comparison to a 130nm technology node, where the number of IP blocks was about 100, the projected number of IP blocks in a 32nm is around 10,000. Aside from the inherent complexity of SoC design with a large number of IP blocks, certain CAD tools will face excessive runtimes as the number of IP blocks in a single SoC increases. To understand the severity of the problem, HotSpot Floorplanner [21] was used to floorplan voltage island based SoCs with the large number of cores. Figure 1.4 shows the runtime for the floorplanning with increasing number of IP blocks and voltage islands. In the SoCs, the cores[1] and their connections were generated using a technique similar to [23]. With twice as many cores and islands, it is observed that the time required for floorplanning increases by 3X. Thus, performing floorplanning iterations at each step of a heuristic-based approach will be very costly. As the number of cores in a SoC increases, the set of possible voltage assignments

[1] The terms IP block and core are used interchangeably here and refer to the same circuit block that is integrated in the SoC.

can also become quite large. The number of options would grow even further if more supply voltages are permitted. Many of the possible partitions do not lead to an acceptable solution. Attempting to floorplan assignments that do not meet power or area constraints wastes a lot of valuable time. More efficient methods to partition and floorplan a large set of IP blocks into voltage islands are needed and such methods are pursued in this work.

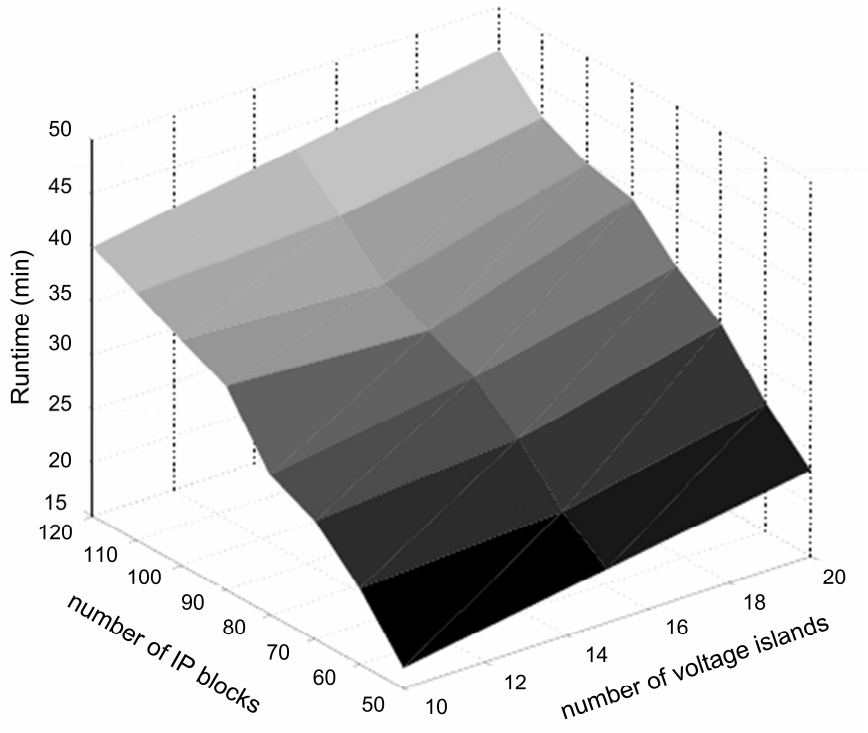


Figure 1.4: Floorplan time vs. number of IP blocks vs. number of islands

Voltage island design is dependent on the floorplanning of the IP blocks. During floorplanning, it is common for designers to impose constraints on the placement of the blocks. For example, a designer may want to place a constraint on the maximum separation between two modules if there are many interconnections between them, or place blocks close to power pins, or align them vertically in the middle of the chip for bus-based routing [24], just to name a few. Methods that incorporate such constraints are developed in this thesis for voltage island design.

1.3. Research Objectives

Having described the motivation and background for this research, a succinct thesis statement is as follows: *this research develops a holistic method of minimizing SoC power using voltage islands while including design constraints such as area, performance, power grid noise and placement requirements at multiple stages of the design flow*.

The following research objectives were identified for this work:

1) To develop *a new application-driven methodology* for choosing optimal supply voltages for each IP block of a SoC based on an application Power State Machine (PSM). The goal is to identify the suitable threshold voltage and then establish the supply voltage ranges that satisfy timing constraints for every IP block.

2) To propose a methodology for *choosing the finite set of supply voltages* to be implemented on the chip. The novelty of the approach is in the voltage assignment table generation approach and a cost function for voltage selection which incorporates information about power and the connectivity among cores. Additionally, certain placement constraints are to be applied to island formation in order to provide more realistic scenarios in SoC design.

3) To develop *a new fast floorplan-aware voltage partitioning approach* to select the most attractive solutions for floorplanning to reduce the power, area and run time. Moreover, such solutions must satisfy the power grid noise budget.

4) Explore the effect of design constraints on the SoC. Specifically, *placement constraints and power grid noise constraint will be incorporated* in the overall power

reduction methodology. In particular, the choice of proper amount of decoupling capacitance in order to meet IR drop noise constraint has been addressed in this work.

Figure 1.5 compares the proposed design methodology with the previous techniques for SoC-based Voltage Island Design.

SoC Based Voltage Island Design[1][21]
Begin
L1: Given the voltage assignment table and power value of each core in an SoC
L2: Use Heuristic Based Approach for voltage assignment to each core.
L3: Voltage Island Floorplanning
(a) Simulated Annealing for Floorplanning [21]
(b) Perturb Floorplan and Merge Islands [1]
L4: Perform cost calculation for each newly formed island
L6: If termination criteria are not met go to L2
end

Proposed Design Methodology
Begin
S1: Given the PSM of the application and power values of the cores in the SoC
S2: V_{DD} and V_T Selection *(Chapter 3)*
(a) Using activity factor find V_T of each core
(b) Using Power, Energy and Delay as design metrics perform V_{DD} selection.
S4: Address floorplanning issues *(Chapter 4)*
(a) Power Grid Noise
(b) Placement Constraint
S5: Voltage Island floorplanning using a two stage voltage assignment technique. *(Chapter 5)*
end

Figure 1.5: Voltage Island Design comparison

1.4. Thesis Organization

Chapter 2 identifies the different sources of power dissipation in synchronous CMOS circuits. A comprehensive literature review of recent low-power design is provided. In particular, multi-V_{DD} and multi-V_T techniques are described. In addition, a brief overview of the floorplanning technique in physical design is provided.

Chapter 3 presents an application-driven approach for identifying the suitable threshold and supply voltage for each IP block. The application to run on the SoC is first modeled as a Power State Model and is considered as the starting point of the design methodology. Based on the relative activity of the cores, the threshold voltage is assigned and the range of possible supply voltage is identified as well. The next task is to identify a few supply voltages that can

be implemented on the chip. Ideally, the larger the number of supply voltages, the greater is the potential power savings. However, in reality, only a small number of voltages can be used on the chip due to the associated cost for power grid design and the usage of pins. Here, a heuristic method is proposed for selection of the suitable supply voltages for the chip.

In Chapter 4, two additional constraints are considered. First, the challenges of satisfying the power grid noise constraint are addressed. Designers typically focus on satisfying power grid noise constraint after the final floorplan is available. Due to the higher complexity of power grid design in multi-V_{DD} chips, an early stage decap allocation methodology has been proposed in order to guarantee that power supply noise issues can be avoided after the final floorplan. Placement constraints, such as restricting memory blocks at certain corners of chip, as well as many other similar constraints listed earlier, are becoming common practice. In this chapter, the potential effect on power saving due to such constraints is explored as well.

Chapter 5 proposes a novel technique for voltage assignment to the cores as well as two-stage approach for floorplanning. The area and power design space is explored and a technique aimed at finding a valid solution that is close to the optimal solution in a fast manner is developed. Moreover, the power savings achieved at different stages of the design are quantified.

Chapter 6 provides conclusions and directions for future work.

CHAPTER 2 : BACKGROUND

2.1. Overview of Chapter

This chapter provides a background and overview of the sources of power consumption on a chip and presents several techniques for power optimization at different levels of the design flow. As voltage island design imposes a constraint on the floorplanning of the IP blocks a brief overview on floorplanning is provided. Classical floorplanning algorithms as well as their extension to address the different emerging floorplan challenges are discussed as well. Section 2.2 briefly reviews the components of power starting with the basic concepts. Section 2.3 discusses some of the relevant and most advanced methodologies for power savings. Section 2.4 reviews the different methods of floorplanning.

2.2. Sources of Power Consumption

There are two sources of power dissipation in synchronous CMOS digital circuits: active and static. Active power is consumed when the output of a CMOS circuit switches and static power is the power consumed when the circuit is idle but powered on. Let P_{active} be the active power while charging or discharging parasitic capacitances during node voltage transitions and P_{static} be the static power during idle mode of the circuit. Then, the following high-level equation characterizes the key factors in the power dissipation on a chip [25][26]:

$$P_{total} = P_{active} + P_{static}$$
$$P_{total} = \underbrace{(\alpha.C_L.V_{DD}{}^2 + Q_{SC}.V_{DD}) f_{clk}.N}_{P_{active}} + \underbrace{N(1-\alpha)V_{DD}I_{leakage}}_{P_{static}} \quad (2.1)$$

where α is the activity factor, C_L is the average load capacitance per gate, Q_{sc} is the short-circuit charge flowing from supply to ground during transition of the output of the gate, f_{clk} is

the clock frequency, N is the effective number of 2-input NAND gates and $I_{leakage}$ is the average NAND leakage current. Note that this equation could be applied to different sections or blocks of the chip and then summed up to produce more accurate results. This is due to the fact that the effective values of α, C_L, Q_{sc} and N are difficult to quantify for an entire chip. In the following subsections these power sources is analyzed in greater detail.

2.2.1. ACTIVE POWER

Active power is the power consumed during the switching of gates. Active power can be further subdivided into dynamic power (including glitches) and short-circuit power. Both are dependent on the switching activity. The CMOS complex gate structure considered here is shown in Figure 2.1(a). The simplest example of such circuit is the inverter shown in Figure 2.1(b).

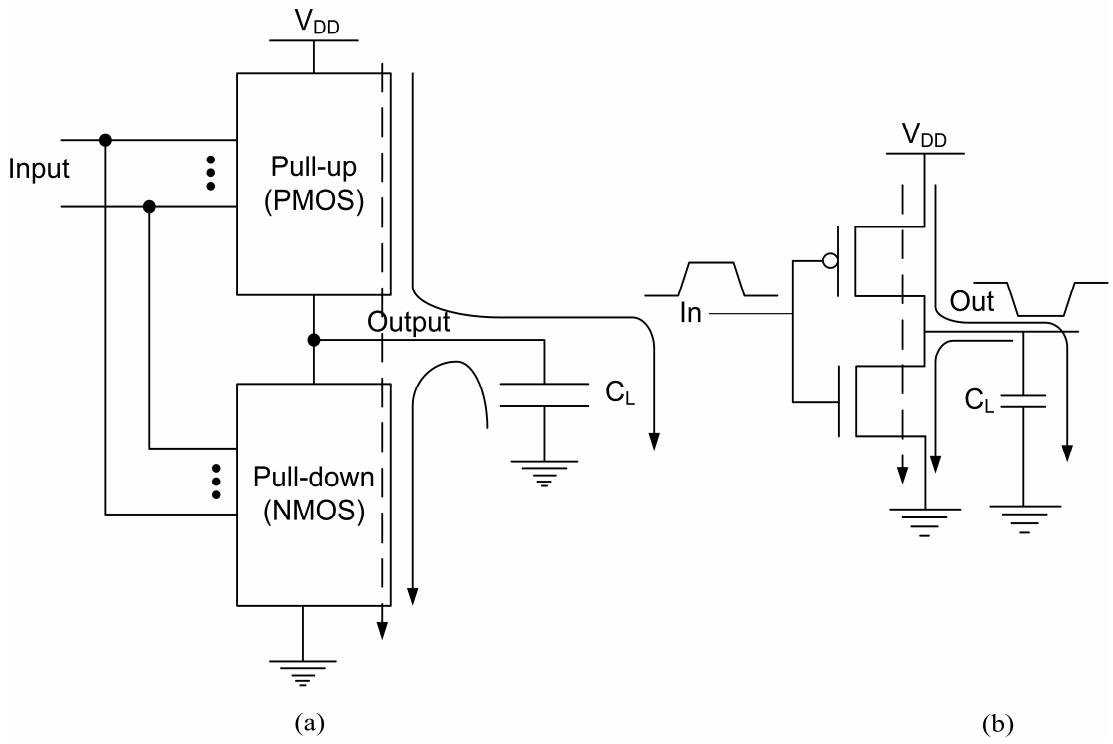


Figure 2.1: Active Power (a) CMOS Logic (b) Simple Inverter

Dynamic power is dissipated by charging and discharging the output capacitance C_L. During a low-to-high transition at the output, supply current charges C_L through one or more PMOS transistors. The capacitance C_L of a gate consists of the self-capacitance, wire capacitance of the fanout net and the capacitances of the gates connected to the fanout net. During a high-to-low transition at the output, the stored charge at C_L is discharged through one or more NMOS transistors. In case of the inverter, after one input transition from 0 to 1 and back to 0 the inverter has transferred a charge of $C_L V_{DD}$ from the power supply to ground. Assuming that the average switching activity of the inverter is α, the clock frequency being f_{clk}, the average dynamic power consumption of the inverter is:

$$P_{dynamic} = \alpha C_L {V_{DD}}^2 f_{clk} \tag{2.2}$$

For larger circuits, the C_L is the total switching capacitance and α is the average of the fraction of gates switching over all the clock cycles. The value of f_{clk}, the clock frequency, is at maximum the inverse of the critical-path delay (D_{crit}) of the circuit. The critical-path delay can be modeled as:

$$D_{crit} = \sum_i \frac{K_i V_{DD}}{(V_{DD} - V_T)^{\varphi}} \tag{2.3}$$

which is the sum of all gate delays in the path, K_i and φ are fitting parameters similar to the approaches in [27][28].

During a switching event short-circuit power is dissipated when there is a momentary direct path from supply to ground. The finite slope of the input signal causes this current, I_{SC} (dotted line in Figure 2.1), to flow when the input voltage is between V_{TN} and (V_{DD}- V_{TP}), when both devices are turned on (V_{TN} and V_{TP} are the threshold voltages of NMOS and PMOS transistors

respectively). The amount of current flowing depends upon the relative size of the PMOS and NMOS devices, the value of the output capacitance as well as the rise and fall times. The following expression models short-circuit power with no external load [29]:

$$P_{short} = \frac{\beta}{12}(V_{DD} - 2V_T)^3 \tau f_{clk} \tag{2.4}$$

where β is the gain factor of the MOS transistor and τ is the rise (or fall) time of the inverter. As the short-circuit current is a strong function of the ratio of the input and output slopes, there exists a tradeoff between dynamic power of the previous gate and short-circuit power of the next gate [25]. At the circuit level, it is useful to keep the rise and fall time of all signals within a constant range to minimize the power dissipation due to short-circuit current. For most circuits, short-circuit power is approximately 10% of the active power [30].

Glitch power is expended when the inputs to a gate do not arrive at the same time causing unwanted transitions of the output signal before the final value is reached. This behavior is due to different delays on signal paths leading to the different inputs of a gate. Glitches tend to propagate through the fanout gates and cause unintended transitions in the subsequent stages, increasing power dissipation even further [31]. For example, glitches are a major contributor of power in complex structures such as adders and multipliers. In a typical combinational circuit, glitch power can contribute up to 40% of the active power [32]. Both glitches and short-circuit power can be incorporated in the "α" of the dynamic power in Eqn. (2.2).

In the past technologies, dynamic power was the major component of power consumption of all VLSI circuits. However, with the emergence of deep submicron technology, power due to

device leakage when they are off contributes significantly to the total power consumption [33]. This falls into the category of static power which is described in the next section.

2.2.2. STATIC POWER

Static power is associated with the non-switching periods of internal nodes of a circuit between switching events. Therefore, this power consumption, caused by the various forms of leakage currents of transistors, does not contribute to any useful computation and in this sense is wasted energy. Device scaling, and the associated reduction of threshold voltage, channel length and gate oxide thickness [33] have introduced the different forms of leakage current and caused static power to be a dominant part of the total power consumption of the chip power. Static power can be represented as:

$$\begin{aligned} P_{static} &= I_{leakage} V_{DD} \\ I_{leakage} &= I_{SUBTH} + I_G + I_{REV} + I_{GIDL} \end{aligned} \tag{2.5}$$

where $I_{leakage}$ is the sum of subthreshold leakage (I_{SUBTH}), gate leakage (I_G), p-n junction reverse bias leakage (I_{REV}) and gate-induced drain leakage (I_{GIDL}). An NMOS transistor with the different leakage components are shown in Figure 2.2.

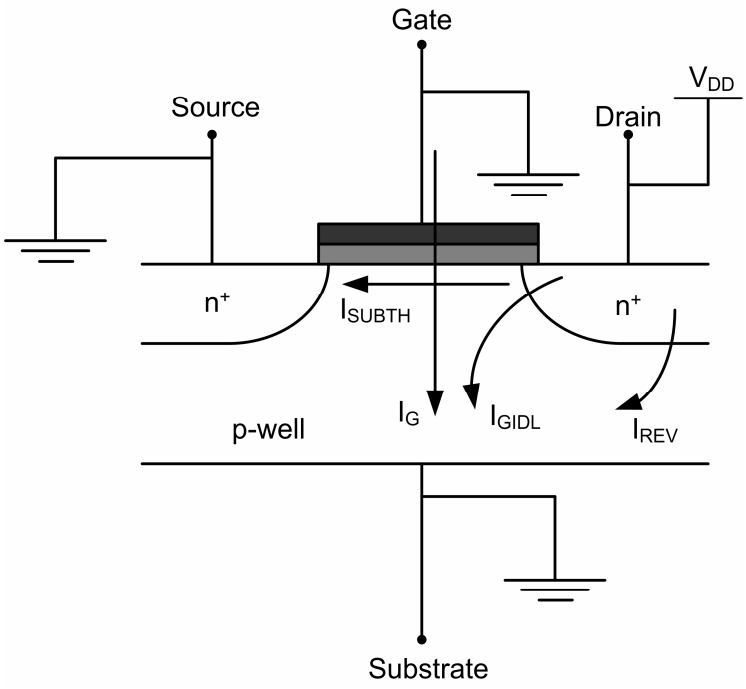


Figure 2.2: Leakage current components of an NMOS transistor [34]

Of the different sources of leakage current, subthreshold leakage is currently the main source of static power [33]. This leakage current occurs in the weak inversion region of transistor operation where current flows between the source and drain nodes even with a gate voltage that is smaller than the threshold voltage. In Figure 2.3, subthreshold leakage current flows through P1, indicated with a solid arrow, when the corresponding input is “1” and, likewise, through N2 when the input is “0”.

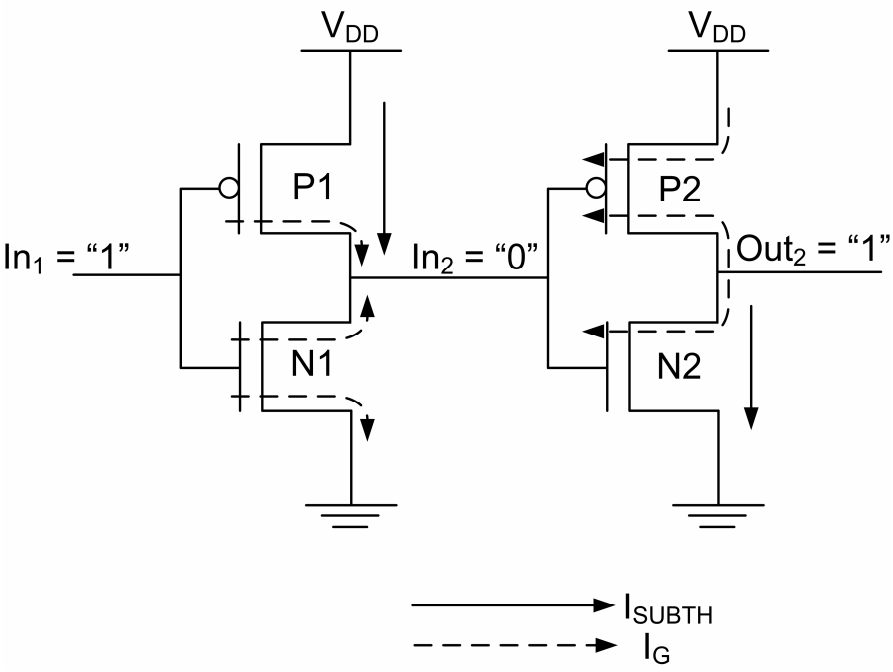


Figure 2.3: Subthreshold and Gate leakage in CMOS circuit

For a single gate, the subthreshold leakage can be modeled using a simplified expression [35]:

$$I_{leakage} = Ae^{\frac{q}{nkT}(V_{GS}-V_T)}(1-e^{\frac{-qV_{DS}}{kT}}) \tag{2.6}$$

where A is a technology-dependent constant, V_{GS} is the gate-to-source voltage, and V_{DS} is the drain-to-source voltage. Due to exponential relation between $I_{leakage}$ and V_T, an increase in V_T would sharply reduce the leakage current, but would also slow down the device as a direct consequence.

Due to thinning of gate oxide with technology scaling, leakage current through the insulator (dotted line in Figure 2.3) is increasing in its contribution to the total static power [36]. Unlike subthreshold leakage, which only exists in weakly turned-off transistors, gate leakage of some form exists no matter whether the transistor is turned on or turned off [37]. Gate Leakage current can be expressed as [38]:

$$I_{gate} = W_{eff} L_{eff} A(\frac{V_{ox}}{T_{ox}})^2 \exp\left(\frac{-B\left(1-\left(1-\frac{V_{ox}}{\Phi_{ox}}\right)^{\frac{3}{2}}\right)}{\frac{V_{ox}}{\Phi_{ox}}} \right) \quad (2.7)$$

where V_{ox} is the potential drop across the thin oxide, Φ_{ox} is the barrier height for the tunneling particles, W_{eff} and L_{eff} are the effective width and length of the gate, T_{ox} is the oxide thickness and A and B are physical parameters. As technology scales, T_{ox} is reduced causing a significant increase in gate leakage. With high-k dielectrics, this problem may be mitigated in future technologies [39].

Other sources of leakage current such as junction tunneling current, bulk punchthrough and gate-induced drain current are secondary sources of leakage current. However, one should note that each of these terms may vary in relative importance in future technologies. For leakage power reduction, the work presented in this dissertation aims at reduction of subthreshold leakage current as it is the dominant source of leakage power today.

2.3. Low Power Design Techniques

As discussed in Chapter 1, low power design methodologies can be implemented at different stages of the design process. The following subsections briefly describe these techniques from dynamic and leakage power reduction perspectives. As multi-V_{DD} and multi-V_T design challenges are explored extensively in this thesis, these power optimization methods have been discussed in greater detail.

2.3.1. DYNAMIC POWER REDUCTION TECHNIQUES

Dynamic power can be reduced by reducing any of the terms in Eqn. (2.2). Thus, designers have devised methods of reducing one or more of the parameters that affect dynamic power. Generally, there are two main approaches for power reduction: power supply reduction and switched capacitance (C_{eff}) reduction; switched capacitance is the product of the activity α and the total load capacitance (C_L). These techniques can be implemented at the architecture, logic design and circuit design levels. The following subsections provide an overview of these techniques.

2.3.1.1. Clock Gating

A significant portion of the dynamic power in a chip can be due to the distribution network of the clock. Because of the switching of clock buffers during every clock cycle, the clock signals in a digital computer can consume between 15%-45% of the chip power [40]. Additionally, flip-flops and registers receiving the clock signal can also cause some gates within them to dissipate dynamic power even though their primary inputs may not have changed. The most common way to reduce this power is to turn off the clock when it is not required. The benefit of using clock gating technique is two-fold. First, when the clock is stopped, no power is consumed by the combinational logic, as their inputs remain unchanged. Second, the gated clock-line as well as the flops and registers do not consume any additional power.

Clock gating was first proposed in [41] and studied in greater detail in [10][42][43]. In [41], using initial specification as a state transition graph of the circuit to be synthesized, conditions under which the next state and the output signals do not change are identified first. In [10], the authors presented a technique for saving power in idle circuits by stopping the clock fed to

them. In [42], a precomputation-based methodology is used to generate a signal to control the load enable pin of the flip flops in the data path. In [43], the authors use a latch to gate the clock in control-dominated circuits. Though clock gating can be implemented using simple control circuitry, caution must be taken on glitches, clock loading effects and additional clock skews.

2.3.1.2. Operand Isolation

Operand isolation is another method of reducing unnecessary activity in the circuit and thereby reducing dynamic power. This method prevents sections of the circuitry from accepting changes in their inputs unless they are expected to respond to those changes [44]. This technique may not be as effective for primitive logic but it saves power in more complex functions.

Adders, multipliers, etc., are likely candidates for implementation of isolation logic. Another good application is circuit parts that have to be active only once after many clock cycles. One of the key challenges in operand isolation is the availability of a signal to indicate that a module is performing a computation that is not redundant. In [45], such a signal is extracted from the logic description of the circuit based on the *don't care* conditions of the modules to be isolated. A comprehensive technique for automating operand isolation at the RT-level has been proposed in [46]. Experimental results show up to a 30% reduction in dynamic power using this technique [46].

2.3.1.3. Gate-level Transformation

Gate-level transformation is a logic optimization technique where a small group of gates are replaced by a logically-equivalent set in order to reduce dynamic power. There are a number

of techniques that can be implemented and the majority of them are performed by the synthesis tools. Following is an overview of some of the methods.

Re-mapping: This technique optimizes the logic expressions in order to reduce dynamic power. The idea is to "hide" nodes with higher switching activity within the cells, thereby having smaller switching capacitance. In [47], a two-level approach for mapping has been proposed. In the first step, the power-delay curve (power vs. arrival time) for each node is computed. Next, a mapping solution is proposed according to the curves and the required time at the primary inputs. Results in [47] show 18% reduction in power with 16% increase in area.

Pin Swapping: When gates with multiple inputs have different input capacitances, pin swapping swaps the inputs in such a way as to map high activity nets to inputs with low capacitance and low activity nets to inputs with high capacitance. The area cost here is low but the overall improvement may not be significant, depending on the circuit.

2.3.1.4. Path Balancing

Path delay balancing is used to reduce glitch power. The connectivity among gates has a strong correlation to the overall switching activity and hence the dynamic power dissipation. To avoid spurious transitions, the delays of paths that converge to the same gate must be made equal, or as close to equal as is practical. Another popular technique for path balancing is inserting delay elements in the faster input paths so that the delay on each path equalized [48][49][50]. This modification would not contribute any additional delay to the critical path but would reduce the number of spurious transitions. Buffer insertion must be done in a judicious manner as it increases the effective capacitance thereby increasing the dynamic power. Glitches can also be eliminated using a *hazard filtering* technique [51]. Where a glitch

might occur, the delay of the offending gate is increased such that it is at least equal to the differential delay of the inputs.

2.3.1.5. Gate Sizing

Gate sizing determines the device widths for each gate. The basic rule is to use smaller transistors that satisfy the delay constraints. Thus, to reduce dynamic power, gates with higher toggle rates must be made as small as possible. The optimization method is especially challenging when there is a chain of gates on a path as there are multiple choices for each gate that can be used to realize the maximum delay. Each choice of gate provides a different area-power-delay cost and thus it is not immediately obvious which choice would provide the best solution. A polynomial formulation using Elmore delay models is used in traditional gate sizing approaches. Heuristic-based greedy approaches [52][53][54][55][56][57][58] can be used to solve such a polynomial problem. In general, though heuristic algorithms are relatively fast, they cannot guarantee to reach a global optimum. However, a linear programming method is proposed [59] in which a piecewise linear delay model is adopted to achieve a global optimal solution. A nonlinear programming approach [60] gives the most accurate optimal solution but has a higher runtime cost.

Additionally, short-circuit power should be considered while sizing the gates. For a given transition time, a larger gate would contribute to a larger short-circuit current. In contrast, the output of a larger gate has sharper slope thereby reducing the short -circuit power of the fanout gates. Some degree of circuit balancing is needed to produce an optimal solution. Thus, careful sizing of gates should be used to reduce overall power as opposed to only dynamic or short-circuit power.

2.3.1.6. Transistor Sizing

Transistor sizing is similar to gate sizing with the exception that in gate sizing all the transistors are sized together with the same factor while in transistor sizing each individual transistor is sized individually. For a gate on a critical path, only a few of its transistors contribute to the largest intrinsic gate delay, and so the rest of the transistors can still be sized to reduce the capacitances. Optimizations by sizing individual transistors have been explored in [61][62][63][64].

2.3.1.7. Voltage Scaling

Due to the quadratic relationship between V_{DD} and active power (Eqn. 2.2), reducing the supply voltage is perhaps the most attractive approach for active power reduction. The major shortcoming of this solution is the increase in the delay of gates (Eqn. 2.3). The lost throughput can be compensated using the following three methods [65]:

1) Redesign the circuit using pipelining and parallelism
2) Reduce V_T in order to compensate for V_{DD} reduction
3) Assign lower V_{DD} values to non-critical paths

The first technique using pipelining/parallelism is an architectural optimization. Though the lost speed can be regained in the form of higher throughput, the new implementation might require more area and generate greater switching activity. In the second case, lowering V_T enhances the gate overdrive of the transistors but also causes an exponential increase in leakage power. Thus, there is a complex tradeoff between the V_{DD} and V_T selection for total power reduction. The third method poses challenges for power routing within blocks due to

the use of multiple supply voltages, and requires careful consideration due to voltage level shifting in these circuits.

Supply voltages can be reduced dynamically or in a static manner. Dynamic Voltage Scaling (DVS) reduces supply voltage during circuit operation based on the throughput requirements of the system. Multiple supply voltage implementation is a static technique that pre-assigns different supply voltages to different IP blocks/subcircuits/gates based on the throughput requirement. The following subsections discuss these techniques in further detail.

2.3.1.8. Dynamic Voltage Scaling

Dynamic voltage scaling [66] is used in systems where peak performance is not required for the entire duration of circuit operation. This is seen in portable devices such as cell phones, PDAs, etc., which typically contain three operating modes: computation-intensive mode, low-speed deadline-driven mode and idle mode [67]. Compute-intensive tasks, such as MPEG video and audio decompression, demand maximum throughput. In contrast, tasks such as text processing, data entry, etc., have relaxed deadlines and require a fraction of the system throughput. Finishing these tasks early has no benefits and thus dynamic voltage scaling can be applied at the cost of reduced speed. DVS utilizes this slack time by lowering the supply voltage as well as the clock frequency to achieve quadratic savings in energy. Commercial chips like IEM926 have an *Intelligent Energy Manager* (IEM) that implements DVS. The IEM has software as well as a hardware unit that monitors the system workload. Using information from the operating system, as well as historical data from the execution of the software, the IEM makes predictions on future workloads. Additionally on the hardware side, there is an Adaptive Power Controller (APC) used for implementation of DVS [68]. Figure 2.4 shows the block diagram of how DVS is implemented in that particular chip. Depending on

the task deadline the APC dynamically sets the corresponding supply voltage for the system processor.

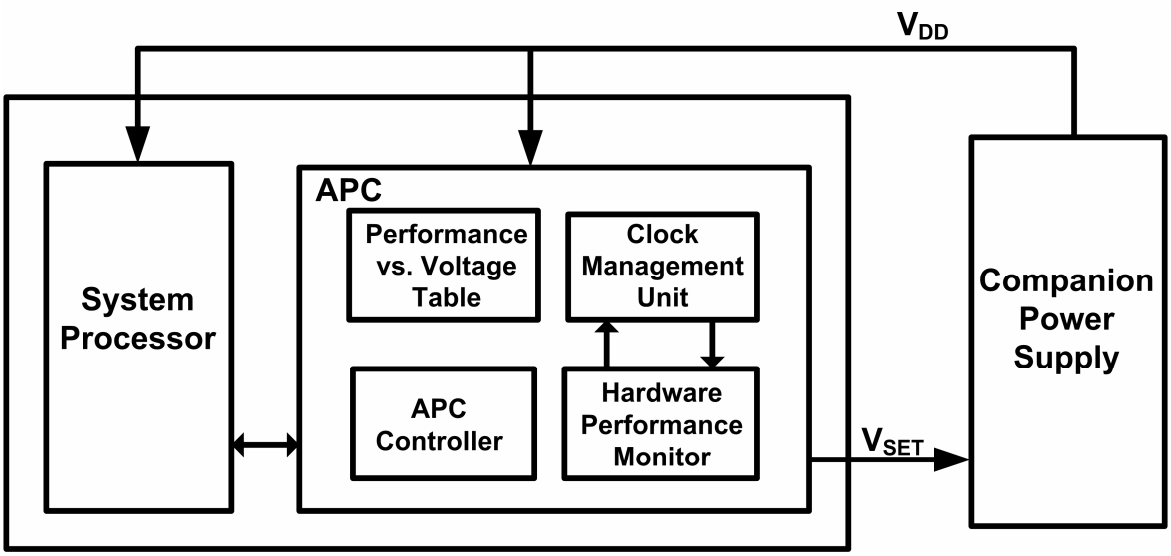


Figure 2.4: Implementation of DVS in the IEM926 [68]

2.3.1.9. Multi-V_{DD} Approach

In comparison to DVS, multi-V_{DD} design is a static approach for reducing dynamic power. A separate supply voltage can be assigned to every module of the design (i.e., the voltage island technique) or, in the extreme, on a gate-by-gate basis (Dual-V_{DD}). First, the dual-V_{DD} approach is discussed followed by the voltage island technique.

Dual-V_{DD}: This approach assigns supply voltages at a small granularity by setting the supply voltage for each cell within a block. Every cell in a module can be assigned to one of the two supply voltages, high-V_{DD}(V_{DDH}) or low-V_{DD}(V_{DDL}) [69]. Gates that are on the critical path of the block are assigned to the V_{DDH} while gates on the non-critical path are assigned to V_{DDL}. When gates operating at V_{DDL} are fanins to gates operating at V_{DDH}, level converters are required in order to avoid short-circuit power. As level converters contribute to additional power consumption, minimizing converters is an additional design issue [70]. The two

methods for V_{DD} assignment to gates are: 1) Clustered Voltage Scaling (CVS) [71] and 2) Extended Clustered Voltage Scaling (ECVS) [72]. In CVS, the cells driven by each supply voltage are clustered such that level conversion is only required at the output flip-flops. Figure 2.5 shows the basic implementation technique.

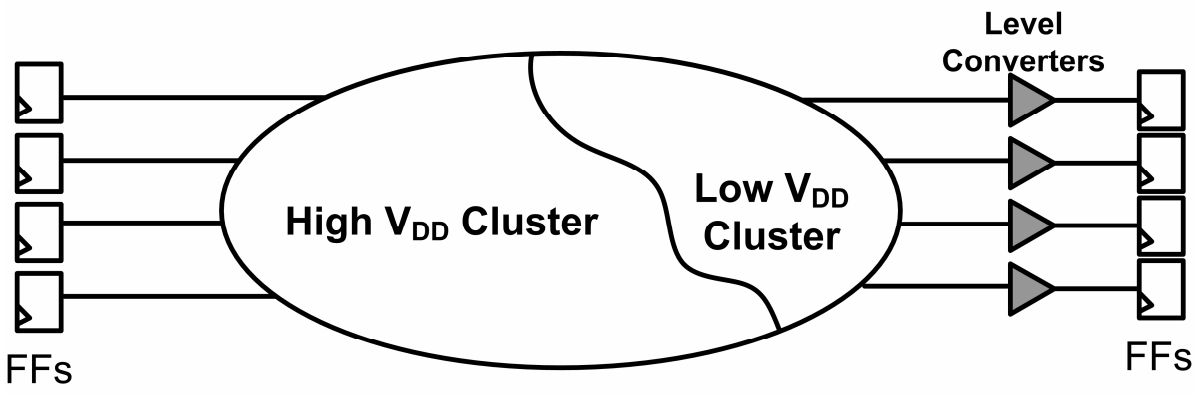


Figure 2.5: CVS implementation

Extended clustered voltage scaling (ECVS) [72] removes restrictions on level converter assignment by allowing level conversion anywhere, and the supply voltage assignment to the gates is much more flexible. Though the algorithm of ECVS is more complicated than that of CVS, it provides greater dynamic power savings.

Voltage Island Design: The block-based multi-V_{DD} design using voltage islands was first proposed in [13]. This technique requires IP blocks with same supply voltage to form contiguous groups called "*islands*". This facilitates the use of different supply voltages for different parts of the chip thereby reducing overall power consumption. Figure 2.6 shows a SoC having 15 IP blocks before and after creating voltage islands. In Figure 2.6(a), all the blocks are assigned to 1.5 V, while in Figure 2.6(b), the performance-critical blocks are assigned to 1.5 V, and the other blocks are assigned to either 0.9 or 1.2 V, depending on their speed requirements. The overall power dissipation of the design in Figure 2.6(b) can be far

less than that in Figure 2.6(a) due to reduced dynamic power. Available white-space is used for placing level shifters, routing and the allocation of decoupling capacitance to reduce power supply noise.

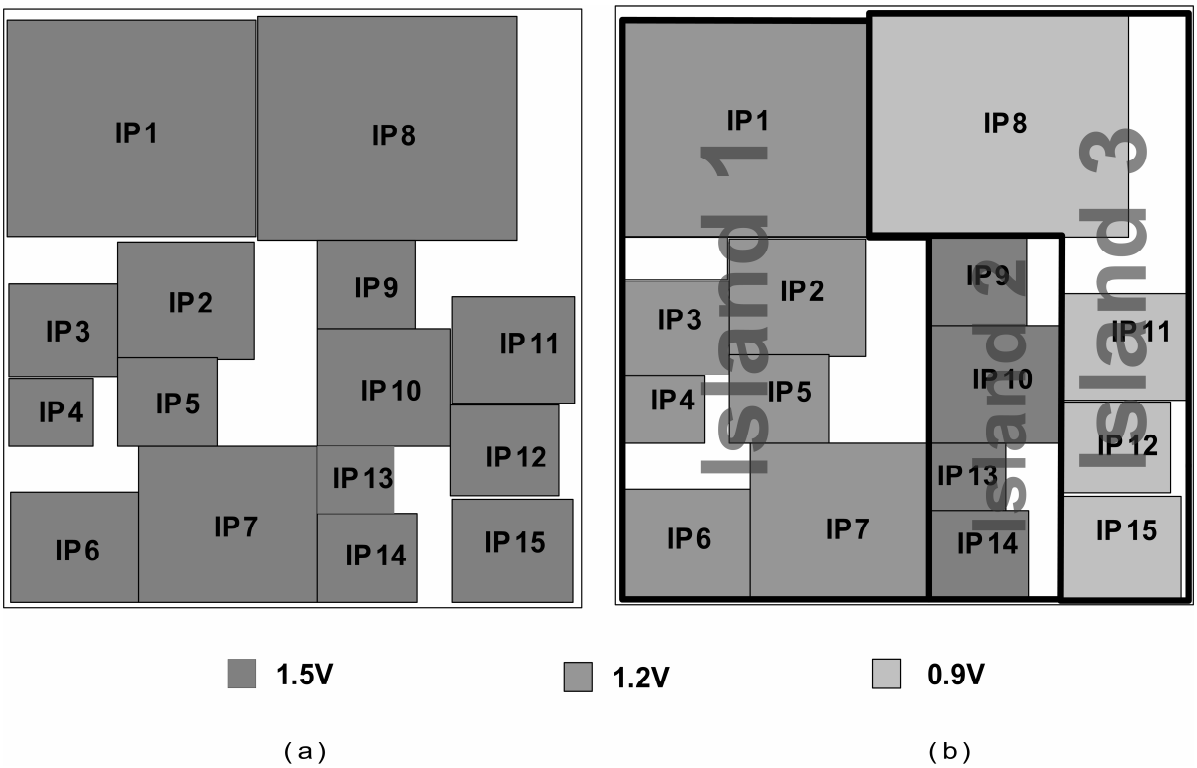


Figure 2.6: SoC with (a) Single Voltage Island and (b) Three Voltage Islands

Voltage island design can be implemented at either the *floorplan/placement* stage or *post-floorplan/post- placement* stage. Following is a brief overview of the different proposed techniques for voltage island design in each stage.

Floorplan/Placement stage: Voltage island design challenges at the floorplan/placement stage have been studied extensively in [13][1][21][73][74][75][76]. An approach involving perturbation for voltage assignment of cores, and simulated annealing for floorplanning is described in [1]. A similar approach is taken in [21] with the additional objective of achieving a thermally-balanced, island-based SoC design. Partitioning based on genetic algorithms, and

floorplanning based on simulated annealing have been used to reduce hotspots and achieve more uniform temperatures across the chip. In [74], based on the block timing requirements, voltage assignment is performed first. Next, level shifters are inserted and floorplanning is performed using a power network "aware" floorplanner such that the sum of perimeters of voltage islands is minimized. Fine-grained voltage assignment during placement of standard cells has been proposed in [76]. Simultaneous voltage assignment, voltage island generation and floorplanning have been considered in [75]. The work in [73] extends the idea by incorporating timing constraints in the voltage assignment and floorplanning problem.

Post-floorplan/Post-placement stage: In [78][79][80][81], voltage islands are created from fixed floorplans as shown in Figure 2.6. Considering placement adjacencies of the logic blocks, in [78][80], voltage islands are created such that power versus design-cost tradeoffs under performance requirement constraints is optimized. In [79], a 0-1 integer-linear programming (ILP) method is proposed for voltage assignment to the cores in the SoC. Next, from a finite choice of floorplan solutions, the candidate floorplan that allows island creation based on the voltage assignment is chosen. In [81], delay budgeting and the corresponding voltage assignment use a zero-slack algorithm and then, based on physical proximity, the actual voltage islands are created.

The major impact of using multiple supply voltages is the need to treat the supply voltage as another design constraint. In voltage island design, IP blocks assigned to unique islands require proper delivery of the specific supply voltage. If the supply voltage is generated off-chip, proximity to its corresponding supply voltage pins must be ensured. On the other hand, on-chip supply voltage generation requires extra routing to each island. Similar to CVS and

ECVS, communication between islands require level converters. Clock-tree generation must take into account buffers residing in different islands. Designers must also account for coupling noise between adjacent lines driven by high and low voltages arising from the different V_{DD} values.

2.3.2. STATIC POWER REDUCTION TECHNIQUES

One of the negative effects of aggressive device scaling is the relative increase in static power. Lowering V_T is one of the ways of achieving faster devices. As explained earlier, this causes an exponential increase in static power due to the exponential relation between subthreshold leakage current and V_T (refer to Eqn. (2.6)). The problem becomes even worse when the temperature of the chip increases. Higher temperature increases the subthreshold leakage and, in the worst case, can exceed the power budget. Also, due to the reduction in gate-oxide thickness for sub-100nm CMOS, gate leakage is also contributing significantly to the overall leakage power. However, for future technology nodes, it is expected that high-k dielectrics would mitigate the increase of gate leakage as it appears to be the only effective way of reducing gate leakage [7].

In general, circuit blocks are constantly turning on and off, depending on the computation requirements of an application. Due to this "bursty" nature of operation, spend a significant amount of time in their idle mode. Different circuit techniques have been proposed to reduce leakage power of such circuits. Leakage optimization can be performed at *design time* using dual-V_T and transistor stacking methods. Such approaches exploit the delay slack in non-critical paths to reduce leakage power. *Runtime* techniques such as standby input vector selection, variable threshold CMOS (VTCMOS) and power gating reduce leakage power

when circuits are not required to run at the highest performance level. These techniques are discussed in the following sections.

2.3.2.1. Transistor Stacking

Leakage current flowing through two serially stacked off CMOS transistors is always less than a single off device. This phenomenon is known as transistor stacking [38][82][83]. In Figure 2.7, turning off both M1 and M2 causes the intermediate node V_M to have a positive value. This causes the threshold voltage of M1 to increase due to body effect, and also reduces the drain-source voltages on the two devices. All these effects contribute to a reduction of subthreshold current flowing through M1 and M2.

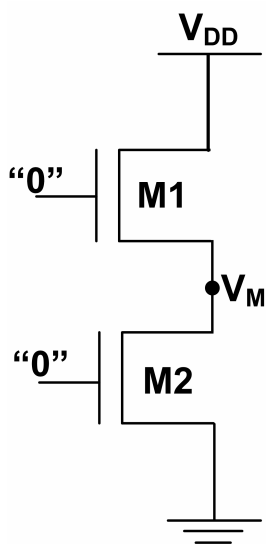


Figure 2.7: Transistor Stacking

Thus, replacing one off transistor with serially-connected, stacked transistors reduces leakage current significantly [84][85]. The disadvantage of adding the extra transistor is performance degradation and higher dynamic power due to increased capacitance.

2.3.2.2. Standby Input Vector Selection

Due to the transistor stacking effect, leakage current also depends upon the input vectors presented to a gate. Functional blocks such as NAND, NOR and complex gates inherently have stacks of transistors. Thus, if the input vector of a circuit entering in the standby mode is chosen carefully, then leakage power can be minimized by maximizing the number of stacked transistors in the off state. Thus, selection of proper input vector is critical to leakage power saving. Numerous techniques for choosing the sleep vector have been proposed in [86][87][88][89][90]. Of course, applying such an input vector to minimize leakage consumes additional dynamic power. This technique should be used only when this extra power can be offset by the leakage power saving in the standby mode.

It must be noted that the leakage power reduction by natural/forced transistor stacking is principally due to reduction in subthreshold leakage current. In fact, the traditional method of stacking transistors to reduce subthreshold leakage current might increase thin-oxide gate leakage power [38]. For example, in some cases a three-transistor stack (NMOS) with input "100" may result in less leakage than the same gates with "000" inputs, even though "000" inputs would maximize subthreshold leakage reduction. In the future, this may not be an issue with high-κ dielectrics that will reduce thin-oxide gate leakage.

2.3.2.3. Dual-V_T Technique

Today's ASIC designers can choose standard cells from cell libraries with different V_T. During design synthesis, a high V_T is assigned to some cells in the non-critical paths and low V_T transistors are assigned to cells in the critical paths [91] so that performance is not sacrificed. In this way, leakage power savings can be obtained while maintaining the desired

performance. Figure 2.8(a) shows an example of a dual-V_T implementation and Figure 2.8(b) shows the path distribution of a 32-bit adder using dual-V_T.

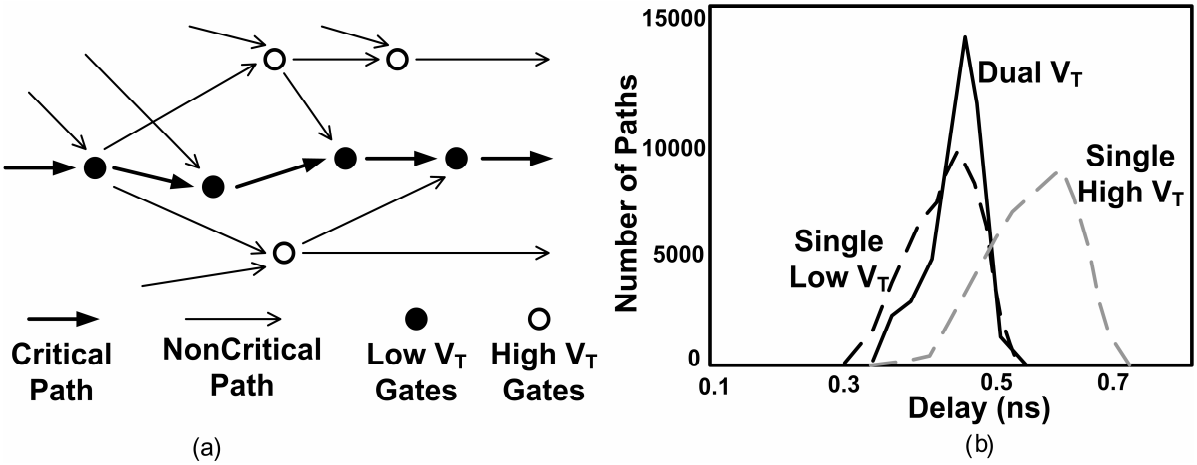


Figure 2.8: Dual V_T implementation [91]

The potential saving that can be achieved using this technique depends on the number of gates assigned to a high V_T. Numerous deterministic [92][93][94][91] and ILP-based [95][96] techniques utilize the slack in the non-critical paths to assign a high V_T. It should be noted that no additional circuitry is required to implement this technique. Additionally there is no performance or power penalty. This technique is also effective in reducing leakage power in standby mode. Design techniques to improve performance by increase the size of the high V_T transistors have been proposed in [92][93][96][97]. However, increasing the size of the transistors increases dynamic power and area. Therefore, there is a tradeoff between dynamic power and the use of low V_T transistors.

In general, transistors with different values of V_T differ in their doping profile. Additionally, a higher oxide thickness, t_{ox}, can be used to obtain high-V_T devices. Multi-t_{ox} CMOS can optimize subthreshold as well as gate leakage. An algorithm for selecting and assigning optimal transistor oxide thicknesses is described in [98].

2.3.2.4. Variable Threshold CMOS (VTCMOS)

VTCMOS is a body-biasing technique for leakage power reduction [99]. The threshold voltage of a short-channel MOSFET transistor in the BSIM model is given by [100]:

$$V_T = V_{T0} + \gamma(\sqrt{\phi_s - V_{BS}} - \sqrt{\phi_s}) - \theta_{DIBL} V_{DD} + \Delta V_{NW} \qquad (2.8)$$

where V_{T0} is the zero-bias threshold voltage, the terms Φ_s, γ and θ_{DIBL} are constants for a given technology, V_{BS} is the voltage applied between the body and source terminals of the transistor and ΔV_{NW} is a constant model that models narrow-width effects. Using Adaptive Body Biasing (ABB) [101][102][103][104][105], the threshold voltage of the transistors can be increased in the standby mode while reduced dynamically in the active mode of operation depending upon the required performance level. The threshold voltage is changed by controlling the V_{BS} (in Eqn. (2.8)) as shown in Fig 2.9(a). ABB can be applied to the entire die or it can be specified on a block-by-block basis and controlled individually. There may be some limits on the usefulness of this technique in future generations as large reverse biases on the drain/source junctions will increase junction leakage, and perhaps force the junctions into the breakdown region of operation. However, at present, it is still a viable option for leakage reduction.

Dynamic Threshold Voltage Scaling (DVTS) is a method of controlling the threshold voltage using substrate biasing proposed in [106]. A block diagram of the DVTS method is shown in Figure 2.9(b). The desired clock speed is decided by the clock speed scheduler. The DVTS controller then adjusts the body-bias voltage to have the system run at the same clock frequency.

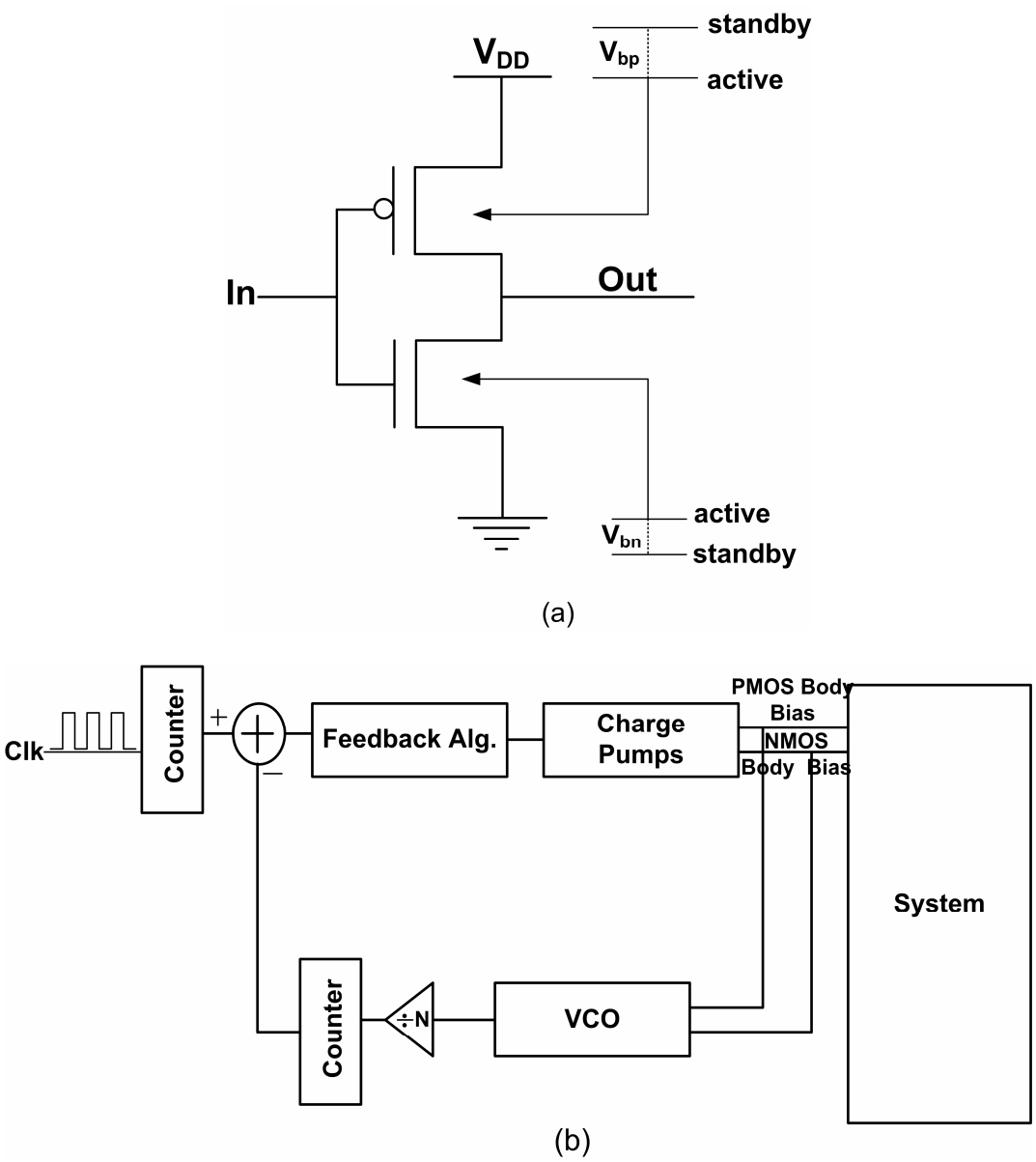


Figure 2.9: VTCMOS (a) Circuit implementation (b) System implementation

In order to provide the body-bias voltage to the appropriate circuits, routing of the body-bias grid is required. This adds to the overall area of the chip and especially to routing congestion in the metal layers. And, as mentioned above, recent results show that the effectiveness of reversing the body biasing has decreased significantly due to the increase in band-to-band tunneling leakage at the source/substrate and drain/substrate p-n junctions [107]. Another method, which has the same effect as body-biasing is to raise the NMOS source voltage while

tying the NMOS body to ground [108]. This method eliminates the requirement for a triple-well process because the substrate of the logic and the control circuit can be shared. In addition to increasing the source voltage of the NMOS transistors, the body of the PMOS is raised to keep the charge at the storage node constant. Figure 2.10(a) illustrates the circuit implementation while Figure 2.10(b) shows the voltage values in active and standby mode.

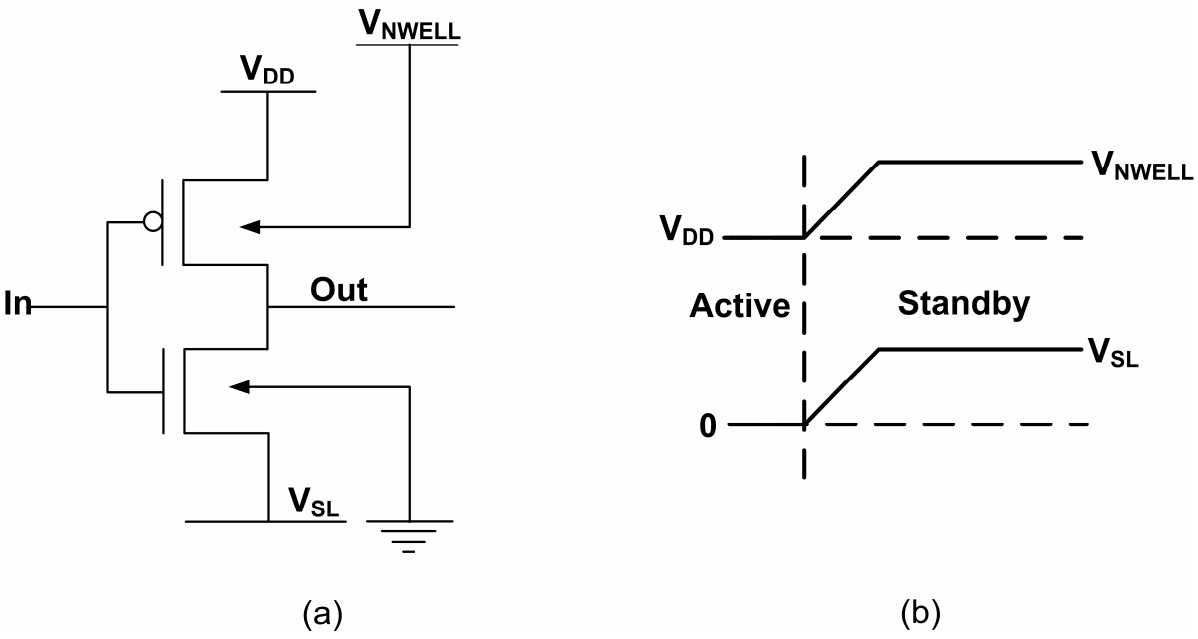


Figure 2.10: Body Biasing by changing source voltage with respect to body voltage

2.3.2.5. Power Gating

Power Shut-Off is another major approach for leakage power reduction, which is also applied in this work. This technique is implemented by inserting a high-V_T transistor between the power supply and the original circuit [109][110][111][112][113]. Figure 2.11 shows high-level Power Gating implementation of IP blocks in a SoC.

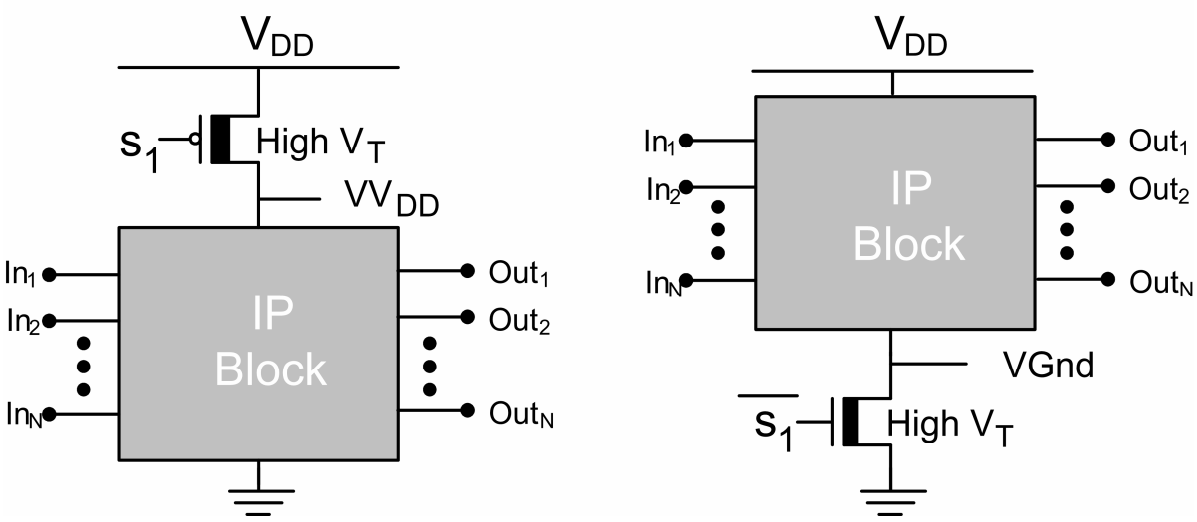


Figure 2.11: Power Gating

As shown in the figure, the sleep transistor can be implemented using a high-V_T PMOS device (or header) or a high-V_T NMOS (or footer). However, footers are usually preferred because the NMOS on-resistance is small and thus it can be sized smaller as compared to PMOS [114]. The goal is to switch between active and sleep modes by selectively turning the sleep transistor on/off. Decisions on shutting down the power of a block can be embedded in software as part of the operating system or can be enabled using hardware such as a system-level power management module. Figure 2.12 depicts the activity profile of a system where only dynamic power optimization is performed using *clock gating* [7]. Fig 2.13 shows the same example with *power gating* [7]. The response time between active (WAKE) and sleep (SLEEP) mode is significant and cannot be ignored. The maximum leakage power saving is obtained well into the sleep mode due to the thermal condition of the block before entering the sleep mode. However, insertion of large sleep transistors incurs area and delay penalties. Moreover, if data retention is required in the standby mode, then extra circuitry would be needed to store the data [109].

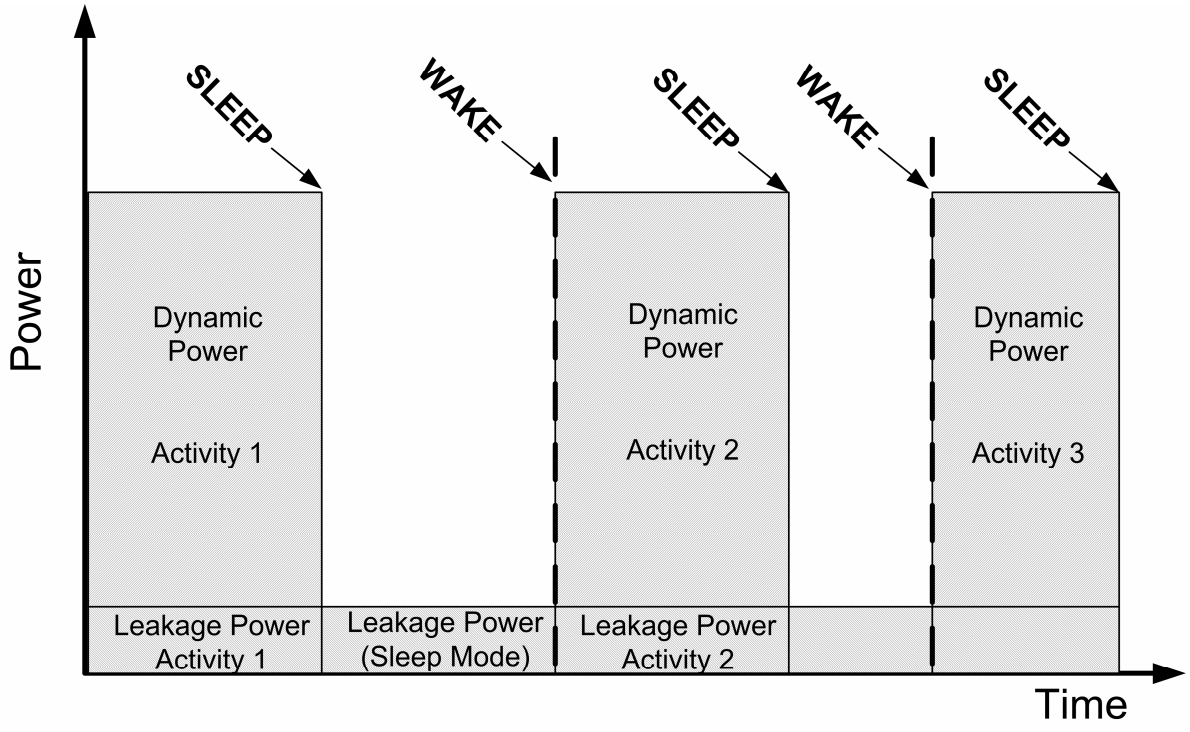


Figure 2.12: Activity Profile without Power Gating [7]

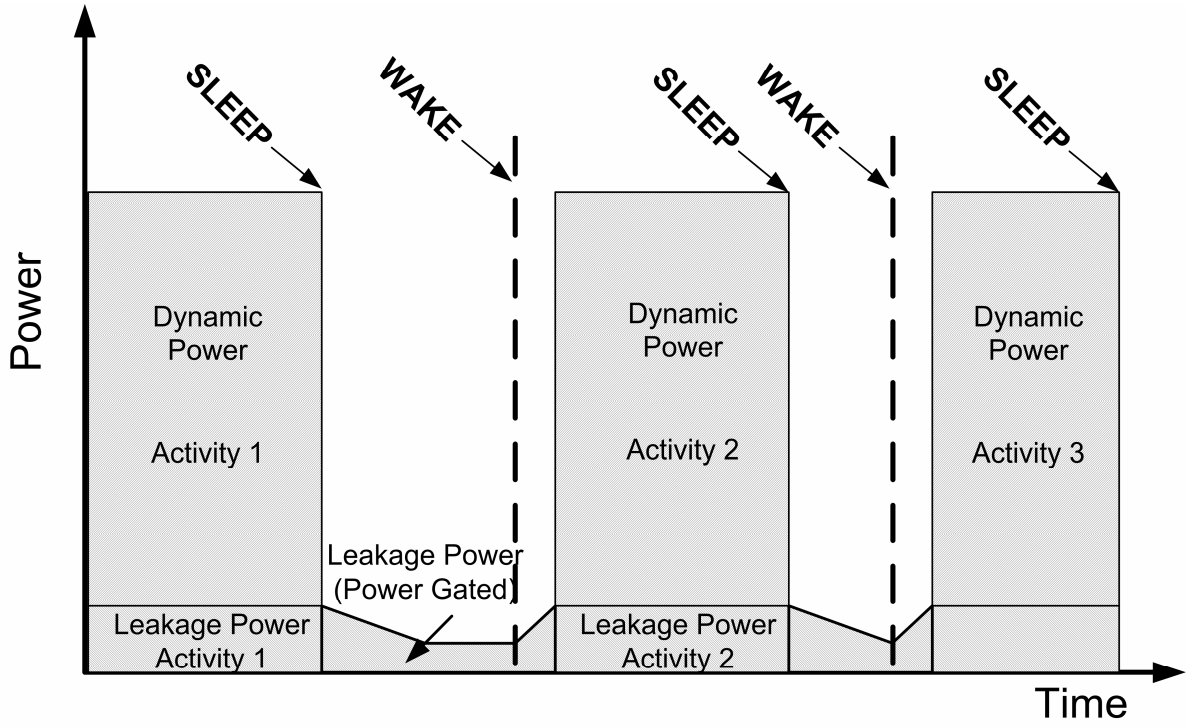


Figure 2.13: Activity Profile with Power Gating [7]

2.4. Floorplanning

Floorplanning is one of the essential design steps of physical design. The classical floorplan objective seeks to pack all IP blocks, such that the enclosing layout region has minimum area (with no overlapping blocks) while satisfying the aspect ratio constraint of each block and timing requirements of the chip.

Formally, the floorpanning problem can be stated as [115]: Let $B = \{b_1, b_2, \ldots, b_n\}$ be a set of n rectangular circuit modules, whose respective area, maximum and minimum aspect ratio are A_i, r_i and s_i. The connectivity matrix C_{nxn} specifies the connection between each module. With the goal of minimizing the total chip area and wire-length, the floorplanning tool determines the relative position of the n non-overlapping rectangular circuit modules in a SoC. Other costs to improve routability, reduce IR drop and even out thermal variations across the chip have been included in some of the modern floorplanners.

The topological representation of the floorplan is critical to the effectiveness and optimization of the floorplanning algorithm. VLSI floorplans can be classified into two categories [116]: (1) *slicing floorplans* (Figure 2.14(a)) and (2) *non-slicing floorplans* (Figure 2.15(a)). The following subsections describe each of these floorplans in relative details.

2.4.1. SLICING FLOORPLANS

A floorplan that can be obtained by recursively partitioning a rectangle into two parts either by a vertical line or a horizontal line is called slicing floorplan. Such a floorplan can be represented in the form of a binary tree, called *slicing tree* (Figure 2.14(b)) [117]. Each internal node of a slicing tree is labeled with either a *V* or *H*, denoting horizontal and vertical cut, and each leaf represents a circuit module in the floorplan. Given a slicing tree, the

corresponding Polish expression (Figure 2.14(c)) can be obtained by performing post-order traversal of the tree [118]. It must be noted that a slicing floorplan can have more than one slicing tree depending on the order of cutline selection.

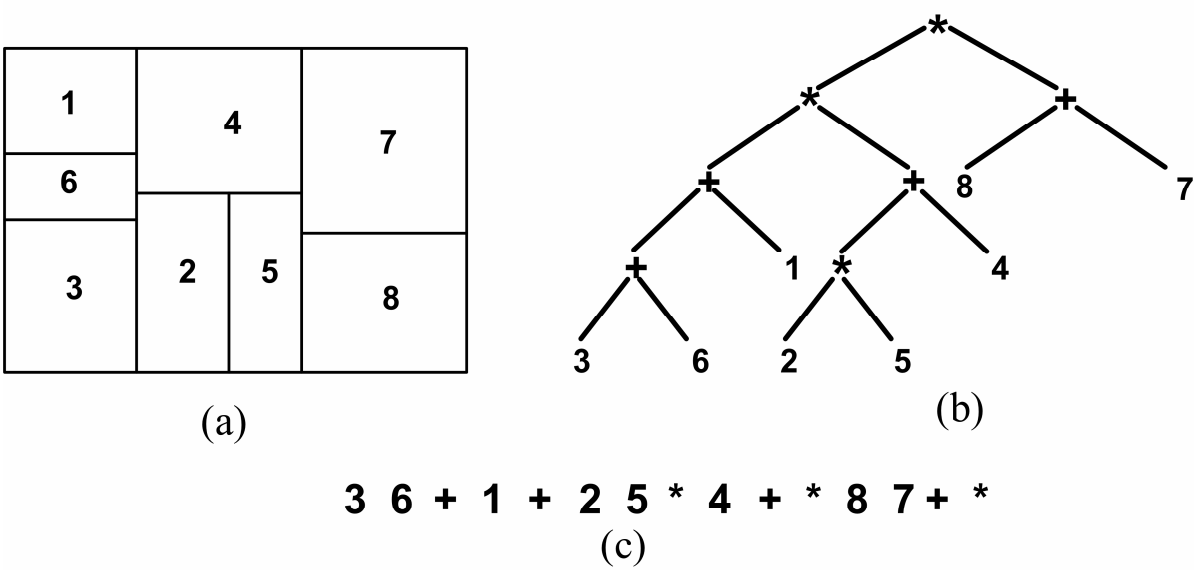


Figure 2.14: (a) Slicing Floorplan (b) Slicing Tree and (c) Normalized Postfix expression

2.4.2. NON-SLICING FLOORPLANS

A non-slicing floorplan cannot be represented by any valid slicing tree. Instead, a *horizontal constraint graph (HCG)* and a *vertical constraint graph (VCG)* are used to model a non-slicing floorplan [119]. The HCG and VCG define the horizontal and vertical relationships between the blocks, respectively. The HCG and VCG of Figure 2.15(a) are shown in Figure 2.15(b) and Figure 2.15(c).

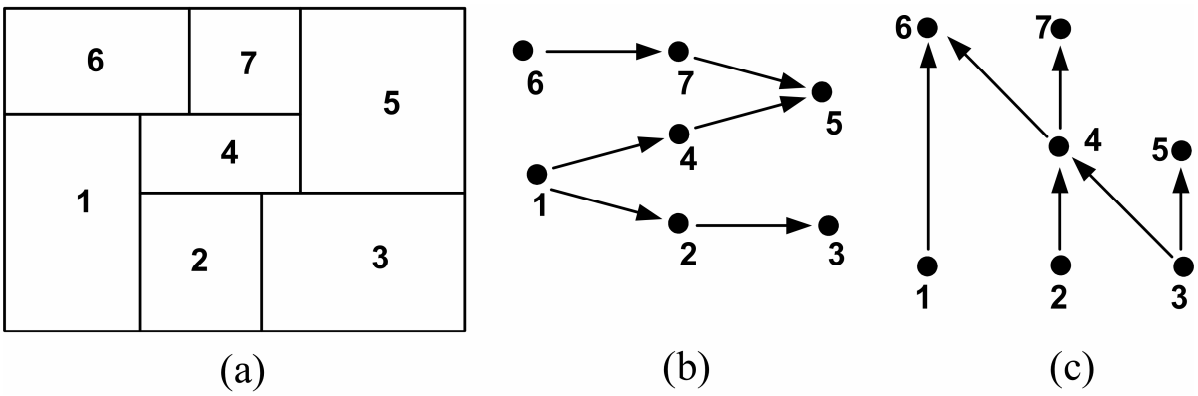


Figure 2.15: (a) Non-Slicing Floorplan (b) HCG and (c) VCG [119]

Some of the other classical floorplan representations are Corner Block List [120], Q-Sequence [121] and Twin Binary Tree [122]. With technology advancement, designs are getting much more complex. Floorplan representations that can handle larger circuit sizes are O-tree [123], B*-Tree [124], Corner Sequence (CS) [125], Sequence Pair (SP) [126], Bounded-Sliceline Grid (BSG) [127], Transitive Closure Graph (TCG) [128], Transitive Closure Graph with a Sequence (TCG-S) [129] and Adjacent Constraint Graph (ACG) [130].

2.4.3. FLOORPLAN ALGORITHMS

The floorplanning problem is known to be NP complete and therefore a number of different heuristic-based methods have been developed. Some of the popular approaches to floorplanning are briefly mentioned in the following subsections. Details of the methods may be found in the associated references.

2.4.3.1. Simulated Annealing

Simulated Annealing (SA) is one of the most popular and widely-used approaches for floorplan optimization [131]. Starting from an initial solution, a cooling scheduled-controlled walk through the search space is performed to find solutions with a lower cost, but

occasionally accepting a result with a higher cost, until no sizeable improvement is obtained in the cost function or the number of simulations exceeds a pre-specified value. The final floorplan solution depends upon (1) the choice of initial solution (2) the choice of cooling schedule (3) the perturbation function and (4) termination criteria. Area and wire length are the two most commonly used costs in floorplanners based on SA.

2.4.3.2. Integer Linear Programming

In this method, the floorplanning problem is modeled as a set of linear equations using 0/1 integer variables [132]. As the area objective is nonlinear, this method attempts to minimize the height for a given bound of floorplan width. In order to ensure that two blocks do not overlap, an overlap constraint is included. This method is exponential in time complexity and thus is not attractive for chips with large number IP blocks. To cope with the problem size, methods such as progressive [133] and divide-and-conquer [134] approaches have been proposed.

2.4.3.3. Genetic Algorithms

A Genetic (evolution-based) Algorithm for floorplanning has been developed in [135] and is also a very popular approach in this application. Using suitable techniques for solution encoding and setting the evolution function, crossover mechanism and mutation function, the floorplanner attempts to create a good compact floorplan in an evolutionary manner. A unified method to handle multiple constraints (such has pre-place constraint, range constraint and alignment constraint) has been proposed in [24]. The thermal impact of the floorplan incurred due to extremely high compaction have been investigated in [136].

2.4.3.4. Knowledge-based Floorplanning

SoCs with a large number of IP blocks often have multiple objectives to be optimized. In such a scenario, designers often use artificial intelligence, more recently referred to as knowledge-based methods, for floorplanning. An early rule-based expert system for floorplanning was proposed in [137]. Placing blocks in a rectangular grid graph with a set of rules that take shape and connectivity into account have been developed in [138]. PIAF [139] adopts a combination of algorithmic and knowledge-based approach for a top-down floorplanning system.

2.4.4. MODERN FLOORPLAN CONSTRAINTS

As pointed in [140], classical floorplanning falls short of addressing the new circuit properties and requirements. Some of the new emerging trends in floorplanning are discussed in the next subsections.

2.4.4.1. Fixed-outline constraint

Classical floorplanning techniques assume variable-die model wherein the boundary of the chip is not fixed. In a fixed-outline version, the dimensions of the bounding chip rectangle are established before floorplanning. Thus, the aspect ratio of the chip boundary is an input constraint rather than an optimization criterion. If the bounding rectangle is tight then the floorplanning problem can become much harder to solve. To handle the fixed-outline constraint, an aspect ratio penalty is included in the SA-based floorplan cost function [141]. Additionally, in order to better guide a local search, new perturbations are added in [142][143].

2.4.4.2. Hierarchical Floorplanning

One of the principle drawbacks of SA-based floorplanning technique is the lack of scalability of SA-based technique with the number of blocks in the chip. A fast hierarchical floorplanner based on recursive bipartitioning is developed in [144]. In order to address the lack of global information among the different sub-regions in a hierarchical floorplanner, multilevel floorplanning [145] performs floorplanning in two stages: bottom-up coarsening and top down un-coarsening.

2.4.4.3. Power Supply Aware Floorplanning

As supply voltage scales with every technology generation, IR drop in power/ground networks become an important concern. Placing power hungry cores far away from the power pads exacerbates the problem. Thus, considering power and clock network at the floorplanning stage is advantageous [146]. A methodology to simultaneously synthesize the power network and carry out floorplanning has been proposed in [147]. Chips with more than one supply voltage have additional placement constraints that must be included in the floorplanner.

Two different floorplanners have been used in this thesis. First, a SA-based thermal aware floorplanner [21] is used for voltage island design. This floorplanner uses area, wirelength and thermal factors while calculating the suitability of a floorplan solution. Thermal floorplanning is performed in two stages: voltage island level floorplanning followed by chip level floorplanning. As opposed to traditional floorplanning, thermal floorplanning incurs longer run-time due to overhead of the temperature estimation tool. But, this approach achieves a better thermal reduction of the final floorplan. The second floorplanner [24] allows designers to plan the position of a set of IP blocks in the SoC. Simulated Annealing using a sequence pair representation is used by the floorplanner. The placement constraints such as pre-place

constraints, range constraints, boundary constraints etc. can be specified as a constraint graph. Though originally intended for SoC with single V_{DD}, this floorplanner was used in a two-step design flow to be able to floorplan voltage island based SoC.

2.5. Summary

This chapter first described how power is consumed in conventional CMOS design. Next, an overview of the different low power design techniques currently used for power reduction was presented. These techniques are implemented at different stages of the design process. The use of multiple V_{DD} and multiple V_T are attractive solutions for power optimization. Scaling the supply voltage is the most effective way of reducing dynamic power. In throughput constrained systems, lowering of V_T might be necessary to maintain the required performance. As leakage power increases exponentially with V_T reduction, this leads to a potential conflict. In the remainder of this thesis, a holistic methodology is proposed for optimization of both dynamic and leakage power in a SoC design framework based on voltage islands.

Though all the power reduction techniques described in this chapter are promising, not all of them can be implemented in this design framework. For example, gate-level transformations to reduce dynamic power would require redesigning the IP blocks themselves, which is beyond the scope of this work. Adaptive body biasing to reduce leakage would require special circuitry to control the body bias that is also outside the scope of this work. Instead, this work proposes power reduction based on different techniques for power reduction using multiple V_{DD} and V_T. Though static allocation of V_{DD} and V_T is explored here, dynamic control may be added for further power reduction. Based on the work described here, this would further improve the power reduction and presents a potentially fruitful area for future work.

This chapter also provided an overview of the floorplanning technique of SoC design. The floorplanner identifies the relative position of the IP blocks in the SoC with area and wirelength minimization as the principle goals. Simulated Annealing is the most popular optimization algorithm used by the different floorplanners. A thermal-aware floorplanner and a floorplanner supporting placement constraints have been used in the proposed design flow.

CHAPTER 3 : SUPPLY AND THRESHOLD VOLTAGE SELECTION

3.1. Overview

This chapter proposes a methodology for the selection of the supply (V_{DD}) and threshold (V_T) voltages. Choosing suitable values for V_{DD} and V_T is crucial; these parameters have significant impact on the power dissipation of the chip. In general, the value of V_T primarily affects static power, while the value of V_{DD} has the largest impact on dynamic power. However, values for these parameters cannot be chosen independently, since the optimum value of each parameter depends on the value of the other. In this work, V_T is chosen for each IP block assuming that power gating techniques are employed, and voltage island technique is employed to support different values of V_{DD} in different parts of the chip.

This work provides a comprehensive power reduction methodology starting from the application to be executed. The key is to use information about the application that is embedded in the Power State Machine (PSM). The PSM represents the different states in which the SoC can exist at any given point of time. In particular, the PSM is used to calculate the relative activity of each core of the SoC. Based on the target delay of each IP block, its activity is used to determine the suitable threshold voltages to be assigned to the blocks. The V_T selection implicitly determines the range of V_{DD} for every core. A few preferred supply voltages are identified for each IP in their associated ranges. Among these preferred supply voltages of all the blocks, a finite number of supply voltages are chosen to be implemented on the chip using heuristic techniques. Parts of this work have been published in [148].

3.2. Preliminaries

This section describes the representation of an application running on an integrated circuit. Though simple, this representation provides sufficient detail to expose the power behavior of the system.

3.2.1. TASK GRAPH AND TASK MAPPING

The application to be implemented on an integrated circuit is often represented as a *task graph*. A task graph [149] is a Directed Acyclic Graph (DAG), in which each vertex represents a task that is to be performed by the application, and each edge represents either a control or data dependency. Figure 3.1 shows an example task graph which consists of eight tasks, along with start and end vertices. Data dependencies are represented as solid lines, and control dependencies are represented as dotted lines. In this example, there is only one control dependency; upon completion of the application, the system loops back to the start state.

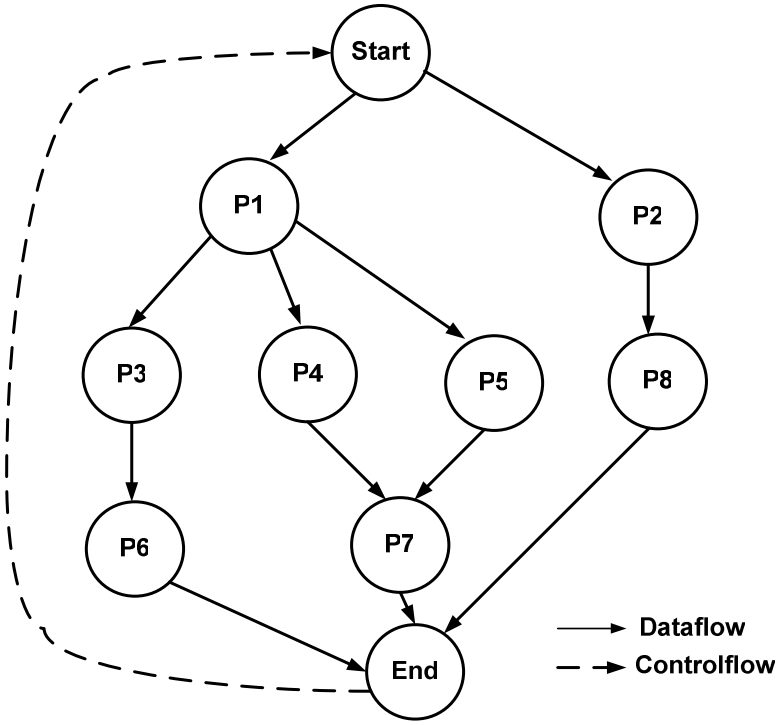


Figure 3.1: Task Graph

This task graph is implemented on a SoC using a set of IP blocks (often called *cores*). In general, each core can implement one or more of the tasks in the task graph. Mapping of tasks to cores is done either to maximize throughput (throughput-driven) for a power constrained system or minimize power dissipation (deadline-driven) for a delay constrained system. This work optimizes power for deadline-driven systems. Given each task to be completed within a certain deadline, the goal is to minimize power of the overall system. Figure 3.2 shows an example SoC that could be constructed to implement the task graph in Figure 3.1. In this implementation, each task is assigned to a different core, and the cores are connected using a generic interconnect. The power management unit (to be discussed later in this chapter) connects to all cores, and controls the power modes of the system.

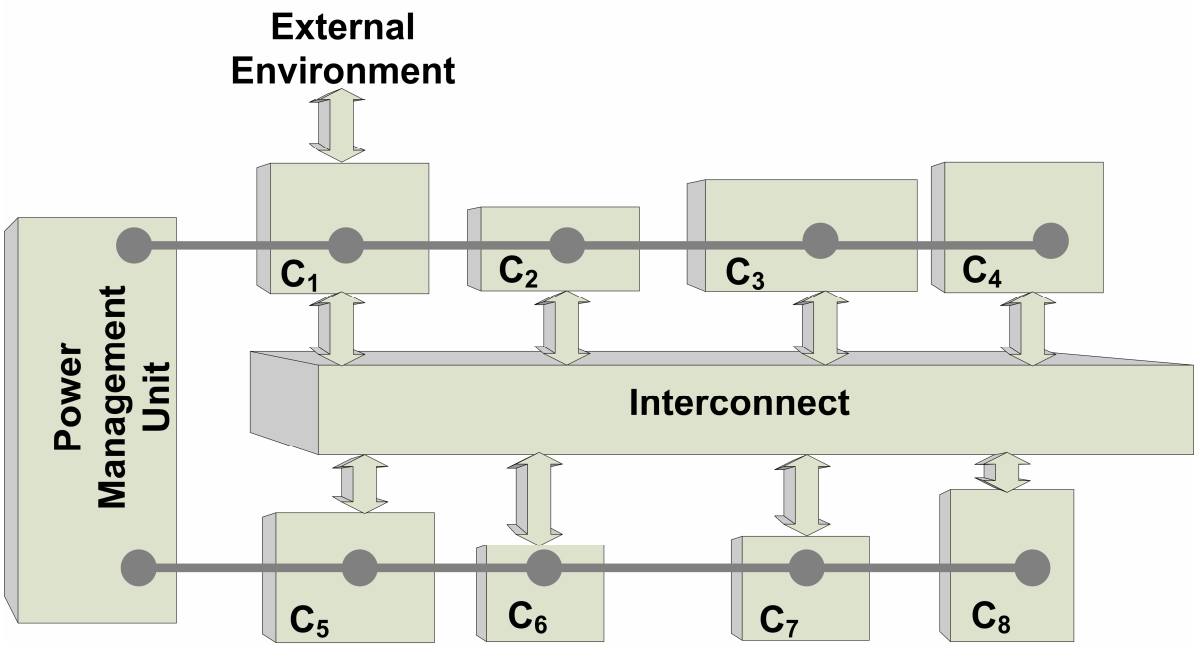


Figure 3.2: Generic System-level Design

3.2.2. CORE-LEVEL POWER MODELING

The power models that are used differ in the level of detail [150]. At the most abstract level, the power of each IP block is considered to be a constant value. Obviously, this ignores the

dynamic behavior of IP blocks during execution of the application but still provides useful information for high-level modeling. Based on the set of tasks mapped to a core, the power behavior of the core is then modeled. In this chapter it is assumed that each core can be in one of the three possible states: *active*, *idle* and *sleep*. In the *active* mode of operation the core is executing a task. The core is inactive in *idle* and *sleep* state. In the *sleep* mode the supply voltage is cut-off from the core to reduce leakage power while in *idle* mode the core remains connected to the power supply. Due to timing and power overhead, cores enter the *sleep* mode only when it is expected to be inactive for a significant amount of time. Based on the tasks performed, each IP block operates through different states as it interacts with the external environment and the other blocks. Transition from one power state to the other is controlled by the external inputs and the power manager. It would be possible to include other models as well (such as "drowsy" mode [151]).

3.2.3. SYSTEM-LEVEL POWER MODELING

As described above, at any given time, each core in the system is in one of the three modes. The set of modes of all cores at a given time is termed as *power state* of the system. Figure 3.3 shows an example of a system with three cores; in this example, C_1 and C_2 are in their idle mode, while C_3 is in its active mode.

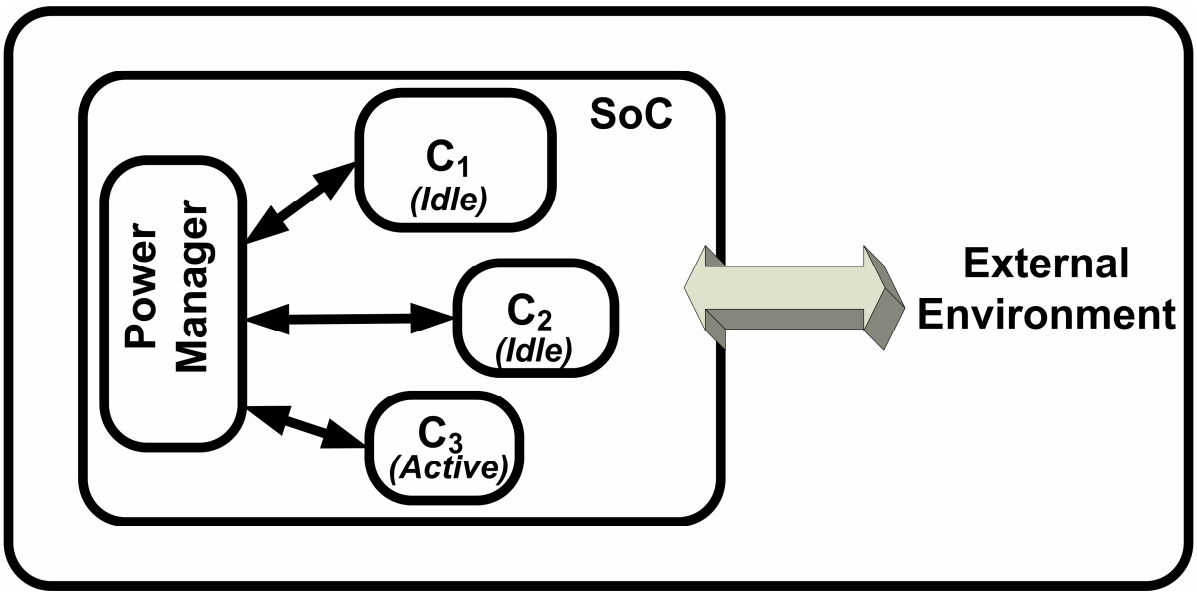


Figure 3.3: Different power modes of cores in a SoC

The dynamic behavior of the systems's power modes can be captured in a *Power State Machine* (PSM) similar to [152]. Such a model is similar to the representations proposed for emerging Common Power Format (CPF) [153] and Unified Power Format (UPF) [154] standards.

Figure 3.4 shows an example PSM. Each vertex in the PSM represents a particular power state (which, as described above, is characterized by the mode of each core in the system). In this example, in *State 1*, C_1 and C_2 are both idle, and C_3 is active, while in *State 4*, C_1 is active, and C_2 and C_3 are idle (the power states for other vertices are not shown in the diagram for clarity). Edges represent transitions between the states. Additionally, *Init* is the initial state of the system from which the chip can change to S_1, S_2 or S_3.

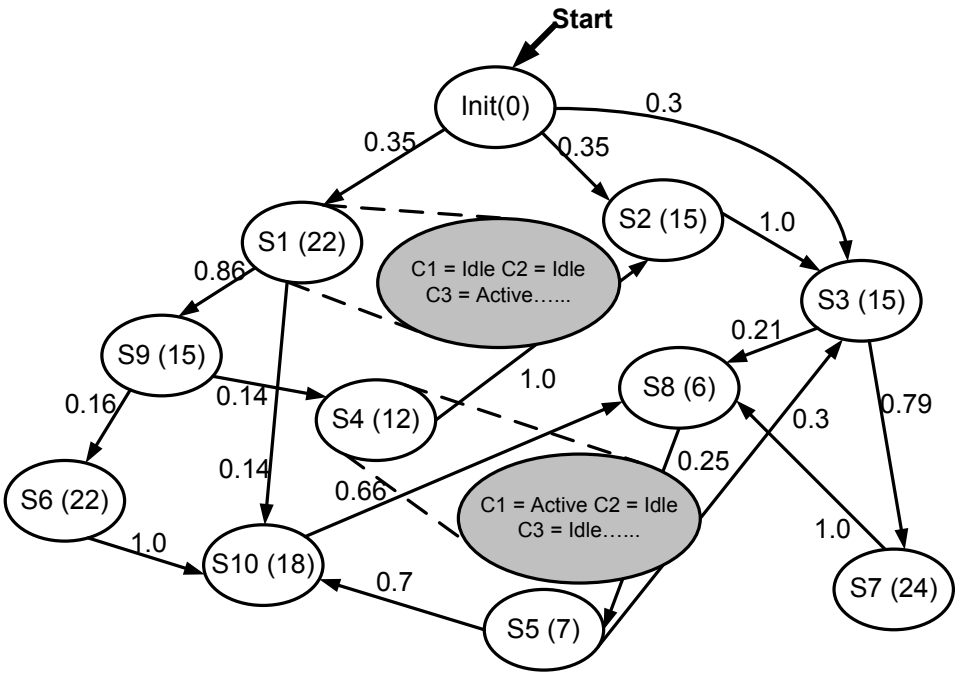


Figure 3.4: Power State Machine

More precisely, the PSM is represented as a directed graph of the form G = (∏, E, Φ, Γ), where ∏ is the set of all states, E is the set of edges, Φ is the set of edge weights and Γ is the set of state weights. Each state $S_i \in \prod$ describes the state of the cores of a SoC at any given time instant and each edge $a_{i,j} \in E$ represents a valid transition from state S_i to S_j. It also assumed that such transitions are controlled by inputs from both the power manager as well as external inputs [155]. Each state in the PSM describes the operating mode of the cores in the SoC. Weight $w(a_{i,j}) \in \Phi$ is the transition probability from state S_i to next states S_j ($S_i, S_j \in \prod$ and $i \neq j$) such that

$$\sum_j w(a_{i,j}) = 1 \tag{3.1}$$

with $w(a_{i,j})=0$ when no edge exists between S_i and S_j.

To represent the relative time period the SoC remains in a particular state, weights $W(S_i) \in \Gamma$ are also assigned to the individual states. The weight $W(S_i)$ represents the average time from

entering state S_i until leaving it. Thus, the PSM provides an application-driven abstraction of the different power modes in which the cores might operate along with their respective probabilities.

Using the PSM of an application and an estimate of the power dissipation by each core in each power mode, it is possible to estimate the overall average power dissipation of the SoC. A simulation-based approach is considered for power estimation. The weight $W(S_i)$ was assigned to each individual state at the beginning of the simulation. For a given PSM, each simulation starts at the *Init* state and the transition from one state to the other is governed by the probability of each edge. Once a sufficiently large simulated data is obtained and all states have been reached at least once, the overall activity factor (α_i) and the overall idle factor (β_i) of each core C_i is calculated as follows:

$$\alpha_i = \frac{\sum_{j=1}^{q} W(simstate(j)) \times m_{i,j}}{\sum_{r=1}^{q} W(simstate(r))} \quad \text{and} \quad \beta_i = \frac{\sum_{j=1}^{q} W(simstate(j)) \times n_{i,j}}{\sum_{r=1}^{q} W(simstate(r))}$$

$$\text{where} \qquad m_{i,k} = \begin{cases} 1, & if\ C_i\ is\ Active\ in\ state\ S_k \\ 0, & otherwise \end{cases} \tag{3.2}$$

$$\text{and} \qquad n_{i,k} = \begin{cases} 1, & if\ C_i\ is\ Idle\ in\ state\ S_k \\ 0, & if\ C_i\ is\ Sleep\ or\ Active\ state\ S_k \end{cases}$$

where q is the total number of states visited in the simulation and *simstate* is list of the visited states [2]. Intuitively α_i represents the proportion of time that the cores are in their active mode (averaged over all states), and β_i represents the proportion of time that the cores are in idle

[2] It must be noted that PSM represents a Markov chain and this analysis can also be done using statistical methods [197].

mode, again averaged over all states. The proportion of time that the cores are in sleep mode is then (1 - α_i - β_i).

If the power dissipation of each core can be estimated in each power mode, then the overall power dissipation by each core C_i can be represented as:

$$P_i = \alpha_i P_{A,i} + \beta_i P_{I,i} + (1 - \alpha_i - \beta_i) P_{S,i} \tag{3.3}$$

where $P_{A,i}$, $P_{I,i}$ and $P_{S,i}$ are the average power consumptions of stand-alone in active, idle and sleep modes, respectively. It must be noted that it is not necessary for each IP block to have all the three modes of operation. Moreover, if there are other power modes available (such as the drowsy mode described in [151]), the extension of this information is clear.

3.3. Power Optimization Framework

With the background provided, SoC power optimization through the use of multiple V_{DD}'s and V_T's, based on the PSM, can now be described. Based on timing constraints, every core C_i in the SoC has a minimum supply voltage at which it can operate, V_{DDL}^i. The maximum on-chip supply voltage, V_{DDH}^*, also termed as the *chip-level supply voltage*, is the minimum voltage at which the timing constraints of all the cores are satisfied. The maximum possible value of V_{DDH}^* is technology specific and is the maximum possible V_{DD} for the technology generation. Thus, any core C_i can operate with a supply voltage between V_{DDL}^i and V_{DDH}^*. The first objective is to find V_{DDL}^i for each core in order to establish the range of possible supply voltages. Here, similar to [21] and [1], it is assumed the inputs and the outputs of the cores are registered so that no critical path will span across two different cores. Therefore, for a given core C_i, the V_{DDL}^i must satisfy the timing constraints of that particular core. As delay is

related to V_{DD} and V_T, the V_{DDL}^i is based on the V_T^i for a given IP block. Thus, to identify the V_{DDL}^i for each core C_i, the V_T^i for the block must be determined first.

3.3.1. BLOCK-LEVEL MULTI-V_T DESIGN

With technology scaling below 90nm, the use of multiple-V_T libraries is standard for reducing leakage power. Many libraries today offer three versions of their cells: Low-V_T (LVT), Standard-V_T (SVT) and High-V_T (HVT) [7]. Each one has a different delay-leakage tradeoff, as follows. HVT cells typically have higher delay but lower leakage power while the converse is true for LVT cells. For SVT cells, the delay and power numbers lie between the HVT and LVT cells. A dual-V_T design technique selectively places low-V_T devices only on the speed critical paths of a circuit and high-V_T devices on non-critical paths to reduce power [156].

A practical limitation of such an approach is that there can be many critical paths in a circuit which reduces the effectiveness of the approach [157]. Process variations can also convert non-critical paths to critical paths, and vice-versa. Moreover if the HVT and LVT cells are not sufficiently far apart, some HVT transistors may be faster than some LVT transistors in the presence of such variations [158]. Consequently, this leads to larger leakage. As device scaling continues, the conventional dual-V_T approach also suffers yield loss due to process variations and would tend to overestimate leakage savings [159]. To avoid such design concerns, this work assumes that all transistors within each IP block have the same V_T (but transistors in different IP blocks have different values of V_T). This would make the design flow less susceptible to such problems arising from process variations. Specifically, designers do not have to perform additional simulations, beyond the usual set, to ensure the design constraints are not violated in presence of process variations. In order to enable further leakage savings, a

power gating technique is assumed in conjunction with the multi-V_T approach. In this thesis, the cores with LVT or SVT are power-gated using HVT sleep transistors.

3.4. V_T selection

The lowest supply voltage for core C_i (i.e., V_{DDL}^i) is governed by the choice of V_T^i . To understand how V_{DDL}^i changes for different V_T libraries, the variable activity circuit [160] of Figure 3.5 is considered. It has a NAND-based ring oscillator at the top level and NAND chains in subsequent levels. Its internal nodes and select inputs are used to control the amount of activity in the overall circuit. Initially, the circuit is simplified to consider only a single level. By setting *Select 0 = 1* and all other Select lines to *0*, a ring-oscillator circuit is obtained.

Figure 3.6 shows HSPICE simulation results in 90nm technology of delay vs. supply voltage for the ring oscillator using the HVT, LVT and SVT libraries, respectively. If the target critical path delay is 1ns, then V_{DDL}^i for LVT, SVT and HVT are 0.67V, 0.82V and 0.98V, respectively. This is highlighted with the arrows in Figure 3.6. Thus, the choice of V_T^i influences the value of V_{DDL}^i.

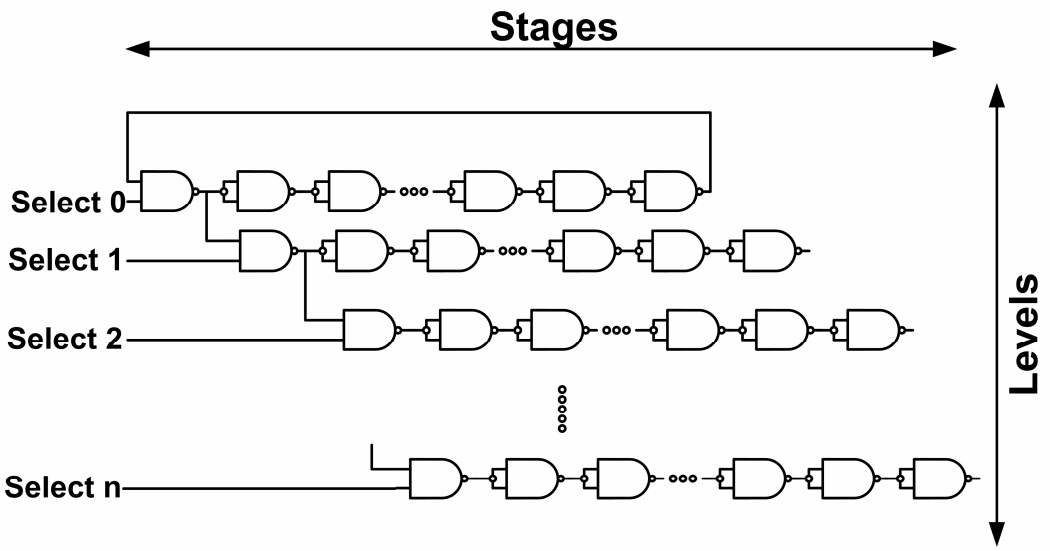


Figure 3.5: Variable Activity Circuit

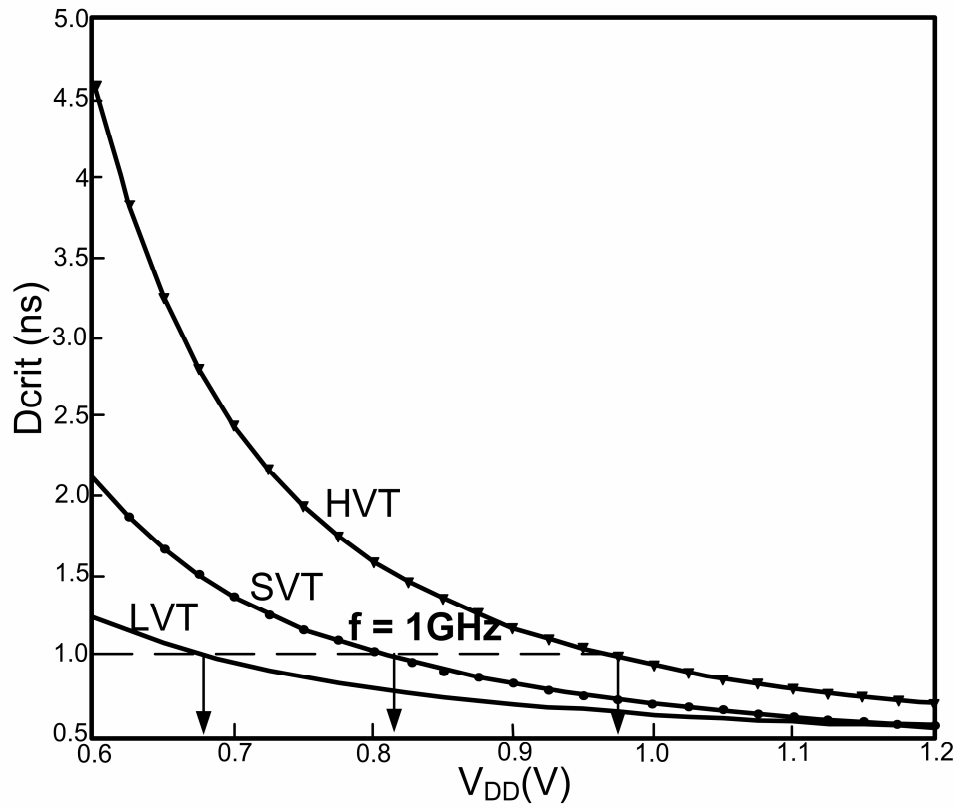


Figure 3.6: Delay vs. V_{DD}

IP blocks with different activity ratios are modeled by appropriately choosing the Select input for subsequent levels of the circuit in Figure 3.5. The power consumption levels of the variable activity circuit are shown in Figure 3.7 (for 1% activity) and Figure 3.8 (for 100% activity). The minimum possible V_{DD} to meet critical path delay of 1ns is denoted by the intersection of the dashed line with the power curves. For 1% activity factor HVT cells consume 70% less power than LVT cells. On the other hand, for 100% activity factor and 1ns critical-path delay (Figure 3.8), LVT is a better choice than HVT as there is a 58% power reduction.

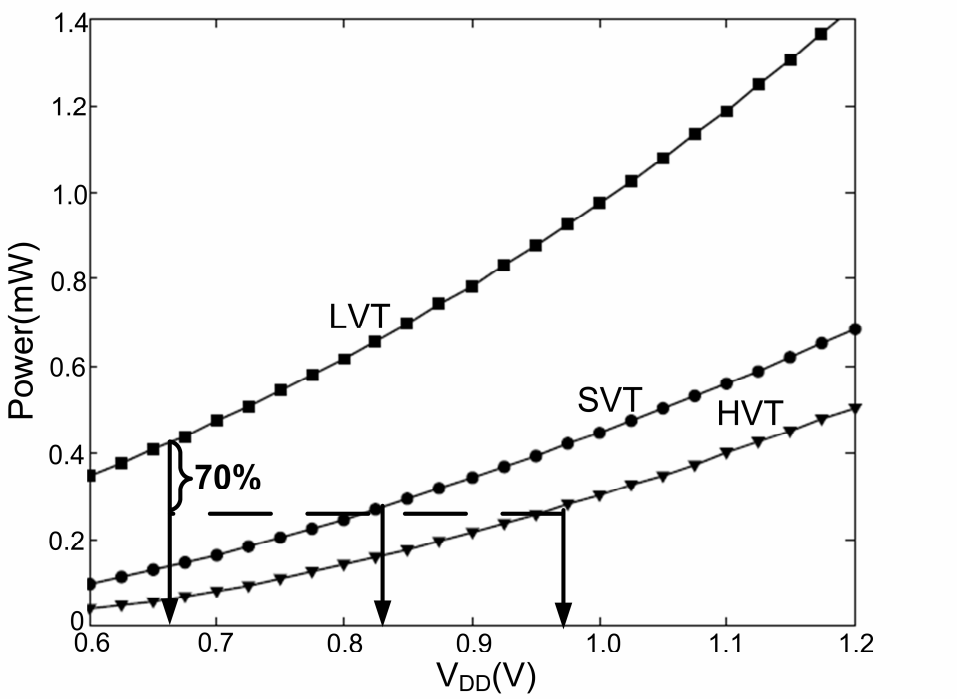


Figure 3.7: Power consumption for 1% activity

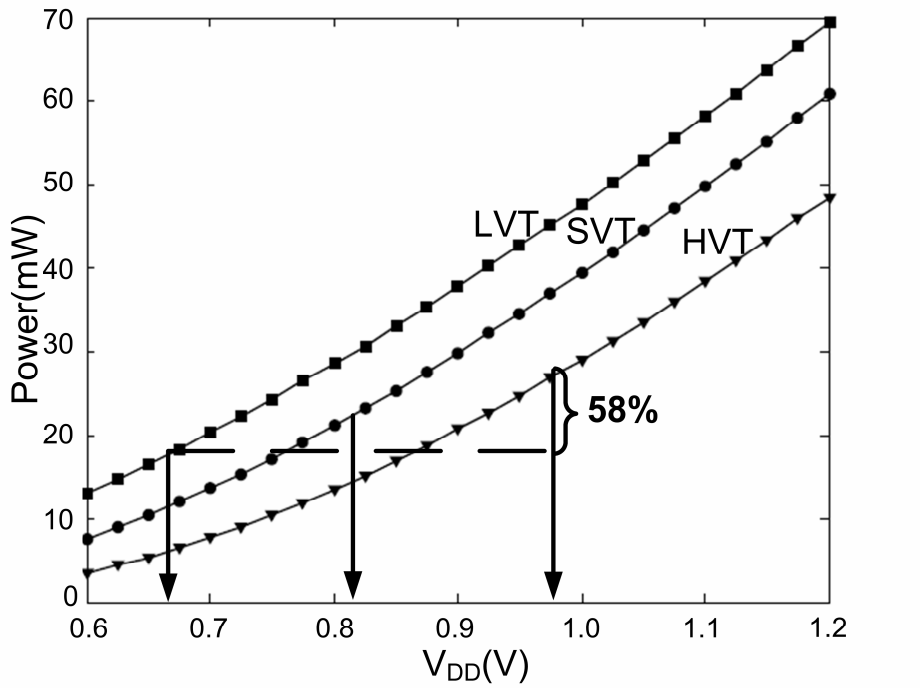


Figure 3.8: Power consumption for 100% activity

This provides the needed insight for V_T selection: based on delay and activity of each core the appropriate V_T for each block can be chosen. Moreover, the range of supply voltages over which the core is permitted to operate is identified. In typical high-performance IC's, there are

a few blocks with high activity while others have relatively low activity. Cores having low activity (hence, higher leakage power) should be implemented using HVT with no power gating. Cores having high activity (and thus higher dynamic power) should be designed using SVT or LVT and lower supply voltages, with HVT power-gating transistors. However, if the required supply voltage for HVT implementation is greater than V_{DDH}^*, the maximum allowable supply voltage, SVT cells should be used instead of HVT cells. For example, in Figure 3.8, if the V_{DDH}^* is 0.9V, the timing cannot be met using HVT and so SVT should be chosen instead of HVT. Moreover, if the LVT cells are chosen and V_{DDL}^i is less than ($V_{DDH}^*/2$), then LVT is replaced by SVT cells. This is because the gate overdrive of circuits below ($V_{DDH}^*/2$) is too low to be useful. Also, circuits with such low V_{DD} might encounter noise issues from adjacent IP blocks operating at V_{DDH}^*.

Thus, the range of supply voltages that can be assigned to each core will depend upon the type of V_T-cell used for implementation and its overall activity. In general, the cores with high activity must be realized using LVT cells while cores having low activity should use HVT cells. Based on delay specifications and the maximum and minimum allowable supply voltages, the designer may have to choose SVT instead of the HVT or LVT. At a high level, the power and delay variation due to changes in supply and threshold voltage can be derived for any arbitrary core using power and delay macromodels [161][162].

3.5. Challenges in Multiple Supply Voltage Design

Multiple supply voltages aim at reducing the overall chip power/energy. While a large number of supplies would provide optimal results, there are a few practical limitations to the number of supply voltages that can be realized on-chip. These problems include routing of multiple

supply voltages, area and delay overhead of level shifters, and lack of design tools and methodologies for implementation [163]. Due to the complexity of multiple voltage islands, the tasks of floorplanning and power planning can be quite challenging. When the different supply voltages are delivered from off-chip source, separate power pads are required for each supply voltage. This would limit the number of possible power supplies. Additional power rails are required when multiple supply voltages are delivered by on-chip voltage regulators. This places an extra burden on power grid designers as they have to consider IR drop and other power supply noise related issues of all the power grids. As libraries have to be characterized for different supply voltages, timing analysis in multi-V_{DD} is far more complex than single-V_{DD} design. Level shifters, isolation cells, enable level shifters, retention flip flops, always ON cells, etc., not only add to the area of the chip but also make routing of signals difficult. Moreover, clock tree synthesis must be aware of the different power domains. Therefore, it is important to choose a small number of voltages.

A set of practical expressions for the optimal number of supply voltages was derived in [164]. It was concluded that three supply voltages should be used. When additional supply voltages are used, the amount of achievable energy savings saturates. As the power supply is reduced and V_T/V_{DD} increases, the effect of power reduction diminishes as well. In the next section, a new heuristic method for choosing a small number of supply voltages is described.

3.6. V_{DD} Selection

Depending upon the V_T chosen for each IP block, the range of possible supply voltages is set. The next step in the design process is to choose a finite number of supply voltages from this range for each core and generate the voltage assignment table. As an example, a SoC with 8

cores (C_1, C_2, ..., C_8) is considered. It is assumed that the chip-level supply voltage is V_{DDH}^* = 1.2V. The threshold voltage for each core is determined using the technique described in the previous section. Next, V_{DDL}^i is identified using the target delay for each core, C_i.

Thus, any voltage assigned to core C_i between V_{DDH}^* and V_{DDL}^i is a valid assignment. From the designer's perspective, there are a few voltages in this range that are of some significance. These are the voltages where the optimal value of one of the various products of power and delay occur, such as power-delay product (PDP, or energy), energy-delay product (EDP) [165], or the power-energy product (PEP) [166]. Of the three metrics, EDP prioritizes delay over power while PEP optimizes power over delay, and PDP optimizes both equally [166]. The optimal value of each of these quantities would give rise to a different V_{DD}. Based on whichever metric is to be optimized, any of these metrics can be used to pick a V_{DD} within the voltage range. For the purposes of this work, the optimal PDP value was chosen for the optimization [166]. This useful intermediate operating point is termed as V_{DDM}^i. Figure 3.9 plots power, delay and energy curves for different V_{DD} values. V_{DDL}^i, V_{DDM}^i and V_{DDH}^* have been identified in the plot.

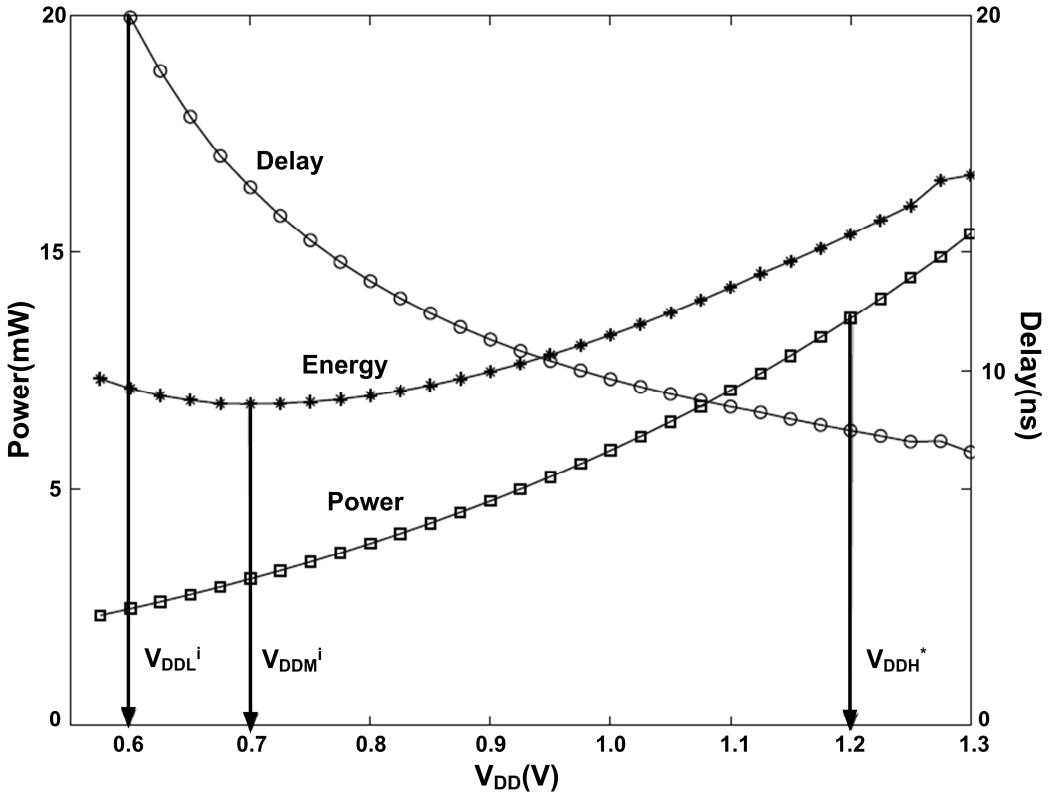


Figure 3.9: Power, Delay and EDP vs. V_{DD} for a core

These types of graphs can be generated for a core if delay, dynamic power and leakage power for a given (V_{DD}, V_T) pair is known. The accuracy has been shown to be within 10-20% of the actual values across process and temperature ranges [162].

The V_{DD} for the optimum energy point (V_{DDM}^{i}) can be located between V_{DDL}^{i} and V_{DDH}^{*}, depending on how the degree of activity of the core and the target critical path delay. Figure 3.10 plots the energy versus V_{DD} for the variable activity circuit with different activity as well as frequency. Here the energy is normalized with the minimum energy possible for each circuit. For each curve, V_{DDM}^{i} corresponds to the V_{DD} with minimum energy value. In general, with increased activity, V_{DDM}^{i} is closer to V_{DDL}^{i}. For example, when comparing any of the circuits for 10% activity and 1% activity, it is observed that V_{DDM}^{i} decreases with higher activity. On the other hand, a slower operating frequency and hence larger leakage, leads to a

larger V_{DDM}^{i}. In Figure 3.10, the V_{DDM}^{i} of a circuit shifts towards V_{DDH}^{*} as one moves from the variable activity circuit with 101-stage to 301-stage to 501-stage oscillators, each of which has a lower frequency than the previous one. The threshold voltages for the circuits in Figure 3.10 are all the same.

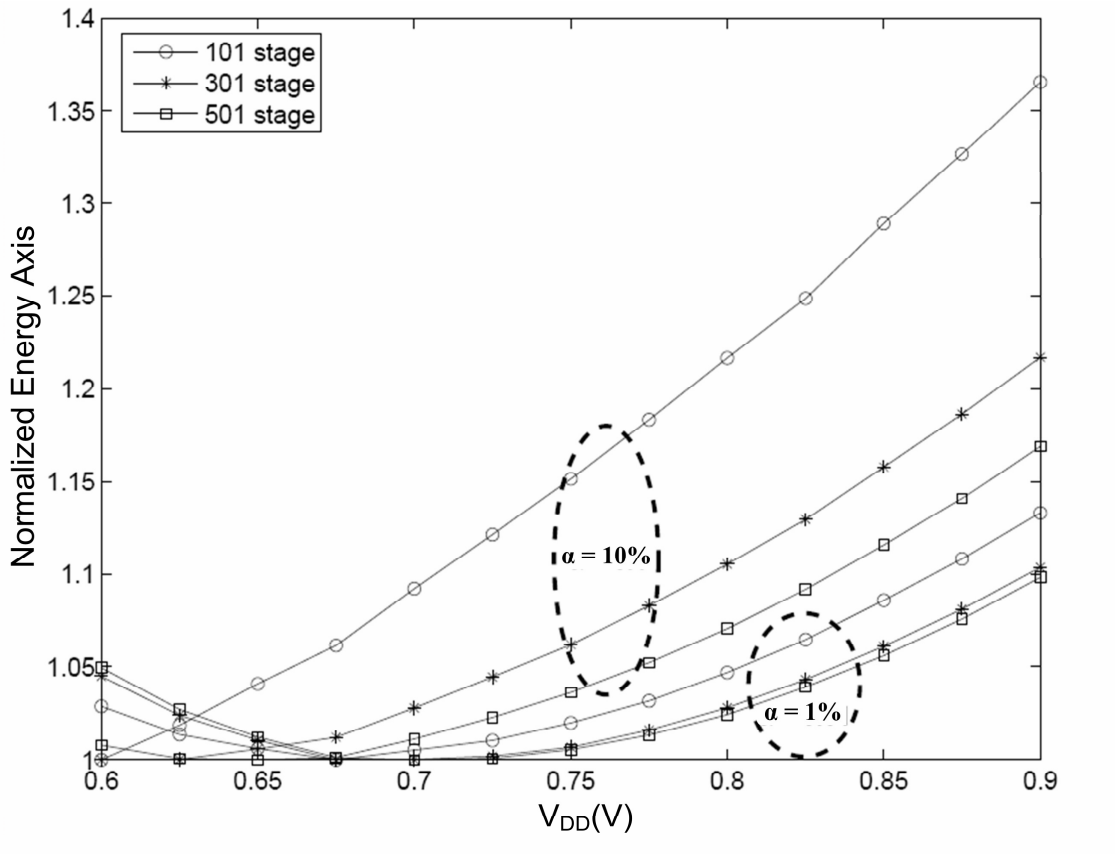


Figure 3.10: Energy vs V_{DD} for variable activity and delay

It is to be noted that if V_{DDM}^{i} is less than V_{DDL}^{i}, or V_{DDM}^{i} is greater than V_{DDH}^{*}, then only two voltages, V_{DDL}^{i} and V_{DDH}^{*}, are used in subsequent steps of method. Some cores may satisfy timing with only V_{DDH}^{*}, so only one voltage level is used.

The resulting values can be used to construct the *Optimal Voltages Table*. Table 3.1 shows this for the 8 cores of the example considered here. The voltage values are rounded to the nearest 0.1V since minor variations do not greatly affect the results. For every core, the energy at each selected supply is also stated in the table.

Table 3.1: Optimal Voltages Table

Cores	V_{DDL} (V)	E_{VDDL} (nJ)	V_{DDM} (V)	E_{VDDM} (nJ)	V_{DDH} (V)	E_{VDDH} (nJ)
C_1(HVT)	0.6	7.8	-	-	1.2	10.7
C_2(LVT)	0.7	4.7	0.8	4.4	1.2	8
C_3(LVT)	0.6	7.1	0.7	6.8	1.2	10.4
C_4(SVT)	0.6	4	0.8	3.4	1.2	5.3
C_5(HVT)	0.6	2.3	-	-	1.2	5.5
C_6(LVT)	0.8	2.8	-	-	1.2	4.6
C_7(LVT)	0.9	2.4	-	-	1.2	3
C_8(HVT)	-	-	-	-	1.2	6.5

In this table, a total of 5 different supply voltages are used implying that up to 5 islands could be created; the larger the number of islands, the more the power savings. However, the optimal power solution may create problems when attempting to generate the feasible floorplan. The floorplanner attempts to reduce wire length by placing highly-connected blocks close together, but the voltage islands prefer blocks with the same supply to be placed contiguously. Even if floorplanning is successful, the design complexity of delivering a large number of supply voltages to different parts of the chip may still be problematic, as mentioned before. Thus, fewer supply voltages are usually selected.

3.6.1. VOLTAGE ASSIGNMENT TABLE GENERATION

To generate the voltage assignment table, the following voltage selection problem is to be solved: *Given n possible supply voltages, choose m voltages (where $m \leq n$) that minimizes chip energy.*

In the example, n=5 and m=3 is considered. Of the possible set of supply voltages, one automatic choice is V_{DDH}^{*} = 1.2V since C_8 requires it. The next task is to choose $m\text{-}1 = 2$ other voltages ranging between 0.6V and 0.9V. Of all the possible supply voltages, one possibility is to pick the two lowest V_{DD}s. However, cores that cannot be assigned to these two supply

voltages are forced to operate at the highest voltage, thus reducing the achievable energy savings. Therefore, the solution requires consideration of many other possibilities to determine the other two voltages. Formally, the problem for choosing *m* supply voltages from *n* voltage choices of a SoC having *N* cores can be defined as:

$$\begin{aligned} &\text{minimize } \sum_{i=1}^{N}(E_i(x_p)) \\ &\text{subject to } \sum_{j=2}^{n} x_j = m-1 \\ &\text{where } x_j = \begin{cases} 1, & \text{if } j^{th} \text{ voltage is chosen} \\ 0, & \text{otherwise} \end{cases} \quad \forall\ 0 < j < n \\ &\text{and } p \text{ is the smallest value for which } x_p = 1 \text{ and } d_i(p) \le D_i \end{aligned} \tag{3.4}$$

where $E_i(x_p)$ is the energy consumption of the i^{th} core in the p^{th} island, $d_i(p)$ is the critical path delay of core C_i with v_p being the supply voltage, D_i is the delay constraint for core C_i and the first voltage (v_1) is the chip-level supply voltage, in the list of ordered supply voltages ($v_1 > v_2 > v_3 > \ldots > v_{n-1} > v_n$).

The voltage selection problem is an NP-hard problem and has been addressed in [167][168][169]. A α^2-approximation algorithm [167] and an optimal voltage selection algorithm [168] attempts to choose a small number of supply voltages from a large voltage set. In terms of time-complexity, the method proposed in [167] is more efficient than [168]. However, [167] is an approximation algorithm where the final solution can be α times worse than the optimal solution. The value of α is the ratio of maximum to minimum supply voltage. A removal-cost method for voltage selection for MPSoC designs under process, voltage and temperature variation is proposed in [169]. The energy function considered in [167][168] is a monotonic function as the dynamic component is only considered. Since the actual energy function (comprised of both dynamic and leakage power) is not always a monotonically

increasing function of the supply voltage in the $(V_{DDL}^{i}\text{-}V_{DDH}^{*})$ range, V_{DDL}^{i} is set to be equal to V_{DDM}^{i} for the cases where $(V_{DDL}^{i} < V_{DDM}^{i} < V_{DDH}^{*})$. For example, as in Figure 3.9, the V_{DDL}^{i} of the core is moved from 0.6V to 0.7V. By doing this, the problem is similar to that addressed [167][168]. The voltage selection problem is now solved using an improved heuristic method, outlined in Figure 3.11, using the following notation:

- V: array of supply voltages chosen for implementation.
- $\min_{j=1 \text{ to } m} \{E_i(V(j))\}$: Selects minimum energy of C_i for the m-permissible V_{DD}s in the V array.

The algorithm begins by choosing the m highest supply voltages and progressively replaces them with the n-m lower supply voltages. If one of the higher supply voltages is replaced by a lower one, then the higher V_{DD} is not considered in the rest of the algorithm (rendering this technique to be greedy). The process continues until the m voltages are selected. In the algorithm itself, L2-L4 is used for initialization. Next, the n-m remaining lower supply voltages (loop of variable "p") are considered from high to low. When each new voltage v_p is introduced, each island in array V ranging from 2 to m (loop of variable *"k"*) is considered. In order to compute the energy for new voltage choices, the array V is copied in a new array V' in line L10. Next, L11 replaces the k^{th} entry of V' with v_p. In L12, a complete reassignment of the cores to the supply voltages in V' is done and then the total energy is recomputed. If, by introducing v_p, the new energy value is less than the previous minimum energy value then the energy and the corresponding supply voltage is stored in lines L13-L15. To explore other solutions, each of the m-1 supply voltages in V is replaced by v_p. If v_p gives a better solution, the array V and the energy value $esum(m,p)$ is updated in lines L16-L18. The final solution

using the proposed heuristic technique for choosing *m* voltages from a set of *n* possible supply voltages is e_{min}* in line L19.

Algorithm for Voltage Selection:

Begin

L1: *Given energy information and possible supply voltages of all cores;*
L2: *N := number of cores /* set N to the total number of cores */*
L3: *V := [v_1,..., v_m] /*Initialize the V array with m-highest voltages */*
L4: $esum(m,m) := \sum_{i=1}^{N} \min_{j=1 \text{ to } m} \{E_i(V(j))\}$ */* Sum min. energy of each core */*
L5: **for** *p:= m+1 to n* **do** */* for rest of the possible lower voltages */*
L6: *esum(m,p) := esum(m,p-1) /*copy previous value*/*
L7: e_{min}* := esum(m,p) /* assume it is the min energy */*
L8: *k* := 0*
L9: **for** *k := 2 to m* **do** */* keep v_1, loop through rest of the cases */*
L10: *V' := V /* initialize V' array */*
L11: *V'(k) := v_p /*replace k^{th} entry with next possible voltage*/*
L12: $e_{\min} := \sum_{i=1}^{N} \min_{j=1 \text{ to } m} \{E_i(V'(j))\}$ */*Recompute sum of min. energy*/*
L13: **if** *(e_{min}* > e_{min})* */* If e_{min} is lower, then store it and */*
L14: *e_{min}* := e_{min} /* the supply voltage to be replaced with v_p */*
L15: *k* := k*
L16: **if** *(k* > 0)*
L17: *esum(m,p) := e_{min}* */* update with a lower energy value */*
L18: *V(k*) := v_p /* update the voltage choice array */*
L19: *e_{min}* := esum(m,n) /* final energy with m islands */*

end

Figure 3.11: Outline for Supply Voltage Selection Algorithm

To use this algorithm, the energy values of the cores at all the *n* supply voltages must be computed. At this stage, any core may be assigned to any of these voltages as long as the timing constraints are satisfied. So the *Voltage-Energy Table* is generated, as shown in Table 3.2 based on the earlier Table 3.1. However, due to the additional constraint on V_{DDL}^{i}, not all possible supply voltages are considered for all cores. For example, in case of C_3, the energy at V_{DD}= 0.6V is omitted as minimum energy is reached at V_{DDM}^{i} = 0.7V. So V_{DDL}^{i} is set to 0.7V. Similar effects are seen on C_2 and C_4.

Table 3.2: Voltage-Energy Table

Cores	Energy (nJ)				
	@0.6V	**@0.7V**	**@0.8V**	**@0.9V**	**@1.2V**
C_1	7.8	8	8.2	8.4	10.7
C_2	-	-	4.4	5.7	8
C_3	-	6.8	7.0	7.5	10.4
C_4	-	-	3.4	3.9	5.3
C_5	2.3	3.8	3.9	4.1	5.5
C_6	-	-	2.8	3.2	4.6
C_7	-	-	-	2.4	3
C_8	-	-	-	-	6.5

Figure 3.12 shows the step-by-step results of applying the proposed technique to the example in Table 3.2. The supply voltages for the voltage islands are included with each energy value. Results from the proposed approach for three different cases: m = 1, 2 and 3 are shown. For example, in the first square, 54 nJ is the total energy assuming only one island with V_{DD}=1.2V. The square with the lowest chip energy is 37.2 nJ and the resulting supply voltages are 0.6V, 0.8V and 1.2V. These supply voltages are the *permissible* voltages that can be implemented on the chip. As discussed before, the lowest two supply voltages were not the best choice to minimize power.

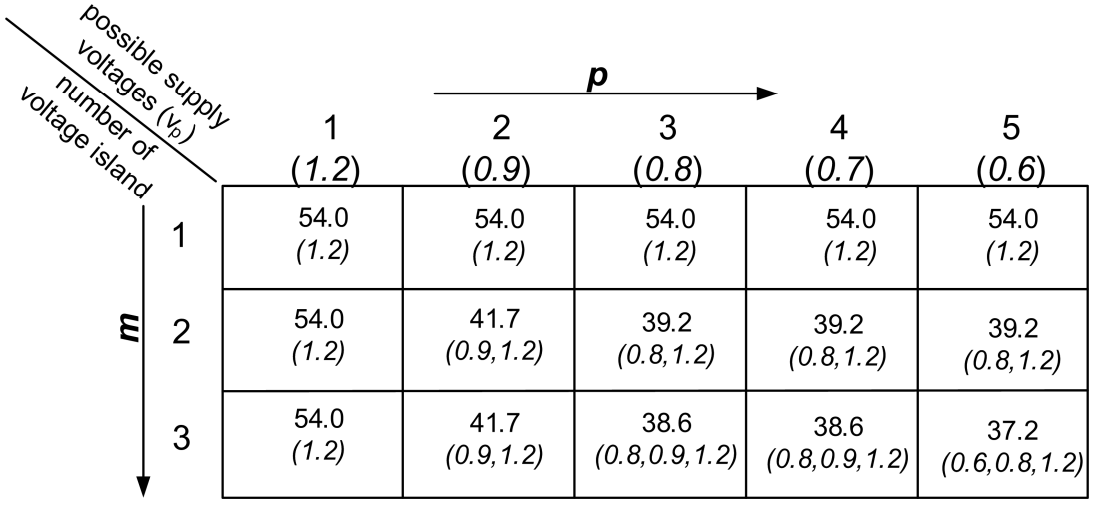


Figure 3.12: Energy-Based voltage choice

After running a large number of cases, it was found that any deviations from the optimal solution incurred by the use of a greedy algorithm, as well as setting V_{DDL}^{i} to V_{DDM}^{i} (for the non-monotonic cores to make them monotonic), are relatively small. On average, the energy found by the proposed solution is 1% greater than that by the optimal solution when 3 islands are chosen, and is indistinguishable from the globally optimal solution when the number of islands is 4 or more. Practical situations where this method gives a suboptimal solution are unlikely to occur. Because V_{DD} values below $V_{DDH}^{*}/2$ are not practical, certain solutions are ruled out. Further, the range between the minimum possible V_{DD} (= $V_{DDH}^{*}/2$) and V_{DDM}^{i} is small and thus there is lower probability that the optimal solution is significantly better. However, if the energy curves are not steep below V_{DDM}^{i}, then the constraint is limiting the ability to find the optimal solution. In the above example, the greedy heuristic technique actually produces the correct globally optimal solution.

Although any voltage between V_{DDH}^{*} and $V_{DDH}^{*}/2$ can be chosen as a feasible solution, in reality, the voltage levels are discretized. This limits the maximum number of possible supply voltages to be equal or less than 10. When complete enumeration is performed in order to choose the best possible voltage set, the total number of cases that have to be explored is 126 (n=10 and m=5). Formally, the complexity of voltage selection problem reduces to $O(NmT + N(^{n-1}C_{m-1}))$ where T is the time overhead to determine the energy consumed by a particular running at a certain V_{DD}. Thus, due to the restriction on the possible set of supply voltages, the voltage selection problem is reduced to a polynomial time problem as opposed to a NP-hard problem.

Using the permissible voltages, a *Voltage Assignment Table* can be constructed from Table 3.2 to produce Table 3.3. Each core can be assigned to any one the possible choice of voltages

during floorplanning. For example after floorplanning is done the V_{DD} of C_1 can be 0.6V or 0.8V or 1.2V. On the other hand C_7 and C_8 must have V_{DD} = 1.2V. The proposed method for V_{DD} assignment to each core during floorplanning is discussed in Chapter 5.

Table 3.3: Voltage Assignment Table

Cores	Supply Voltages (V)
C_1	0.6,0.8,1.2
C_2	0.8,1.2
C_3	0.8,1.2
C_4	0.8,1.2
C_5	0.6,0.8,1.2
C_6	0.8,1.2
C_7	1.2
C_8	1.2

It is to be noted that if a core's lowest supply voltage is not permissible, it must be replaced by a higher voltage that is on the permissible list. The highest voltage is always available in this case and included in the table.

3.7. Summary

In this chapter, the methodology for selection of *permissible* V_{DD} and V_T values to be implemented on the SoC has been developed and described in detail. The activity of each IP block is extracted from the application using a Power State Model. The technique of assigning V_T to each IP block using the activity information has been illustrated. In general, HVT cells are used for low activity cores while LVT cells are used for high activity cores. Based on the maximum and minimum supply voltage constraint, HVT or LVT cells are replaced by SVT cells. For each core, depending upon the delay constraint and the activity factor the suitable threshold voltages should be chosen.

Using the assigned V_T and the target delay, the range of V_{DD} that can be assigned to each core is identified. Next, a few meaningful supply voltages are identified in this range. The supply

voltage that satisfies the delay constraint of the IP block and the maximum supply voltage for the given technology are the two limits of this voltage range. This work uses the energy curve to identify the third supply voltage in the voltage range. Other design metrics can be used as well. In fact, a mix of EDP, PEP and PDP might produce a better result than using a single design metric. A full investigation of suitable choice of metric is considered as future work. If the preferred set of supply voltages of all IP blocks is less or equal to the number of supply voltages possible on the chip, then this set of voltages is considered in the next design phase. For the case where the preferred set is larger, a novel approach for voltage selection has been proposed. It is to be noted that the minimum difference in voltage among islands is considered to be 0.1V. Additionally, the range of supply voltage is considered to be between V_{DDH}^{*} and $V_{DDH}^{*}/2$. Due to these constraints the number of islands is not significantly high and total enumeration of all possible solution can be performed in order to find the best choice for supply voltages. Using the chosen voltage set, the Voltage Assignment Table is constructed. This table is the starting point for majority of the published work on Voltage Island Design. This chapter showed how such a table should be generated considering the application to be executed as well as the different threshold voltages that can assigned to each IP block.

CHAPTER 4 : SUPPLY NOISE AND VOLTAGE SELECTION ISSUES IN FLOORPLANNING

4.1. Overview

This chapter focuses on two issues related to voltage island-based floorplanning of the IP blocks in the SoC. First, power supply noise is addressed by ensuring enough white-space would be available to insert decoupling capacitances after the final floorplanning is completed. Second, a further enhancement of the voltage selection methodology is proposed such that highly-connected blocks are placed in the same island in order to reduce the total wirelength of the SoC and increase the likelihood that the floorplanner can produce a feasible solution in fewer iterations. Considering such constraints not only affects the total chip area but also contribute to the energy savings. Moreover, the time required for floorplanning the IP blocks can be significantly reduced.

Floorplanning is the step in the CAD flow in which cores are assigned positions relative to each other within a SoC. Typically, algorithms which perform floorplanning attempt to minimize the total white-space in the final layout. In this work, floorplanning of the cores is attempted once the suitable supply voltages to be implemented on the chip have been identified. After floorplanning, any available white-space can be utilized for level shifter insertion and decoupling capacitance (decap) placement. Decap allocation is crucial for satisfying power supply noise. Typically, the amount of white-space left after floorplanning is not sufficient to allow enough decoupling capacitance, leading to power supply problems [170]. When this occurs, the designer must manually modify the floorplan to create the extra white-space. In this thesis, an early estimation method of decap area allocation is proposed

such that extra white-space can be reserved in advance for decaps in individual islands. This alleviates the designer from making changes to the floorplan in the late stage of the design in order to insert decaps.

In order to estimate the required decap area, some idea of the current levels and voltage drops in the power grid are required by the CAD tool. However, early in the design phase, it is almost impossible to have an accurate model of the current profile of the different cores of the SoC. In such a situation, modeling the current profile as a triangular waveform is common practice [171]. Experimental results have been demonstrated based on a large number of configurations to guide the development of a CAD tool that performs early voltage-island planning (eVIP) [20]. This approach uses a 1-D analysis which is essentially an early stage simulation of the power grid.

Additionally, the choice of the supply voltage for each core has direct impact on the floorplaning problem. Cores with similar supply voltage must be placed adjacent to one another to form islands, leading to additional constraints that must be satisfied by the CAD tool. Choosing supply voltages such that cores with high connectivity are most likely to be assigned to the same island would tend to allow the floorplanner to generate a floorplan with a smaller area. Island creation, total area/wire-length reduction, power optimization and other floorplan constraints are related to one another and must all be considered during the supply voltage selection process. The voltage selection technique discussed in the previous chapter considered energy minimization only. Since it ultimately impacts the floorplanner, a better approach to voltage selection by including connectivity in the algorithm is proposed in this chapter. Parts of this work have been published in [20][172].

4.2. Decoupling Capacitance Allocation

Power grid design in multi-voltage chip has been addressed in [173][174][17]. Design challenges related to dual-supply voltage power grids have been addressed in [173][17]. A new floorplanner supporting voltage islands with special consideration of the power delivery integrity and the proximity of cores to suitable power pads was proposed in [174]. However, decoupling capacitance and inductance was ignored in [174]. This section addresses the problem of choosing the proper decap values to suppress power grid noise of each voltage island before floorplanning is attempted.

To perform simulations of voltage island-based power grids at the pre-floorplan stage of design, a tool that models each voltage island as a 1-D power grid has been developed. Such a model is similar to the power delivery model in [173][175][176][177]. In this tool, the cores are assigned to each voltage island in some manner (discussed in details in Chapter 5) and then modeled as current sources. Power grids are modeled as RLC networks and simulated in HSPICE. Depending upon the peak and average noise, the decap allocation problem is addressed using methods described in this section. The goal is to find a methodology to distribute decaps to produce supply noise within the budgeted limits.

4.2.1. POWER GRID NOISE

Power grid noise can be attributed to two major factors, namely IR drop and Ldi/dt effects [178]. Current flowing through imperfect conductors with nonzero resistance comprise the IR drop term. The Ldi/dt voltage variations at package pins, due to increased rate of switching, have made the situation worse. Together, the voltage drop at any point on the grid can be expressed as:

$$dV = IR + L\frac{di}{dt} \tag{4.1}$$

where dV is the voltage drop, R is the total resistance from the power supply to the switching node and L is the inductance of the power supply network. Decoupling capacitance is allocated when dV is greater than 10% of V_{DD} [179].

4.2.2. VOLTAGE ISLANDS AND 1-D POWER GRID REPRESENTATION

In this work a simple representation for power grid analysis of a SoC is required. Recall that the *voltage assignment table* developed in the previous chapter provides the list of possible voltage assignments for each core. Table 4.1 shows one such table for an SoC with 12 cores, each of which can be assigned to between 1 to 3 different voltage grids, GV_{DD1}, GV_{DD2}, and GV_{DD3}. The assignment process requires multiple iterations of decap allocation until an acceptable solution is found. For each core (C_i), early estimation of the area and average power consumed is usually available, and this is assumed in this work. The required decoupling capacitance would be added to the total chip area. The objective of this work, in such a scenario, is to find the suitable amount of decap for each possible assignment such that the area overhead can be minimized and accounted for at the floorplanning stage of each island. Since this is done iteratively where many assignments will be evaluated, a simple network representation of each power grid is required.

Table 4.1: Voltage Choices For Different Cores

Core Name	Voltage Island Choices
C_1, C_6, C_7, C_{10}	GV_{DD1},GV_{DD2}
C_3, C_5, C_9	GV_{DD2}
C_2, C_{11}, C_{12}	GV_{DD2},GV_{DD3}
C_4, C_8,	GV_{DD3}

Figure 4.1 shows a schematic of a power distribution network in a voltage island design environment. A macromodel of the entire power grid and its corresponding elements, logic blocks, decaps and I/O is approximated as a simple one-dimensional network. The package

leads are represented as a combination of inductors (L_{VDD} and L_{VSS}) and resistors (R_{VDD} and R_{VSS}). In this work, a flip-chip style packaging is assumed and typical values from such packaging are used to implement them. The power grid is represented as resistors ($R_{meshVDD}$ and $R_{meshVSS}$). The mesh resistance and the package intrinsic values (such as the number of solder bumps, etc.) are distributed among the grids based on the area ratio of the cores assigned to each grid. The cores (whether memory, logic, I/O or analog) are represented as time-varying current sources. Further discussion on the modeling of IP blocks as current sources is provided in the next section.

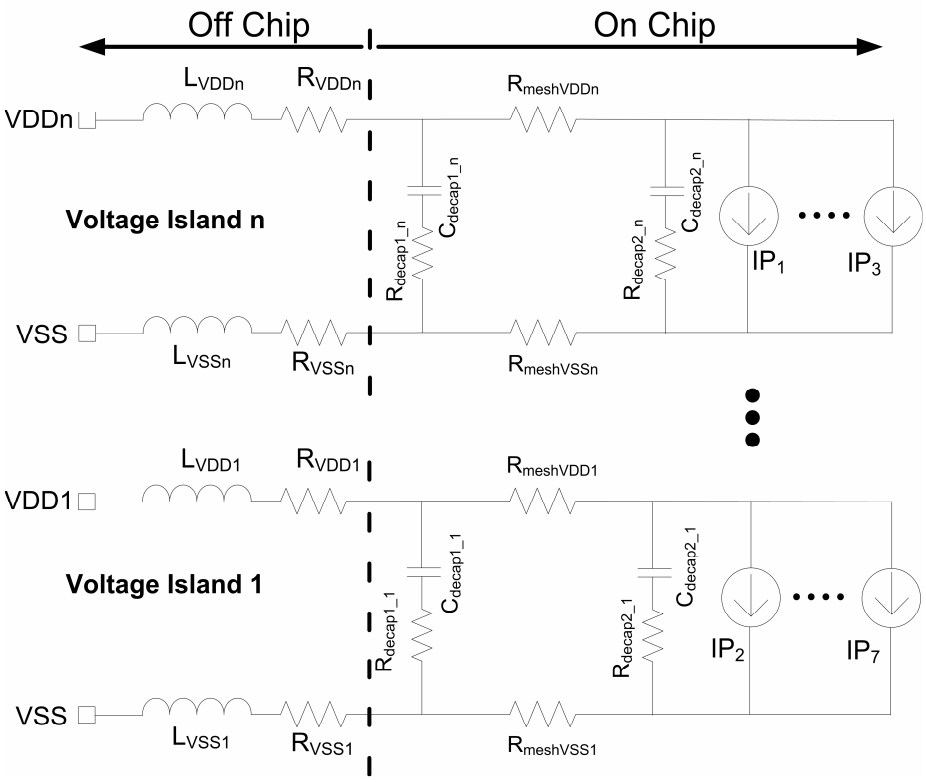


Figure 4.1: 1-D analysis of Voltage Island

In order to obtain a high-level assessment of the worst-case and average voltage drops of the power grid, transient analysis of the 1-D grid is performed in HSPICE. This provides initial estimates of worst-case peak current, supply noise encountered in the power grid as well as the

amount of decap required for a given design. While the use 1-D power grid model (Figure 4.1) can lead to a certain degree of over design, a suitable scale factor can be applied relative to a more accurate 2-D power grid simulation used later in the design cycle based on correlating 1-D and 2-D simulations to bound the error, and hence limit the degree of over design.

4.2.3. CURRENT MODEL AND NOISE CORRELATION

Modeling the current profile of each core as a triangular or trapezoidal-based waveform is common for this type of early analysis [171][180]. In this thesis, each core is modeled as a single current source with a triangular waveform. However, the triangular waveform must be modified for different supply voltages. To determine the nature of those changes, simple HSPICE simulations of the variable activity circuit were performed (Figure 4.2). The selected supply voltages were 1.8V, 1.5V and 1.2V. As seen in Figure 4.2, the rise and fall times of the current and the peak current change significantly with V_{DD}. The new model for the triangular waveform was developed and adjusted based on this type of simulation performed on a number of different circuits until a good correlation was obtained.

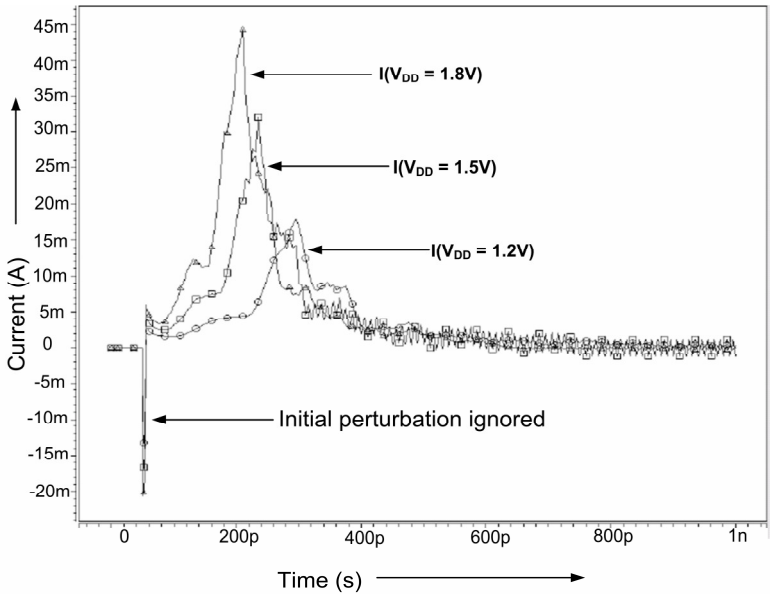


Figure 4.2: HSPICE simulation of current waveform

As shown in Figure 4.3, the current of each IP block is modeled as a symmetric triangular waveform, for which the peak value and time-to-peak change linearly as a function of V_{DD}. The representation has three parameters: the peak current, I_{peak}, the time-to-peak, t_{peak} and the total time required for charging and discharging the capacitance, t_{period}. Normally I_{peak} is modeled as 3-7 times the average current, I_{avg} [146]. In Figure 4.3, T_i is the time when the current reached its maximum value (I_{peak}). Each core has a parameterized triangular waveform as part of its model. The parameters can be adjusted to select any one of the waveforms shown in Figure 4.3. When the cores are assigned to different voltage islands, their corresponding current waveforms are modified in this manner to reflect the change in supply voltages. By using such current profile models and including the other power grid parasitics, a somewhat pessimistic supply noise estimate can be obtained from HSPICE simulations.

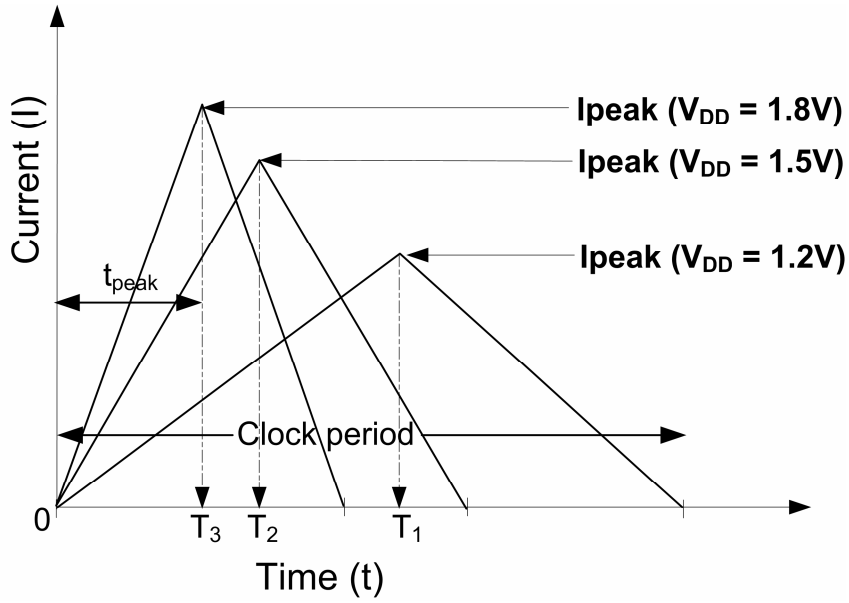


Figure 4.3 : Voltage Island current model

As discussed earlier, there are two sources of power grid noise: IR drop and Ldi/dt. To understand the effect of the current parameters on each of these components, a fixed time window is considered. In this amount of time, the increase in the average current drawn in by

the cores (keeping all other parameters constant) would lead to an increase in the total current demand. This will directly cause an increase in the current flow through the power grid causing the increase in the IR drop noise factor. Reducing t_{period} causes a larger amount of current to flow through the power grid in the same amount of time. This would also cause an increase in IR drop. The effect of Ldi/dt can also be understood by changing t_{period}. With a shorter t_{period}, the steepness of the triangular curve, as it reaches I_{peak}, increases. As the Ldi/dt noise factor is dependent on the slope of the waveform, the Ldi/dt noise level is smallest when

$$t_{peak} = \frac{t_{period}}{2} \tag{4.2}$$

since this value minimizes the steepness of both the rising and the falling edge [171]. Moreover, the steepness of the waveform depends on t_{peak}. Variations in the peak value would cause variations in the noise contributed to the power grid due to the di/dt factor. Thus, the variation of these parameters would enable the designer to understand how the noise in power grid is changing and thus have a quantitative estimate of the amount decoupling capacitance needed to reduce it.

Next, a correlation between the triangular waveform and top-level design parameters is established. Since this analysis is being carried out at a very early phase of design, only rough estimates of power and area of the different cores are available to the designer. The relation between average dynamic power and average current (I_{avg}) of a circuit is expressed as:

$$P_{dyn-avg} = \alpha C V_{DD}{}^2 f = I_{avg} V_{DD} \tag{4.3}$$

Thus, the relationship between the average current drawn by the circuit, the total load capacitance, activity factor and frequency of operation is:

$$I_{avg} = \alpha C V_{DD} f \tag{4.4}$$

For a fixed supply voltage and a fixed frequency of operation, the factor (αC) decides how much current the circuit is drawn from the supply rail. In other words, (αC) is the amount of capacitance that is being charged and discharged. So the amount of charge (Q) that needs to be delivered in one clock cycle is:

$$Q = (\alpha C)\Delta V \tag{4.5}$$

This charge is delivered from the capacitors nearby which will cause a drop in voltage (ΔV), the noise in the power rail. If this noise is more than 10% of V_{DD}, then it can be reduced by the insertion of decoupling capacitance and Eqn. (4.5) can thus be re-written as:

$$Q = (\alpha C + C_{decap})\Delta V' \tag{4.6}$$

where $\Delta V'$ is the new noise value. The relation of charge (Q) and current drawn by the circuit is given by:

$$Q = \int_0^{t_{period}} I(t)dt \tag{4.7}$$

where $I(t)$ is the current profile for one clock period. Once the required C_{decap} is found from HSPICE simulation, it can be translated to the amount of white-space required as:

$$A_{decap} = \frac{C_{decap}}{C_{ox}} \tag{4.8}$$

where C_{ox} is the unit area capacitance of a MOS capacitor [170]. For each island, this additional area is added to the silicon area of the cores in that island.

In a SoC with single V_{DD}, I_{avg} is fixed. In voltage island design, cores can be assigned to more than one V_{DD}. To find the best voltage assignment to the cores, in the floorplanning stage, the cores are moved from one island to the other. This leads to change in the I_{avg} of the core, thereby affecting the supply noise as well. To understand how such variations affect noise and decap amount, Figure 4.4 shows the variation of power supply noise versus amount of decoupling capacitance and the average current. For a fixed decap, the power supply noise increases linearly with I_{avg}. For a fixed I_{avg}, the power supply noise can be reduced by increasing the decap amount by a commensurate amount. Based on the noise tolerated by the power grid and the amount of current drawn by the circuit, appropriate amount of decap can be calculated using Figure 4.4.

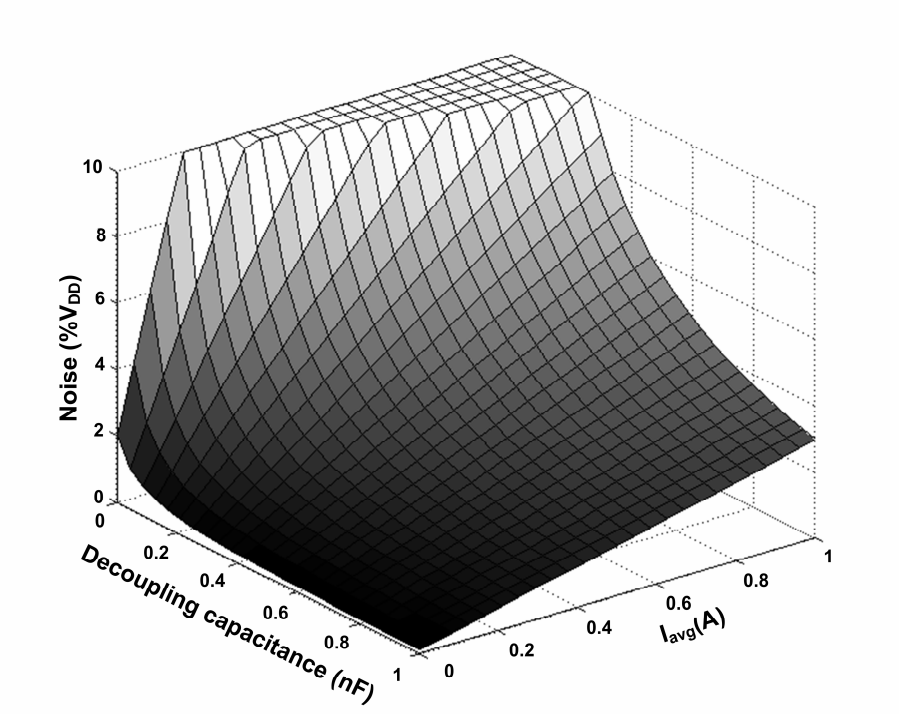


Figure 4.4: Power Grid Noise vs. Decap vs. I_{avg}

4.2.4. DECAP SELECTION IN VOLTAGE ISLAND SOCS

Before floorplanning the voltage islands, the individual IP blocks are each assigned to different legal voltage islands and the appropriate amount of decap for each individual island is determined. Three different techniques: *linear increment*, *binary search* and *exponential method*, have been developed for finding the suitable amount of decap. Before providing the details of each technique, a representative example of a voltage island-based SoC is described.

An industrial design [3] with 21 logic cores and 32 memory cores is used as the case study. First, the choice of voltages is restricted to 0.8, 1.0V, 1.2V and 1.5V. Next, cores are assigned to the different islands such that the assignment meets the power and timing budgets. Then, the power grids are simultaneously simulated using HSPICE and the power supply noise is noted. If the power grid noise satisfies the noise budget then no decap allocation is needed. Otherwise, a decap value needs to be assigned to each island. The objective is to find a methodology that will quickly calculate the decap value for this purpose.

The following subsections describe each of the iterative techniques followed by plots of the decap and voltage noise traces.

4.2.4.1. Linear Increments

As the name suggests, in this technique, the decap value for each power grid is increased from an initial value by a fixed amount (normally a few nanofarads) until the noise margin of the each grid is within acceptable limits. If the noise in any grid is above the acceptable limit at the beginning, then it slowly decreases with the addition of decap. The stopping criterion is when the average noise in every grid is below the 10% of the supply voltage of the particular

[3] Details of the chip are not revealed due to IP restrictions

grid The traces of how decap value and corresponding noise levels change for one unique voltage assignment to the grids is shown in Figure 4.5. As shown in Figure 4.5(a), the decap amount is increased progressively until the noise budget is satisfied. For V_{DD}=0.8V, the decap value stays fixed since the noise level is about 0.04V (only 5%) as seen in Figure 4.5(b). For the other voltages, the increase of decap value stops at a higher value (see inset of Figure 4.5(a)) once the noise margin is within the 10% of the supply voltage of the particular grid. The amount of time required to calculate the appropriate decap amount depends upon the step size. Larger step size would require less time for computation but may result in over design while a smaller step size would give a more accurate result with longer design time. In general this method of computation is very time consuming.

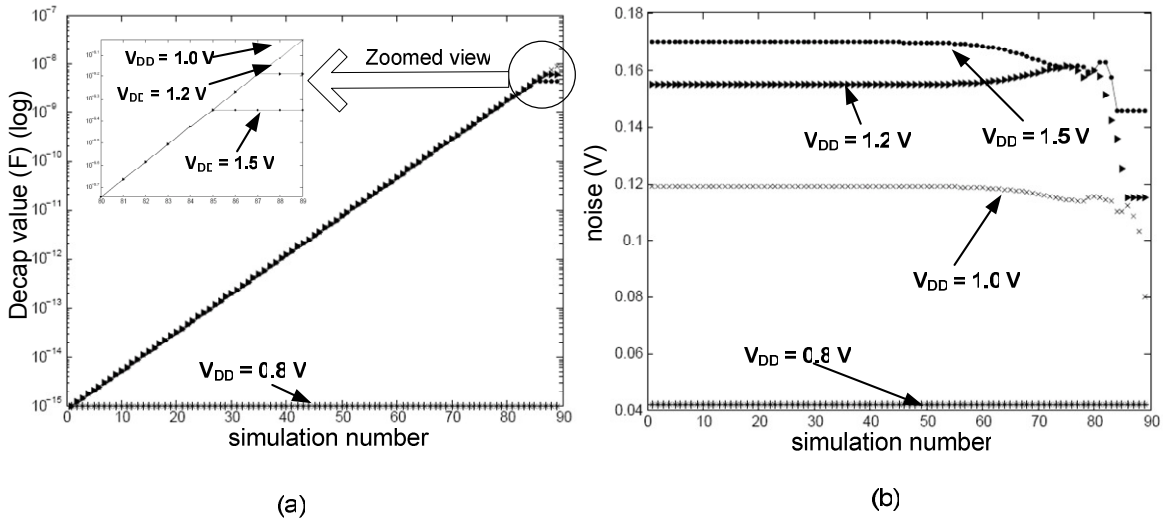


Figure 4.5: Results for Linear Incremental increase in decap value

4.2.4.2. Binary Search

In this technique, the maximum and minimum decap for each power grid is provided. A conventional binary search technique is applied to find the required decap value. Figure 4.6(a) shows one such trace of the decaps and Figure 4.6(b) shows the corresponding noise trace. The first two simulations, 1 and 2, establish the noise levels for the minimum and maximum

values. In this case, the minimum and maximum values symmetrically bound the solution. Since the maximum amount of decap of simulation 2 is very high, with each simulation step, the amount of decap allocated steadily decreases (Figure 4.6(a)) until the noise margin is approximately equal to the 10% of the supply voltage, at which point the decap value is no longer reduced any further (horizontal lines in Figure 4.6(a)). Another stopping criterion is when the amount of decap is below a certain margin to have any significant effect on area. In that case, the simulation stops (as in case for V_{DD}= 0.8V), since the power grid noise is already well below the 10% budget. Clearly, the binary search approach provides about an order of magnitude of runtime improvement based on these results.

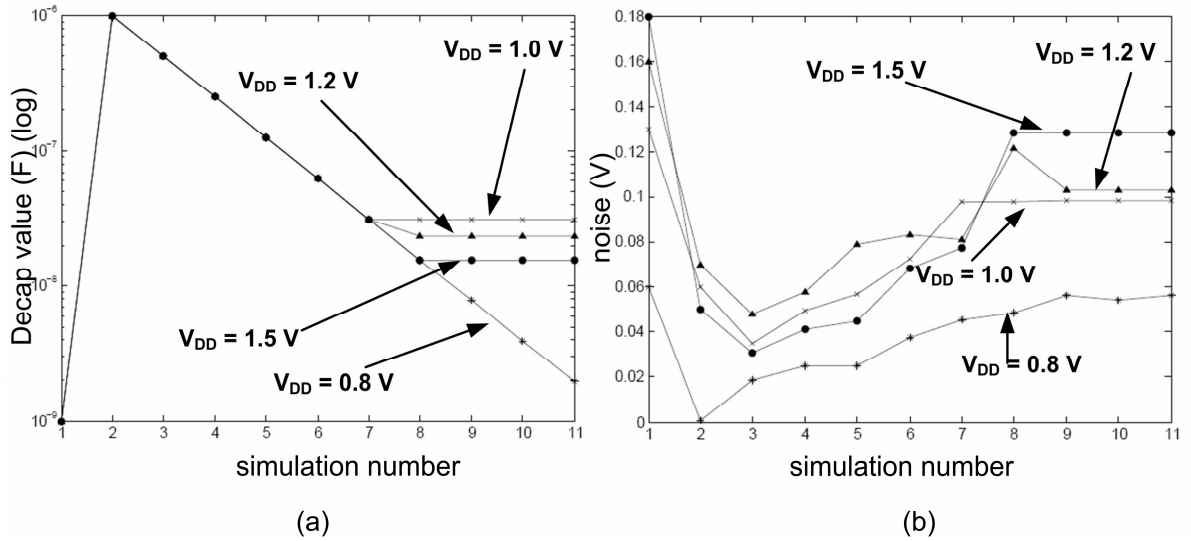


Figure 4.6 : Results for Binary Search method for decap selection

4.2.4.3. Exponential Increase

From Figure 4.4, it was noticed that noise decreases rapidly with the increase in the decap amount. Starting from a small decap amount, it seemed appropriate to use an exponential stepping algorithm to change the decap amount. Figure 4.7(a) shows the decap trace while Figure 4.7(b) shows the voltage noise trace. Unlike the previous cases, the rate of increase of the decap values will not be same for the different grids. This is because the absolute value of

the noise in each grid is used to calculate the new decap amount for each particular grid using the exponential relation. The previous two techniques used the absolute value to decide whether to increase the decap amount or stop. The actual amount of increase was not dependent on the noise. However, only a few iterations were needed to produce the correct results.

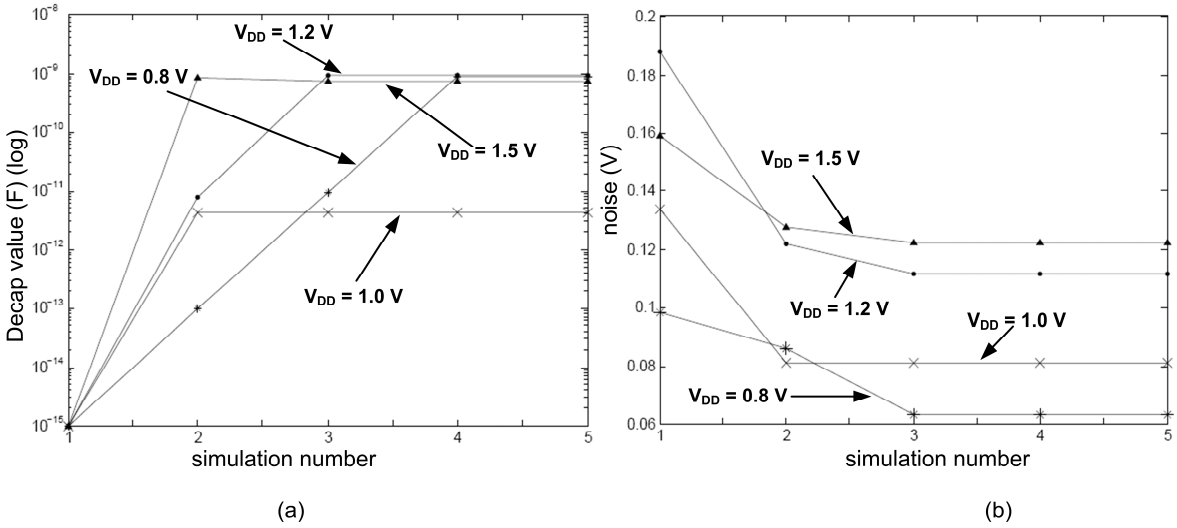


Figure 4.7: Results for Exponential Search method for decap selection

These techniques were performed on a large number of different voltage assignments and solutions for three different SoC were used to illustrate the three techniques. To obtain a more general comparison of each method, Table 4.2 compares the average number of simulations needed to obtain the decap from some initial estimate. As seen in the results, the exponential methodology is much more efficient than the other two techniques so this method is used for decap allocation.

Table 4.2: Comparison Of Search Techniques

Technique used	Average Number of Iterations Required to Reach Solution.
Binary Search	~20
Linear Increase	~66
Exponential Increase	~6

4.2.5. EFFECT OF CORE REASSIGNMENT ON DECAP VALUE

In order to find the best set of voltage assignments during voltage island design, multiple floorplanning attempts must be made with cores reassigned from one grid to another during the process. Each iteration would likely cause a change in the average current demand in corresponding power grids. As a side-effect, any change in the noise levels will require changing the amount of decoupling capacitance to satisfy the noise constraints. The exponential method of finding the required amount of decap is suitable before the first floorplanning attempt. For subsequent floorplanning attempts, the relationship between noise, average current and decap can be utilized to more accurately estimate the needed amount of decap. This can reduce the full design time by diminishing the number of HSPICE simulations necessary to find the appropriate decap value.

To understand how all these factors interact with one another, a very simple situation in which a single core is moved from one power grid (GV_{DD1}) to another grid (GV_{DD2}) is considered and no other change is made to the configuration. In that case, the amount of average current in the grid GV_{DD2} would increase, assuming GV_{DD2} is higher than GV_{DD1}. This would also cause the noise in GV_{DD2} to increase by an amount that can be determined from Figure 4.8. In order to illustrate the process further, Figure 4.8 plots the noise contour (as % of supply voltage of GV_{DD2}) with current on the x-axis and decap value on the y-axis. Each straight line in the plot signifies the decap-current combination needed to have the fixed amount of noise. The point “A” shows a combination of average current and decap amount for having 10% noise margin. Now when a core is added to the grid, the average current increases and the design moves to point “B”. Here, the noise has also increased to 15% of the supply voltage. To return the noise level back to 10% limit, an additional decap must be placed on the grid.

The amount is exactly the difference between "B" and "C". The new solution would be at point "C" where the noise budget is satisfied. Similarly, it is possible to track the change in the average current, noise and decap value for grid GV_{DD1} and adjust the decap value based on the noise margin. In that case, the decap value would be reduced since a core has been removed from the grid.

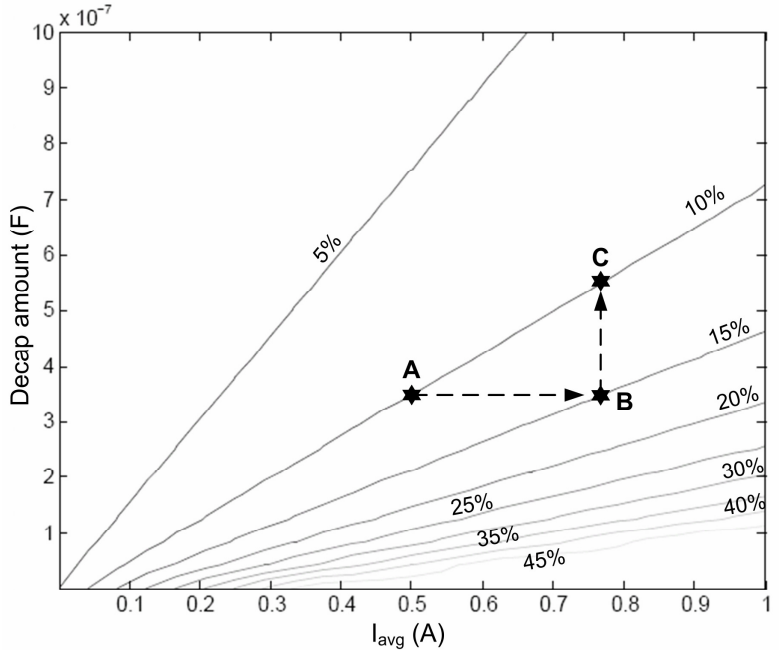


Figure 4.8: Effect of moving IP block from one island to the other

4.2.6. FIXED DECAP ALLOCATION PROBLEM

In the early phase of design, designers may decide on the total area of the chip. In such a case, a slightly different problem arises in terms of decap distribution. If an estimate of the area of each core is available and the total chip area is known then this implicitly places a constraint on the maximum area available for decaps. Given that a fixed amount of decap is available, the problem is to find the best method of splitting the total amount of decap into the different grids such that the minimum noise in each grid is achieved. This section addresses this fixed decap distribution problem.

For each voltage assignment, three distribution techniques for decap assignment were attempted: one based on power, one based on area and a third based on power density (power/area). For the unique voltage assignment with minimum power of the industrial design, the maximum noise percentages of all the voltage islands are provided in Table 4.3. From noise perspective, splitting decap based on power density ratio is the only option since the others do not satisfy the noise constraint.

Table 4.3: Comparison of Noise and Power

Distribution factor	Max. Noise (% of V_{DD})	Satisfies Power Grid Noise Constraint
Area	17	**no**
Power	12	**no**
Power Density	8.5	**yes**

For each voltage assignment, the *allocation ratio* for each grid was also noted. The allocation ratio is the percentage of the total amount of decap allocated to a particular grid. The results are shown in Figure 4.9 for the grid with supply voltage at V_{DD}=1.0V. For every allocation ratio up to 50%, the amount of noise in the power grid is least if power density is used as the metric. Beyond this point, the power metric crosses the density metric but both exceed the noise budget. Thus, it is evident that the best way of fixed decap distribution is to use the power density of the cores.

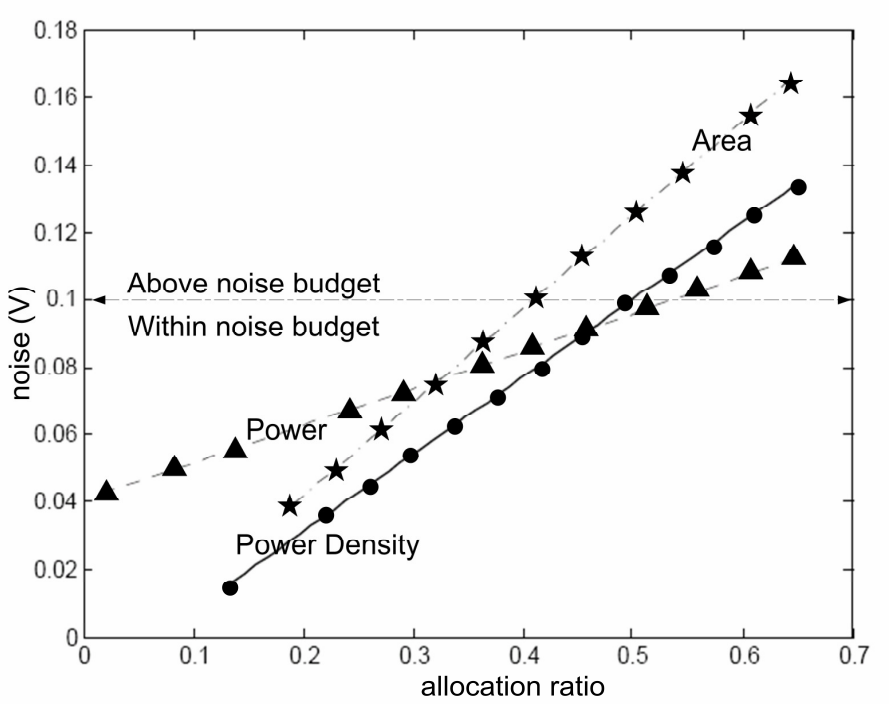


Figure 4.9: Comparison of three metrics

4.3. Floorplan Constraints

As the number of cores on a single chip increases, the number of supply voltages required by cores may be greater than the number of supplies that are realizable on a single chip. Arbitrarily increasing the number of voltage domains does not necessarily improve the power, and sometimes may even reduce the impact of power optimization [167]. Thus, a method for carefully choosing a finite number of supply voltages from a large set of choices was proposed in the previous chapter. The choice of supply voltage was based on reduction of the overall chip energy. Precise information about the floorplan was unavailable at that stage of the design. Therefore, the choice of supply voltages was made assuming that every core is placed at the lowest possible island. As cores with same supply need to be placed contiguously, supply voltage selection effectively decides if two cores should be placed in the same island and in the vicinity of one another. However, during floorplanning, the primary goal is to reduce the wire-length/area while satisfying the timing constraints. This increases the potential

of cores with high connectivity being placed in the same island and it seems worthwhile to include this as part of the voltage selection process. Therefore, this work proposes to incorporate connectivity information during voltage selection in order to determine if it has any useful impact on improving the overall energy consumption.

In the floorplanning step, it is common that a designer will want to control or lock down the positions of some modules for various reasons thereby affecting the final chip floorplaning and voltage island formation [12]. Such requirements arise from the various emerging design constraints such as placing analog blocks in certain sections of chip where the noise is relatively low, or lining up modules horizontally or vertically in order to facilitate pipeline data transfer [182]. Additionally, designers might impose boundary constraints on some IP block pairs in order to facilitate the input-output connections [183]. Figure 4.10 shows an example of SoC design with placement constraint imposed on two IP blocks (C_5 and C_7).

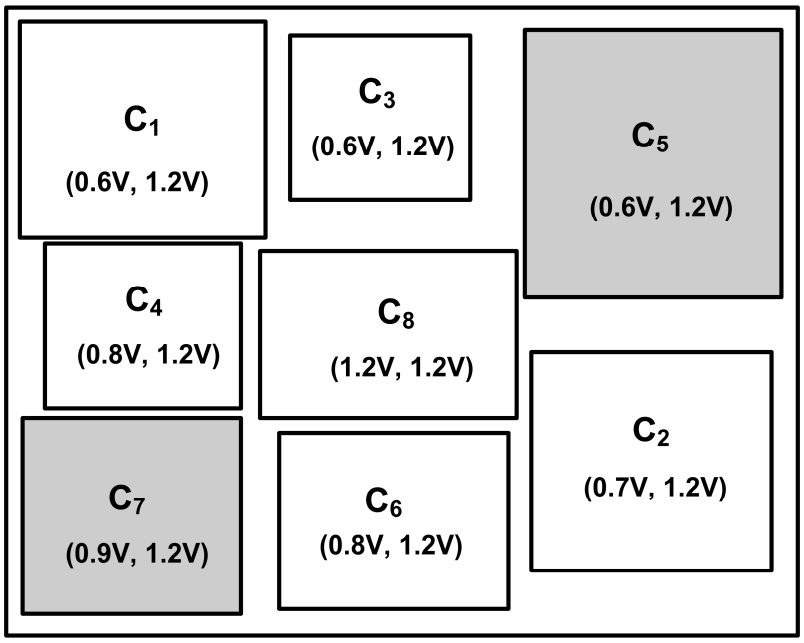


Figure 4.10: A SoC with placement constraint

In the figure, every core C_i, is associated with a pair of supply voltages (V_{DDL}^{i}, V_{DDH}^{*}) where V_{DDL}^{i} is the minimum supply voltage that meets the timing constraints of the core and V_{DDH}^{*}

is the chip-level supply voltage. Core C_i could be assigned to an island with V_{DD} between V_{DDL}^{i} and V_{DDH}^{*}. Additionally core C_5 and C_7 have placement constraints and are required to be placed at the specified locations. Being at the two far ends of the chip, it is highly unlikely that cores C_5 and C_7 will eventually end up in the same island. In the previous chapter, such placement constraints have not been considered while choosing the supply voltages. In [1], these constraints have been imposed on the floorplanner after the initial voltage choice for individual cores have been already made. In this work, such constraints have been incorporated into the voltage selection process.

4.3.1. CONNECTIVITY-AWARE SUPPLY VOLTAGE SELECTION

As connectivity among the cores is an important factor in the floorplanning, Figure 4.11 shows a hypothetical set of connections between the cores as a weighted graph. The edge weight w_{ij} is the number of wires between core C_i and C_j.

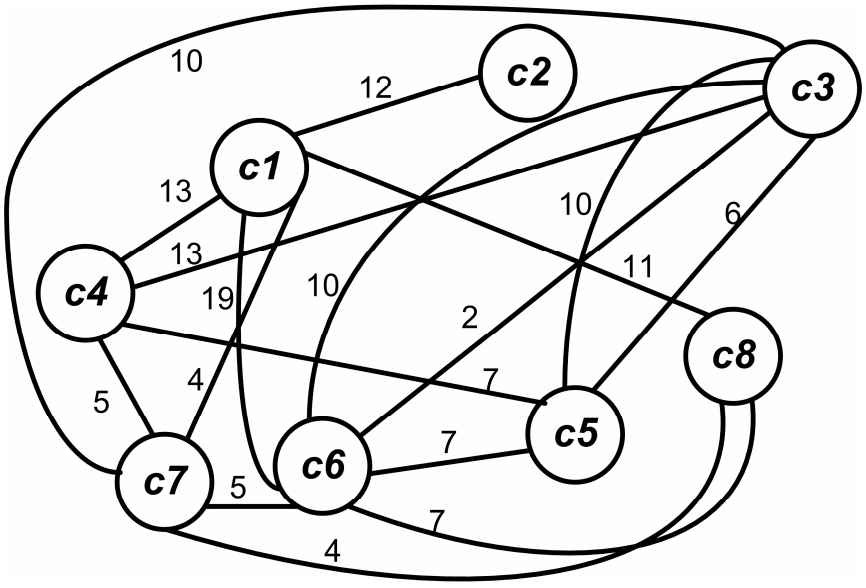


Figure 4.11: Connections among the cores

In order to incorporate the placement constraints and connectivity information, a new cost function f_i for every core C_i in a SoC with r cores is proposed as:

$$f_i = \sum_{j=1}^{r} (\gamma_{ij} e_{v(i,j)} + s_{ij}) \tag{4.9}$$

$$\text{where } \gamma_{ij} = \frac{\sum \text{edges to/from } C_i \text{ to core } C_j}{\sum \text{edges to/from } C_i}, \qquad j \neq i \text{ and}$$

$$e_{v(i,j)} = \text{ energy of core } C_i \text{ in the lowest permissible supply where both } C_i \text{ and } C_j \text{ can be placed}$$

where s_{ij} is the extra cost due to placement constraint imposed on the core C_i with respect to C_j. Formally, the problem for choosing m supply voltages from n voltage choices of a SoC having r cores can be redefined as:

$$\begin{aligned} &\text{minimize } \sum_{i=1}^{N} (f_i(x_p)) \\ &\text{subject to } \sum_{j=2}^{n} x_j = m - 1 \\ &\text{where } x_j = \begin{cases} 1, & \text{if } j^{th} \text{ voltage is chosen} \\ 0, & \text{otherwise} \end{cases} \quad \forall\ 0 < j < n \\ &\text{and } p \text{ is the smallest value for which } x_p = 1 \text{ and } d_i(p) \leq D_i \end{aligned} \tag{4.10}$$

The larger the value of γ_{ij}, in Eqn. (4.9), the higher the chances of C_i and C_j being placed in the same island. Hence, the function f_i includes the connectivity information among the cores when deciding the total energy consumption of the chip. Also, the value of s_i depends upon the constraints for a particular core. For example, during voltage selection, if any possible supply voltage can be assigned to both cores C_i and C_j , s_{ij} represents the additional cost of placing the cores in dissimilar islands due to the placement constraint.

The voltage selection problem with the new cost function is solved using the same heuristic technique discussed in the previous chapter. The energy values of Table 3.2 in the previous chapter and the connectivity information of Figure 4.11 are used for calculating the cost

function in Eqn. (4.9). Figure 4.12 shows the step-by-step results of the voltage selection method using the modified cost function in Eqn. (4.9).

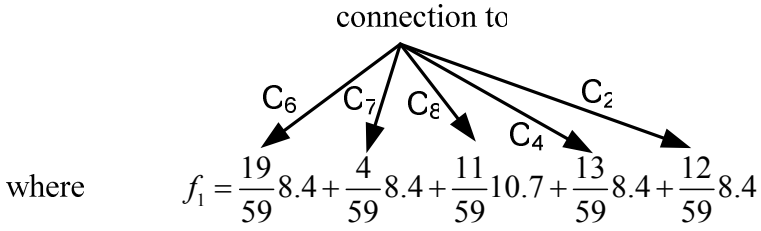
possible supply voltages

No. of voltage islands	1 (1.2)	2 (0.9)	3 (0.8)	4 (0.7)	5 (0.6)
1	54.0 *(1.2)*	54.0 *(1.2)*	54.0 *(1.2)*	54.0 *(1.2)*	54.0 *(1.2)*
2	54.0 *(1.2)*	43.8 *(0.9, 1.2)*	41.1 *(0.8, 1.2)*	41.1 *(0.8, 1.2)*	41.1 *(0.8, 1.2)*
3	54.0 *(1.2)*	43.8 *(0.9, 1.2)*	39.6 *(0.8, 0.9, 1.2)*	39.2 *(0.7, 0.9, 1.2)*	39.2 *(0.7, 0.9, 1.2)*

Figure 4.12: Connectivity-based voltage choice

The results are produced one column at a time from left to right. The optimal supply voltages for the voltage islands are included with each energy value. The first row and first column are trivial solutions as only one supply is chosen. The final solution is the square containing the minimum energy. In Figure 4.13, the f_i for each core is calculated and sum of all cores is minimized to compute z. The calculation of f_1 for the cell (2,2) is shown below:

$$z = f_1 + f_2 + f_3 + f_4 + f_5 + f_6 + f_7 + f_8 + s_{57}$$
$$= 8.8 + 5.6 + 7.8 + 3.9 + 4.1 + 3.4 + 2.5 + 6.3 + 1.4 = 43.8$$



where $f_1 = \frac{19}{59}8.4 + \frac{4}{59}8.4 + \frac{11}{59}10.7 + \frac{13}{59}8.4 + \frac{12}{59}8.4$

Figure 4.13: Function calculation

Similarly, f_i was calculated for all other cores. When placement constraints are considered, the assignment of cores to the islands is further restricted. For the example in Figure 4.13, it is ensured that C_5 and C_7 are always in different islands. The values of s_{57} are set such that C_5

and C_7 are most likely to be connected to different supply voltages. In order to do so, in Figure 4.13, C_7 is placed in the island with V_{DD}= 1.2V while C_5 is placed in the island with V_{DD} =0.9V. The cost of forcing C_7 to be placed at a higher voltage has been included in s_{57}.

Using the *Energy-based* cost function from the previous chapter, the optimal set of V_{DD}'s to be implemented on the chip were {0.7V, 0.8V, 1.2V}. The new optimal set using the proposed *Connectivity-based* cost function is {0.8V, 0.9V, 1.2V}. Comparing the two cases, the optimal energy using the new cost function is larger than the previous cost function but it is more likely to produce a suitable floorplan. Although some higher supply voltages are chosen in the new approach, it was found that for a predetermined die size, the final floorplan solution in new approach consumes less energy and area compared to previous techniques (discussed in Chapter 5). This is because assigning cores to the lowest possible V_{DD} leads to an area that is over-budget. For an acceptable final floorplan solution, some cores have to be assigned to a higher V_{DD} leading to higher chip power/energy. Figure 4.14 shows the final floorplan solution of the above example.

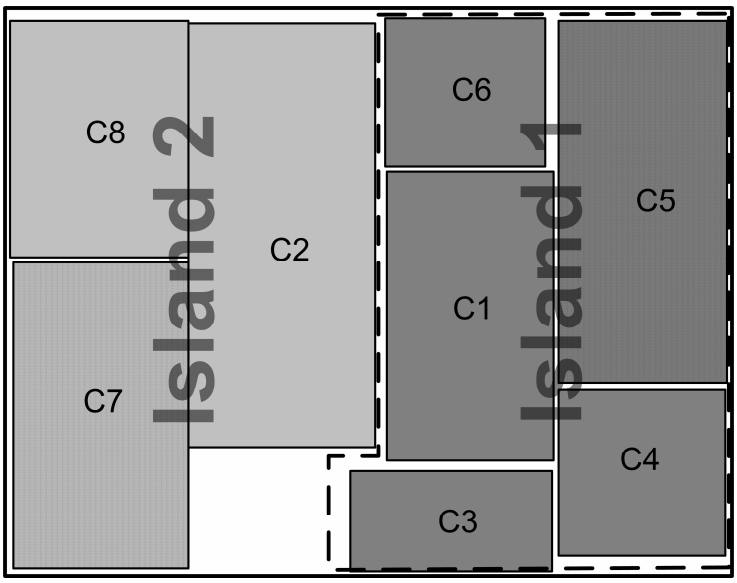


Figure 4.14: Voltage Island Floorplan with placement constraints

A compromise between the energy and connectivity objectives computes the energy of C_i as follows:

$$f_i = \eta e_{iv(l)} + (1-\eta)\sum_{j=1}^{r}(\gamma_{ij} e_{v(i,j)} + s_{ij}) \tag{4.11}$$

where η is between 0 and 1, $e_{iv(l)}$ is the energy consumption of C_i in it's lowest possible island. The variation of energy consumption as a function of the number of voltage islands for three different cases is shown in Figure 4.15.

The maximum number of islands is considered to be 20 and the SoC is composed of 100 cores. Except for the case where there is only one supply voltage, the *Connectivity-based* assignment predicts larger energy consumption than the *Energy-based* case. In the third case, where $\eta = 0.5$, the energy consumption lies between the two boundary conditions. To generate the Voltage Assignment Table, a suitable value of η should be chosen based on the power/energy budget, area budget and the number of possible supply voltages to be realized on the chip.

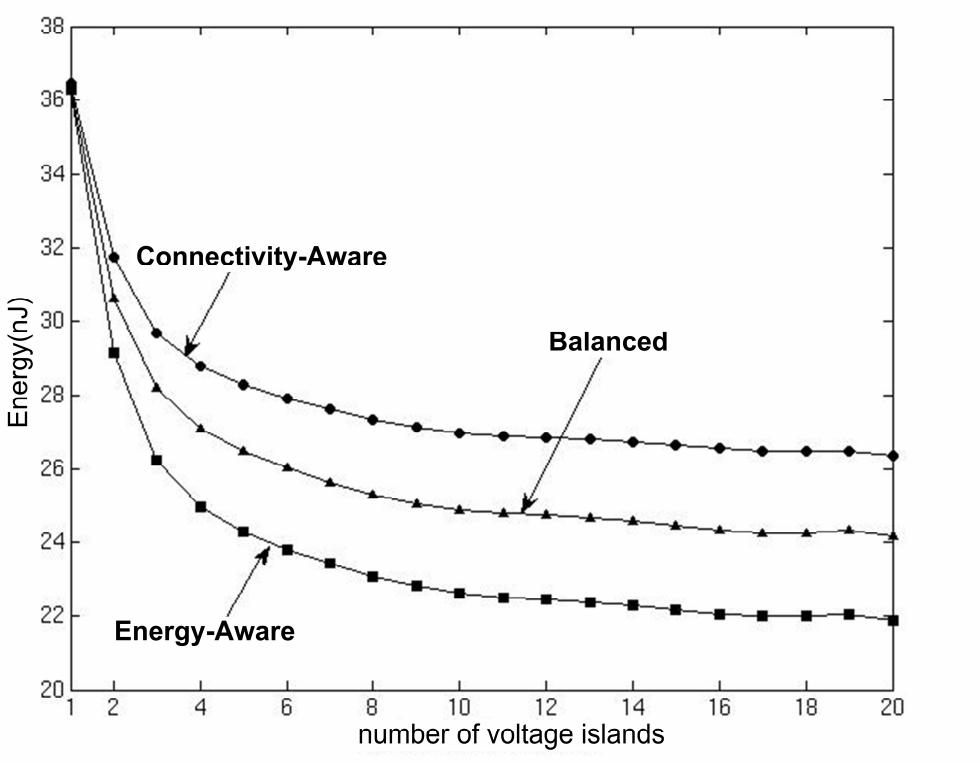


Figure 4.15: Comparison of three voltage assignments

4.4. Summary

This chapter has focused on the design constraints that need to be addressed at the floorplan stage of the design. As the time required for floorplanning is increasing significantly with the increase in the number of IP blocks in the SoC, the steps described in this chapter would produce a feasible floorplan and reduce unnecessary floorplan iterations due to area over budget.

The inter-relationships between noise in the power grid, decap allocation and average current have been explored. There is a linear relation between noise and average current. The methodology to be adopted to meet the power budget when a single core is moved from one grid to the other has also been described. Based on the results presented in Section 4.2.4, the exponential search should be applied to find the bounds on the decap value and the binary search can be applied to obtain the suitable value.

Next, placement restrictions on the cores in the SoC were addressed. Due to such constraints, not all cores can be placed in the voltage island with minimum possible V_{DD}. This would not only increase the energy consumption of the core/s on which the restriction is imposed but also affect the neighboring cores that are connected to it. These factors have been included in the cost function used for voltage selection, and hence in the overall floorplanning process.

CHAPTER 5 : VOLTAGE ISLAND FLOORPLANNING

5.1. Overview

The contribution of this thesis is a voltage island aware floorplanning algorithm. The Voltage Assignment Table developed from the previous chapters provides a starting point for the selection of supply voltages for each core. In this design phase, the cores are assigned to unique islands and floorplanning is performed. To find a feasible solution that meets the design constraints, a very large design space needs to be explored. Due to the numerous choices of voltage assignment to each core, this task must be performed in a judicious manner. The seemingly better solutions, purely from a power perspective, are the ones that have more voltage islands. However, these solutions have a higher area cost, hence the tradeoff. Thus, the power and area budget should be included in the assignment process. Furthermore, connectivity between cores plays an important role in the quality or ability to complete the floorplanning operation and should also be included in the approach.

Due to the existence of an enormous number of choices for voltage assignment to the cores, not all possible solutions can be explored. A two-phase voltage assignment methodology has been proposed in this work that attempts to floorplan only those assignments that can potentially lead to a good solution with a high probability of completing the floorplan operation. The intuition behind a two-phase voltage assignment is that aggressive power optimization of high power cores would maximize the benefits of power optimization while connectivity-aware voltage assignment of low-power cores would yield floorplan solutions with a smaller area overhead.

After the assignment phase, floorplanning is performed hierarchically. First, the islands are floorplanned to form macroblocks and then the chip-level floorplanning is performed. During island floorplanning, the area overhead for decap allocation is included. This ensures that the designer will not have to make late changes in the floorplan to include extra decoupling capacitance. Parts of this work have been published in [184].

Results from all of the proposed techniques from the previous chapters as well as the floorplanning methodology are included in this chapter. The power savings at different stages of the design have been included as well to identify the aspects of the approach that have the highest impact.

5.2. Design Exploration

This section discusses the different optimization goals that govern the floorplanning of voltage islands. The principal design objectives are power and area reduction while ensuring that timing constraints are satisfied. The timing constraints have been considered in the voltage table generation process. Therefore, all solutions that do not meet timing are automatically removed from the *Voltage Assignment Table* (VAT) during the generation process. The constraint on the number of supply voltages has also been included in the VAT. The supply noise issue in terms of decap area requirements is included while exploring different assignments of cores to islands. It is tempting to prefer the solution that would provide the lowest chip power consumption. However, there are also other system level considerations that will drive the final decision. In order to facilitate the selection process, two cost functions, one for power (C_P) and one for area (C_A), are defined:

$$C_P = P_C + P_{LS} + P_I$$
$$\text{and } C_A = A_C + A_{LS} + A_{decap} + A_{VI} \tag{5.1}$$

where P_C, P_{LS} and P_I are the power values of the cores, level shifters and interconnect, respectively, while A_C, A_{LS}, A_{decap} and A_{VI} are the areas of the cores, level shifters, decap and power grid layout for the voltage islands, respectively. From a power perspective, the voltage assignment solutions that meet the system power budget can be considered as the set of feasible assignments. Of these feasible solutions, the ones that require the fewest voltage islands should be considered first. This is because of the high cost associated with adding each extra island. It implies either extra power supply voltages for the chip or additional on-chip voltage generation circuits. Moreover, the power grid routing and additional level shifting, as well as the placement of the blocks within the voltage island presents further problems.

As a case study, an industrial chip having 52 cores has been used for this analysis with design parameters for 90nm technology was considered. Figure 5.1 plots each voltage assignment in terms of power consumption and the corresponding number of islands. Each point in the plot represents a unique voltage assignment. Those solutions above 15mW can be removed from consideration as they violate the power constraint. In this example, there is no voltage island configuration consisting of only one voltage that satisfies the power budget. With two or three islands, the budget can be satisfied, but many of them exceed the power limit and are not further considered. As mentioned earlier, the best solution purely from a power perspective is the one that has the maximum number of islands. However, the goal is to identify configurations with fewer islands, as specified by the designer to reduce the overall design complexity.

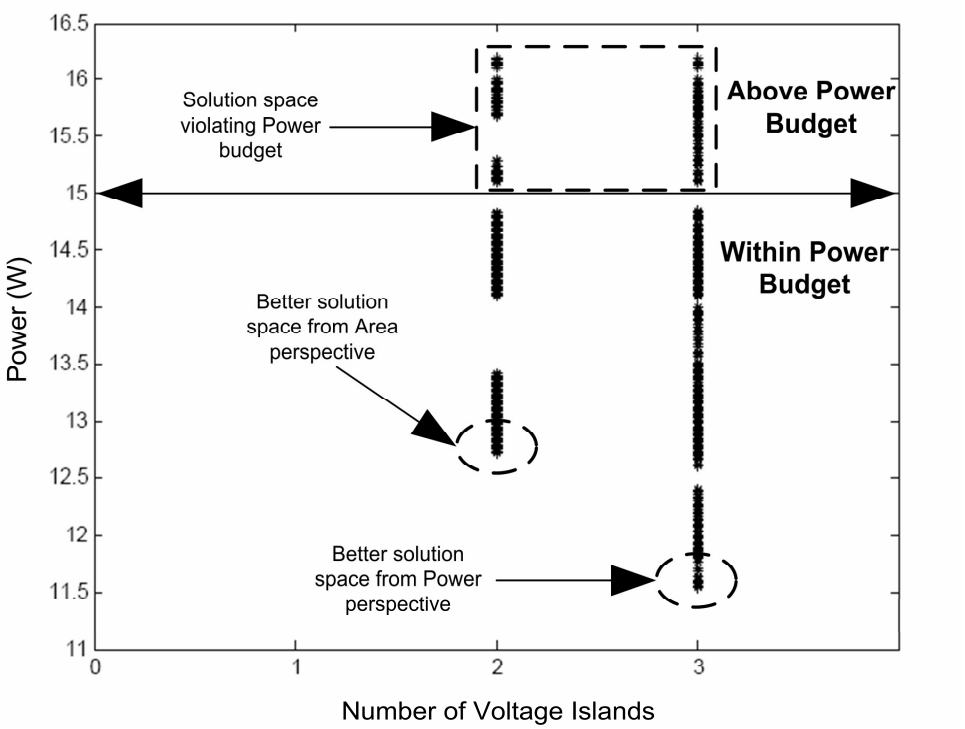


Figure 5.1: Power vs. number of Voltage Islands

The number of voltage level converters is another important factor that needs to be taken into account. With more islands, there is a requirement for more level converters between the islands. They contribute to the total area as well as the overall chip power consumption. A solution with fewer level converters is preferable. The total chip area is composed of the core area, the level shifters, the power grid routing, and the additional area for decaps. The core area is fixed for all the cases. The power grid routing and level shifters vary depending on the size of each island and the number of connections between them. The decap area is also variable and depends on the power grid noise in each configuration. Again, the configuration with the lowest amount of decap is the better solution in terms of overall area.

Based on the above analysis, a methodology for assigning the cores to the different islands is as follows. The first step is to determine whether the voltage assignment solution is feasible from the power and timing perspectives. All assignments that do not satisfy the power

constraint are removed from further consideration. Starting with the assignment that would consume lowest possible power, the number of level shifters needed must be calculated. Then, using the 1-D simulation tool, the configuration is simulated to estimate the decap requirements. A calculation of the total chip area, including core areas, area of the level shifters and additional area for the voltage island generation as well as power routing is performed. All solutions that meet area and power budgets are considered in the floorplanning stage.

Figure 5.2 shows the solution space using the power-area cost factors. Each "o" is a valid voltage assignment. Solutions near the origin are highly desirable. Therefore, the designer (or in this case the floorplanning tool) should start from origin and move diagonally away and consider each assignment in that sequence.

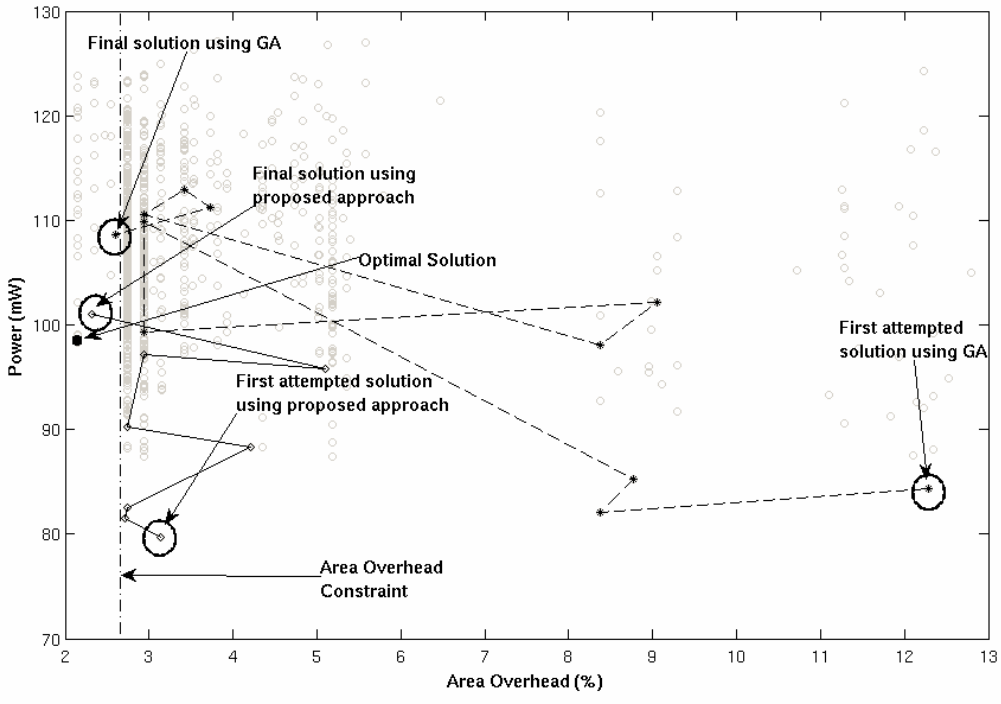


Figure 5.2: Comparison of design exploration

The sequence of solutions attempted using the proposed approach is shown using solid lines while solutions attempted using genetic algorithm [21] has been shown as dotted lines. As an overview of the proposed technique, solutions are explored from lowest-power solutions and work upwards from there. In the figure, the first attempted solution and the final optimal solution for the proposed technique as well as for the genetic algorithm-based approach have been identified. The optimal solution is also marked in the figure. Note that voltage assignments far from optimal are often attempted by the genetic algorithm. The proposed technique is specifically targeting low power, low area partitions to avoid any floorplan attempts that are unlikely to produce a desirable result and result in long runtimes. The next subsections discuss the methodology for voltage assignment as well as floorplanning in detail.

5.3. Voltage Assignment for Floorplanning

As mentioned throughout this thesis, floorplanning in presence of multiple voltage islands is much more complex than floorplanning of IP blocks with a single V_{DD}. The complexity stems primarily from the need for cores with similar supply voltage to be placed contiguous to one another in order to form islands. Different floorplan techniques have been proposed at different stages of the design. In [80][79][185] voltage islands were created starting with a fixed floorplan. Floorplanning techniques considering the thermal distribution during voltage island design have been proposed in [186][21].

Based on the voltage assignment table, cores are assigned to one of the permissible voltage islands. Next the islands are floorplanned individually and then the floorplaning of all the islands at the chip level is considered. From power/energy perspective, assigning cores to their minimum possible supply voltage is an attractive option. However, the final floorplan may

contain more white-space than permitted due to chip area budget. In such a situation, voltage assignment is performed again to get the next best solution and then the floorplanning process is repeated.

A two-step approach of voltage assignment and floorplanning in [1][21] are summarized in Figure 5.3.

Algorithm [21]:
Begin
L1: Given the voltage assignment table and power value
L2: Use Heuristic Based Approach for voltage assignment to each core.
L3: Use Simulated Annealing for Floorplanning each island
L4: Perform cost calculation for each newly formed island
L5: Attempt chip level floorplanning.
L6: If termination criteria are not met go to L2
end

Algorithm [1]:
Begin
K1: Given the voltage assignment table and initial floorplan
K2: Perturb initial floorplan solution
K3: Attempt to merge islands.
K4: Perform cost calculation for each newly formed island
K5: If termination criteria are not met then go to K2
end

Figure 5.3: Comparison of design exploration

The algorithms for voltage island implementation in [21] and [1] attempt floorplanning in the inner loop of a heuristic-based voltage assignment algorithm. Unfortunately, as the number of IP blocks in a single chip increases with technology scaling [2], runtimes associated with floorplanning can increase significantly [25]. Both [1] and [21] note that their approaches require multiple floorplanning steps and that the execution time is only acceptable for small number of cores. Thus, the efficiency of the floorplanner is crucial for reducing the design time and this was a key consideration in this work in identifying only the best candidates before floorplanning. Voltage assignment of the IP blocks is considered an important step in this design flow. In the proposed approach, careful voltage assignment places highly-

connected cores in the same island thereby reducing the total wirelength of the chip and increases the likelihood of a successful floorplanning process. Next, the island formation process, cost function and the floorplanning operations are all described in the following sections and subsections.

5.3.1. PROPOSED VOLTAGE ASSIGNMENT METHODOLOGY

With the background discussion of the overall methodology fully described, the details of the proposed approach to voltage island formation at the floorplanning stage are described. The actual assignments of voltages to cores are performed at the island formation stage of the design. In typical high-performance IC's, there are a few blocks with high activity while others have relatively low activity [187]. Additionally, on investigating the power consumption of cores in chips designed for different applications it was found that the number of high power consuming cores was approximately 10-20% of the total number of blocks in the SoC [188][189][190]. This work uses such power consumption profiles among the cores for voltage assignment purposes. Using methods described in previous chapters, the power for each core is computed and ordered from highest to lowest power. Starting with N cores, the V_{DD} assignment is carried out in two phases.

1. *High-Power Blocks (Phase I)*: The first k out of N cores that have the largest contribution to the total chip power is selected. The lowest available V_{DD} from Voltage Assignment Table is assigned to each of the high-power cores and the design flow proceeds to *Phase II*. On the next iteration of *Phase I*, the core that results in the lowest increase in power is moved to its next highest V_{DD}. This process continues on subsequent iterations until all voltage options are exhausted.

2. *Low-Power Blocks (Phase II)*: Voltage assignment for the remaining (*N-k)* cores that have a relatively smaller contribution to the overall power is performed using a cost function (described in the next subsection) that considers both connectivity and power.

Voltage Assignment in *Phase I* is related to power optimization. The deterministic process sequences through the cores and voltages in a pre-determined order in an attempt to provide maximum gain in reducing power. The value of *k* should be in the range of 50-60 or less since the deterministic approach can potentially explore all possible combinations until a suitable result is found. The heuristic technique in *Phase II* is more focused on the floorplan challenges. If there are a large number of cores, the permutations and combinations are exponentially large, so a heuristic approach makes the assignment a tractable problem. In some cases, a low power block may be moved from its lowest possible supply voltage to a higher voltage due its greater connectivity to the blocks in the higher voltage island. Using *Phase I* and *Phase II*, voltage assignment is repeated until the floorplan with an acceptable chip area is found. On every new assignment, the total achievable power reduction is diminished as the high power cores are progressively assigned to higher supply voltages. Thus, the sooner the floorplan provides an acceptable solution, the better is the solution. The power penalty for assigning some of the low power cores to the higher supply voltage is amortized by the reduced number floorplan attempts to meet area budget, thereby reducing the overall power of the SoC and runtime of the iterative process. The cost function used for voltage assignment is *Phase II* is discussed next.

5.3.2. COST FUNCTION

For the *k* highest-power cores, which should comprise 70-90% of the total power, initial islands are based on their lowest permissible voltages listed in the VAT. For the remaining *N-k* cores that have smaller activity and lower power, voltage assignment is done using a cost function that progressively chooses better connectivity over lower power. From a floorplanning area perspective, solutions that produce the highest connectivity within voltage islands, and fewest connections between them (i.e., requiring the fewest level shifters) are preferred. The proposed cost function is designed to prioritize power for the initial cores, and then place an increasing emphasis on connectivity for the rest of the cores to ensure that a feasible floorplan can be generated.

The remaining *N-k* cores are sequenced in a sorted order, with the highest power cores assigned first followed by the lower power cores. Taking each core in the sequence, the incremental chip power increase for each possible assignment to a permissible island is calculated. The ratio of wires connected to the core from a permissible island to the total number of wires connected to the core is also calculated. Given certain weighting factors to these power and connectivity terms, the cost of assigning the given core to each of the islands is computed. The core is then assigned to the island with the lowest cost. Formally, if "*s*" blocks have been pre-assigned to different islands (including the first *k* cores from *Phase I*) and *C* is the next core to be placed, then the assignment problem with cost estimate (E_i) can be represented as:

$$\min E_i = \chi_1 \frac{P_C(i)}{\sum_{j=1}^{s} P_{Cj}} - \chi_2 \frac{W_{Ci}}{W_{Ctotal}}, \quad i = 1 \; to \; r \tag{5.2}$$

where the block can be placed in r permissible islands, $P_C(i)$ is the power consumption of the block when placed in i^{th} island, P_{Cj} is the power consumption of a previous j^{th} block that has already been assigned, W_{Ci} is the number of wires from core C to island i and W_{Ctotal} is the total number of wires of core C. As more cores are assigned to the islands, the denominator of the power term will grow and, therefore, the second term will dominate the first term in Eqn. (5.17). Eventually, connectivity will take control of the assignment over power.

By adjusting χ_1 and χ_2, the assignment of the (N-k) blocks can be entirely power-based (χ_1=1 and χ_2=0) or connectivity-based (χ_1=0 and χ_2=1), or any combination in between. The resulting chip power, area and required number of level shifters depend on the choice of χ_1 and χ_2. To illustrate this for a chip comprised of 1000 cores, a number of experimental runs were carried out using different values of χ_1 and χ_2. The results of these experiments in terms of chip power and the number of level shifters are shown in Table 5.1.

Table 5.1: Power vs. Level Shifters

χ_1	χ_2	Power (W)	# Level Shifters
1	0	13.6	243
0.67	0.33	13.8	192
0.5	0.5	14.0	172
0.33	0.67	14.2	154
0	1	14.3	137

To balance the two factors, equal values of χ_1 and χ_2 (=0.5) were used for most of this work. However, the effect of using other values, depending upon the relative importance of power, area is also explored in the later sections.

5.4. Floorplanning

Once the IP blocks are assigned to unique voltage islands and the amount of required decap for the individual islands is estimated, the next task is to perform floorplanning. Similar to the floorplanning technique in [1][21], floorplanning is performed in two stages as follows:

1. *Island Floorplanning*: Except for the island with the chip level supply voltage (usually it is the highest supply voltage, i.e., V_{DDH}^{*}), each island is floorplanned individually. The total area for each island and the amount of extra white-space is noted. The white-space required for the power supply noise constraint is compared to the existing white-space in the islands. Accordingly either the total area of the island is increased or remains same. Placement constraints can be included in the floorplanning of the cores in the island.

2. *Chip-level Floorplanning*: This is the final stage of floorplanning for the entire SoC. Since *all but one* of the islands that have already been floorplanned, they are considered as hard macro blocks. It is assumed that the chip-level supply voltage (i.e. V_{DDH}*) is available throughout the chip and thus contiguous island formation is not a strict criterion for the cores residing in the island with V_{DDH}*. In case of on-chip V_{DD} generation, such a condition holds true. If the different supply voltages are delivered from off-chip sources via dedicated pins then the cores with V_{DDH}* must be contiguous as well and must be performed in the previous floorplanning stage. In this case, chip level floorplanning involves the floorplaning of all the islands in the SoC.

Similar to the previous floorplan stage, any new constraints can be included in this stage of the design. If the constraint is on a core that has already been placed within the island, then the

constraint is imposed with respect to the entire island. Once the final floorplan is obtained, the pre-placed islands are reinstated by the cores in the respective island. Decap allocation for the chip level supply voltage is performed next. Similar to the previous step, the total area of the chip might be increased to accommodate the total decap. Finally, the level shifters are placed at appropriate locations to complete the design.

Figure 5.4 shows the two phased floorplanning methodology for voltage island design.

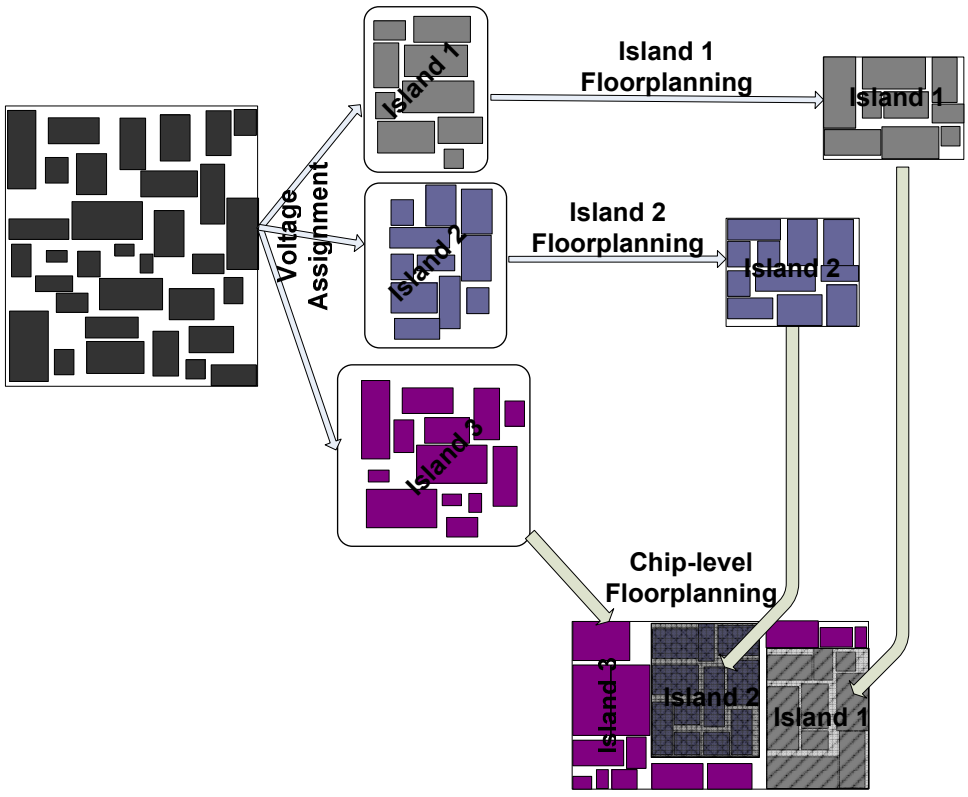


Figure 5.4: Voltage Island Floorplanning

At the end, if the total chip area is greater than the chip budget, then the floorplan is discarded and voltage assignment is performed again. It should be noted that for subsequent floorplan attempts, if cores form islands that had already been created at some previous stage, then the

floorplan of the island is reused. This results in fewer floorplans iterations and thus reduces total design time.

5.5. Overall Design Flow

The overall design flow for voltage island design using multiple V_{DD} and multiple V_T is provided in a detailed algorithmic format in Figure 5.5.

Algorithm: Voltage Island Design:
Begin
L1: Given the PSM of the cores of an SoC, calculate the relative activity of each core;
L2: Assign threshold voltage to each core based on activity factor;
L3: Calculate the possible supply voltages and corresponding energy consumptions for every core;
L4: Generate the *Voltage Option Table*;
L5: ***if*** the number of required voltage choice is greater than the number of V_{DD} that can be implemented on the chip ***then***
L6: Use *Voltage Selection Methodology* to choose the supply voltages in order to minimize the total chip energy;
L7: Generate *Voltage Assignment Table*;
L8: *Phase I*: Assign *best* possible supply voltage, that is yet to be attempted for floorplanning, to the cores with higher activity for power optimization;
L9: *Phase II*: Assign cores having small activity (and thus nominal contribution to overall chip power) to the existing islands using the cost function based algorithm;
L10: Use HSpice simulation to allocate white-space to each island for placing decoupling capacitance;
L11: Attempt the voltage assignment solution for floorplanning;
L12: ***if*** *feasible* floorplan reached ***then***
L13: Solution reached;
L14: Output Voltage Assignment and Floorplan;
L15: Return;
L16: ***else***
L17: Go to Step L8;
end

Figure 5.5: Overall design flow

This approach was tested on a number of MCNC [191] and GSRC [192] benchmark circuits. As the designs in the benchmarks were not originally intended for voltage islands, additional information was added. For all the circuits, a 90nm CMOS design kit was used in order to

model dynamic power, leakage power, threshold voltage, power and area of level shifters. A representative (but artificial) PSM was generated for each benchmark using an approach similar to [193]. The power consumption and delay of each core at 1.2V was randomly assigned between 1mW and 2W, similar to the approach used in [194]. The activity of each core was derived from the PSM. Next, the V_T was assigned to each core (L2). Table 5.2 compares the power saving obtained using multiple V_T and power gating technique as opposed to a single V_T.

Table 5.2: Power saving due to Multi-V_T Design

Circuit	No. of HVT cores	No. of SVT cores	No. of LVT cores	Power (single V_T)	Power (multi V_T & power gating)	Power Saving (%)
apte	3	5	1	1.85	1.15	37
n10	0	9	0	1.32	1.21	8
xerox	3	6	1	1.88	1.7	10
hp	3	6	2	2.11	1.79	15
n30	9	18	3	1.70	1.33	21
ami33	15	14	4	2.08	1.60	23
ami49	14	30	5	4.12	3.70	12
n50	3	45	2	1.72	1.53	11
n100	36	54	10	4.95	3.099	37
n200	18	160	22	6.70	4.95	26
n300	54	216	30	10.77	8.55	20
AVG						**20**

On average 20% of total power can be saved when multiple V_T and power gating are used in place of single V_T design. Note that *n10* did not result in any savings due to the use of different V_T libraries. The ony savings were achieved by the use of power gating. It must be noted that the savings can vary considerably with changes in these values.

The next task is to identify the range of allowable V_{DD} for each core and obtain the 3 interesting supply voltages (L3). Voltage selection is performed in L4-L6 of Figure 5.5. The previous voltage selection technique was compared to the proposed approach [167]. As [167] selects only 3 supply voltages, comparison for 3 V_{DD}'s is considered here. Each technique chooses a distinct set of supply voltages. The power savings obtained using the proposed algorithm for voltage selection as compared to the previous approach in [167] is shown in Table 5.3. The power consumption for each circuit is the sum of power of each individual core when paced in the lowest possible supply voltage from the chosen set. The average power savings using the new approach is an additional 4% in comparison to the previous approach in [167].

Table 5.3: Comparison of Voltage Selection Algorithms

Circuit	**Power (Previous approach** [167]**)(W)**	**Power (Proposed approach)(W)**	**Power Savings (%)**
apte	1.39	1.37	1
ami33	1.70	1.65	3
n50	1.69	1.64	3
n100	3.31	3.05	8
n200	5.51	5.51	0
n300	9.71	8.82	11
AVG	-	-	**4**

For some circuits, this work has occasionally resulted in the use of more than three supply voltages. Using the chosen V_{DD}'s, the *Voltage Assignment Table* is generated in L7. Step L8 and L9 performs voltage assignment to the cores using the two-phase approach discussed in the previous section. Step L10 attempts to estimate the average IR drop and the corresponding decoupling capacitance requirement in each island prior to floorplanning. Since there is no

floorplan available during this step, a simple 1-D power grid that satisfies worst-case IR drop for each island is used. For each voltage assignment, dynamic IR drop simulation is carried out and the decaps are adjusted until the average voltage drop satisfies the allotted budget (5-10%) for each grid. The total amount of decap is then translated to the amount of required white-space.

Finally, floorplanning is attempted with the partitioned cores to create voltage islands. In each island, additional white-space is reserved for decap allocation after completing the floorplanning operation. If a feasible floorplan is obtained, then it is considered to be a desirable solution and the results are accepted. If not, the design flow goes back to *Phase I* and repeats the process.

The floorplanning methodology was implemented on the same benchmark circuits as in Table 5.2. Similar to [1], the possible number of supply voltages allowed on chip was assumed to be proportional to the number of cores. The proposed approach was tested using the HotSpot Floorplan Tool that included the LAPACK package [195]. The experiments were run using Intel Pentium 4, 2.4 GHz machine with 2GB RAM running on Debian Linux (version 2.6.18).

The flow presented in [21] is the main comparison point due to the availability of the floorplanning tool as well as their use of the same benchmark circuits. As the Voltage Assignment Table is the starting point in [21], the same table generated in step L7 is used as a starting point for their design flow. In the proposed approach, equal importance was given to both power and area by assigning $\chi_1 = \chi_2 = 0.5$. Similar priorities were given to the coefficients in the fitness function in [21].

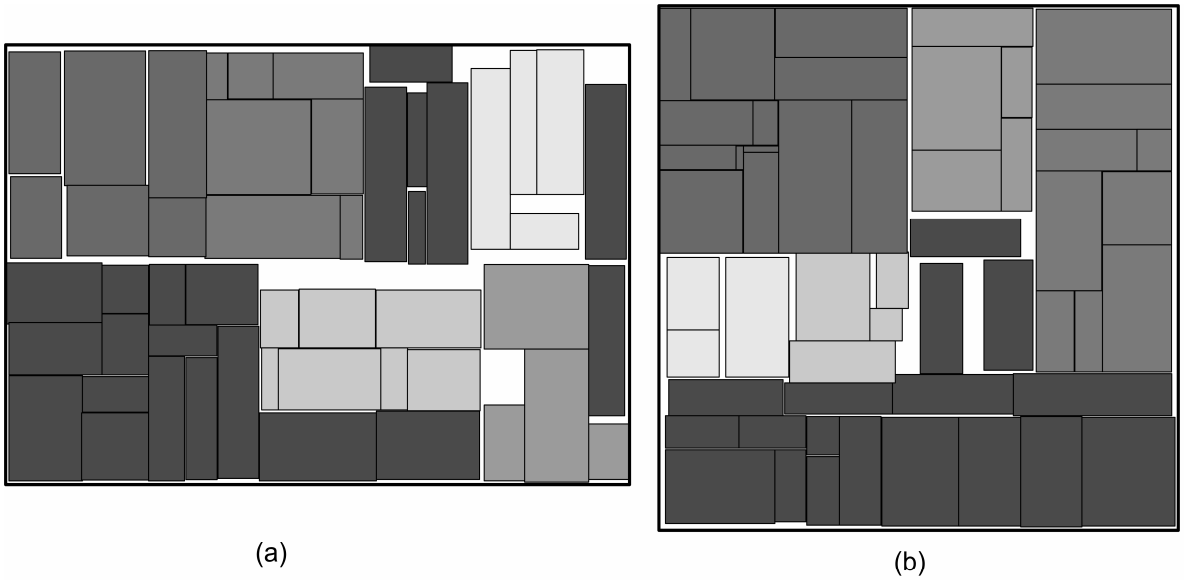
Table 5.4 compares the runtime and the quality of results (number of voltage islands, area, and power) of proposed approach with [21]. Both methods were required to use the same number of islands. Columns 10-12 compare the savings of the proposed technique against [21]. The power and area values of the new method are significantly better than those obtained using the flow in [21] with an average improvement of 8-10%. This is due to the fact that the approach properly accounted for high power blocks and high connectivity blocks in the cost function when making the voltage assignment of cores to islands.

Table 5.4: Power and Area Comparison

Circuit	**No. of Cores**	**#vi**	**Flow in** [21]			**Proposed Flow**			**Savings**		
			Area (μm^2)	**Power (W)**	**CPU (min)**	**Area (μm^2)**	**Power (W)**	**CPU (min)**	**Area (%)**	**Power (%)**	**CPU (X)**
apte	9	3	49.15	1.71	17	47.00	1.51	5	4.3	11.6	3.4
n10	10	3	3.70	1.82	13	3.16	1.89	8	14.5	-3.8	1.6
xerox	10	3	20.10	2.1	15	20.10	2.1	5	0	0	3
hp	11	4	9.08	2.23	35	8.92	2.06	15	1.8	7.6	2.3
n30	30	6	6.27	1.92	40	5.45	1.41	20	13	26.5	2
ami33	33	6	1.55	2.07	41	1.34	1.86	22	13.5	10.1	1.8
ami49	49	6	45.10	4.46	50	39.87	4.02	20	11.6	9.8	2.5
n50	50	6	4.07	2.26	59	3.83	2.04	25	5.8	9.7	2.3
n100	100	9	2.27	4.53	90	2.14	4.09	31	5.9	9.7	2.9
n200	200	15	2.40	8.2	160	2.11	7.31	60	13.7	10.8	2.7
n300	300	20	4.18	12.4	180	3.81	10.88	90	12.1	12.2	2
AVG									**8.7**	**9.4**	**2.4**

The runtime is 2.4X better on average. This is due to the reduced number of floorplan iterations required to generate a feasible floorplan. Note that voltage assignments far from the optimal are often attempted in the genetic algorithm [21]. The new approach is specifically designed to avoid such floorplan attempts, as they would only add to the runtime. The incremental cost is relatively small for ordering the cores and finding the preferable voltage assignments for which floorplanning should be attempted.

In the case of *xerox*, the two techniques produced the same power and area solution though the technique proposed in this thesis was much faster than in [21]. For the circuit *n10* the power consumption was slightly higher for the proposed result. Such anomalies are possible for the smaller circuits. Figure 5.6 compares the floorplans of n50 obtained using [21] and the proposed approach. Both floorplans used 6 islands, but the power consumption and specific cores assigned to the islands were different. The new solution in Figure 5.6(b) had an area that is 5.8% less and a power that is 9.7% less than the corresponding result of Figure 5.6(a).



Case	Island 1	Island 2	Island 3	Island 4	Island 5	Island 6	Total
(a)	0.08 W	0.17 W	0.10 W	0.29 W	0.34 W	1.28 W	2.26 W
(b)	0.07 W	0.11 W	0.17 W	0.37 W	0.48 W	0.84 W	2.04 W

Figure 5.6: Floorplan of *n50* using (a) technique in [21] (b) proposed approach

Investigating the thermal effects on the floorplan of each benchmark (reported from the HotSpot floorplanner), it was found that the average temperature in the proposed approach is approximately 9-10% lower than [21]. Additionally, the peak temperature was approximately 13% lower. The reason is that the new approach assigns more cores to islands with lower supply voltages. This leads to an overall reduction in dynamic and leakage power which, in

turn, lowers the on-chip average and peak temperature [196]. A comparison of average and peak temperature of six benchmark circuits is shown in Figure 5.7.

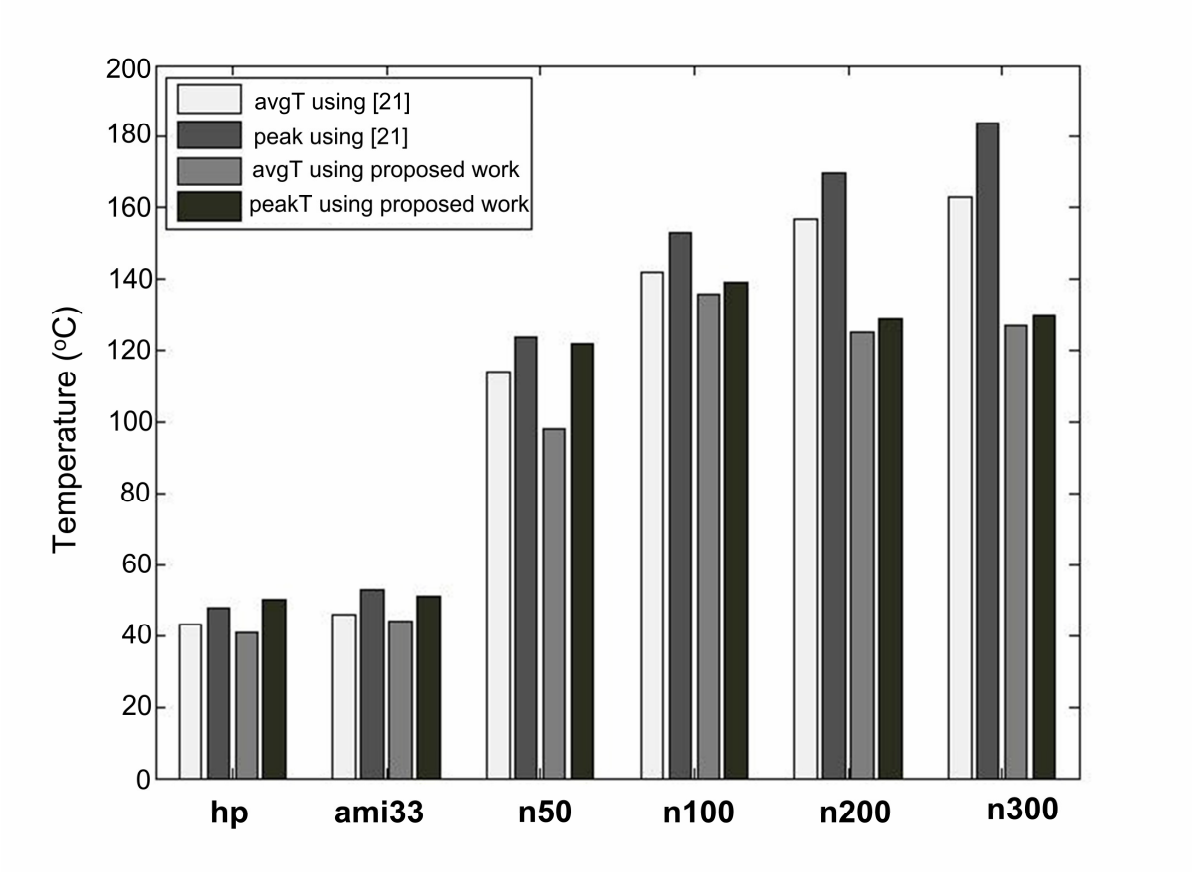
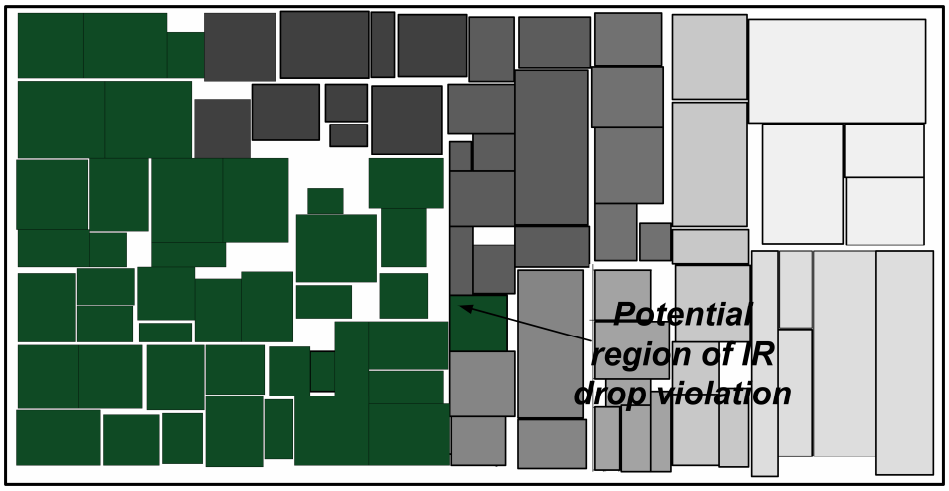


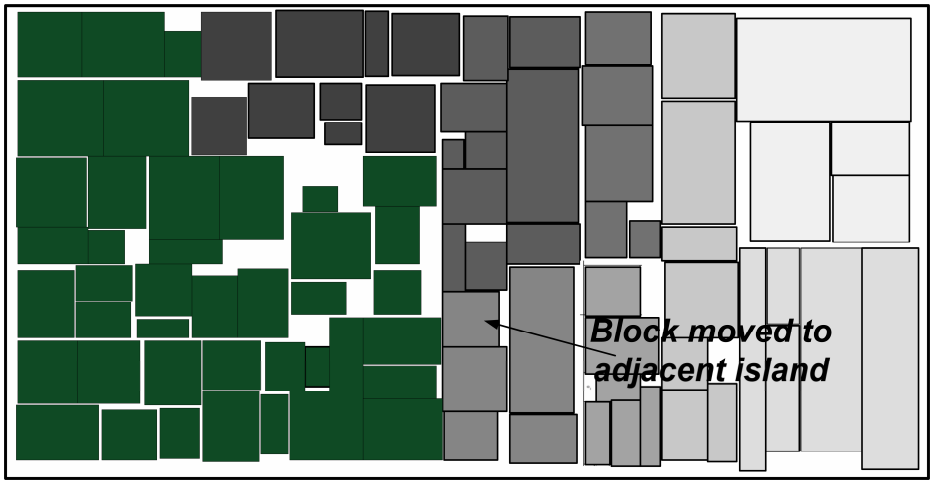
Figure 5.7: Average and Peak temperature comparison

The floorplan solutions of all the benchmarks were carefully inspected and it was observed that, in some cases, there were blocks that belonged to one island but were surrounded on three sides by other islands. This problem can be seen in circuit (*n100*), as shown in Figure 5.8(a). Voltage island design not only requires floorplaning of blocks belonging to similar islands in a contiguous fashion but also must ensure that there are no power grid design problems due to the final floorplaning. Such floorplans may produce an IR drop problem when a 2-D power grid analysis is performed during the final validation of the design. To avoid such potential problems, this block is moved to another adjacent islands and the

incremental cost in power, level shifters and power grid design is assessed. The lowest cost solution is used, as long as only two sides are adjacent to other islands. The solution selected for this case is shown in Figure 5.8(b) and indicates the final adjustment to a different island.



(a) Floorplan of *n100* with potential IR-drop problem



(b) Floorplan of *n100* without IR-drop problem

Figure 5.8: IR drop consideration

To investigate the decap allocation problem, two floorplan solutions involving 3 voltage islands were considered. In the first case, Figure 5.9(a), the floorplanning does not incorporate decaps. In the proposed approach, Figure 5.9(b), IR drop analysis was already carried out and extra white-space was included for decap allocation in each island. Though Figure 5.9(a)

requires a slightly smaller area, there is an IR drop violation. However in the latter case, proper allocation of decaps to each island ensures that the supply noise constraint can be met, along with power, thermal and area constraints.

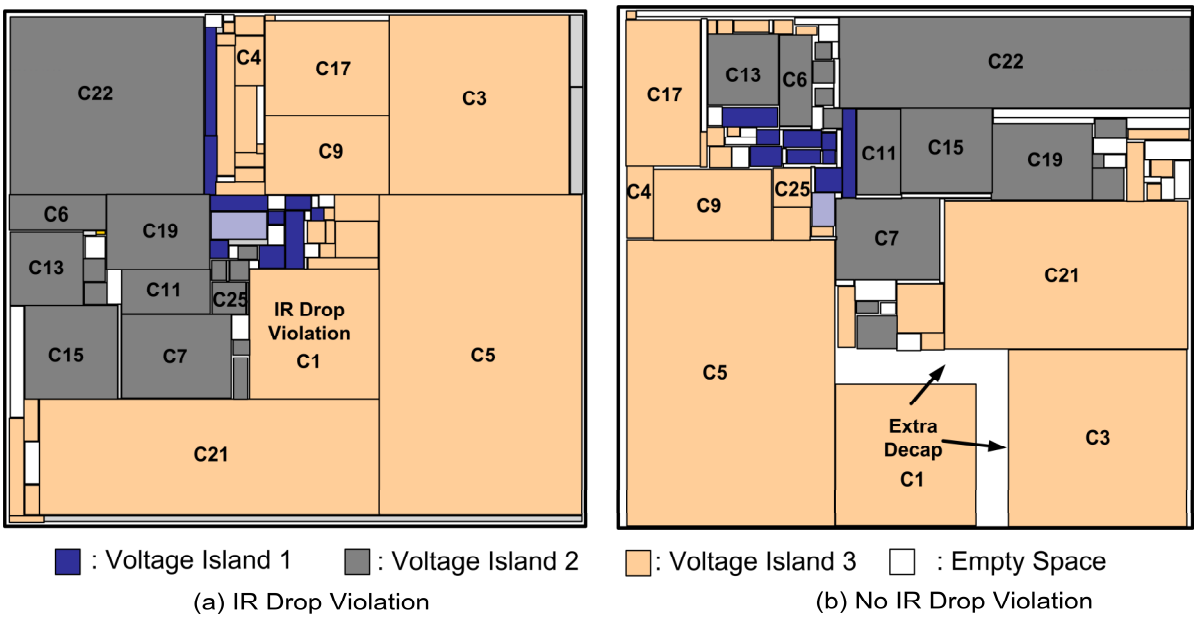


Figure 5.9: Floorplan solutions from (a) without and (b) with decap consideration

In all the previous results reported here, the value of χ_1 and χ_2 were both 0.5. The value of these two parameters has a considerable effect on the power of the chip. Moreover, for a fixed area budget, the amount of permissible overhead affects the voltage island formation. This indirectly affects the power savings possible. Figure 5.10 shows the relation between the parameters (χ_1 and χ_2), power consumption and area overhead.

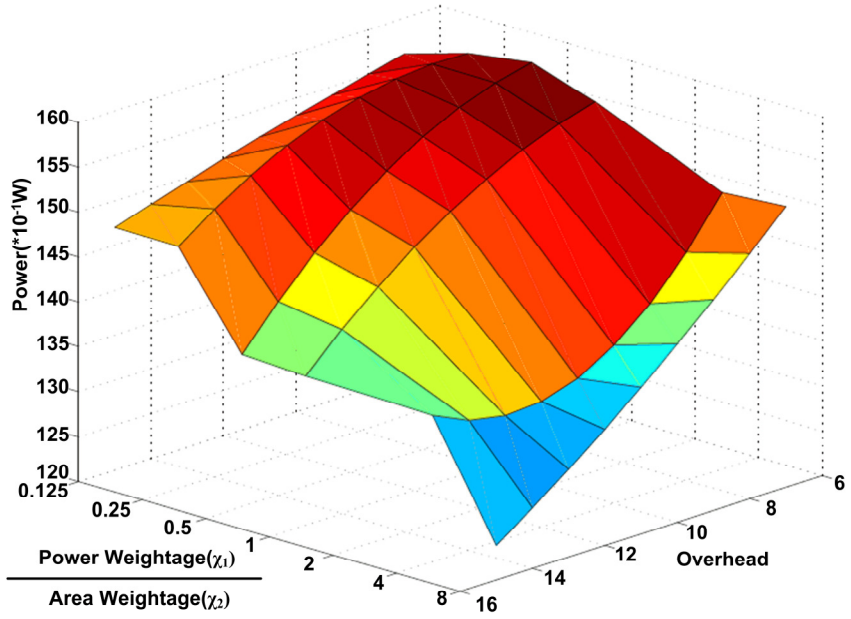


Figure 5.10: Design parameters vs. Power vs. Overhead

The greater is the chip area, the higher is the power savings. Power is directly related to the ratio of (χ_1 /χ_2). For overhead ranging between 6-16% the total chip power consumption can vary by as much as 40%. Thus, based on the allowable overhead and the target power budget, the proper values of the parameters should be chosen carefully to satisfy these design tradeoffs.

5.6. Connectivity-Aware Voltage Island Design

In Table 5.4, the *energy-aware* cost function of Chapter 3 (Equation 3.4) was used for the voltage selection algorithm rather than the connectivity-aware cost function of Chapter 4, as given in Eqn. (4.9). Results obtained using the cost function in Chapter 4 are discussed in this section. There were no placement constraints on the IP blocks to be placed in the SoC in the previous results in Table 5.4. However in reality, such constraints exist and can limit the amount possible power saving. Therefore, typical floorplan constraints were imposed on the design to provide more realistic results. Due to unavailability of benchmark circuits with

floorplan constraints, random assignment of constraints was performed on the benchmark circuits used in the previous sections. The floorplanner proposed in [183] was used in this case due to the capability of including placement constraints on the cores.

Table 5.5 compares the results when *connectivity-aware* cost function is used instead of *energy-aware* cost function for voltage selection. The voltage assignment of the cores in the floorplanning stage is identical to the approach discussed in the previous sections in both cases.

Table 5.5: Connectivity Aware vs. Energy-Aware

Circuit	No. of islands	Flow using Energy-Aware Technique		Flow using Connectivity Technique		Savings	
		Power (W)	Area Overhead (%)	Power (W)	Area Overhead (%)	Power (%)	Area Overhead (%)
apte	2	1.1	2.7	1.0	2.7	9.0	0
hp	2	2.6	9.3	2.2	5.1	15.3	4.2
ami33	2	10.6	7.7	9.7	7.6	8.5	0.1
ami49	3	6.6	9.4	6.3	7.5	4.5	1.8
n50	3	15.7	5.9	15.1	4.4	3.8	1.5
n100	4	13.0	7.0	12.4	6.5	4.6	0.5
n200	5	19.0	11.9	17.5	10.9	7.8	1
n300	5	18.7	12.6	17.9	11.6	9.6	1

Columns 7-8 compare the savings of connectivity-aware approach to energy-aware technique. For simplicity, the placement constraints were randomly assigned to a subset of the cores in the design. The results show that the power values are significantly better than those obtained using the previous voltage selection scheme with an average power improvement of 7%. In comparison to the energy-aware methodology higher supply voltages were chosen for the connectivity-aware methodology. However as smaller number of cores were required to be placed in the highest V_{DD} for the connectivity-aware solution, the total power consumption of the SoC was less than that of the energy-aware technique. The die size for each circuit was

predetermined and the area overhead is reported. On average, in the new approach the area overhead is less than the previous technique. This implies more area can be allocated for other resources such as test logic, buffers, etc. Figure 5.11 compares the floorplans of hp obtained using the two methods. A pre-placement constraint has been imposed on one of the cores. The solution in Figure 5.11(b) has an area that is 4.2% less and a power that is 15.3% less than the corresponding result of Figure 5.11(a). These results make intuitive sense and demonstrate that connectivity and energy should be used in both voltage selection and island formation to produce the highest-quality results. This is a key research result of the work in this thesis.

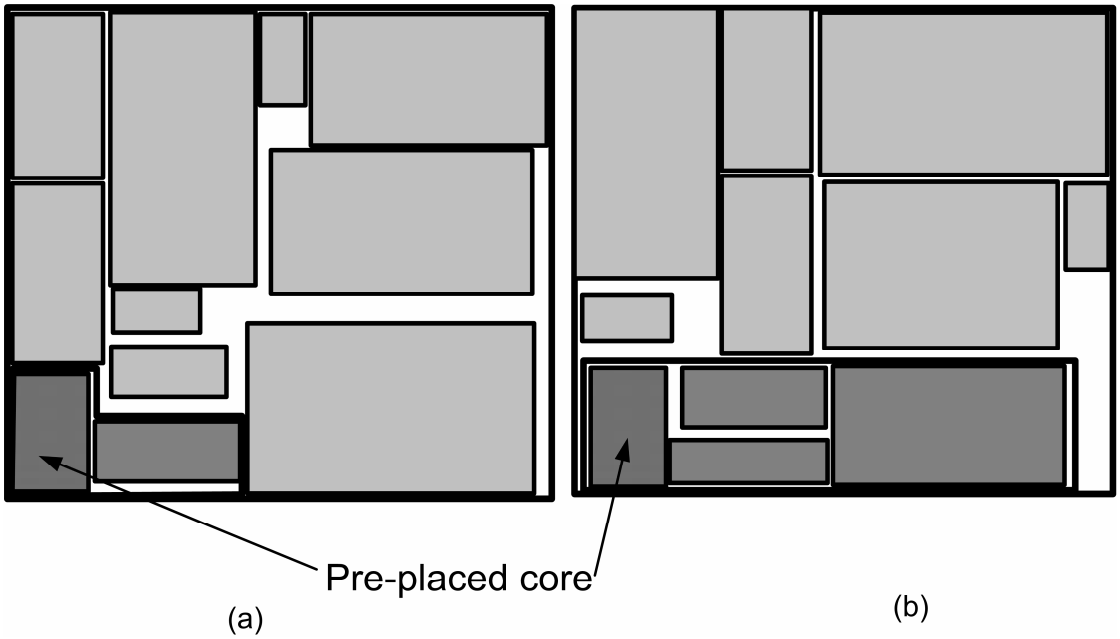


Figure 5.11: (a) Energy-aware (b) Connectivity-aware floorplan

5.7. Summary

In this chapter, the final phase of design process has been described. The design space exploration also provided some insight into the parameters that control the final results of voltage island designs. A two-phase voltage assignment technique has been used in the new approach. Next, a two-stage floorplanning operation, beginning with the islands and then the chip-level floorplanning is performed. Depending upon the white-space available, the area of

the islands is adjusted to include the area for decaps. Experimental results for the entire design flow have demonstrated the improved quality in the overall results. On average, the use of multiple V_T and power gating can reduce almost 20% of power. On average, a proper choice of supply voltages leads to another 4% reduction in power. The floorplanning technique proposed here offers another 10% in reduction of power. However, it must be noted that in the entire design flow a fixed area and power budget were considered. These constraints significantly affect the savings obtained. Based on the given design budget, proper choice of design parameters is essential to obtain high-quality results using these techniques.

CHAPTER 6 : CONCLUSIONS AND FUTURE WORK

Technology scaling offers excellent opportunities for enhanced circuit performance and higher levels of integration. However, the adverse effects of scaling are higher power consumption and increased design complexity. With the growing popularity of battery-driven, hand-held devices as well as the push for high-performance processors, it has now been established that power optimization is central to all chip designs. The increasing gap between battery technology and integrated circuit design, as well as the consumer demand for enhanced performance and functionality for mobile devices has made power the principle design constraint. Moreover, due to higher device and interconnect density, increase in power density is also an important concern for future technologies. Local hotspots due to large temperature gradients within the die lead to degradation in the reliability and performance of the SoC. Thus, power reduction is now at the forefront in the design of high performance integrated circuits. Low power and, indeed, low energy design methodologies are increasingly becoming the primary focus for current and future generation VLSI chips.

There are several power reduction techniques that can be applied at different stages of the system-on-chip design. Proper design choices early in the design phase usually have a more significant impact on the different design goals than those made late in the design process. However, in fact, the most design methodologies now incorporate suitable techniques at every phase of the design process. Ideally, optimization achieved at any stage of the design process should not be offset by changes made in the latter stages of the design process. Realization of such low-power design methodologies increases the design complexity of chips. In this

dissertation, a comprehensive methodology for power reduction has been proposed for voltage island-based SoC design.

This work proposes a floorplan methodology for voltage islands in SoCs that selectively emphasizes chip power reduction and overall chip area reduction. Using this new methodology, the solutions obtained in this research are far better than preexisting techniques in terms of power and area as well as reducing the amount time invested for floorplanning thereby reducing the total design cycle. Reducing design time and focusing on the numerous floorplanning design challenges that are critical to successful System-on-Chip design are fundamental contributions in this thesis. Due to the increased number of cores, the floorplanning time has increased considerably and is projected to get worse. The new approach significantly reduces the number of floorplan iterations. The final floorplan solution satisfies all design constraints, meets area budget and reflects all the power optimizations performed at the earlier design phases.

6.1. Research Summary and Contributions

Multiple supply and threshold voltage techniques are widely used for low power SoC design. By taking a holistic approach to this problem, the goal is to reduce the overall power consumption of the SoC. The main focus of this thesis is to address the different design challenges of multiple supply and threshold voltages early in the design process. At first, the application is modeled as a Power State Machine (PSM). Such a representation portrays a true picture of the working of the different components of the SoC while executing the application. *To the best of the author's knowledge, this work for the first time proposes a design*

methodology that begins with the application PSM to be executed and ends at the floorplan stage of the SoC.

A new method of extracting the activity of every core in the SoC from the PSM has been proposed. Next, the activity of the cores is used as a guiding factor to decide the different threshold voltages. The availability of three different threshold voltages, namely HVT, LVT and SVT, has been assumed in this work. Cores having high activity are assigned to LVT while cores with smaller activity use HVT cells. During this design phase, it is assumed that the critical paths of the cores exist from the input to output of the cores only and do not span across two or more cores. If the minimum supply voltage satisfying the delay requirements is less/more than the minimum/maximum allowable V_{DD} for a given technology, then SVT cells are used instead. Additionally, the power consumed by each core to execute in the application is calculated and used in the latter stages of the design. Compared to earlier designs with single V_T, the average power savings of about 20% can be achieved using a different V_T for each core and using power gating technique for shutting of idle cores.

Supply voltage selection is performed after the threshold voltage for each core has been established. The minimum possible V_{DD} for every core is determined by the delay constraint and the maximum V_{DD} is the chip-level supply voltage. Any voltage between these two values is legal for the given core. This work further investigated this range to identify a few other interesting supply voltages. The minimum V_{DD} is preferred if power optimization is the principle objective. The overall goal is the reduction of the energy or energy-delay product, or any other power-delay metric preferred by the designer. Based on the designer's optimization goals, the important supply voltages are identified for every core in the SoC. This thesis uses

the minimum energy point as the preferred V_{DD} for each core, along with the lowest and the highest possible V_{DD}, as the possible set of supply voltages Therefore, as many as three values of V_{DD} may be associated with each core, but only one voltage will ultimately be selected for the core.

If each core in the SoC is allowed to have it's preferred V_{DD}, then the required number of supply voltages may be quite large. Typically, the number of possible V_{DD}'s on the SoC does not exceed over three or four distinct supply voltages in designs today. This constraint is due to the added complexity in the generation and distribution of the power supplies, as well as verification of the entire SoC. This work proposed a novel algorithm for voltage selection. Given the large set of preferred V_{DD}'s, the algorithm identifies the supply voltages that should be implemented on the chip. Two different cost functions have been used in the selection process. Depending upon the design constraint, the cost function can be entirely power-driven or connectivity-driven, or some combination of the two. Reduction in overall power consumption of the SoC is the focus for power-driven cost function.

In the case of the connectivity-driven function, the focus is on area and other floorplan constraints that may require some cores be placed at specific location on the chip. Results show that overly-constrained floorplan requirements can easily offset the potential power savings achievable during voltage selection. Moreover, proper voltage selection affects the final floorplan solution and thereby affects the final power and area savings as well. Thus, this work considers such constraints early in the design process. The proposed voltage selection technique described here is on average 4% better than the pre-existing methodology for voltage selection.

Once the V_{DD}'s to be implemented on the chip have been identified, the next task is to attempt floorplanning of the cores. For simple chips with a single supply voltage, if the available white-space is insufficient for decap allocation, then the cores are manually shifted to create the extra white-space. However, floorplanning of chips with multiple supply voltages is far more complex. Due to the power grid design challenges, cores with similar supply voltage are placed contiguous to one another. This leads to the formation of voltage islands which have the same power supply value.

For voltage island design, floorplanning is done in two stages. First, the islands are floorplanned individually and then the final chip is floorplanned considering each island as a macro block. The area overhead for floorplanning is due to decaps, level shifters, isolation cells and power gating cells. The chip area reserved for level shifters, isolation cells and power gating cells can be computed in advance. The decap area depends upon the power grid noise as well as the available white-space after the floorplanning of the cores. Identifying extra white-space becomes a difficult task in such island-based floorplans. Moreover, as opposed to the chips with single V_{DD}, the white-space at the boundary of two islands cannot be shared. Thus, the white-space allocation becomes a non-trivial task.

To ensure that such difficulties do not arise, a 1-D power grid design methodology has been proposed to calculate the amount of decap required by each island. The extra white-space for decaps is compared to the amount of white-space available after floorplanning of each island and the area of the islands are adjusted accordingly. If the total area overhead required for decap allocation, level shifters, power gating cells violates the area budget, then chip-level floorplanning of cores would be a useless task. In effect, it constitutes a wasted iteration and

increases the design time. The proposed technique ensures that only valid solutions are attempted for floorplanning by checking for such violations and removing them from further consideration.

Finally, the voltage assignment of the cores to the different islands is performed in two stages. Optimization for the high power cores, identified from the PSM, is solely based on power. For the rest of the cores, voltage assignment is performed from high to low power cores in a sorted fashion using a cost function that progressively gives higher importance to connectivity over power. The rationale behind this two-stage approach is to focus on power optimization for the subset of cores which have maximum impact on the total power consumption of the chip and focus on connectivity-based island formation for the rest of the cores in order to satisfy the needs of the floorplanner constraints. At first, voltage assignments that consume least power are attempted for floorplanning and, if the final floorplan solution violates the area budget, then the next best voltage assignment of the cores is used with a commensurate, but incremental, increase in power. The first floorplan solution that satisfies the area budget constitutes the final solution for the SoC. This proposed approach leads to a further 10% reduction in power, 9% reduction in area and over 2X reduction in floorplan time compared to previously reported results in the literature.

6.2. Research Limitations

The entire design flow requires designers to specify a number of design parameters at different stages of the design process which is typical of all design flows. Parameters for mapping applications to PSM, the states in which each core can exist, the number of possible V_{DD} and V_T values that can be implemented, source and distribution methodology of V_{DD}, and the

overall power gating methodology are some of these parameters. Due to the complexity of exploring all such parameters only a handful of them have been included in the analysis. To understand the entire design scope, all such parameters should be analyzed separately and thoroughly to have a better understanding on their individual effects on power, area and timing. Another issue is that, due to the lack of appropriate industrial benchmark circuits to test the effectiveness of the approach, synthetic benchmark circuits were created and used in this research. Though it is common practice to use such methods in an academic setting, it is clear that implementing these solutions on chips in the industry would give more insight into the practicality of the optimization techniques.

Assumptions used in this work related to timing constraints are not always valid in all designs. For example, it is assumed that the critical path can be determined by examining the paths within each core, since their I/O's are registered; however, the critical path of many designs can span over two or more IP blocks. Then, the timing constraint used for each IP block would become difficult to predict due to the dependence on other cores. Additionally, the clock design issues relating to power and area have been ignored. The power consumed for clocking individual core is assumed to be included within the power consumption of each core. Clock distribution across different islands must be investigated as well to determine if there is an additional impact on the overall design.

The source of the different V_{DD}'s and the mechanisms of power gating signal generation have been ignored as well. The different V_{DD}'s may be generated on chip or be supplied externally. The power management system that is expected to turn different cores ON/OFF based on their operating state has been considered to be readily available. Such design complexities should

be included in the refinements to the design flow as they may potentially affect the different design decisions.

6.3. Future Work

During the latter stages of the work described here, new directions for further research and new ideas have surfaced that would expand the scope and impact of the work. Some of these ideas that represent as possible future work are discussed in this section:

1) Multi-Clock Design: As the clock tree consumes significant power, the clock design should be done in a judicious manner. Due to different timing requirements, voltage island designs inherently require multiple clocks. Thus, in addition to voltage islands, use of frequency islands is a common feature in low-power chips. Research in this direction should investigate the challenges in clock design in multi-V_{DD} SoC and address the multiple clock tree design technique, clock skew management and inter-clock domain communication infrastructures.

2) 1-D vs. 2-D Power Grid Analysis: The decap allocation problem was addressed using a 1-D power grid model. However, the actual design is a 2-D structure. The assumption here is that, once the floorplan has been completed, a 2-D power grid verification step is performed to ensure that the noise budgets are met. However, on average, the predicted amount of decap using the 1-D tool would be pessimistic in comparison to the 2-D tool. In order to have the 1-D tool provide a more accurate solution, the noise metric needs to be calibrated to a 2-D tool and adjusted accordingly so that excessive over design does not occur for the decap requirements. Also, the 2-D current profile of the core could be mapped in an appropriate manner to the 1-D model.

3) Dynamic vs. Static Voltage Control: Dynamic control of supply and threshold voltage is also a useful technique for power reduction. The complexity of dynamic control is far more than static control. The benefits of dynamic control as opposed to static control of voltage islands need to be investigated. Additionally, the possibility of having a mixture of dynamic and static control of different sets of cores on the same chip must be explored. By using such a technique, few tasks can be assigned to static voltage islands while others can be executed on cores whose supply voltage can be tuned to the changing performance requirements. This opens up an interesting array of new topics to pursue as future work.

4) Power Management Verification: As different low-power techniques are incorporated into more and more designs, designers are left wondering how to verify the effectiveness of these power saving strategies. Some combinations of low-power design options could inadvertently produce errors that are bound to the typical logic failures. Due to verification challenges of such errors, aggressive implementation of low-power techniques at the micro level of design still remains to be realized. This work would effectively merge two computer-aided design methodologies in VLSI by building novel formal techniques that perform global and local verification of low-power techniques. This would not only reduce verification time and increase design confidence but also would create new opportunities for further aggressive power reduction in future nanoarchitectures.

REFERENCES

[1] H. Jingcao, S. Youngsoo, N. Dhanwada, and R. Marculescu, "Architecting voltage islands in core-based system-on-a-chip designs," *Proceedings of International Symposium on Low Power Electronics and Design*, 2004, pp. 180-185.

[2] "International Technology Roadmap for Semiconductors," *http://www.itrs.net*, 2007.

[3] S. Borkar, "Design challenges of technology scaling," *IEEE Micro*, vol. 19, 1999, pp. 23-29.

[4] J. Srinivasan, S. Adve, P. Bose, and J. Rivers, "The impact of technology scaling on lifetime reliability," *Proceeding of IEEE International Conference on Dependable Systems and Networks*, 2004, pp. 161-170.

[5] R. Mahajan, C. Chia-pin, and G. Chrysler, "Cooling a Microprocessor Chip," *Proceedings of the IEEE*, vol. 94, 2006, pp. 1476-1486.

[6] K. Lahiri, A. Raghunathan, S. Dey, and D. Panigrahi, "Battery-driven system design: a new frontier in low power design," *Proceedings of Asia and South Pacific Design Automation Conference*, 2002, pp. 261-267.

[7] M. Keating, D. Flynn, R. Aitken, A. Gibbons, and K. Shi, *Low Power Methodology Manual*, New York, USA: Springer-Verlag, 2007.

[8] A. Bellaouar and M.I. Elmasry, *Low-power digital VLSI design: Circuits and Systems*, Springer, 1995.

[9] M. Srivastav, S. Rao, and H. Bhatnagar, "Power Reduction Technique Using Multi-Vt Libraries," *Proceedings of IEEE International Workshop on System-on-Chip for Real-Time Applications*, 2005, pp. 363-367.

[10] G. Tellez, A. Farrahi, and M. Sarrafzadeh, "Activity-driven clock design for low power circuits," *Proceedings of IEEE International Conference on Computer Aided Design*, 1995, pp. 62-65.

[11] T. Kobayashi and T. Sakurai, "Self-adjusting threshold-voltage scheme (SATS) for low-voltage high-speed operation," *Proceedings of IEEE Custom Integrated Circuits Conference*, 1994, pp. 271-274.

[12] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, and J. Yamada, "1V high-speed digital circuit technology with 0.5μm multi-threshold CMOS," *Proceedings of International ASIC Conference and Exhibit*, 1993, pp. 186-189.

[13] D. Lackey, P. Zuchowski, T. Bednar, D. Stout, S. Gould, and J. Cohn, "Managing power and performance for system-on-chip designs using Voltage Islands,"

Proceedings of IEEE International Conference on Computer Aided Design, 2002, pp. 195-202.

[14] C. Kihwan, R. Soma, and M. Pedram, "Dynamic Voltage and Frequency Scaling based on Workload Decomposition," *Proceedings of International Symposium on Low Power Electronics and Design*, 2004, pp. 174-179.

[15] Cadence Design Systems, "Architecting, Designing, Implementing, and Verifying Low-Power Digital Integrated Circuits," 2007.

[16] R. Puri, D. Kung, and L. Stok, "Minimizing Power with Flexible Voltage Islands," *Proceedings of International Symposium on Circuits and Systems*, IEEE, 2005, pp. 21-24.

[17] M. Popovich, E.G. Friedman, M. Sotman, and A. Kolodny, "On-Chip Power Distribution Grids With Multiple Supply Voltages for High-Performance Integrated Circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, 2008, pp. 908-921.

[18] B. Chatterjee, M. Sachdev, S. Hsu, R. Krishnamurthy, and S. Borkar, "Effectiveness and scaling trends of leakage control techniques for sub-130 nm CMOS technologies," *Proceedings of International Symposium on Low Power Electronics and Design*, 2003, pp. 122-127.

[19] S. Sapatnekar and H. Su, "Analysis and optimization of power grids," *IEEE Design & Test*, vol. 20, 2003, pp. 7-15.

[20] D. Sengupta and R. Saleh, "Constraint-based voltage island partitioning," *Proceedings of IEEE Northeast Workshop on Circuits and Systems*, 2007, pp. 1050-1053.

[21] W. Hung, G. Link, N. Vijaykrishnan, N. Dhanwadaf, and J. Conner, "Temperature-aware voltage islands architecting in system-on-chip design," *Proceedings of IEEE International Conference on Computer Design*, 2005, pp. 689-694.

[22] P. Wielage and K. Goossens, "Networks on silicon: blessing or nightmare?," *Proceedings of Euromicro Symposium on Digital System Design. Architectures, Methods and Tools*, 2002, pp. 196-200.

[23] C. Chin-Chih, J. Cong, and X. Min, "Optimality and scalability study of existing placement algorithms," *Proceedings of IEEE Asia and South Pacific Design Automation Conference*, 2003, pp. 621-627.

[24] E. Young, C. Chu, and M. Ho, "Placement Constraints in Floorplan Design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, 2004, pp. 735-745.

[25] D. Hodges, J. Horace, and R. Saleh, *Analysis and Design of Digital Integrated Circuits*, McGraw-Hill Science/Engineering/Math, 2003.

[26] S. Turgis, N. Azemard, and D. Auvergne, "Explicit evaluation of short circuit power dissipation for CMOS logic structures," *Proceedings of International Symposium on Low Power Electronics and Design*, 1995, pp. 129-134.

[27] D. Markovic, V. Stojanovic, B. Nikolic, M. Horowitz, and R. Brodersen, "Methods for true energy-performance optimization," *IEEE Journal of Solid-State Circuits*, vol. 39, 2004, pp. 1282-1293.

[28] V. Zyuban and P.N. Strenski, "Balancing hardware intensity in microprocessor pipelines," *IBM Journal of Research and Development*, vol. 47, 2003.

[29] H. Veendrick, "Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits," *IEEE Journal of Solid-State Circuits*, vol. 19, 1984, pp. 468-473.

[30] K. Nose and T. Sakurai, "Analysis and future trend of short-circuit power," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, 2000, pp. 1023-1030.

[31] B. Nikolic, J.M. Rabaey, and A.P. Chandrakasan, *Digital integrated circuits: A design perspective*, Pearson Education, 2003.

[32] A. Ghosh, S. Devadas, K. Keutzer, and J. White, "Estimation of average switching activity in combinational and sequential circuits," *Proceedings of ACM/IEEE Design Automation Conference*, 1992, pp. 253-259.

[33] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proceedings of the IEEE*, vol. 91, 2003, pp. 305-327.

[34] C. Piguet, *Low-power Electronics Design*, CRC Press, 2004.

[35] K. Roy, "Leakage power reduction in low-voltage CMOS designs," *Proceedings of International Conference on Electronics, Circuits and Systems*, 1998, pp. 167-173.

[36] Y. Bin, W. Haihong, C. Riccobene, X. Qi, and L. Ming-Ren, "Limits of gate-oxide scaling in nano-transistors," *Proceedings of IEEE Symposium on VLSI Technology*, 2000, pp. 90-91.

[37] W. Liqiong, K. Roy, and V. De, "Low voltage low power CMOS design techniques for deep submicron ICs," *Proceedings of International Conference on VLSI Design*, 2000, pp. 24-29.

[38] A. Agarwal, S. Mukhopadhyay, C. Neau, R. Cakici, C. Kim, and K. Roy, "Gate leakage reduction for scaled devices using transistor stacking," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 11, 2003, pp. 716-730.

[39] L. Dongwoo, D. Blaauw, and D. Sylvester, "Gate Oxide Leakage Current Analysis and Reduction for VLSI Circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, 2004, pp. 155-166.

[40] M. Pedram, "Power minimization in IC design: principles and applications," *Proceedings of ACM Transactions on Design Automation of Electronic Systems*, 1996, pp. 3-56.

[41] L. Benini and G. De Micheli, "Automatic synthesis of low-power gated-clock finite-state machines," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, 1996, pp. 630-643.

[42] M. Alidina, J. Monteiro, S. Devadas, a. Ghosh, and M. Papaefthymiou, "Precomputation-based sequential logic optimization for low power," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 2, 1994, pp. 426-436.

[43] L. Benini, G.D. Micheli, E. Macii, M. Poncino, and R. Scarsi, "Symbolic synthesis of clock-gating logic for power optimization of control-oriented synchronous networks," *Proceedings of IEEE European Design and Test Conference*, 1997, pp. 514-520.

[44] A. Correale, "Overview of the power minimization techniques employed in the IBM PowerPC 4xx embedded controllers," *Proceedings of International Symposium on Low Power Electronics and Design*, 1995, pp. 75-80.

[45] V. Tiwari, S. Malik, and P. Ashar, "Guarded evaluation: pushing power management to logic synthesis/design," *Proceedings of International Symposium on Low Power Electronics and Design*, 1995, pp. 221-226.

[46] M. Munch, B. Wurth, R. Mehra, J. Sproch, and N. Wehn, "Automating RT-level operand isolation to minimize power consumption in datapaths," *Proceedings of Design, Automation and Test in Europe Conference and Exhibition*, 2000, pp. 624-631.

[47] T. Chi-Ying, M. Pedram, and A. Despain, "Power efficient technology decomposition and mapping under an extended power consumption model," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 13, 1994, pp. 1110-1122.

[48] V.D. Agrawal, M.L. Bushnell, G. Parthasarathy, and R. Ramadoss, "Digital Circuit Design for Minimum Transient Energy and a Linear Programming Method," *Proceedings of International Conference on VLSI Design*, 1999, pp. 434-439.

[49] S. Kim, J. Kim, and S. Hwang, "New path balancing algorithm for glitch power reduction," *IEE Proceedings - Circuits, Devices and Systems*, vol. 148, 2001, pp. 151 - 156.

[50] T. Raja, V.D. Agrawal, and M.L. Bushnell, "Minimum Dynamic Power CMOS Circuit Design by a Reduced Constraint Set Linear Program," *Proceedings of International Conference on VLSI Design*, 2003, pp. 527-532.

[51] V. Agrawal, "Low-power design by hazard filtering," *Proceedings of International Conference on VLSI Design*, 1997, pp. 193-197.

[52] A. Srivastava, D. Sylvester, and D. Blaauw, "Power minimization using simultaneous gate sizing, dual-Vdd and dual-Vth assignment," *Proceedings of ACM/IEEE Design Automation Conference*, 2004, pp. 783-787.

[53] C. Chunhong and M. Sarrafzadeh, "Simultaneous voltage scaling and gate sizing for low-power design," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 49, 2002, pp. 400-408.

[54] J. Singh, V. Nookala, Z. Luo, and S. Sapatnekar, "Robust gate sizing by geometric programming," *Proceedings of ACM/IEEE Design Automation Conference*, 2005, pp. 315-320.

[55] A. Murugavel and N. Ranganathan, "A microeconomic model for simultaneous gate sizing and voltage scaling for power optimization," *Proceedings of International Conference on Computer Design*, 2003, pp. 276-281.

[56] A. Murugavel and N. Ranganathan, "Gate sizing and buffer insertion using economic models for power optimization," *Proceedings of International Conference on VLSI Design*, 2004, pp. 195-200.

[57] O. Coudert, "Gate sizing for constrained delay/power/area optimization," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 5, 1997, pp. 465-472.

[58] O. Coudert, R. Haddad, and S. Manne, "New algorithms for gate sizing: a comparative study," *Proceedings of ACM/IEEE Design Automation Conference*, 1996, pp. 734-739.

[59] M. Berkelaar and J. Jess, "Gate sizing in MOS digital circuits with linear programming," *Proceedings of the European Design Automation Conference*, 1990, pp. 217-221.

[60] V. Mahalingam and N. Ranganathan, "A Nonlinear Programming Based Power Optimization Methodology for Gate Sizing and Voltage Selection," *Proceedings of IEEE Computer Society Annual Symposium on VLSI*, 2005, pp. 180-185.

[61] M. Borah, R. Owens, and M. Irwin, "Transistor sizing for low power CMOS circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, 1996, pp. 665-671.

[62] W.H. Kao, N. Fathi, and C. Lee, "Algorithms for automatic transistor sizing in CMOS digital circuits," *Proceedings of ACM/IEEE Design Automation Conference*, 1985, pp. 781-784.

[63] J. Shyu, A. Sangiovanni-Vincentelli, J. Fishburn, and A. Dunlop, "Optimization-based transistor sizing," *IEEE Journal of Solid-State Circuits*, vol. 23, 1988, pp. 400-409.

[64] C.H. Wu, N.V. Zanden, and D. Gajski, "A new algorithm for transistor sizing in CMOS circuits," *Proceedings of IEEE European Design Automation Conference*, 1990, pp. 589-593.

[65] G.K. Yeap, *Practical Low Power Digital VLSI Design*, Springer, 1997.

[66] K. Flautner, D. Flynn, D. Roberts, and D. Patel, "IEM926: an energy efficient SoC with dynamic voltage scaling," *Proceedings of IEEE Design, Automation and Test in Europe Conference*, 2004, pp. 324-327.

[67] J.M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits (2nd Edition)*, Prentice Hall, 2003.

[68] K. Flautner, D. Flynn, and M. Rives, "A Combined Hardware-Software Approach for Low-Power SoCs: Applying Adaptive Voltage Scaling and Intelligent Energy Management Software," *Proceedings of System-on-Chip and ASIC Design Conference*, 2003.

[69] C. Chen, A. Srivastava, and M. Sanafzadeh, "On gate level power optimization using dual-supply voltages," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 9, 2001, pp. 616-629.

[70] B. Amelifard, A. Afzali-Kusha, and A. Khadernzadeh, "Enhancing the Efficiency of Cluster Voltage Scaling Technique for Low-power Application," *Proceedings of IEEE International Symposium on Circuits and Systems*, 2005, pp. 1666-1669.

[71] K. Usami and M. Horowitz, "Clustered voltage scaling technique for low-power design," *Proceedings of International Symposium on Low Power Electronics and Design*, 1995, pp. 3-8.

[72] K. Usami, K. Nogami, M. Igarashi, F. Minami, Y. Kawasaki, T. Ishikawa, M. Kanazawa, T. Aoki, M. Takano, C. Mizuno, M. Ichida, S. Sonoda, M. Takahashi, and N. Hatanaka, "Automated low-power technique exploiting multiple supply voltages applied to a media processor," *Proceedings of IEEE Custom Integrated Circuits Conference*, 1997, pp. 131-134.

[73] Q. Ma and E.F. Young, "Network flow-based power optimization under timing constraints in MSV-driven floorplanning," *Proceedings of IEEE International Conference on Computer Aided Design*, 2008, pp. 1-8.

[74] L. Wan-Ping, L. Hung-Yi, and C. Yap-Wen, "Voltage Island Aware Floorplanning for Power and Timing Optimization," *Proceedings of IEEE International Conference on Computer Aided Design*, 2006, pp. 389-394.

[75] Q. Ma and E.F. Young, "Voltage island-driven floorplanning," *Proceedings of IEEE International Conference on Computer Aided Design*, 2007, pp. 644-649.

[76] L. Bin, C. Yici, Z. Qiang, and H. Xianlong, "Power driven placement with layout aware supply voltage assignment for voltage island generation in dual-Vdd designs,"

Proceedings of Asia and South Pacific Conference on Design Automation, 2006, pp. 582-587.

[77] L. Bin, C. Yici, Z. Qiang, and H. Xianlong, "Power driven placement with layout aware supply voltage assignment for voltage island generation in dual-Vdd designs," *Proceedings of Asia and South Pacific Conference on Design Automation*, 2006, pp. 582-587.

[78] R.L. Ching, E.F. Young, K.C. Leung, and C. Chu, "Post-placement voltage island generation," *Proceedings of ACM/IEEE International Conference on Computer Aided Design*, 2006, pp. 641-646.

[79] W. Mak and J. Chen, "Voltage Island Generation under Performance Requirement for SoC Designs," *Proceedings of Asia and South Pacific Design Automation Conference*, 2007, pp. 798-803.

[80] H. Wu, Y. Wang, and M. Wong, "Post-placement voltage island generation under performance requirement," *Proceedings of IEEE International Conference on Computer Aided Design*, 2005, pp. 309-316.

[81] W. Huaizhi, M. Wona, and L. I-Min, "Timing-constrained and voltage-island-aware voltage assignment," *Proceedings of ACM/IEEE Design Automation Conference*, 2006, pp. 429-432.

[82] S. Mukhopadhyay and K. Roy, "Accurate modeling of transistor stacks to effectively reduce total standby leakage in nano-scale CMOS circuits," *Proceedings of Symposium on VLSI Circuits*, 2003, pp. 53-56.

[83] N. Hanchate and N. Ranganathan, "A new technique for leakage reduction in CMOS circuits using self-controlled stacked transistors," *Proceedings of International Conference on VLSI Design*, 2004, pp. 228-233.

[84] M. Johnson, D. Somasekhar, C. Lih-Yih, and K. Roy, "Leakage control with efficient use of transistor stacks in single threshold CMOS," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 10, 2002, pp. 1-5.

[85] L. Yong and G. Zhiqiang, "Timing analysis of transistor stack for leakage power saving," *Proceedings of International Conference on Electronics, Circuits and Systems*, 2002, pp. 41-44.

[86] A. Abdollahi, F. Fallah, and M. Pedram, "Leakage current reduction in CMOS VLSI circuits by input vector control," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, 2004, pp. 140-154.

[87] C. Xiaotao, F. Dongrui, H. Yinhe, Z. Zhimin, and L. Xiaowei, "Fast Algorithm for Leakage Power Reduction by Input Vector Control," *Proceedings of International Conference on ASIC*, 2005, pp. 98-101.

[88] F. Gao and J.P. Hayes, "Exact and heuristic approaches to input vector control for leakage power reduction," *Proceedings of IEEE International Conference on Computer Aided Design*, 2004, pp. 527-532.

[89] L. Yuan and G. Qu, "A combined gate replacement and input vector control approach for leakage current reduction," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, 2006, pp. 173-182.

[90] S. Naidu and E. Jacobs, "Minimizing stand-by leakage power in static CMOS circuits," *Proceedings of Design, Automation, and Test in Europe*, 2001, pp. 370-376.

[91] L. Wei, Z. Chen, K. Roy, M. Johnson, Y. Ye, and V. De, "Design and optimization of dual-threshold circuits for low-voltage low-power applications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 7, 1999, pp. 16-24.

[92] M. Ketkar and S.S. Sapatnekar, "Standby power optimization via transistor sizing and dual threshold voltage assignment," *Proceedings of IEEE International Conference on Computer Aided Design*, 2002, pp. 375-378.

[93] P. Pant, R.K. Roy, and A. Chatterjee, "Dual-threshold voltage assignment with transistor sizing for low power CMOS circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 9, 2001, pp. 390-394.

[94] Q. Wang and S.B. Vrudhula, "Static power optimization of deep submicron CMOS circuits for dual VT technology," *Proceedings of IEEE International Conference on Computer Aided Design*, 1998, pp. 490-496.

[95] D. Nguyen, A. Davare, M. Orshansky, D. Chinnery, B. Thompson, and K. Keutzer, "Minimization of dynamic and static power through joint assignment of threshold voltages and sizing optimization," *Proceedings of International Symposium on Low Power Electronics and Design*, 2003, pp. 158-163.

[96] F. Gao and J.P. Hayes, "Total power reduction in CMOS circuits via gate sizing and multiple threshold voltages," *Proceedings of ACM/IEEE Design Automation Conference*, 2005, pp. 31-36.

[97] S. Sirichotiyakul, T. Edwards, C. Oh, J. Zuo, A. Dharchoudhury, R. Panda, and D. Blaauw, "Stand-by power minimization through simultaneous threshold voltage selection and circuit sizing," *Proceedings of ACM/IEEE Design Automation Conference*, 1999, pp. 436-441.

[98] N. Sirisantana, L. Wei, and K. Roy, "High-performance low-power CMOS circuits using multiple channel length and multiple oxide thickness," *Proceedings of International Conference on Computer Design*, 2000, pp. 227-232.

[99] T. Kuroda, T. Fujita, S. Mita, T. Nagamatsu, S. Yoshioka, K. Suzuki, F. Sano, M. Norishima, M. Murota, M. Kako, M. Kinugawa, M. Kakumu, and T. Sakurai, "A 0.9-V, 150-MHz, 10-mW, 4 mm2, 2-D discrete cosine transform core processor with

variable threshold-voltage (VT) scheme," *IEEE Journal of Solid-State Circuits*, vol. 31, 1996, pp. 1770-1779.

[100] S.M. Martin, K. Flautner, T. Mudge, and D. Blaauw, "Combined dynamic voltage scaling and adaptive body biasing for lower power microprocessors under dynamic workloads," *Proceedings of IEEE International Conference on Computer Aided Design*, 2002, pp. 721-725.

[101] H. Ananthan, C.H. Kim, and K. Roy, "Larger-than-Vdd forward body bias in sub-0.5V nanoscale CMOS," *Proceedings of International Symposium on Low Power Electronics and Design*, 2004, pp. 8-13.

[102] K. von Arnim, E. Borinski, P. Seegebrecht, H. Fiedler, R. Brederlow, R. Thewes, J. Berthold, and C. Pacha, "Efficiency of body biasing in 90-nm CMOS for low-power digital circuits," *IEEE Journal of Solid-State Circuits*, vol. 40, 2005, pp. 1549-1556.

[103] W. Elgharbawy, P. Golconda, A. Kumar, and M. Bayoumi, "A New Gate-Level Body Biasing Technique for PMOS Transistors in Subthreshold CMOS Circuits," *Proceedings of International Symposium on Circuits and Systems*, 2005, pp. 4697-4700.

[104] L. Xiaomei and S. Mourad, "Performance of submicron CMOS devices and gates with substrate biasing," *Proceedings of IEEE International Symposium on Circuits and Systems*, 2000, pp. 9-12.

[105] M. Sumita, S. Sakiyama, M. Kinoshita, Y. Araki, Y. Ikeda, and K. Fukuoka, "Mixed body-bias techniques with fixed Vt and Ids generation circuits," *Proceedings of International Conference on Integrated Circuit Design and Technology*, 2005, pp. 233-234.

[106] C. Kim and K. Roy, "Dynamic VTH Scaling Scheme for Active Leakage Power Reduction," *Proceedings of Conference on Design, Automation, and Test in Europe*, 2002, pp. 163-167.

[107] A. Keshavarzi, S. Ma, S. Narendra, B. Bloechel, K. Mistry, T. Ghani, S. Borkar, and V. De, "Effectiveness of reverse body bias for leakage control in scaled dual Vt CMOS ICs," *Proceedings of International Symposium on Low Power Electronics and Design*, 2001, pp. 207-212.

[108] H. Mizuno, K. Ishibashi, T. Shimura, T. Hattori, S. Narita, K. Shiozawa, S. Ikeda, and K. Uchiyama, "An 18- uA Standby Current 1.8-V, 200-MHz Microprocessor with Self-Substrate-Biased Data-Retention Mode," *IEEE Journal of Solid-State Circuits*, vol. 34, 1999, pp. 1492-1500.

[109] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, and J. Yamada, "1-V power supply high-speed digital circuit technology with multithreshold-voltage CMOS," *IEEE Journal of Solid-State Circuits*, vol. 30, 1995, pp. 847-854.

[110] B. Calhoun, F. Honore, and A. Chandrakasan, "A leakage reduction methodology for distributed MTCMOS," *IEEE Journal of Solid-State Circuits*, vol. 39, 2004, pp. 818-826.

[111] M. Anis, M. Mahmoud, and M. Elmasry, "Efficient gate clustering for MTCMOS circuits," *Proceedings of IEEE International ASIC/SOC Conference*, 2001, pp. 34-38.

[112] B. Calhoun, F. Honore, and A. Chandrakasan, "Design methodology for fine-grained leakage control in MTCMOS," *Proceedings of International Symposium on Low Power Electronics and Design*, 2003, pp. 104-109.

[113] W. Hyo-Sig, K. Kyo-Sun, J. Kwang-Ok, P. Ki-Tae, C. Kyu-Myung, and K. Jeong-Taek, "An MTCMOS design methodology and its application to mobile computing," *Proceedings of International Symposium on Low Power Electronics and Design*, 2003, pp. 110-115.

[114] J. Kao, A. Chandrakasan, and D. Antoniadis, "Transistor sizing issues and tool for multi-threshold CMOS technology," *Proceedings of ACM/IEEE Design Automation Conference*, 1997, pp. 409-414.

[115] N.A. Sherwani, *Algorithms for VLSI Physical Design Automation*, Springer, 1998.

[116] R. Otten, "Automatic Floorplan Design," *Proceedings of ACM/IEEE Design Automation Conference*, 1982, pp. 261-267.

[117] R. Otten, "Efficient floorplan optimization," *Proceedings of IEEE International Conference on Computer Design*, 1983, pp. 499-502.

[118] D.F. Wong and C.L. Liu, "A new algorithm for floorplan design," *Proceedings of ACM/IEEE Design Automation Conference*, 1986, pp. 101-107.

[119] L. Wang, Y. Chang, and K. Cheng, *Electronic Design Automation: Synthesis, Verification, and Test*, Morgan Kaufmann, 2009.

[120] H. Xianlong, H. Gang, C. Yici, G. Jiangchun, D. Sheqin, C. Chung-Kuan, and G. Jun, "Corner block list: an effective and efficient topological representation of non-slicing floorplan," *Proceedings of IEEE International Conference on Computer Aided Design*, IEEE, 2000, pp. 8-12.

[121] K. Sakanushi, Y. Kajitani, and D. Mehta, "The quarter-state-sequence floorplan representation," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 50, 2003, pp. 376-386.

[122] E.F. Young, C.C. Chu, and C. Shen, "Twin binary sequences: a non-redundant representation for general non-slicing floorplan," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, 2003, pp. 457-469.

[123] G. Pei-Ning, C. Chung-Kuan, and T. Yoshimura, "An O-tree representation of non-slicing floorplan and its applications," *Proceedings of IEEE Design Automation Conference*, IEEE, 1999, pp. 268-273.

[124] Y. Chang, Y. Chang, G. Wu, and S. Wu, "B*-Trees: a new representation for non-slicing floorplans," *Proceedings of ACM/IEEE Design Automation Conference*, 2000, pp. 458-463.

[125] L. Jai-Ming, C. Yao-Wen, and L. Shih-Ping, "Corner sequence - A P-admissible floorplan representation with a worst case linear-time packing scheme," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 11, 2003, pp. 679-686.

[126] H. Murata and E.S. Kuh, "Sequence-pair based placement method for hard/soft/pre-placed modules," *Proceedings of IEEE International Symposium on Physical Design*, 1998, pp. 167-172.

[127] C. Song, H. Xianlong, D. Sheqin, M. Yuchun, C. Yici, C. Chung-Kuan, and G. Jun, "Evaluating a bounded slice-line grid assignment in O(nlogn) time," *Proceedings of International Symposium on Circuits and Systems*, 2003, pp. 708-711.

[128] L. Jai-Ming and C. Yao-Wen, "TCG: a transitive closure graph-based representation for non-slicing floorplans," *Proceedings of ACM/IEEE Design Automation Conference*, 2001.

[129] J. Lin and Y. Chang, "TCG-S: orthogonal coupling of P*-admissible representations for general floorplans," *Proceedings of ACM/IEEE Design Automation Conference*, 2002.

[130] H. Zhou and J. Wang, "ACG-Adjacent Constraint Graph for General Floorplans," *Proceedings of Internationation Conference on Computer Design*, 2004.

[131] C. Sechen and A. Sangiovanni-Vincentelli, "The TimberWolf placement and routing package," *IEEE Journal of Solid-State Circuits*, vol. 20, 1985, pp. 510-522.

[132] S. Sutanthavibul, E. Shragowitz, and J.B. Rosen, "An analytical approach to floorplan design and optimization," *Proceedings of ACM/IEEE Design Automation Conference*, 1991, pp. 187-192.

[133] M. Cho and D.Z. Pan, "BoxRouter: a new global router based on box expansion and progressive ILP," *Proceedings of ACM/IEEE Design Automation Conference*, 2006, pp. 373-378.

[134] F.M. Johannes, "Partitioning of VLSI circuits and systems," *Proceedings of ACM/IEEE Design Automation Conference*, 1996, pp. 83-87.

[135] M. Rebaudengo and M. Reorda, "GALLO: A genetic algorithm for floorplan area optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, 1996, pp. 943-951.

[136] W. Hung, Y. Xie, and N. Vijaykrishnan, "Thermal-aware floorplanning using genetic algorithms," *Proceedings of International Symposium on Quality of Electronic Design*, 2005, pp. 634-639.

[137] A. Dickinson, "Floyd: A knowledge-based floorplan designer," *Proceedings of IEEE International Conference on Computer Design*, 1986, pp. 176-179.

[138] H. Watanabe, "Flute an Expert Floor Planner for Full-Custom VLSI Design," *IEEE Design & Test*, vol. 4, 1987, pp. 32-41.

[139] M.A. Jabri and D.J. Skellern, "PIAF: a knowledge-based/algorithm top-down floorplanning system," *Proceedings of ACM/IEEE Design Automation Conference*, 1989, pp. 582-585.

[140] A.B. Kahng, "Classical floorplanning harmful?," *Proceedings of International Symposium on Physical Design*, 2000, pp. 207-213.

[141] T. Chen and Y. Chang, "Modern floorplanning based on fast simulated annealing," *Proceedings of International Symposium on Physical Design*, 2005, pp. 104-112.

[142] S. Adya and I. Markov, "Fixed-outline floorplanning: enabling hierarchical design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 11, 2003, pp. 1120-1135.

[143] C. Lin, D. Chen, and Y. Wang, "Robust fixed-outline floorplanning through evolutionary search," *Proceedings of Asia and South Pacific Conference on Design Automation*, 2004, pp. 42-44.

[144] J. Cong, M. Romesis, and J. Shinned, "Fast floorplanning by look-ahead enabled recursive bipartitioning," *Proceedings of Asia and South Pacific Design Automation Conference*, 2005, pp. 1119-1122.

[145] L. Hsun-Cheng, C. Yao-Wen, H. Jer-Ming, and H. Yang, "Multilevel floorplanning/placement for large-scale modules using B*-trees," *Proceedings of Design Automation Conference*, 2003, pp. 812-817.

[146] Y. Joon-Seo, B. Seong-Ok, and K. Chong-Min, "A floorplan-based planning methodology for power and clock distribution in ASICs [CMOS technology]," *Proceedings of ACM/IEEE Design Automation Conference*, 1999, pp. 766-771.

[147] L. Chen-Wei and C. Yao-Wen, "Power/Ground Network and Floorplan Cosynthesis for Fast Design Convergence," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, 2007, pp. 693-704.

[148] D. Sengupta and R.A. Saleh, "Application-driven voltage-island partitioning for low-power system-on-chip design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, 2009, p. 10.

[149] C.V. Ramamoorthy and M.J. Gonzalez, "Recognition and representation of parallel processable streams in computer programs-II (Task/process Parallelism)," *Proceedings of ACM Annual Conference/Annual Meeting*, 1969.

[150] S. Eike and S. Frank, "Towards Activity Based System Level Power Estimation," *Design and ReUse*, 2003.

[151] L.T. Clark, R. Patel, and T.S. Beatty, "Managing standby and active mode leakage power in deep sub-micron design," *Proceedings of International Symposium on Low Power Electronics and Design*, 2004, pp. 274-279.

[152] R.A. Bergamaschi and Y.W. Jiang, "State-based power analysis for systems-on-chip," *Proceedings of ACM/IEEE Design Automation Conference*, 2003, pp. 236-241.

[153] "Common Power Format," *http://www.si2.org/?page=766*.

[154] "Unified Power Format," *http://www.unifiedpowerformat.com/*.

[155] L. Benini, R. Hodgson, and P. Siegel, "System-level power estimation and optimization," *Proceedings of International Symposium on Low Power Electronics and Design*, 1998, pp. 173-178.

[156] A. Dansky and et al., "760 MHz G6 S/390 microprocessor exploiting multiple Vt and copper interconnects," *Proceedings of IEEE International Solid-State Circuits Conference*, 2000, pp. 96-97.

[157] J. Kao, S. Narendra, and A. Chandrakasan, "Subthreshold leakage modeling and reduction techniques," *Proceedings of IEEE International Conference on Computer Aided Design*, 2002, pp. 141-148.

[158] L. Clark, "Mapping statistical process variations toward circuit performance variability: an analytical modeling approach," *Proceedings of ACM/IEEE Design Automation Conference*, 2005, pp. 658-663.

[159] A. Agarwal, K. Kang, S. Bhunia, J.D. Gallagher, and K. Roy, "Device-Aware Yield-Centric Dual-Vt Design Under Parameter Variations in Nanoscale Technologies," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, 2007, pp. 660-671.

[160] A. Wang, A. Chandrakasan, and S. Kosonocky, "Optimal supply and threshold scaling for subthreshold CMOS circuits," *Proceedings of IEEE Annual Symposium on VLSI*, 2002, pp. 7-11.

[161] E. Macii, M. Pedram, and F. Somenzi, "High-level power modeling, estimation, and optimization," *Proceedings of ACM/IEEE Design Automation Conference*, 1997, pp. 504-511.

[162] R. Saleh, "Delay macromodeling and estimation for RTL," *Proceedings of International Symposium on Circuits and Systems*, IEEE, 2008, pp. 2430-2433.

[163] J. Chang and M. Pedram, "Energy minimization using multiple supply voltages," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 5, 1997, pp. 436-443.

[164] M. Hamada, Y. Ootaguro, and T. Kuroda, "Utilizing Surplus Timing for Power Reduction," *Proceedings of IEEE Conference on Custom Integrated Circuits*, 2001, pp. 89-92.

[165] R. Gonzalez, B. Gordon, and M. Horowitz, "Supply and threshold voltage scaling for low power CMOS," *IEEE Journal of Solid-State Circuits*, vol. 23, 1997, pp. 1210-1216.

[166] D. Sengupta and R. Saleh, "Generalized Power-Delay Metrics in Deep Submicron CMOS Designs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, pp. 183-189.

[167] H. Liu, W. Lee, and Y. Chang, "A Provably Good Approximation Algorithm for Power Optimization Using Multiple Supply Voltages," *Proceedings of Design Automation Conference*, IEEE, 2007, pp. 887-890.

[168] R. Dick, "TAPHS: thermal-aware unified physical-level and high-level synthesis," *Proceedings of IEEE Asia and South Pacific Design Automation Conference*, 2006, pp. 879-885.

[169] S. Majzoub, R. Saleh, S.J. Wilton, and R. Ward, "Removal-Cost Method: An Efficient Voltage Selection Algorithm for Multi-Core Platforms under PVT," *Proceedings of IEEE System-on-Chip Conference*, 2009.

[170] K. Roy, "Decoupling capacitance allocation and its application to power-supply noise-aware floorplanning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, 2002, pp. 81-92.

[171] H. Chen and D. Ling, "Power Supply Noise Analysis Methodology For Deep-submicron Vlsi Chip Design," *Proceedings of ACM/IEEE Design Automation Conference*, 1997, pp. 638 - 643.

[172] D. Sengupta and R. Saleh, "Supply voltage selection in Voltage Island based SoC design," *Proceedings of IEEE International SOC Conference*, 2008, pp. 219-222.

[173] S. Kulkarni and D. Sylvester, "Power distribution techniques for dual vdd circuits," *Proceedings of Asia and South Pacific Conference on Design Automation*, 2006, pp. 838-843.

[174] Y. Cai, B. Liu, J. Shi, Q. Zhou, and X. Hong, "Power Delivery Aware Floorplanning for Voltage Island Designs," *Proceedings of International Symposium on Quality of Electronic Design*, 2007, pp. 350-355.

[175] J. Yi-Min, C. Kwang-Ting, and D. An-Chang, "Estimation of maximum power supply noise for deep sub-micron designs," *Proceedings of International Symposium on Low Power Electronics and Design*, 1998, pp. 233-238.

[176] H. Chen and J. Neely, "Interconnect and circuit modeling techniques for full-chip power supply noise analysis," *IEEE Transactions on Components, Packaging, and Manufacturing Technology: Part B*, vol. 21, 1998, pp. 209-215.

[177] S. Zhao and K. Roy, "Frequency domain analysis of switching noise on power supply network," *Proceedings of IEEE International Conference on Computer Aided Design*, 2000, pp. 487-492.

[178] N. Chang, "Challenges in power-ground integrity," *Proceedings of ACM/IEEE International Conference on Computer Aided Design*, 2001, pp. 651-654.

[179] K. Arabi, R. Saleh, and X. Meng, "Power Supply Noise in SoCs: Metrics, Management, and Measurement," *IEEE Design & Test*, 2007, pp. 236-244.

[180] S. Pant, E. Chiprout, and D. Blaauw, "Power Grid Physics and Implications for CAD," *IEEE Design & Test*, vol. 24, 2007, pp. 246-254.

[181] Y. Joon-Seo, B. Seong-Ok, and K. Chong-Min, "A floorplan-based planning methodology for power and clock distribution in ASICs [CMOS technology]," *Proceedings of ACM/IEEE Design Automation Conference*, 1999, pp. 766-771.

[182] F. Young, M. Wong, and H. Yang, "On extending slicing floorplan to handle L/T-shaped modules and abutment constraints," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, 2001, pp. 800-807.

[183] F. Young, D. Wong, and H. Yang, "Slicing floorplans with boundary constraints," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, 1999, pp. 1385-1389.

[184] D. Sengupta and R. Saleh, "Application-driven floorplan-aware voltage island design," *Proceedings of ACM/IEEE Design Automation Conference*, 2008, pp. 155-160.

[185] H. Wu, M.D. Wong, I. Liu, and Y. Wang, "Placement-Proximity-Based Voltage Island Grouping Under Performance Requirement," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, 2007, pp. 1256-1269.

[186] Y. Cai, B. Liu, Q. Zhou, and X. Hong, "A Thermal Aware Floorplanning Algorithm Supporting Voltage Islands for Low Power SOC Design," *Integrated Circuit and System Design*, Springer, 2005, pp. 257 - 266.

[187] D. Frank, S. Kosonocky, and R. Dennard, "Supply voltage strategies for minimizing the power of CMOS processors," *2002 Symposium on VLSI Technology. Digest of Technical Papers (Cat. No.01CH37303)*, IEEE, 2002, pp. 102-103.

[188] Ö. Paker, J. Sparsø, N. Haandbæk, M. Isager, and L.S. Nielsen, "A Low-Power Heterogeneous Multiprocessor Architecture for Audio Signal Processing," *Journal of VLSI Signal Processing Systems*, vol. 37, 2004, pp. 95-110.

[189] D. Milojevic, L. Montperrus, and D. Verkest, "Power Dissipation of the Network-on-Chip in Multi-Processor System-on-Chip Dedicated for Video Coding Applications," *Journal of Signal Processing Systems*, vol. 57, 2009, pp. 139-153.

[190] K. Lahiri and A. Raghunathan, "Power analysis of system-level on-chip communication architectures," *Proceedings of International Conference on Hardware Software Codesign*, 2004, pp. 236-241.

[191] F. Brglez, D. Bryan, and K. Kozminski, "Combinational profiles of sequential benchmark circuits," *Proceedings of IEEE International Symposium on Circuits and Systems*, 1989, pp. 1929-1934.

[192] "GSRC: http://vlsicad.eecs.umich.edu/BK/GSRCbench/."

[193] R. Dick, D. Rhodes, and W. Wolf, "TGFF: task graphs for free," *Proceedings of International Workshop on Hardware/Software Codesign*, 1998, pp. 97-101.

[194] T. Ching-Han and K. Sung-Mo, "Cell-level placement for improving substrate thermal distribution," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, 2000, pp. 253-266.

[195] "LAPACK-Linear Algebra PACK-age," 2009.

[196] M. Ghoneima and Y. Ismail, "The importance of including thermal effects in estimating the effectiveness of power reduction techniques," *Proceedings of IEEE Custom Integrated Circuits Conference*, 2005, pp. 294-297.

[197] W.K. Ching and M.K. Ng, *Markov chains: models, algorithms and applications*, Springer, 2006.